On the relative complexity of approximately counting H-colourings

by

Steven M. Kelk

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Computer Science

July 2003

THE UNIVERSITY OF
WARWICK

Contents

List of Figures

Acknowledgments

A particular debt of gratitude must go to my supervisor Leslie Goldberg, who has been a consistently excellent supervisor throughout the duration of this PhD. I have benefitted immensely from Leslie's professionalism, her technical insights and guidance, and her understanding on those occasions when events from beyond my academic life have threatened to interfere with my PhD work.

Other people I wish to offer specific thanks to include my parents and my brother, my long-suffering housemates (Tim Heron, Andy Cooke, Simon McIlhinney and Caroline Rogers), my good friend Laura Lawson, and the various people I have been fortunate to share an office with at one time or another (Graham Cormode, Jon Sharp, Harish Bhanderi and Christopher Roe.)

Of course, there are countless other people who deserve a vote of thanks, too many to mention by name. Suffice to say, I am immensely grateful to all those people who have helped me out in one way or another over the last few years. Sincere thanks.

Steven Kelk

Declarations

The bulk of this thesis is my own and has not been published previously. In this section I shall declare exceptions to this statement, highlighting not only previously published work but also those parts of the thesis that have not been published in the literature but are the result of collaboration or communication with others. The formal declaration (required by the University of Warwick's submission guidelines) follows after these.

Chapter 1 (plus Appendix A.1) is a detailed literature survey and, as such, refers to a large number of papers by other authors. In particular, a number of definitions and H-colouring results from [8] are reproduced in the later part of this chapter.

In Chapter 2, the complexity classifications for 2-vertex and 3-vertex graphs were communicated to me informally by Dyer, Goldberg, Greenhill and Jerrum (DGGJ). (The proofs presented have in some cases been adapted and/or formalised by myself.) Various other parts of this chapter repeat results and proofs taken from the paper [8] by DGGJ; where this occurs it is explicitly marked in the text to delineate it from my original work. Lemma 2.14 was a collaborative effort with Dyer and Goldberg.

Sections 3.4 and 3.5 from Chapter 3, and the whole of Chapter 4, are based on the earlier publication [14]. I was equal-status co-author on this paper with Goldberg and Paterson. (The paper was written during my period of study.)

Chapter 6 contains an algorithm by Jerrum, communicated informally to this author.

Chapter 7 again reproduces some work from [8], and elaborates on a number of sketch proofs communicated to me by DGGJ. (In particular, the $\#H \leq_{\mathsf{AP}} \#LargeIS$ proof in Section 7.3.2, and the "crossing property" observation.) As in Chapter 2, work not originally attributable to me is explicitly marked to distinguish it from my original work. Lemma 7.5 and its associated lemmas were jointly developed by myself in collaboration with Dyer and Goldberg.

Finally, here's the formal bit:

This thesis is entirely my own work apart from the exceptions listed above. This thesis has not been previously submitted in any form for a degree at Warwick or any other university.

Abstract

The infinite suite of H-colouring counting problems affords us an excellent opportunity to refine our understanding of how the complexity hierarchy for approximate counting problems is structured. In particular, the more we understand about how different H-colouring problems are distributed throughout the approximate counting complexity hierarchy, the more nuanced our understanding about the hierarchy becomes for more general approximate counting problems. In the context of this thesis, "relative complexity" refers primarily to the complexity of different H-colouring problems *relative to each other*, but also (to a lesser extent) relative to non-H-colouring approximate counting problems.

In this thesis we build on the foundations laid by earlier authors and look to enhance our understanding, as far as possible, of which H-colouring problems are of similar complexity for approximate counting. Against this backdrop, the major original contribution of this thesis can be summarised as follows:-

- An almost complete complexity classification of those H-colouring problems where H has 4 or fewer vertices;
- A wide range of new gadgets, utility lemmas and reduction techniques;
- A suite of general classification lemmas which establish the complexity of various different families of H, with a particular emphasis on identifying the H-colouring problems which are "hardest" to approximately count;
- A framework (developed in conjunction with others) wherein the very closely related problem of approximately uniformly *sampling* H-colourings can be studied;
- The subsequent discovery that there is a non-trivial complexity lower bound for approximately uniformly sampling H-colourings with no non-trivial components. This has implications for approximately counting H-colourings;
- The collection of a body of evidence which suggests that the complexity hierarchy for approximately counting H-colourings (and thus for more general approximate counting problems) may well be significantly more nuanced than a simple "easy", "intermediate" and "hard" partitioning.

Chapter 1

The context

1.1 Introduction

This thesis is dedicated to the topic of determining the relative complexity of approximately counting H-colourings. In order that the reader can fully appreciate the original work in this thesis (which starts in Chapter 2) and place it in a wider context we use this chapter to conduct a literature survey of the subject area and introduce key definitions.

1.2 Decisions, counting, approximate counting

1.2.1 From decisions to exact counting

In the field of computer science, decision problems have a long and rich history. The class of NP-complete decision problems has been well studied, and the unresolved question "does $P = NP$?" continues to be a cornerstone problem in the study of algorithmic efficiency.

Whereas decision problems are concerned with questions such as "does there exist...?", the related field of exact enumeration problems (which we henceforth call *counting* problems) is instead concerned with questions such as "how many...?" To this end, in 1979 Valiant (see [29]) defined the complexity class $\#P$, which loosely speaking is the counting analogue of NP. To formally explain the set of problems captured by $\#P$ it is necessary to introduce a few definitions.

Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation on binary strings. R is a *p-relation* if

1. for all $(x, y) \in R$, $|y| = O(\text{poly}(|x|))$,
2. there is a polynomial-time decision procedure for determining whether $(x, y) \in R$.

A language L is in NP iff there exists a p-relation R such that

$$x \in L \Longleftrightarrow \exists y \text{ such that } (x, y) \in R$$

Phrasing this in more conventional NP parlance, y is a (polynomial-size) certificate that x is in L, in the sense that given y, a deterministic polynomial-time Turing machine can check that x is in the language L. So, to offer a concrete example, the decision problem SAT is in NP because the relation

$$R_{SAT} = \{(x, y) : x \text{ is a boolean formula and } y \text{ is a satisfying truth assignment of } x.\}$$

is a p-relation.

Now, let f be a function from $\{0,1\}^*$ to $\mathbb{N}$. We say that $f \in \#P$ iff there exists a p-relation R such that for all $x \in \{0,1\}^*$, $f(x) = |\{y : (x, y) \in R\}|$. Thus a function f is in $\#P$ iff $f(x)$ is the number of certificates verifying that x is in some NP language i.e. when given an instance of some specific NP problem, f counts the number of solutions (which in equivalent to saying that $f(x)$ is the number of accepting computations on the Turing machine deciding x.) Informally, $\#P$ is the class of counting problems where it does not take too long (i.e. no more than polynomial time in the size of the input) to verify whether an object in the potential domain of solutions is in fact a solution. As such, $\#P$ captures most reasonable counting problems: there are not many natural counting problems where it takes an exponentially long time to recognise what you are trying to count! To continue with our earlier example of R_{SAT}, it is not difficult to see that the problem of counting satisfying assignments of a boolean formula is in

$\#P$. This counting problem is one of the most important problems we shall encounter in this thesis, so we formally define it here. (The CNF restriction does not affect its membership in $\#P$.)

Name: $\#SAT$

Instance: A boolean formula ϕ in CNF (Conjunctive Normal Form)

Output: The number of satisfying assignments to ϕ.

A counting problem is said to be $\#P$-hard if all counting problems in $\#P$ are polynomial-time Turing-reducible to it. If (in addition) the problem is in $\#P$ itself then the problem is said to be $\#P$-complete. The set of $\#P$-complete problems includes problems such as computing the permanent of a 0-1 (i.e. boolean) matrix (which is equivalent to counting perfect matchings in a bipartite graph) and counting the number of satisfying solutions of a boolean formula (i.e. $\#SAT$). These (and other $\#P$-complete) problems are widely thought to be intractable (i.e. cannot be solved in polynomial time) so $\#P$-completeness is considered a very strong indication of intractability.[1]

Let us look once again at the problem $\#SAT$. Given that the decision counterpart of $\#SAT$ is NP-complete, it comes as no surprise that $\#SAT$ is, in a general sense, considered intractable. To see this, note that a polynomial-time oracle for $\#SAT$ could be used to solve the decision problem SAT: answer "yes" to the satisfiability question iff $\#SAT$ returns a non-zero answer. However, this informal evidence of intractability does not constitute proof that $\#SAT$ is $\#P$-complete. Rather, the $\#P$-completeness of $\#SAT$ follows because there exist proofs of Cook's theorem (which states that any decision problem in NP can be reduced to the decision problem SAT in polynomial time) which use *parsimonious* reductions. (A parsimonious reduction is one that "preserves the number of solutions", so a parsimonious reduction from a problem $\#X$ to $\#Y$ might be a function f that transforms the input σ to $\#X$ such that

[1]If every function in $\#P$ was shown to be polynomial-time solvable then this would cause the *polynomial hierarchy* to collapse, an eventuality thought extremely unlikely.

$\#X(\sigma) = \#Y(f(\sigma))$.) Therefore, by extension of Cook's theorem, it follows that all problems in $\#P$ are polynomial-time reducible to $\#SAT$, which (combined with the fact that $\#SAT \in \#P$) confirms that $\#SAT$ is $\#P$-complete. Now it could well be that all NP-complete decision problems have $\#P$-complete counting analogues; we have no counter-examples to this conjecture but, because we cannot yet prove that all NP-completeness proofs can be made parsimonious we do not yet know if this is a theorem.

It is also interesting to note that some decision problems that are "easy" (i.e. are in P) become "hard" (i.e. $\#P$-complete) when switched to the counting world. Valiant showed that this is the case for $\#Matchings$, the problem of counting the number of perfect matchings in a given graph. Another example, more pertinent to this thesis, involves *independent sets*. An independent set of a graph $G = (V(G), E(G))$ is any subset I of $V(G)$ such that all vertices in I are pairwise non-adjacent in G. The decision problem "does G have an independent set?" is trivial because all graphs at least have an independent set of size zero. However, consider the counting analogue, $\#IS$, which we will encounter on numerous occasions throughout this thesis:

Name: $\#IS$

Instance: A graph G

Output: The number of independent sets in G

We know $\#IS$ to be $\#P$-complete, even for low-degree graphs [9].

In terms of counting non-trivial combinatorial structures (such as graph colourings, shortest paths etc.) we do know of a very small number of problems in $\#P$ which can be solved exactly in polynomial time. These problems include counting spanning trees in a graph and counting perfect matchings in a planar graph. See Kasteleyn's survey article [24] for details of these two problems, and see any standard student text (such as [4]) for details of other polynomial-time exact counting algorithms. However, combinatorial counting problems do not in general seem to lend themselves to polynomial-time solutions.

1.2.2 Approximate counting

The need for practical computing solutions, particularly in fields such as statistical physics and engineering, has been a strong driving force behind the development and study of approximate counting algorithms, i.e. those designed to return approximate (as opposed to exact) solutions. This field of study is particularly vital where the exact counting version of a problem is thought to be intractable i.e. where the exact counting problem is $\#P$-complete. The primary focus of this thesis is on the *relative* complexity of approximate counting algorithms, and in particular on the relative complexity of approximate H-colouring counting problems, a suite of problems which we shall introduce shortly. (Recognizing the potential ambiguity of the term "relative complexity", we clarify the meaning of this term in due course.) First, some further definitions are necessary; much of this section is heavily based on [8].

A *randomized approximation scheme* (RAS) for a function $f : \Sigma^* \to \mathbb{N}$ (where Σ^* is the domain from which instances of the problem are drawn) is a probabilistic Turing machine that takes as input a pair $(x, \epsilon) \in \Sigma^* \times (0, 1)$ and produces as output an integer random variable Y satisfying the condition

$$\Pr(e^{-\epsilon} \leq Y/f(x) \leq e^{\epsilon}) \geq 3/4$$

A randomised approximation scheme is said to be *fully polynomial* if it runs in time no greater than poly($|x|$,ϵ^{-1}). The phrase "fully polynomial randomized approximation scheme" [23] is usually abbreviated to $FPRAS$. Thus, an $FPRAS$ is an efficient approximate counting algorithm. It is natural to question what happens to problems that are $\#P$-complete (i.e. hard to count exactly) once they are switched to the approximate-counting world.

There is a growing body of literature which investigates the feasibility of developing an $FPRAS$ for certain $\#P$-complete counting problems. Many such attempts utilise what is known as the *Markov Chain Monte Carlo* (MCMC) method. This is a method which, under certain circumstances, permits the generation of an approximately-uniform sample from a state space by simulating a random walk (consisting of no more than a polynomial number of steps) on that state space. The idea is then to use this efficient, approximate sampler to build an $FPRAS$, exploiting the fact that for some problems in $\#P$ approximate counting and approximate sampling are of equivalent complexity. (We shall discuss this approximate sampling/approximate counting relationship in more depth later on.) Some examples of this technique in use include approximately counting the number of k-colourings of a low-degree graph [18] and a number of papers which put forth positive and negative results regarding the approximability of the independent set counting problem for certain degree bounds ([25], [9], [6]). Overviews of the MCMC method can be found in works such as [19].

Interestingly, it has been shown that different $\#P$-complete problems respond differently in the switch to the approximate counting domain. For example, it is considered highly unlikely that $\#SAT$ is $FPRAS$able because the existence of an $FPRAS$ for $\#SAT$ would mean the decision problem SAT could be solved in randomized polynomial time.[2] In contrast, some $\#P$-complete problems (albeit not that many) have been identified that have an $FPRAS$; two examples of such problems include counting matchings in a graph G and counting satisfying assignments of a boolean formula in Disjunctive Normal Form (DNF). (See [19] and [22].) Significantly, Jerrum, Sinclair and Vigoda recently showed that the computation of the permanent (widely considered the "canonical" $\#P$-complete problem) has an $FPRAS$ [21].

Before proceeding any further with our analysis of approximate counting problems, it is necessary to detour briefly into the realm of approximately uniform *sampling* problems.

[2] That is, it would mean $RP = NP$, where RP is the class of decision problems which can be solved in polynomial time by a probabilistic algorithm with one-sided error probability.

1.2.3 Approximate sampling

The study of approximate counting problems is heavily intertwined with the study of approximate sampling problems. This is largely attributable to the seminal work of Jerrum, Valiant and Vazirani (JVV) [16] in which it was shown that for the class of *self-reducible* problems in $\#P$, the problem of approximately counting solutions is interreducible with the problem of sampling solutions approximately uniformly at random. (The fairly restrictive notion of self-reducibility has been generalised to a notion of *self-partitionability* by Dyer and Greenhill in [7]. This is an attempt to more broadly define the set of problems for which the JVV technique works i.e. capturing those problems which are not formally self-reducible but are "effectively" so.) That is, for a self-reducible problem an $FPRAS$ for the counting problem can be turned into an $FPAS$ (*Fully Polynomial Approximate Sampler*) for the analogous sampling problem, and vice-versa.[3] (We discuss the $FPAS$ in greater detail in Chapter 3, but for now it is adequate to understand it as a sampler that returns samples with distribution "close enough" to that which is desired, without taking too long.[4]) We know that problems such as counting independent sets and counting proper colourings are self-reducible; hence the utilisation of the MCMC method to develop efficient approximate counting algorithms for certain variants of these problems.

A natural question that arises is whether approximate counting and approximate sampling are interreducible for all problems in $\#P$. (For many problems in $\#P$ it is not clear whether they are self-reducible or, more generally, whether their approximate counting and approximate sampling counterparts are interreducible by any method.) There is strong evidence that this is *not* the case. In [12], it is observed that there exists a problem in $\#P$ which has an $FPRAS$ but no $FPAS$, unless there is a polynomial time

[3] Self-reducibility is actually a property of the underlying p-relation, so statements such as "the problem of counting independent sets is self-reducible" and "the problem of sampling independent sets is self-reducible" are, in effect, equivalent.

[4] Throughout this thesis our work with sampling tends to be discussed in terms of what we call the $PAUS$ (*Polynomial Almost Uniform Sampler*) rather than the $FPAS$. This is explained further in Chapter 3, but in a nutshell a $PAUS$ is a slightly less demanding version of an $FPAUS$ - which can sometimes be "bootstrapped" to turn it into an $FPAUS$ - and an $FPAUS$ is simply an $FPAS$ that specialises on sampling from the uniform distribution.

algorithm for computing the discrete logarithm.[5] (In the same paper it is noted that it remains an open question as to whether there exists a problem in $\#P$ that has an $FPAS$ but no $FPRAS$.) However, given the fairly contrived nature of the $FPRAS$-but-no-$FPAS$ problem discussed in [12], it is believed by many that the equivalence holds for most, if not all, "reasonable" problems in $\#P$. We return to this question on several occasions throughout this thesis, principally because it is not clear whether our problems of choice (i.e. the suite of H-colouring problems) are, in general, self-reducible[6].

1.2.4 Relative complexity of approximate counting problems

Given the concrete evidence that $\#P$-complete exact counting problems respond to approximation in different ways, the next step is to chart more fully the behaviour of $\#P$-complete problems in the approximation domain. While a comprehensive classification is likely to be very hard to come by, it is more feasible to ask questions such as how many / what type of $\#P$-complete problems have an $FPRAS$? How many / what type of $\#P$-complete problems are hard even when exact counting is relaxed to approximate counting? Do all $\#P$-complete problems respond in one of two ways in the approximation domain (i.e. become $FPRAS$able or stay hard), or is the approximation complexity landscape more nuanced? Of course, to even ask such questions without ambiguity it is necessary to define a formal framework for the comparison of approximate counting problems. In [8], Dyer, Goldberg, Greenhill and Jerrum (DGGJ) set out a formal framework to facilitate "approximation preserving" reductions. We repeat verbatim the relevant definitions from [8].

Suppose $f, g : \Sigma^* \to \mathbb{N}$ are functions whose complexity (of approximation) we want to compare. An *approximation-preserving reduction* (AP-reduction) from f to g is a probabilistic oracle Turing machine M that takes as input a pair $(x, \epsilon) \in \Sigma^* \times (0, 1)$, and satisfies the following three conditions:

[5] In fact it is shown that any *one-way permutation* can be used as a basis for a problem in $\#P$ that has an $FPRAS$ but probably no $FPAS$.

[6] To be more precise, it is not clear (for many H-colouring problems) whether interreducibility follows by any method, JVV or otherwise.

1. Every oracle call made by M is of the form (w, δ), where $w \in \Sigma^*$ is an instance of g, and $0 < \delta < 1$ is an error bound satisfying $\delta^{-1} \leq \text{poly}(|x|, \epsilon^{-1})$;

2. The Turing machine M meets the specification for being a randomised approximation scheme for f whenever the oracle meets the specification of being a randomised approximation scheme for g;

3. The run-time of M is polynomial in $|x|$ and ϵ^{-1}.

If an AP-reduction from f to g exists we write $f \leq_{\mathsf{AP}} g$, and say that f is AP-reducible to g. If $f \leq_{\mathsf{AP}} g$ and $g \leq_{\mathsf{AP}} f$ then we say that f and g are AP-interreducible, and can write $f \equiv_{\mathsf{AP}} g$.

To clarify some terminology, if we say a problem $\#X$ is "$\#Y$-hard" with respect to AP-reducibility we mean $\#Y \leq_{\mathsf{AP}} \#X$; similarly $\#X$ being "$\#Y$-easy" means $\#X \leq_{\mathsf{AP}} \#Y$. We denote the class of problems AP-interreducible with a problem $\#Y$ by $\equiv_{\mathsf{AP}} \#Y$. Owing to the definition of AP-interreducibility, we can often use $\#Y$ and $\equiv_{\mathsf{AP}} \#Y$ interchangeably without confusion e.g. under AP-reducibility the statement "$\#X$ is $\equiv_{\mathsf{AP}} \#Y$-hard" is equivalent to the statement "$\#X$ is $\#Y$-hard". We will make extensive use of the $\equiv_{\mathsf{AP}} \#Y$-hard and $\equiv_{\mathsf{AP}} \#Y$-easy notation so the reader is encouraged to take note of its definition.

Informally, the statement $f \leq_{\mathsf{AP}} g$ means that if we can efficiently approximate g then we can efficiently approximate f. This thesis is (with the exception of our work on *sampling-preserving* reductions) founded wholly on the AP-reduction framework defined above. The authors (of [8]) note that their definition of AP-reduction is just one possible way of defining an approximation-preserving reduction. Given a number of competing considerations, they decided to define AP-reductions in a "liberal" manner i.e. to encourage rather than discourage AP-reductions between counting problems. They note, however, that the examples of AP-reductions demonstrated in [8] rarely use the AP-reduction to its full power:- most reductions decline to use randomisation, make only a single oracle call and perform very little processing on the value returned by the oracle. As a general comment, this heavily restricted use of the AP-reduction continues throughout this thesis.

If a counting problem $\#Y \in \#P$ is such that, for all counting problems $\#X \in \#P$, $\#X \leq_{\mathsf{AP}} \#Y$, we say that $\#Y$ is complete for $\#P$ with respect to AP-reducibility. We have already seen strong evidence that $\#SAT$ is unlikely to be $FPRAS$able, and indeed $\#SAT$ is complete for $\#P$ w.r.t. AP-reducibility. This holds for the same reason that the $\#P$-completeness of $\#SAT$ holds i.e. because for all decision problems X in NP we know how to build a parsimonious reduction from the decision problem X to the decision problem SAT. Furthermore, [8] goes further and proves that if a decision problem X is NP-complete, $\#X$ is complete for $\#P$ w.r.t. AP-reducibility. This powerful theorem is notable for a number of reasons, not least the fact that we don't yet know whether an NP-complete problem always has a $\#P$-complete exact counting counterpart. (The fact that NP-complete problems are "automatically hard" for $\#P$ with respect to AP-reducibility also ties in with the observation that, whilst approximate counting lies just above NP in the polynomial hierarchy, Toda has in contrast proven that $\#P$ contains the *whole* of the polynomial hierarchy [28]. This constitutes strong complexity-theoretic evidence that approximate counting is strictly easier than exact counting.)

Thanks to this theorem, we immediately inherit a large number of counting problems that are all AP-interreducible, and (because they are all AP-interreducible with $\#SAT$) are unlikely to have efficient approximation algorithms. Given its prominence in complexity work, it is sensible to use $\#SAT$ as the chief representative of this class, and as a result we denote the class by writing $\equiv_{\mathsf{AP}} \#SAT$. An advantage of $\#SAT$ being complete for $\#P$ w.r.t. AP-reducibility is that if, for a counting problem $\#X$, we demonstrate $\#SAT \leq_{\mathsf{AP}} \#X$, we immediately have the result that $\#X \equiv_{\mathsf{AP}} \#SAT$. Hence, the expressions $\#SAT \leq_{\mathsf{AP}} \#X$ and $\#X \equiv_{\mathsf{AP}} \#SAT$ effectively mean the same thing, assuming (as is always the case in this thesis) that $\#X \in \#P$.

Also, with reference to the problem $\#IS$, DGGJ show that - as with the switch from the decision world to the exact counting world - some decision problems in P become $\equiv_{\mathsf{AP}}\#SAT$ in the approximate counting world.

Having defined the AP-reduction framework, DGGJ use it to analyse a number of counting problems. We briefly summarise their results in this chapter, with a view to analysing them in more detail at appropriate points throughout this thesis. In summary, they consider a variety of $\#P$-complete counting problems (not known to permit an $FPRAS$) and attempt to position their approximation counterparts in the complexity hierarchy as defined by AP-reducibility. They find that quite a few of the problems considered (such as $\#IS$) are $\equiv_{\mathsf{AP}}\#SAT$, but - intriguingly - they also identify a suite of problems of apparently intermediate complexity. More specifically, they locate an infinite family of seemingly non-$FPRAS$able counting problems that they are equally unable to show are $\equiv_{\mathsf{AP}}\#SAT$. These problems are interreducible with the problem $\#BIS$, defined as follows:

Name: $\#BIS$

Instance: A bipartite graph G

Output: The number of independent sets in G

(Note that $\#BIS$ is equivalent to the $\equiv_{\mathsf{AP}}\#SAT$ problem $\#IS$ when its input is restricted to being bipartite.) We denote the class of problems AP-interreducible with $\#BIS$ as $\equiv_{\mathsf{AP}}\#BIS$. DGGJ put forth some evidence which suggests that it may not be possible to efficiently approximate $\#BIS$. Firstly, they demonstrate that $\#BIS$ is of equivalent complexity to a number of other problems that are not (yet) known to admit an $FPRAS$, such as counting downsets in partial orders and counting satisfying assignments to "restricted Horn" CNF boolean formulas. Secondly, they show that $\#BIS$ - and a number of the problems proven to be interreducible with it - are complete (with respect to AP-reducibility) for a logically-defined subclass of $\#P$ called $\#RH\Pi_1$. We discuss $\#RH\Pi_1$ in a little more detail in Chapter 7, but loosely speaking $\#RH\Pi_1$ is the class of counting problems that can be represented by first-order Boolean formulas where every clause is of the form x, $\neg x$ or $x \to y$ and only universal quantifiers are used.

In [8], some of the problems tested with regard to AP-reducibility are H-colouring problems. Thus, we will shortly introduce the suite of problems which this thesis is dedicated to, but first a note about "relative complexity."

A comment on the meaning of "relative complexity"

Before going any further, it is important to clarify what is meant, in the context of this thesis, by "relative complexity". The AP-reduction framework does not *in itself* say anything at all about the absolute complexity of approximate counting problems categorised within it; it is simply a mechanism for comparing the absolute complexity of approximate counting problems *relative to each other.* So, generally speaking, the intention in this thesis is *not* to comment on the complexity of approximate counting problems (particularly, H-colouring problems) relative to some other complexity measure. Instead, we are mainly concerned with the complexity of different H-colouring approximate counting problems relative to *each other*, and more generally the complexity of different H-colouring approximate counting problems relative to other (potentially non-H-colouring) approximate counting problems.

Of course, the AP-reduction hierarchy does not float completely "freely" in complexity space, because there exist problems at the "bottom" and "top" of the AP-reduction hierarchy which we have a more concrete understanding about. That is, the $FPRAS$able problems are easy in an absolute sense, and the $\equiv_{\mathsf{AP}}\#SAT$ problems are intractable under standard complexity assumptions. (As mentioned earlier, we also have a reasonable understanding of where approximate counting fits in the polynomial hierarchy - i.e. quite low down - and as a consequence we understand that there is (in all likelihood) a significant difference in absolute complexity between the hardest approximate counting problems and the hardest exact counting problems.)

1.3 H-colouring

1.3.1 The definition of the problem

The term H-colouring describes an infinite class of problems with particularly rich qualities. Let $H = (V(H), E(H))$ be a fixed, undirected graph, where each vertex potentially has a loop. Now, consider an undirected, unlooped input graph $G = (V(G), E(G))$. (It should be emphasised that H is not part of the input; on the contrary, H defines the problem. Note also that throughout the rest of the thesis we leave it as implicit that H and G are undirected, and G is unlooped.) An H-colouring of G is a function $C : V(G) \to V(H)$ such that, for all $u, v \in V(G)$, $\{u, v\} \in E(G) \Rightarrow \{C(u), C(v)\} \in E(H)$. Informally, the vertices of H are "colours" and the edges of H determine which colours are allowed to be adjacent on the vertices of G. (Technically speaking, an H-colouring is a homomorphism from G to H.) We let $H(G)$ denote the set of H-colourings of the graph G, and let $\#H(G)$ denote $|H(G)|$ i.e. the number of H-colourings of G.

The above definition of H-colouring is the most basic formulation of the problem. There are many variants, such as *list H-colouring*, where in addition to G the input includes a function $r : V(G) \to 2^{V(H)}$ which restricts each vertex in G to a particular set of colours. (See [5] for more information.) However, in this thesis we focus exclusively on the basic formulation; as will become apparent, this does not leave us short of challenging questions to answer! Observe that every distinct graph H enforces

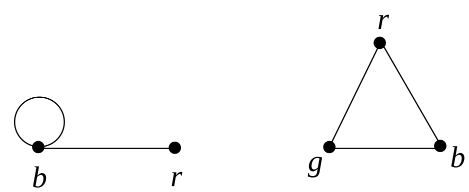


Figure 1.1: *Two examples of H-colouring problems. The H graph on the left encodes independent sets, the H graph on the right encodes proper 3-colourings*

a different set of colouring rules. As such, a number of well-known counting problems can be generalised within the H-colouring framework. For example, where H is a complete unlooped graph on m vertices (i.e. K_m), counting H-colourings is equivalent to counting proper m-colourings of G. (Each colour in H can be adjacent to the other $m-1$ colours, but not to itself, because it lacks a loop.) Similarly, consider the graph H which consists of two vertices r, b, where r and b are connected by an edge, and only b has a loop. (That is, $V(H) = \{r, b\}$ and $E(H) = \{\{b, b\}, \{b, r\}\}$.) Counting H-colourings is in this instance equivalent to counting independent sets i.e. for all G, $\#H(G) = \#IS(G)$. To see why this is, observe that a vertex of G coloured r can only be adjacent to vertices of G coloured b, while a vertex coloured b can be adjacent to b or r vertices. In other words, b vertices are OUT of the independent set, while r vertices are IN. It is worth noting that adding *weights* to the vertices and edges of this graph makes it more generally represent what is known as the *hard-core lattice gas* model from statistical physics. We discuss vertex weighting in much greater detail later on in the thesis (where we note that each vertex-weighted H-colouring problem that uses only rational vertex weights corresponds to some vertex-unweighted H-colouring problem) but, informally, vertex-weighted H-colouring is where each vertex $c \in V(H)$ has a positive "weight" $w(c)$ attached to it. The weight of a colouring $w(C)$ (for $C \in H(G)$) then becomes equal to $\prod_{u \in V(G)} w(C(u))$ and the value that the problem seeks to compute is $Z = \sum_{C \in H(G)} w(C)$ which is also known as the *partition function*.[7] Thus, the standard (unweighted) H-colouring counting problem is equivalent to the problem of computing the partition function where all weights on the colours of H are equal to 1. Unless stated otherwise this thesis presumes H-colouring problems are unweighted. (We do not tackle edge-weighted H-colouring at all.)

1.3.2 A brief history of deciding, exactly counting and approximately counting/approximately sampling H-colourings

The decision and exact counting H-colouring problems have already been well analysed. Hell and Nešetřil have proven [15] the celebrated dichotomy result which states that the

[7]The sampling counterpart to vertex-weighted H-colouring is thus to produce a colouring $C \in H(G)$ with probability $w(C)/Z$.

decision problem for a graph H (i.e. "does G have at least one H-colouring?") is in P iff H is an unlooped bipartite graph or a non-bipartite graph with at least one loop. (Note that, unlike some other treatments of H-colouring, our definition of bipartite is such that bipartite graphs do *not* have loops, so we henceforth drop the "unlooped" prefix in front of the term bipartite.) Otherwise, the decision problem for H is NP-complete. More recently, Dyer and Greenhill have proven [10] that the exact counting problem $\#H$ for a graph H is $\#P$-complete unless every component of H is either a complete bipartite graph or a complete non-bipartite graph with loops on every vertex (in which case it is in P, and H is known to be *trivial*.) Thus, the result of Dyer and Greenhill shows that almost all H-colouring problems are hard to count exactly.[8] As a consequence, this makes H-colouring a very interesting domain of problems to investigate from an approximation point of view.

Here we briefly survey some of the main results concerning direct attempts to approximately count (and sample) H-colourings. Following this we set the context for the rest of this thesis by exploring preliminary results by DGGJ that seek to determine the relative complexity of approximately counting H-colourings.

As alluded to earlier, most attempts to approximately count H-colourings have been tightly interwoven with attempts to approximately sample H-colourings. For certain specific H-colouring problems, such as independent sets and proper colouring, we know that such problems are self-reducible. For graphs such as these, efficient approximate counting is interreducible with efficient approximate sampling. More generally, however, it has been noted that even for H not known to be self-reducible, approximate counting is reducible to approximate sampling. The result by Dyer, Jerrum and Vigoda [11] shows that for *dismantleable* H (see [2]) approximately counting H-colourings can be reduced to approximately sampling H-colourings. However, in addition to the dismantleability constraint their proof is restricted to the case when the maximum degree

[8] Dyer and Greenhill's result also applies to the situation where the maximum degree of G is bounded. With regard to the H-colouring decision problem, the bounded-degree case is more complex [13]

of G is bounded. Dyer, Goldberg and Jerrum have extended this result (see [12]) to show that approximately counting H-colourings is reducible to approximately sampling H-colourings for *all* H and for general graphs G. (The result actually goes further than this and is shown to hold for the case when H has weights on its vertices and edges. As such, they prove that the ability to efficiently approximately sample from the *Gibbs* distribution yields an $FPRAS$ for the partition function.) Thus, we now know that finding an efficient approximate sampler (i.e. an $FPAS$) for an H-colouring problem automatically allows us to build an efficient approximate counter (i.e. an $FPRAS$) for the counting problem. Whether the other direction (i.e. sampling to counting) is true for all H - and not just for self-reducible H - is an open question. We discuss this again in Chapter 3; it may be that approximately counting H-colourings is (in general) strictly easier than approximately sampling them.

Positive approximate sampling (and thus approximate counting) results have been shown for specific variants of certain H-colouring problems. Jerrum's result [18] is a simple example of an $FPAS$ (and thus an $FPRAS$) for the k-colouring problem (which, as we state above, is the H-colouring problem with H comprising a complete unlooped graph on k vertices) where the maximum degree of G is bounded above by a constant which is a particular function of k. Vigoda's paper [30] offers further positive results on k-colourings. Positive results for approximately sampling (and thus approximately counting) independent sets have been put forward by Dyer and Greenhill in [9]. Slightly more generally, the aforementioned paper by Dyer, Jerrum and Vigoda (i.e [11]) proves that, for all dismantleable H and appropriately degree-bounded G, there always exist weights on the vertices of H such that efficient approximate sampling from $H(G)$ is possible.

There are also a number of negative H-colouring results. Dyer, Frieze and Jerrum [6] have shown that the MCMC sampling method is likely to fail (i.e. take exponentially long to converge on a stationary distribution) when used to sample independent sets in a graph with constant maximum degree greater than or equal to 6. They also show that for constant maximum degree greater or equal than 25 it cannot be possible to efficiently approximately sample (and thus, by the self-reducibility of the problem, approximately count) independent sets unless $RP = NP$. One of the main attempts to reason about the approximability of H-colouring more generally has produced a negative result. Cooper, Dyer and Frieze [3] have shown that for all non-trivial H there exist input graphs G for which a "cautious" deployment of the MCMC method fails to produce a sufficiently uniform sample within polynomial time.[9] (Informally, a "cautious" Markov Chain is one that only attempts transitions between members of the state space that are not too far apart.)

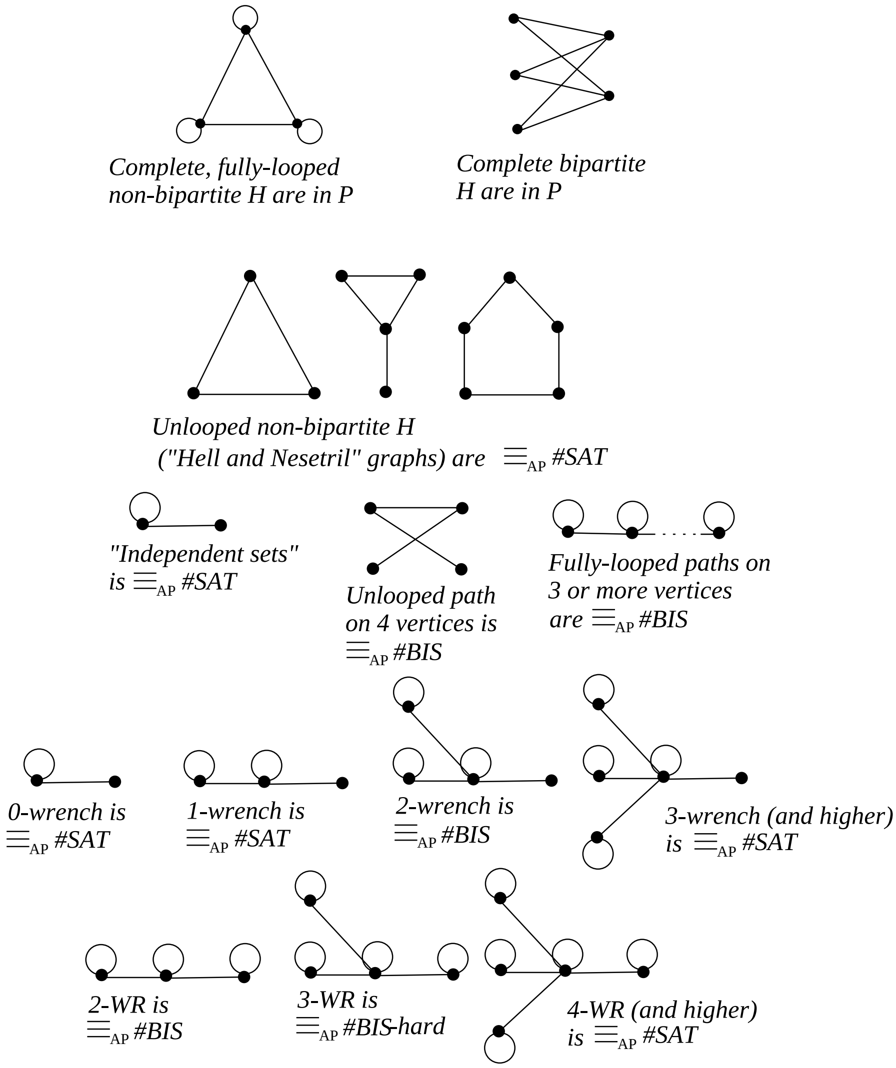
Before moving onto the relative complexity of approximate H-colouring problems, it is worth putting these positive and negative results in context. With regard to the positive results, it should be noted that they do not answer the following question: does there exist a non-trivial H-colouring problem for which efficient approximate sampling and/or efficient approximate counting is possible for *general* G? With regard to the negative results, we have already mentioned briefly that, for H-colouring, the link between approximate sampling and approximate counting is not entirely clear. This is something we discuss again in Chapter 3 but, because it remains a possibility that approximately counting H-colourings is strictly easier than approximately sampling them, it is not yet apparent how far negative approximate sampling results automatically imply negative approximate counting results.

1.3.3 Preliminary results in determining the relative complexity of approximately counting H-colourings

Having surveyed attempts to determine the *absolute* complexity of approximately counting certain H-colourings, we now shift to *relative* complexity. In [8], Dyer, Goldberg, Greenhill and Jerrum consider a number of graphs H and attempt to position the problem $\#H$ within the AP-reducibility hierarchy. We briefly outline their results here; these

[9] Note that the Dyer, Jerrum and Vigoda result [11] is an interesting adjunct to this because - at least for dismantleable H - it constitutes a positive "there exists..." result, which contrasts with the negative "there exists" result of [3].

results will be referred to again (in more detail) throughout the thesis. Just to clarify, their analysis is with respect to the "standard" H-colouring problem i.e. unweighted vertices and no restrictions on G.



Their most general observation is that H-colouring seems to span the (known) complexity hierarchy defined by AP-reducibility. That is, they identify some H for which $\#H$ is "easy" (i.e. $FPRAS$able), some H for which $\#H$ is "hard" (i.e. $\equiv_{\mathsf{AP}}\#SAT$) and some H of apparently "intermediate" complexity (i.e. $\equiv_{\mathsf{AP}}\#BIS$). (We have included a diagram to summarise the main H-colouring classifications from [8].) To elaborate, they note that for trivial H (i.e. complete bipartite graphs or complete fully-looped non-bipartite graphs) the problem $\#H$ is trivially $FPRAS$able. At the other end of the complexity hierarchy, they point out that, because the "Hell and Nešetřil" graphs all have NP-complete decision problems, $\#H\equiv_{\mathsf{AP}}\#SAT$ for such graphs H. They also prove that $\#IS$ is $\equiv_{\mathsf{AP}}\#SAT$. Focusing on more structured families of H, they prove that for all H from the family P_k^* $(k \geq 3)$ - i.e. paths on k vertices with every vertex looped - $\#H \equiv_{\mathsf{AP}} \#BIS$. (The case P_4^* corresponds to what in statistical physics is known as the *Beach* model.) Continuing with the theme of models from statistical physics, they *almost* fully classify the family of H graphs corresponding to the *Widom-Rowlinson* model. That is, they prove that *#1-WR* - which is equal to the fully looped, complete graph on two vertices - is (trivially) $FPRAS$able, *#2-WR*$\equiv_{\mathsf{AP}}\#BIS$, and *#k-WR*$\equiv_{\mathsf{AP}}\#SAT$ for $k \geq 4$. (So the complexity of *3-WR* is left open.) They do, however, manage to completely determine the complexity of the *wrenches*; *#0-wrench* and *#1-wrench* are both $\equiv_{\mathsf{AP}}\#SAT$, *#2-wrench*$\equiv_{\mathsf{AP}}\#BIS$ and *#k-wrench*$\equiv_{\mathsf{AP}}\#SAT$ (for $k \geq 3$.) Finally, for a number of H-colouring problems they are unable to completely classify (namely *3-WR* and the sequence of H-colouring problems equivalent to counting proper q-colourings in bipartite graphs, for $q \geq 3$) they prove that $\#BIS \leq_{\mathsf{AP}} \#H$ i.e. approximating $\#H$ is at least as hard as approximating $\#BIS$.

This thesis seeks to build upon the firm foundations laid by the above H-colouring research. **In short, we seek a greater understanding of how H-colouring problems span the AP-reducibility complexity hierarchy.** Broadly speaking, the work from [8] brings us to a point where we face two key challenges. Firstly, we have to consider a much wider range of H graphs and in the process group them into bands of similar complexity. Secondly, we have to take this information and use it to hone our understanding of how nuanced the complexity hierarchy itself is; the beauty of H-colouring (and one of the main reasons it elicits such attention) is that if there are tiers of complexity within the (infinite) set of H-colouring problems, it shows that the underlying complexity hierarchy is at least as tiered as this. (So the prospect of H-colouring helping us to develop classes other than $FPRAS$able problems, $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ is a distinct possibility.)

In the following Section, we give a brief overview of how we go about this. Before this, however, we recognise that, for the reader not familiar with the field of (approximate) counting and sampling, substantial parts of this chapter may have been quite difficult to follow. To this end, we have provided (in Appendix A.1) a much briefer, more informal "distillation" of this chapter, which will hopefully contextualise matters for the reader who has thus far been unable to grasp the overall thread.

1.4 Summary of thesis

This thesis comprises six core chapters followed by brief comments on possible directions for future research and, finally, appendices. In Chapter 2 (*The relative complexity of approximating $\#H$ where H has 4 or fewer vertices*) we expand our understanding of H-colouring by considering, on a graph-by-graph basis, the relative complexity of the 65 connected H with 4 (or fewer) vertices. With assistance from earlier research[10] we have completely classified 62 of the graphs, and established a complexity lower bound for the remaining 3. In addition to this catalogue, we use the chapter to introduce a large amount of basic terminology and reduction machinery, much of it related to the use of gadgetry. Also, rather than purely cataloguing individual graphs on an ad-hoc basis, we put forward a number of fairly simple classification lemmas (i.e. lemmas which establish a complexity result for a family of H rather than just an individual H) which, while not as sophisticated as the powerful lemmas from Chapters 4 and 5, establish a sound basis for the remainder of the thesis.

In Chapter 3 (*Approximately count - or approximately sample?*) we focus on the aforementioned complexity links between approximate counting and approximate sampling. In particular, we develop a sampling analogue to the approximation-preserving (AP) reduction called a *sampling-preserving* (SP) reduction. (As its name suggests, an SP-reduction from a sampling problem X to Y entails that if you can approximately sample solutions to Y you can approximately sample solutions to X.) We explore the relationship between the SP-reduction and the AP-reduction - noting that the former does not suffer from some of the apparently structural constraints of the latter - and also explain

[10] In particular, earlier work by DGGJ which completely classified those connected H on 3 or fewer vertices.

how these reductions tie in with current understanding of how the *absolute* complexity of approximately counting H-colourings relates to that of approximately sampling H-colourings.

In Chapter 4 (which, like the preceding chapter, contains results and definitions taken from [14], written by this author in conjunction with Goldberg and Paterson) we prove the significant result that, for all H with no trivial components, approximately sampling H-colourings is at least as hard as approximately sampling or counting independent sets in bipartite graphs. Given that there exists *some* complexity-theoretic evidence that the problem of approximately sampling or approximately counting bipartite independent sets[11] might be intractable, this translates as weak provisional evidence that efficiently approximately sampling H-colourings (for H with no non-trivial components) may not be possible.

Moving onto Chapter 5 (*The class* $\equiv_{AP}\#SAT$) we conduct an in-depth exploration of how H-colouring relates to $\equiv_{\mathsf{AP}}\#SAT$, the class of problems in $\#P$ that are hard to efficiently approximate. The backbone of this chapter comprises five general classification lemmas that enable us to sweep quite an extensive range of H-colouring problems into $\equiv_{\mathsf{AP}}\#SAT$. Additionally, we identify what seems to be a recurrent feature of graphs we know to be $\equiv_{\mathsf{AP}}\#SAT$, and use this information to explain why we do not think there exists a $\equiv_{\mathsf{AP}}\#SAT$ bipartite H. The question of approximately counting *partial* H-colourings is also tackled in this chapter, owing to the fact that this variant of H-colouring (which, in common with weighted H-colouring, can be expressed within our standard H-colouring framework) is particularly interesting in the case of H graphs already known to be $\equiv_{\mathsf{AP}}\#SAT$.

The bulk of this thesis assumes that H is connected. In Chapter 6 (*Disconnected* H) we consider some of the complexities that are introduced when H is disconnected.

[11] By the self-reducibility of the problem, approximately counting and approximately sampling bipartite independent sets is of equivalent complexity

Amidst a number of "utility" lemmas we show that the complexity of disconnected H is closely related to the complexity of any "exponentially dominant" components it has. Also, with the help of an observation by Jerrum, we provide a complete classification (with respect to AP-reducibility) for disconnected H on 4 (or fewer) vertices.

Finally, in Chapter 7 (*The complexity hierarchy*) we outline some evidence in support of our suspicion that, with respect to AP-reducibility, the classifications $FPRAS$able, $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ do not completely partition the domain of H-colouring problems. Indeed, while believing that the classes of $FPRAS$able, $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ problems are *probably* mutually distinct from a complexity point of view[12] we explain our conviction that there is a "complexity gap" between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ in which numerous H graphs lie. In addition, we point to a number of graphs that we think are likely candidates for lying in this gap, and communicate a number of reductions between unclassified H that pose some tantalising questions for future work.

Note

There is a technical glossary in the appendices, which we hope will be of some assistance to the reader. A considerable effort has been made to ensure that the thesis is easily readable without recourse to the glossary, but if necessary the glossary can be used to locate the point in the thesis where terms are first defined.

[12] albeit with some doubt hanging over whether $\#BIS$ might actually be $FPRAS$able

Chapter 2

The relative complexity of approximating $\#H$ where H has 4 or fewer vertices

2.1 Preliminaries

In the context of this thesis, this chapter serves two purposes. Firstly, it conveys what is known about the relative complexity of the approximate H-colouring counting problems for graphs H of size less than or equal to 4 vertices. We achieve this by recapping on existing results in this area (which pertain, mostly, to 2 and 3 vertex H), and then demonstrating new results that have emerged from this period of research. The second purpose of this chapter is to help the reader gain familiarity with the methodology of the more ubiquitous reduction techniques deployed throughout the body of this work. This, it is hoped, will instill a sense of familiarity and confidence in the reader which will be of benefit in grasping the more general, advanced results in subsequent chapters. Indeed, many of these later results rest on the ability to combine the simple reductions explored in this chapter in novel and interesting ways. It is worth stating explicitly that, in this context, many of the reductions and results in this chapter are *technically* made redundant by later chapters. Yet failing to demonstrate this early work would be misguided, as it would rob the reader of the opportunity to develop valuable insights into the underlying mechanics of the reduction technology. As a compromise, we have given every graph a number which corresponds to its position in the graph index provided in Appendix A.11. This graph index serves three purposes: it gives every connected H graph on 4 or fewer vertices a unique "handle", it helps convince the reader that we have exhaustively considered all 65 connected graphs on 4 or fewer vertices, and for each graph it pinpoints the location in the thesis where its complexity is originally determined, along with references to later classification lemmas that supersede its original, "ad-hoc" classification.

The chapter is set out as follows. In this section (Section 2.1 - *Preliminaries*) we lay the foundations for classifying H graphs. In particular, we state which classifications we get "for free"; that is, those graphs H which (by virtue of their structure) are already recognised as being either "easy" or "hard" to approximately count. Also in this section we define the ubiquitous "rounding" technique, a fundamental tool for building AP-reductions. In Section 2.2 (*H with 3 vertices*) we demonstrate revised and updated proofs for the class of connected 3-vertex H, based on informal proofs communicated to this author by DGGJ. This section serves as an opportunity for the reader to familiarise themself with many of the most common reduction techniques, concepts and gadgets. Section 2.3 (*4-vertex H for which $\#H$ is interreducible with $\equiv_{AP}\#SAT$*) is almost entirely original work by the author, and proves that most connected 4-vertex H are (in all likelihood) hard to approximately count. Section 2.4 (*4-vertex H interreducible with $\equiv_{AP}\#BIS$*) takes a detailed look at 4-vertex H which are interreducible with a class of apparently intermediate complexity, as well as expanding the classification to more generalised families of graphs. Section 2.5 (*4-vertex H not yet classified*) takes a brief look at the few remaining connected 4-vertex H that have yet to be classified; these are considered in more depth in Chapter 7.

Unless otherwise stated, one of the prevailing assumptions is that the input graph, G, is connected. This restriction does not affect the complexity of the problem, because the connected/disconnected versions of the problem are interreducible.

To see this, note first that the direction reducing from "G connected" to "G may be disconnected" is trivial. The other direction can be achieved, for an input graph G with k connected components $\{G_1, G_2, ..., G_k\}$, by making k separate oracle calls - one for each $\#H(G_i)$ - and then taking the overall product. It follows from the definition of the AP-reduction that using an accuracy of ϵ/k for each $\#H(G_i)$ call is sufficient to achieve ϵ accuracy overall.

The other principal assumption we adopt is that H is connected. From this point onwards, when a graph H is referred to but no mention is made of whether it is connected or disconnected, we assume it is connected. We have just seen that, for the input graph G, the connected/disconnected distinction is not so significant; however, the problem of approximately counting H-colourings for disconnected H is somewhat more difficult to reason about than when H is connected. This is why we dedicate the whole of Chapter 6 to the topic.

We encourage the reader to make a mental note of these assumptions before reading any further. Of the two assumptions, the first (that G is connected) is most robust and we rarely deviate from it. To repeat, where there is no mention otherwise, both H and G are assumed to be connected.

2.1.1 Trivial H

Quite a few of the H graphs with 4 or fewer vertices are *trivial* in the sense that a simple, polynomial-time exact counting algorithm has previously been identified. (Immediately we see that, since it is easy to solve the exact counting problem, such H are trivially $FPRAS$able.) As mentioned in the introduction, Dyer and Greenhill have fully mapped out the complexity of the exact H-colouring counting problem, for all H, in [10]. For connected H, the problem is polynomial-time solvable iff H is a complete bipartite graph or a complete, fully-looped non-bipartite graph, and $\#P$-complete otherwise. The polynomial time algorithms (assuming G is connected) are repeated here:

$H = K_{i,j}$. That is, H is the complete bipartite graph with i vertices on one side of the bipartition and j vertices on the other. If the input G graph is non-bipartite - a property that can be checked in polynomial time using depth-first search - the solution is zero, because by definition a bipartite H cannot colour a non-bipartite G. However, if G is bipartite and has vertex bipartition $(V_L(G), V_R(G))$, the exact solution is $i^{|V_L(G)|}j^{|V_R(G)|} + j^{|V_L(G)|}i^{|V_R(G)|}$. This term arises from the fact that there is a choice as to whether $V_L(G)$ is mapped to the i colours and $V_R(G)$ to the j colours, or vice-versa, and then the colouring is unrestrained.

$H = K_i^*$. That is, H is the complete, fully-looped graph on i vertices, otherwise known as a *looped i-clique*. The solution is $i^{|V(G)|}$ because all possible mappings from G to H are valid.

(By convention we treat the unlooped, single vertex as $K_{1,0}$. In this case, if G has only one vertex, then the solution is 1. For all other G, the solution is zero.)

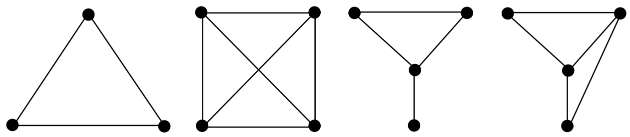
As we might expect, these exact solutions generalise easily to the problem when H is disconnected but *all* of its components are trivial:- simply add the individual solutions together. The full version of Dyer and Greenhill's result, taking into account connected and disconnected H, states that exactly counting H-colourings for a graph H is polynomial-time solvable iff *all* components of H are trivial, and $\#P$-complete otherwise.[1]

2.1.2 H that are already known to be "hard"

As discussed in the introduction, it transpires that when the decision problem for a particular H is difficult, so too is the approximate counting problem for that H. That is, when the decision problem is NP-complete, the approximate counting problem is (in the AP sense) interreducible with $\#SAT$. (This is true not just for H-colouring but also in general; it is Theorem 1 of [8].) This is a powerful observation since it pigeonholes a large number of H graphs as $\equiv_{\mathsf{AP}} \#SAT$ without any further effort. For example, in

[1]The Dyer and Greenhill result also applies when the maximum degree of G is bounded.

[15] Hell and Nešetřil established a famous dichotomy result in which they showed how the set of H-colouring decisison problems could be completely partitioned into P and NP-complete problems. They proved that, in the domain of connected[2] H, only the non-bipartite, unlooped H have NP-complete decision problems. For bipartite H and H with at least one loop the problem is trivially polynomial-time solvable. Recalling that a graph is bipartite iff it has no odd-length cycles, the following 3 and 4 vertex H are - by combining the result of Hell and Nešetřil with Theorem 1 of [8] - immediately seen to be $\equiv_{\mathsf{AP}}\#SAT$:



(From left-to-right, these graphs correspond to graphs 12, 61, 34 and 52 in the graph index.)

Section 2.1.1 and Section 2.1.2 have shown (respectively) that complete, fully-looped graphs and complete bipartite graphs are easy to exactly count[3], while non-bipartite, unlooped graphs are hard to approximately count. This thesis is therefore principally concerned with graphs that do not fit into either of these categories.

2.1.3 Independent sets, and a fundamental proof technique

This brings us to one of the most fundamental H-colouring problems. The 2-vertex H represented by K_2^* minus one loop corresponds exactly to the *independent set* problem. To see this, note that every H-colouring of G maps to a unique independent set of G, with the mapping being that a vertex is IN the independent set iff it is coloured with the unlooped colour from H. (Traditionally we label the looped colour b, as in "blue", and the unlooped colour r, as in "red".) This problem, $\#IS$, is $\equiv_{\mathsf{AP}}\#SAT$. However, though the independent set problem is often encountered in discussions of

[2]Their result also applies to disconnected H, on the basis that a disconnected H is deemed bipartite iff every component is bipartite, and a disconnected H is deemed to have at least one loop iff at least one of its components has at least one loop.

[3]Recall that in this thesis a bipartite graph is by definition unlooped

NP-completeness it is not, in its most general form, an NP-complete decision problem. This follows from the fact that the problem "does G have an independent set?" can be trivially answered "yes" simply by observing that every graph G has an independent set of size zero. Hence, we don't get $\#IS \equiv_{\mathsf{AP}} \#SAT$ for free. The problem does become NP-complete, however, when it is cast in the following form,

Name: $LargeIS$
Instance: A positive integer m and a graph G in which every independent set has size at most m.
Output: "Yes" if G contains a size-m independent set, "no" otherwise.

Hence, its counting counterpart $\#LargeIS$ is $\equiv_{\mathsf{AP}} \#SAT$:-

Name: $\#LargeIS$
Instance: A positive integer m and a graph G in which every independent set has size at most m.
Output: The number of size-m independent sets in G.

To show that $\#IS \equiv_{\mathsf{AP}} \#SAT$, therefore, we demonstrate a reduction from $\#LargeIS$ to $\#IS$. This was shown originally in [8], and that proof is the one given here.

Let m and $G = (V(G), E(G))$ be the input to $\#LargeIS$. Informally, what we are going to do is code up a graph G' (as input to $\#IS$) which "resembles" G, but in which each vertex of G is coded up (in G') as a *set* of vertices polynomially large in n. Thus, it becomes exponentially likely that each such vertex encoding in G' contains (where possible) both r and b vertices (rather than just b.) We argue that a vertex encoding coloured this way corresponds to a vertex from G that is IN the independent set, and consequently that "most" of $IS(G')$ is occupied by colourings corresponding to size-m independent sets. We will then use this fact to extract a good approximation to $\#LargeIS(G, m)$ from our approximation to $\#IS(G')$.

Here is the formal proof. We construct an instance $G' = (V(G'), E(G'))$ of $\#IS$ as follows, where k is a sufficiently large integer (to be specified later in the proof) and $[k] = \{0, 1, ..., k-1\}$.

$$V(G') = V(G) \times [k]$$

and

$$E(G') = \Bigg\{ \{(u,i),(v,j)\} : \{u,v\} \in E(G) \text{ and } i,j \in [k] \Bigg\}$$

Thus, each vertex in G is coded up as a size-k independent set, and an edge in G is coded up by the introduction of a complete bipartite graph between the corresponding size-k independent sets. To keep matters simple we will use the notation $I[u]$ to refer to the size-k independent set in G' corresponding to vertex $u \in V(G)$. Now, consider an IS colouring of G'. If all the vertices in $I[u]$ (for some vertex $u \in V(G)$) are coloured b then, in terms of how $I[u]$ behaves in relation to neighbouring $I[.]$ sets, it simulates the behaviour of a vertex OUT of the independent set. However, if $I[u]$ contains at least one r vertex then $I[u]$ can only be adjacent to other $I[.]$ sets that are coloured entirely b; in other words, it simulates the behaviour of a vertex IN the independent set. Hence, we can map colourings of G' to independent sets in G in the following way: a vertex $u \in V(G)$ is IN the independent set iff $I[u]$ contains at least one r vertex. It follows from this that an independent set in G with i vertices IN the independent set comes up $(2^k - 1)^i$ times as a colouring of G'. Let $\#IS_i(G)$ be the number of size-i independent sets in G. Hence, the following inequality holds:-

$$\#IS_m(G)(2^k-1)^m \leq \#IS(G') = \left(\sum_{0 \leq i < m} \#IS_i(G)(2^k-1)^i \right) + \#IS_m(G)(2^k-1)^m$$

The next stage exploits the fact that, if k is large enough, the contribution of size-m independent sets "outweighs" the contribution of all the other independent sets put together. Observe that an upper bound on the summation term above is $2^n(2^k-1)^{m-1}$, which we derive by assuming (generously) that every possible assignment of $\{r, b\}$ to $V(G)$ is a valid size $m-1$ independent set. Hence, if we divide the above inequality through by $(2^k-1)^m$ we obtain:

$$\#IS_m(G) \leq \frac{\#IS(G')}{(2^k-1)^m} \leq \#IS_m(G) + \frac{2^n(2^k-1)^{m-1}}{(2^k-1)^m}$$

Significantly, if $k \geq n+3$ then the term on the far right hand side drops (and stays) below $1/4$ for all n. Thus, if the $\#IS$ oracle was exact, we could obtain an exact solution to $\#IS_m(G) = \#LargeIS(G, m)$ by returning $\lfloor \#IS(G')/(2^k - 1)^m \rfloor$. However, we are dealing with approximation oracles, not exact oracles, and although the general concept remains the same there are a couple of minor technicalities to clear up. We briefly divert into a discussion of these technicalities before bringing the overall proof to a close.

More specifically, suppose we wish to use the above reduction to obtain an approximation to $\#LargeIS$ with accuracy ϵ. At some point we must decide what accuracy (δ) to use in the call to the $\#IS$ oracle, but putting that issue aside for one moment, we observe that the discontinuous nature of the floor function could cause problems when ϵ is small.

To avoid this we adopt the convention that, when developing our arguments in an exact counting framework (as in this example), the input to the floor function is restricted to being in the range $[N, N+1/4]$ where N is the final, integer answer. (That is, the input to the floor function has fractional part in the range $[0, 1/4]$.)

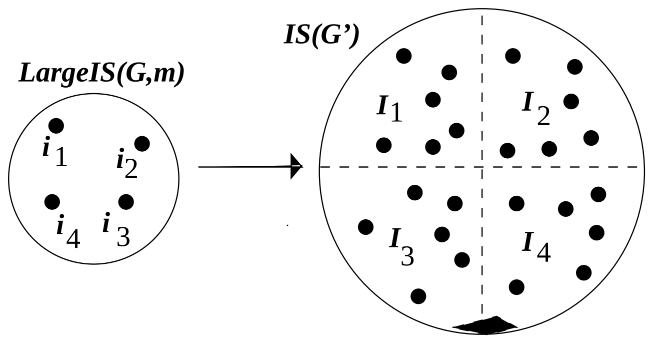
Suppose more generally that the true result N is obtained by *rounding* a fraction Q with $|Q - N| \leq 1/4$. Suppose further that the oracle provides an approximation $\hat{Q}$ to Q satisfying $Qe^{-\delta} \leq \hat{Q} \leq Qe^{\delta}$ (as it is required to do with probability at least 3/4.) Set $\delta = \epsilon/21$, where ϵ is the accuracy required of the final result. There are two cases. If $N \leq 2/\epsilon$, then a short calculation yields $|\hat{Q} - Q| < 1/4$ implying that the result returned is exact. If $N > 2/\epsilon$, then the result returned is in the range $[(N - 1/4)e^{-\delta} - 1/2, (N + 1/4)e^{\delta} + 1/2]$ which, for the chosen δ, is contained in $[Ne^{-\epsilon}, Ne^{\epsilon}]$. Either way we obtain sufficient accuracy.

Following this technical diversion, we now return to the main proof.

Having already shown that (in the exact counting world) the argument of the floor function, $\frac{\#IS(G')}{(2^k-1)^m}$ is in the range $[\#IS_m(G), \#IS_m(G)+1/4]$, the $\#LargeIS \leq_{\mathsf{AP}} \#IS$

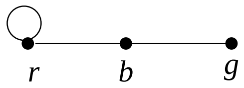
reduction is completed by using accuracy $\epsilon/21$ in the call to the $\#IS$ oracle, dividing the result by $(2^k - 1)^m$, and then rounding. $\square$

In the above proof we approximated $\#LargeIS(G, m)$ by constructing a graph G' such that each element of $LargeIS(G, m)$ corresponded to exponentially many elements in $IS(G')$. This idea of "boosting" desirable elements so that they completely dominate over all other types of elements (in the solution space of the problem we are reducing to) is fundamental to this thesis and is used time and time again. The following diagram clarifies (with respect to the above proof) the technique. (Each element of $LargeIS(G, m)$ comes up very many times as colourings of $IS(G')$. An insignificantly small fraction of the $IS(G')$ solution space, represented by the filled black area at the base of the $IS(G')$ figure, comprises colourings of $IS(G')$ that do not correspond to size-m independent sets in G.)



2.2 H with 3 vertices

Informal work by DGGJ produced a complete classification for the connected, 3-vertex H graphs that are neither trivial nor which are immediately $\equiv_{\mathsf{AP}}\#SAT$ by virtue of their decision counterpart being NP-complete. We reproduce their results here, although in many cases the proofs have been re-engineered and expanded (as original work) to bring them into synchronisation with the 4-vertex proofs that follow. On a point of clarification, the convention adopted throughout this thesis is that the proof pertaining to a particular graph is in the text *after* the graph's picture.



(Graph 8) Status: $\equiv_{\mathsf{AP}}\#SAT$. Assuming that the input graph to $\#IS$ is G, and is connected, we proceed (as with most proofs) by constructing a new graph G' that is used as input in a single call to the $\#H$ oracle. Informally, the idea is to build G' in such a way that "most" of the solution space, $H(G')$, corresponds to independent sets, thus proving that this graph is $\equiv_{\mathsf{AP}}\#SAT$. This will be formalised in due course, but first we describe the construction of G'. The vertex set of G' comprises a copy of $V(G)$ (which we will just refer to as $V(G)$ for brevity), a new vertex x, and a set K containing k disjoint copies of the 2-vertex graph K_2, where the value of k will be discussed a little further on. Thus,

$$V(G') = V(G) \cup \{x\} \cup \bigcup_{i \in [k]} \{p_i, q_i\}$$

The edge set of G' is

$$E(G') = E(G) \cup \bigcup_{u \in V(G)} \Big\{\{x, u\}\Big\} \cup \bigcup_{i \in [k]} \Big\{\{p_i, q_i\}\Big\} \cup \bigcup_{i \in [k]} \Big\{\{x, p_i\}, \{x, q_i\}\Big\}$$

This formal definition of G' provided is somewhat unwieldy so it is useful to study the symbolic representation of G' provided in Figure 2.1; such diagrams are a great help in understanding what are very gadget-heavy, "graphical" reductions. By way of summary, G' is built from G by adding a "nexus" vertex x and connecting x to every vertex in G. In addition, a large set K of disjoint copies of K_2 is introduced, and x is connected to all $2k$ vertices in K also. (The structure K is called a K_2-*cliqueset* gadget[4].)

We now argue that, as long as k is sufficiently large with respect to n (where $n = |V(G)|$), "most" of the colourings in $H(G')$ are such that x is coloured r. First, we demonstrate why this is the case, and then argue the significance of this fact. The initial point to note is that when x is coloured with a colour c, colourings in both K and G are restricted to the subgraph of H induced by the neighbours of c. Thus, if x is coloured r, colourings in K are restricted to the graph with vertex set $\{r, b\}$ and edge set $\{\{r, r\}, \{r, b\}\}$. Likewise, if x is coloured b, K is restricted to the graph with vertex

[4]Naturally, a K_i-*cliqueset* is where copies of K_i are used in K rather than copies of K_2.

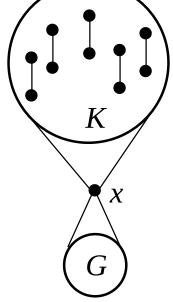


Figure 2.1: *The K_2-cliqueset in action*

set $\{r, g\}$ and edge set $\{\{r, r\}\}$. Finally, if x is coloured g the restriction has vertex set $\{b\}$ and edge set $\emptyset$. For shorthand we refer to these restricted subgraphs as $H[r], H[b]$ and $H[g]$, and often describe (for example) $H[r]$ as "the graph pointed out by r". Now, if K is appropriately large with respect to n, it follows that the colour c most likely to be used to colour x will be the one that maximises the number of $H[c]$ colourings of K. We now show that, for the chosen K, r is indeed that colour. To see this, observe that when x is r each $\{p_i, q_i\}$ in K can independently take one of 3 colourings: (r, r), (r, b), or (b, r). When x is coloured b only one colouring is possible per $\{p_i, q_i\}$, (r, r), and it is not difficult to see that it is in fact impossible for x to be coloured g when $k > 0$. Hence, as long as k is large enough that the number of colourings in the G part of G' can (to all intents and purposes) be ignored, we see that the number of colourings of G' with x coloured r consume an exponentially large fraction of the solution space $H(G')$, because $1^k/3^k$ is exponentially small in k. Furthermore, we see that whenever x is coloured r the colourings allowed in G are precisely independent sets, so in fact each independent set of G occurs 3^k times as a colouring of G'.[5] It follows that the result of the approximation oracle call, $\#\hat{H}(G')$, is *roughly* equal to $\#IS(G)3^k$ plus some exponentially small contribution from colourings where x does not get coloured r. Dividing $\#\hat{H}(G')$ by 3^k and rounding should, therefore, give a very close approximation to $\#IS(G)$. To operationalise this we use the reduction technique described in Section

[5] To see this, consider that if we fix an independent set colouring of G, and fix x as r, we can colour K any of 3^k ways.

2.1.3, which requires that (for n larger than some threshold constant) it must hold that:

$$\#IS(G) \leq \frac{\#H(G')}{3^k} \leq \#IS(G) + \frac{1}{4}$$

If we adopt the notation $H(G|P)$ to mean "the set of colourings of G (under H) for which predicate P holds", and understand $u \to c$ to mean "if u is coloured c", it follows that:

$$\#H(G') = \#H(G'|x \to r) + \#H(G'|x \to b) + \#H(G'|x \to g)$$

$\#H(G'|x \to g)$ is zero, and a pessimistically high upper bound on $\#H(G'|x \to b)$ is $1^k|V(H)|^n$, which we derive by assuming a b colouring of x places no constraints on the colourings possible in G. We already know $\#H(G'|x \to r)$ is $\#IS(G)3^k$. Clearly then, the LHS inequality in the above inequality is trivially satisfied, since we know $\#IS(G) \geq 1$ in all cases. To satisfy the RHS, then, we need to choose k such that:

$$\frac{1^k|V(H)|^n}{3^k} \leq 1/4$$

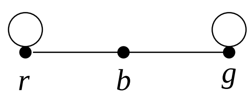
A simple rearrangement of this inequality suggests that, as long as we set k such that

$$k \geq \left\lceil \frac{n \ln(|V(H)|) + \ln(4)}{\ln(3)} \right\rceil$$

then we are done. (The use of the ceiling function here is simply to account for the fact that k must be non-fractional.) Note that the reduction wouldn't be valid if k was exponentially large in n, but a quick inspection of the above inequality shows that k is no bigger than linear in n, which is fine, and this completes the reduction. □

It is worth making a few observations about this reduction at this point, since it is highly typical of others in this chapter. Firstly, such is the ubiquity with which the concept "subgraph induced by mutual neighbours" arises that we generalise the notation used in this reduction (e.g $H[r]$) so that it can also apply to subsets of $V(H)$, e.g. $H[S]$ for some $S \subseteq V(H)$. The definition is as we might expect: $H[S]$ only contains vertices of H that are adjacent to every vertex in S. ($H[S]$ is uniquely defined because, naturally, we insist that *all* colours which can be adjacent to every colour in S, appear in $H[S]$.) Secondly, it is not problematic to be pessimistically generous when calculating upper bounds on the contribution of "small" quantities (such as $\#H(G'|x \to b)$ in this instance.) Indeed, as long as the size of the most significant gadget is big enough, the contributions of colourings from other, smaller parts of the input space (primarily G in this reduction) cannot accumulate to the extent that the most significant gadget is no longer the chief determinant of which types of colouring dominate the solution space. Usually we can be less rigorous than we were here in choosing the size of the most significant gadget; a choice of $k = n^2$ would be perfectly adequate, for example, because the only competition to colourings in K are colourings of G, and (in this instance) these are bound above by a linear exponent in n. Indeed, as will become apparent later, this generalises nicely to arrangements of two or more gadgets, allowing us to refine the colourings they pick out by "stepping" the gadgets progressively down in size.

It is worth pointing out that, had copies of K_1 (i.e. single vertices) been used in K rather than copies of K_2, this would have had the property of making x exponentially likely to be coloured with maximum degree colours from H. Given that, for the H just discussed, both r and b are maximum degree colours, this would have been insufficient to isolate r. We might, however, have argued that because $H[b]$ can only colour a connected G (with $|V(G)| > 1$) in 1 way, $H[r]$ still "wins" overall. There is some validity to this argument, but we choose to steer away from it because it introduces complications into the reduction. Moreover, it is generally better to avoid "doubling-up" encodings of the input space (the G part of G' in this case) as boosting gadgetry. This is because, whereas we can reason about the behaviour of gadgetry we elect to add (such as K) *independently* of the input G, that independence is lost if we start relying on a particular property of the input G.



(Graph 10) Status: $\equiv_{\mathsf{AP}} \#SAT$. This graph is $\equiv_{\mathsf{AP}} \#SAT$, and this can be established with a reduction virtually identical to the previous one. The only difference in the construction of G' is that k needs to be ever so slightly larger, although if the "lazy" option is pursued and we choose $k = n^2$ this is adequate for both reductions. Note that in this instance both $H[r]$ and $H[g]$ can colour K in 3^k ways, and $H[b]$ shows a slight (but still inadequate to match 3^k) improvement on the previous reduction, being able to colour K in 2^k ways: each $\{p_i, q_i\}$ can take either the colouring (r,r) or (g,g). Hence, x is exponentially likely to be r or g, but both point out independent set colourings in G so this isn't a problem, and can be adjusted for by dividing $\#\hat{H}(G')$ by $3^k 2$, rather than 3^k. In other words, we know:

$$2\#IS(G)3^k \leq \#H(G') \leq 2\#IS(G)3^k + |V(H)|^n 2^k$$

So, to use the "rounding" technique, the inequality we must satisfy this time is:

$$\frac{2^k |V(H)|^n}{3^k 2} \leq \frac{1}{4}$$

A quick re-arrangement shows that

$$k \geq \left\lceil \frac{n \ln(|V(H)|) + \ln(2)}{\ln(3/2)} \right\rceil$$

is adequate. □

Both the graphs in Figure 2.2 are $\equiv_{\mathsf{AP}} \#SAT$, but rather than prove each one in-

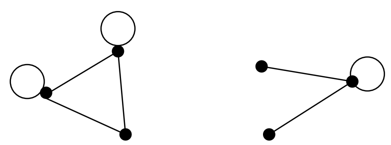


Figure 2.2: *Two examples of weighted versions of the independent set problem, shown in expanded form. (Graphs 14 and 7 respectively.)*

dividually we show that they can both be captured by a slightly more general result, Lemma 2.3, which we demonstrate shortly. First, we must introduce the concept of *weighting*.

Letting $\mathbb{N}^+$ denote $\mathbb{N} \setminus \{0\}$, we say H' is a weighted version of H if, for some weight function $w : V(H) \to \mathbb{N}^+$, H' is isomorphic to the graph H'' defined as follows:-

$$V(H'') = \bigcup_{\substack{d \in V(H) \\ 0 \le i < w(d)}} \{d_i\} \qquad\qquad E(H'') = \bigcup_{\substack{0 \le j < w(e) \\ 0 \le i < w(d) \\ \{d,e\} \in E(H)}} \Big\{ \{d_i, e_j\} \Big\}$$

To clarify this, note that the graph on the left in Figure 2.2 (which has graph number 14 and which we sometimes call the *ear* graph) is the weighted version of the independent set problem with weight 2 on the looped colour. The graph on the right of Figure 2.2 (which has graph number 7 and is often called the *compass* graph) is also a weighted version of the independent set problem, but this time with weight 2 on the unlooped colour. There is a natural relationship between H colourings and H' colourings, expressed by:

$$\#H'(G) = \sum_{Col \in H(G)} \prod_{u \in V(G)} w(Col(u)) \tag{2.1}$$

2.2.1 Compact form, expanded form, identity, equivalence, indistinguishability

Though weighting is a very natural concept, it is important to be absolutely clear about the notational complexities it can introduce. Firstly, there is the issue of whether a graph H is presented in *compact* or *expanded* form. For example, the *expanded* form of the graph on the left in Figure 2.2 would (modulo relabelling) be $H = (V(H), E(H))$ where $V(H) = \{r, b, g\}$ and $E(H) = \{\{r,r\}, \{b,b\}, \{r,b\}, \{r,g\}, \{b,g\}\}$. The *compact* form would be $H = (V(H), E(H), w)$ where $V(H) = \{r, b\}$, $E(H) = \{\{b,b\}, \{r,b\}\}$, $w(r) = 1$, $w(b) = 2$. In other words, the compact form of a graph is derived from the expanded form by maximally collapsing *equivalent* (see below) colours into one representative equivalence class, and attaching a weight equal to the size of that equivalence class. Hence, when represented in compact form colours of H are pairwise non-equivalent in $E(H)$.[6]

Also, when drawn in compact form, a graph has its weights denoted in square brackets [ ], as shown in Figure 2.3.

[6] The compact form of a graph (presented initially in expanded form) is uniquely defined. We provide a short motivation for this in Appendix A.2.

If a graph H is presented in expanded form, two colours c and d from $V(H)$ are *identical* iff $c = d$. In other words, identity is a labelling-level property. What does it mean for two colours to be *equivalent*? Two colours c and d from a graph H (in its expanded form) are equivalent iff their adjacency set is identical[7]. Equivalence is therefore an extremely strong property. Indistinguishability is a similar but slightly weaker property. Two colours c and d in a graph H are *indistinguishable* iff H has an automorphism f such that $f(c) = d$. Equivalence implies indistinguishability, but the opposite way round does not necessarily hold. For example, in the graph $V(H) = \{r, b, g\}$, $E(H) = \{\{b, b\}, \{b, r\}, \{b, g\}, \{r, g\}\}$, r and g are indistinguishable but not equivalent, because $adj(r) = \{b, g\}$ and $adj(g) = \{b, r\}$. The main point to note is that if two colours c, d are either equivalent or indistinguishable, there is no point in trying to distinguish between them using gadgetry, because from that perspective they are "the same".

Unless specified otherwise, all graphs H are assumed to be given in expanded form. However, in some situations it is preferable to represent a graph in compact form. In such instances we state this explicitly and provide the relevant weight function w. Note that, when H is given in compact form, we let the sets $V'(H)$ and $E'(H)$ denote the vertex and edge sets of H in expanded form. This is often useful because from a mathematical point of view the "total number of colours" is $|V'(H)|$, not $|V(H)|$. We can derive the former from the latter by noting that

$$|V'(H)| = \sum_{c \in V(H)} w(c) \tag{2.2}$$

For a graph H in compact form, it is also helpful to define $deg'(c)$ and $adj'(c)$ for $c \in V(H)$. We define $deg'(c) = |adj'(c)|$, where $adj'(c)$ is the set of colours in $V'(H)$ adjacent to c_i, where $c_i \in V'(H)$ is any colour from the equivalence class c represents[8]. It follows therefore that $deg'(c) = |adj'(c)| = \sum_{d \in adj(c)} w(d)$. We call $deg'(c)$ the *effective* degree of c; this quantity is useful when constructing counting equations and so on. (It is useful to note that the effective degree of c is always equal to $deg(c_i)$,

[7] The definition of identity has the expected extension to sets of colours.

[8] It does not matter which colour from the equivalence class is considered; they all have the same adjacency set.

where c_i is any colour in $V'(H)$ from the equivalence class c represents.)

We now demonstrate some observations pertaining to weighting.

Lemma 2.1 *If the graph H' is obtained from H by multiplying all the weights on the colours of H by a constant $W > 0$, $\#H' \equiv_{\mathsf{AP}} \#H$.*

Proof. Firstly, note that it is of no consequence whether H is first written out in its expanded form, and *then* has weight W attached to each colour, or if instead H is first written in its compact form and then has each colour weight multiplied by W: the operations are identical, and as a result both yield the same graph H'. Now, observe that in such an instance:

$$\#H'(G) = W^n \#H(G) \tag{2.3}$$

This can be derived from (2.1) (for example) by setting $w(c) = W$ for all $c \in V(H)$. $\#H'(G)$ is therefore interreducible with $\#H(G)$, because one can be obtained from the other by multiplying or dividing by W^n, as required. Note that, owing to the multiplicative definition of accuracy used in AP-reductions, it is fine to use ϵ (the overall desired accuracy) for the oracle call also, and in addition there is no need to make use of the "rounding" technique. □

Observation 2.2

The domain of w is restricted to $\mathbb{N}^+$ for purely "aesthetic" reasons:- it is difficult to graphically realise a graph with non-integer weights on its colours. Nonetheless, there is no algebraic reason why the domain of w should not be $\mathbb{Q}^+$, where $\mathbb{Q}^+$ is the set of positive, non-zero rationals. Conveniently, Lemma 2.1 allows us to extend the domain of w to $\mathbb{Q}^+$ in a very simple and elegant way. To see this, suppose (for example) H is a 3-vertex graph with colour weights $x, y, z \in \mathbb{Q}^+$. By the definition of rational numbers, these weights can be re-written as $\frac{x_1}{x_2}, \frac{y_1}{y_2}, \frac{z_1}{z_2}$. Now, if we apply Lemma 2.1 and multiply each weight by $x_2 y_2 z_2$, we produce a graph H' such that all the weights on H' are in $\mathbb{N}^+$ and $\#H' \equiv_{\mathsf{AP}} \#H$. The most immediate consequence of this is that we can always assume that a weighted graph draws its weights from $\mathbb{N}^+$, because even if it doesn't it is easy to find another (with which it is interreducible) which does. □

Now, we show that, for all H such that H is a weighted version of the independent set problem, $\#H \equiv_{\mathsf{AP}} \#SAT$. We show this by making minor modifications to the $\#LargeIS \leq_{\mathsf{AP}} \#IS$ reduction in Section 2.1.

Lemma 2.3 *All weighted versions of the independent set problem are $\equiv_{AP} \#SAT$*

Proof. (This result is subsumed by Lemma 5.1.) We assume H is given in compact form, so let $w_b = w(b)$ and $w_r = w(r)$, where b is the looped colour and r is the unlooped colour.

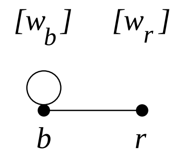


Figure 2.3: *The weighted version of the independent set problem, with weight w_b on b and weight w_r on r, in compact form*

The $\#LargeIS \leq_{\mathsf{AP}} \#IS$ reduction mentioned earlier in Section 2.1.3 can be easily adapted to cater for these weights. We build G' in the same fashion, with a slightly different value of k which we will now derive. As before, a vertex $u \in V(G)$ is IN the independent set iff $I[u]$ contains at least one vertex that is coloured r. Taking weights into consideration, it follows that there are now w_b^k ways of colouring a particular $I[u]$ with just b (as opposed to 1 way previously), and $(w_b + w_r)^k - w_b^k$ ways of colouring it with at least one r (as opposed to $2^k - 1$ ways previously.) Hence, an independent set of G containing i vertices comes up

$$((w_b + w_r)^k - w_b^k)^i w_b^{k(n-i)}$$

times as a colouring of G'. We argue, as before, that the colourings in G' that correspond to maximum (m) size independent sets of G consume an exponentially large proportion of the $H(G')$ solution space. This holds because the contribution of a vertex IN the independent set, $(w_b + w_r)^k - w_b^k$, is exponentially larger than that of an OUT vertex, w_b^k. (To see this note that for k beyond a fixed constant threshold, $(1/2)(w_b +$

$w_r)^k \leq (w_b+w_r)^k - w_b^k$, whilst $(1/2)(w_b+w_r)^k$ nonetheless exponentially outstrips w_b^k.) Thus, we seek to obtain our approximation to $\#LargeIS(G,m)$ by dividing $\#\hat{H}(G')$ by $((w_b+w_r)^k - w_b^k)^m w_b^{k(n-m)}$ and rounding. As is standard, this requires us to show that the number of colourings of G' pertaining to independent sets smaller than m is "small". An upper bound on this number is gleaned if we assume that every such colouring maps to a size $m-1$ independent set, and that every possible assignment of IN/OUT to the vertices of G is valid and a size $m-1$ set. Thus, we have to choose k such that:

$$\frac{2^n((w_b+w_r)^k - w_b^k)^{m-1} w_b^{k(n-m+1)}}{((w_b+w_r)^k - w_b^k)^m w_b^{k(n-m)}} \leq 1/4$$

If we cancel the LHS down, and (using the earlier observation) note that for sufficiently large k the following holds true,

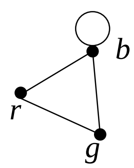
$$\frac{2^n w_b^k}{(w_b+w_r)^k - w_b^k} \leq \frac{2^n w_b^k}{(1/2)(w_b+w_r)^k} \tag{2.4}$$

we can satisfy the initial inequality by taking:

$$k \geq \left\lceil \frac{(n+1)\ln(2) + \ln(4)}{\ln((w_b+w_r)/w_b)} \right\rceil$$

(To be formally precise, we should ensure that k is large enough for (2.4) to hold in the first place, but this is easily achieved by taking k to be sufficiently large to satisfy both inequalities.) $\square$

Lemma 2.3 can thus be used to prove that the graphs in Figure 2.2 are $\equiv_{\mathsf{AP}}\#SAT$.



(Graph 13) Status: $\equiv_{\mathsf{AP}}\#SAT$. This graph, occasionally referred to as the *pyramid*, is also $\equiv_{\mathsf{AP}}\#SAT$. We again demonstrate this with a $\#IS\leq_{\mathsf{AP}}\#H$ reduction, but one different to those used thus far. This time, the vertices of G' include $V(G)$ with the addition of two disjoint, size-k sets of vertices, I_1 and I_2, where k is determined later. The edge set of G' is $E(G)$ plus edges connecting every vertex in I_1 to every vertex in I_2, and every vertex in I_2 to every vertex in $V(G)$. Figure 2.4 clarifies this. (We are going to show how I_2 is exponentially likely to be coloured either $\{b, g\}$ or $\{b, r\}$, pointing out independent set colourings in G.) Note that colourings of G' can be partitioned into

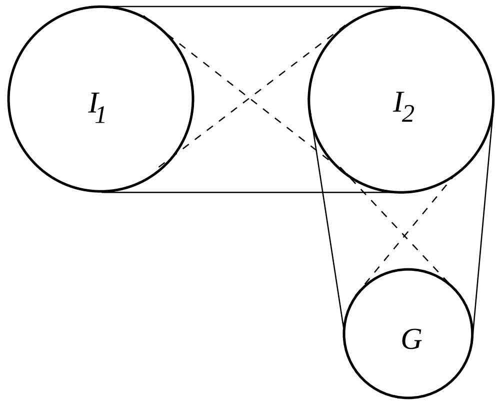


Figure 2.4: *The construction used in the proof for graph 13.*

equivalence classes, on the basis of the *sets* of colours used to colour I_1 and I_2. For example, the subgraph of H pointed out in the G part of G' is dependent not so much on the exact colouring of I_2 but more generally the *set* of colours used to colour I_2. Thus, we can partition colourings in $H(G')$ into the following categories, where for a colouring $Col \in H(G')$ and $X \subseteq V(G')$, $Map(Col, X)$ is defined as $\{Col(u) | u \in X\}$.

$Map(Col, I_1)$	$Map(Col, I_2)$	*Colours allowed in G*
$\{b\}$	$\{r, b, g\}$	$\{b\}$
$\{r, b, g\}$	$\{b\}$	$\{r, b, g\}$
$\{r, b\}$	$\{b, g\}$	$\{r, b\}$
$\{g, b\}$	$\{b, r\}$	$\{g, b\}$

Even though there are more possibilities than the 4 listed above we have deliberately omitted those where $Map(Col, I_1)$ is a *proper* subset of the colours in $H[Map(Col, I_2)]$, or vice-versa. The motivation for this is that such colourings "waste the opportunity" to use all the colours at their disposal; for example, if $Map(Col, I_1) = \{r, b\}$ and $Map(Col, I_2) = \{b\}$ then we could have safely coloured I_1 with $\{r, b, g\}$ as this would have made an extra colour available but not changed the type of colourings pointed out in G. It is usually a fairly trivial exercise to show that such "wasteful" colourings occupy a vanishing fraction of the total solution space:- intuitively, if they are exponentially small compared to a less wasteful counterpart, they are clearly exponentially small as a fraction of the whole solution space.

It is helpful at this point to explain the meaning of the term *configuration*. A configuration is, essentially, a predicate on a set of colourings. For example, each of the 4 entries in the above table is a distinct configuration; the predicate characterising the top entry would be "I_1 is coloured $\{b\}$ and I_2 is coloured $\{r, b, g\}$". Thus, "the number of times a configuration comes up" is simply the number of colourings that satisfy the relevant predicate. We introduce this terminology because it is often helpful, as in the table above, to group together colourings which share some common property:- generally we look to group colourings together that "behave" the same.

So, returning to the configurations as listed above, we now wish to determine which, if any, occupy an exponentially large fraction of the solution space. The first configuration in the above example comes up *at most* $1^k 3^k 1^n$ times. We say *at most* because, if we were being exact, we would write that it comes up $1^k \nu(k, 3) 1^n$ times, where $\nu(a, b)$ is the number of "onto" (i.e. surjective) functions from a set of size a to a set of size b. We will come across $\nu(a, b)$ numerous times throughout this thesis, but for the purpose of this proof it is sufficient to note that b^a is both very close to $\nu(a, b)$ and an upper bound of it. Hence $1^k 3^k 1^n$ is in fact a fairly tight upper bound on the number of times this configuration comes up. (We are satisfied with an upper bound measurement here because we will shortly demonstrate that this configuration and the next are exponentially dominated over by the latter two.)

The second colouring comes up *at most* $3^k 1^k 3^n$ times which we derive from the generous assumption that any colourings are valid inside G. The third and fourth both come up exactly $(2^k - 2)(2^k - 2)\#IS(G)$ times, because when I_2 is coloured with either $\{r, b\}$ or $\{b, g\}$ the subgraph of H pointed out in G is isomorphic to the independent set problem. (Note that $2^k - 2$ is the exact value of $\nu(k, 2)$.) The idea, then, is that if k is sufficiently big compared to n we can ignore the contribution of the terms exponential in n, and focus solely on the terms exponential in k. Then we argue that, because $(2^k - 2)(2^k - 2) \approx 4^k$, and $4 > 3$, the bottom two configurations are maximal i.e they are exponentially dominant over the first two configurations, which only come up approximately 3^k times.

So to proceed we need to show that dividing $\#\hat{H(G')}$ by $2(2^k - 2)(2^k - 2)$ and rounding gives us an adequate approximation of $\#IS(G)$. Thus we need to show that the number of colourings pertaining to the first, second and wasteful configurations combined is less than a quarter of $2(2^k - 2)(2^k - 2)$. With regard to wasteful colourings, we have made our life easy in this instance because using 3^k rather than $\nu(k, 3)$ in deriving upper bounds (for the first two configurations) means wasteful configurations have already been counted once, bar one minor tweak. The tweak in question involves loosening the upper bound for the first configuration from 3^k to $3^k 3^n$, because a wasteful configuration that is a subset of the first configuration may point out more than just b in G.

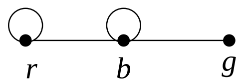
Hence, we need to choose k such that (for n larger than some fixed constant):

$$\frac{3^k 3^n + 3^k 3^n}{2(2^k - 2)(2^k - 2)} \leq 1/4$$

We could prove a tight lower bound for k in the manner displayed in other proofs, but for simplicity we state that $k = n^2$ is more than adequate and this can be easily checked. $\square$

Before continuing, it is helpful to make some observations about the above reduction. Owing to their dominance over the other configurations, the third and fourth configurations are what we generally call *maximal configurations*. The vast majority of proofs in this thesis rely (either explicitly or implicitly) on the construction of gadgets that have maximal configurations with desirable properties, as in this case. Note that a wasteful configuration is by definition not maximal, but a configuration that is not maximal is not necessarily wasteful; for example, neither the first or second configuration is maximal, but neither are they wasteful. On that note, the reader may be wondering whether, beyond its intuitive meaning, the term wasteful has a more formal definition.

Its exact meaning differs from gadget to gadget, but generally speaking wasteful configurations are those which we can automatically disregard because we could have easily "upgraded" the configuration to some exponentially superior configuration. So, the term is really only a mechanism for shortening proofs. However we define it, it is important that - where it is not obvious - we provide evidence that all non-wasteful configurations have been considered. In this case it is easy; consider that, if $C_2 = Map(Col, I_2)$, we have to use all the colours from $H[C_2]$ in I_1 because otherwise the configuration is immediately exponentially inferior. As a result we can focus on C_2. The table considers all subsets of $V(H)$ in I_2 apart from $\{r, g\}$. This is wasteful because the mutual neighbour of $\{r, g\}$ is $\{b\}$, yet the mutual neighbours of b are $\{r, b, g\}$ - so we may as well have coloured I_2 with $\{r, b, g\}$.



(Graph 9) Status: $\equiv_{\mathsf{AP}}\#SAT$. This graph, known as the *wrench* (or, more generally, the *1-wrench*) was shown to be $\equiv_{\mathsf{AP}}\#SAT$ in [8], using a reduction from the problem $\#LargeCut$:

Name: $\#LargeCut$
Instance: A positive integer m and a connected graph G in which every cut has size at most m.
Output: The number of size-m cuts of G

A *cut* of a graph G is a partition of $V(G)$ into two sets, and the size of the cut is defined to be the number of edges that cross the partition divide. In [8] it is observed that $\#LargeCut \equiv_{\mathsf{AP}} \#SAT$ because the decision problem $LargeCut$ is NP-complete. We now provide a version of the reduction provided in [8]. (This proof is quite long because it introduces a number of new concepts. The proof concludes on page 53.)

The reduction proceeds by constructing a graph G' as follows. $V(G')$ comprises:

1. For every vertex $u \in V(G)$, introduce 4 sets of vertices, $A[u], B[u], B'[u]$ and

$A'[u]$. $A[u], A'[u]$ are both of size p, to be determined, and $B[u], B'[u]$ are both of size q, also to be determined.

2. For every edge $\{u, v\} \in E(G)$, introduce disjoint sets $S[uv], S'[uv]$. $S[uv]$ and $S'[uv]$ are both of size t, to be determined.

The edge set of $V(G')$ is:

1. For every vertex $u \in V(G)$, connect every vertex in $A[u]$ to every vertex in $B[u]$, and connect every vertex in $A'[u]$ to every vertex in $B'[u]$. Add q edges so as to introduce a perfect matching between the sets $B[u]$ and $B'[u]$. (That is, make sure each vertex in $B[u]$ is connected to exactly one counterpart in $B'[u]$, and vice-versa.)

2. For every edge $\{u, v\} \in E(G)$, connect every vertex in $B[u]$ to every vertex in $S'[uv]$. Connect every vertex in $B'[u]$ to every vertex in $S[uv]$. Connect every vertex in $S'[uv]$ to every vertex in $B'[v]$, and every vertex in $S[uv]$ to every vertex in $B[v]$.

Thus, Figure 2.5 symbolises the encoding of vertices u and v in $V(G)$, and the $S[uv], S'[uv]$ part symbolises the encoding of the edge $\{u, v\}$. As in the previous proof we now look

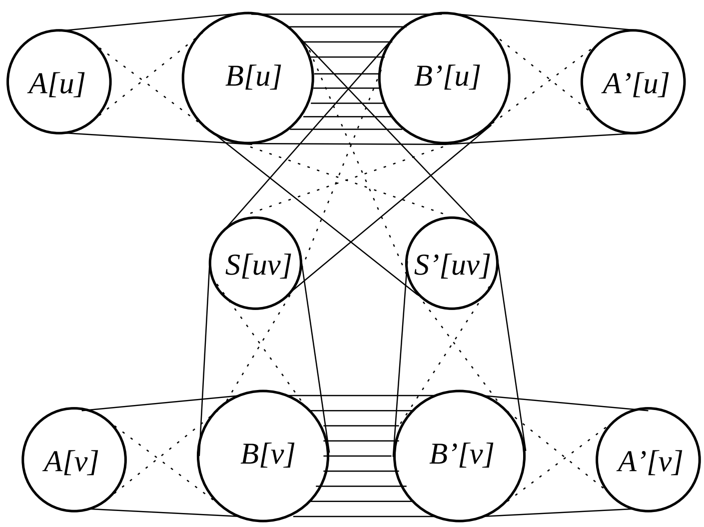


Figure 2.5: *Vertex and edge encodings for the $\#LargeCut$ reduction*

at the different configurations of colours possible in G'. The value t will be set significantly smaller than both p and q so in trying to work out which colourings dominate it is reasonable to ignore (up to a point) the S and S' sets. Note that, because of the slightly unusual means of connecting $B[u]$ and $B'[u]$ we in fact count the number of different *coloured edges* that span these two sets.

For example, suppose the exact set of edges (from H) used to colour the edges between $B[u]$ and $B'[u]$ is $\{(r,r),(r,b),(b,r),(b,b),(b,g)\}$. This can happen in $\nu(q,5) \approx 5^q$ ways. Observe that the set of edges from H appearing between $B[u]$ and $B'[u]$ dictate the set of colours that appear in $B[u]$ and $B'[u]$; in this example $B[u]$ would be coloured with $\{r,b\}$ and $B'[u]$ would be coloured with $\{r,b,g\}$. However, it is technically incorrect to claim the converse i.e. that the colours appearing in $B[u]$ and $B'[u]$ dictate the set of edges appearing between $B[u]$ and $B'[u]$. For example, we could colour $B[u]$ with $\{r,b\}$ and $B'[u]$ $\{r,b,g\}$ but only using the edges $\{(r,r),(b,r),(b,b),(b,g)\}$. Hence, to avoid confusion, we introduce ":" notation: what we mean by "$B[u] : B'[u]$ is coloured $X : Y$" (for $X, Y \subseteq V(H)$) is that all the edges with left endpoint in X and right endpoint in Y occur at least once on the edges between $B[u]$ and $B'[u]$, that the set of colours appearing in $B[u]$ is exactly X and that the set of colours appearing in $B'[u]$ is exactly Y. Thus, not all $X : Y$ are possible, because if (say) $X = \{r\}$ and $Y = \{b,g\}$ it's impossible to colour $B[u] : B'[u]$ such that $B'[u]$ is coloured with both b and g.

Thus, if we assume $B[u]$ and $B'[u]$ are both of size q, and that $X : Y$ is possible, $X : Y$ comes up $\nu(q, edgespan(X,Y)) \approx edgespan(X,Y)^q$ times, where

$$edgespan(X,Y) = \left| \left\{ (c_i, c_j) | c_i \in X \wedge c_j \in Y \wedge \{c_i, c_j\} \in E(H) \right\} \right|$$

The following table details the varying contributions of the different configurations; for convenience we henceforth drop the cumbersome Map notation for describing the set of colours occurring within a particular set of vertices, because in most contexts this is self-evident. The column "exact" denotes the exact number of times the relevant configuration comes up, on the basis that if $(C_1, C_2 : C_3, C_4)$ is a (possible) configuration on $(A[u], B[u] : B'[u], A'[u])$, it comes up $\nu(p,|C_1|)\nu(p,|C_4|)\nu(q, edgespan(C_2, C_3))$ times. The significance of the far right-hand column is explained in due course.

$A[u]$	$B[u] : B'[u]$	$A'[u]$	$edge-$ $span$	exact	approx	$p = 2s$ $q = 7s$
$\{r, b, g\}$	$\{b\} : \{b\}$	$\{r, b, g\}$	1	$\nu(p, 3)^2$	9^p	81^s
$\{r, b, g\}$	$\{b\} : \{r, b\}$	$\{r, b\}$	2	$\nu(p, 3)\nu(p, 2)\nu(q, 2)$	$6^p 2^q$	4608^s
$\{r, b, g\}$	$\{b\} : \{r, b, g\}$	$\{b\}$	3	$\nu(p, 3)\nu(q, 3)$	$3^p 3^q$	19683^s
$\{r, b\}$	$\{r, b\} : \{b\}$	$\{r, b, g\}$	2	$\nu(p, 3)\nu(p, 2)\nu(q, 2)$	$6^p 2^q$	4608^s
$\{r, b\}$	$\{r, b\} : \{r, b\}$	$\{r, b\}$	4	$\nu(p, 2)^2\nu(q, 4)$	$4^p 4^q$	262144^s
$\{r, b\}$	$\{r, b\} : \{r, b, g\}$	$\{b\}$	5	$\nu(p, 2)\nu(q, 5)$	$2^p 5^q$	312500^s
$\{b\}$	$\{r, b, g\} : \{b\}$	$\{r, b, g\}$	3	$\nu(p, 3)\nu(q, 3)$	$3^p 3^q$	19683^s
$\{b\}$	$\{r, b, g\} : \{r, b\}$	$\{r, b\}$	5	$\nu(p, 2)\nu(q, 5)$	$2^p 5^q$	312500^s
$\{b\}$	$\{r, b, g\} : \{r, b, g\}$	$\{b\}$	6	$\nu(q, 6)$	6^q	279936^s

As before we have neglected to include "wasteful" configurations; we will define them and sweep up their (exponentially small) contribution in due course. To be formally tight we need to show that the table exhaustively lists non-wasteful configurations; we leave this until the end of the proof.

If we now inspect the "approx" column of the above table, it is clear we have two degrees of freedom in determining the contributions of the various configurations: our choice of p and q, and more specifically the ratio of these two values. The far right-hand column shows the respective contributions if p, q are set such that $q = 7s$ and $p = 2s$ (for some sufficiently large s). Clearly, when p and q are chosen in this manner, the configurations that exponentially dominate (in s) are the sixth and eighth configurations, which are in fact symmetries of each other. This is important. We will argue shortly that, because of this dominance, "most" colourings of G' are such that *every* vertex encoding is coloured with either the sixth or the eighth configuration[9]. Within this set of colourings (i.e. where every vertex encoding is coloured with either the sixth or the eighth configuration) there is a further exponential hierarchy, which is determined by the number of colourings possible in the edge encodings i.e. the S, S' sets. To explain, observe that when configuration 6 is adjacent to configuration 6, both S and S' must be

[9]To clarify, this does not mean they must all be coloured with the same configuration.

coloured $\{b\}$, a contribution of 1^t, and this is also true when configuration 8 is adjacent to configuration 8. When configuration 6 is adjacent to configuration 8, however, one of S, S' is coloured $\{r, b\}$ and the other $\{b\}$, a superior contribution of 2^t. Hence, if we think of vertex encodings coloured with configuration 6 as being on one side of the cut, and those coloured with configuration 8 as being on the other, it follows that those colourings of G' which correspond to size-m cuts sit on the very top of the exponential hierarchy. We call such colourings *full* colourings to denote the fact that (as we shall show) there are far more of them than other types of colouring.

For ease of notation let $X = 312500$. Now, it follows that each size m-cut comes up Z times as a colouring of G', where Z is defined as follows:-

$$Z = 2(\nu(p,2)\nu(q,5))^n 2^{tm}$$

(The factor 2 is present because it is arbitrary which configuration is used to denote which side of the cut.) Thus, the idea is that if we divide the result of our oracle call (with G' as input) by Z and round we come up with a value sufficiently close to $\#LargeCut(G, m)$. To prove this we need to show (in the usual style) that there are fewer than $Z/4$ non-full colourings. Now, the set of non-full colourings can be partitioned into two categories; the set where every vertex encoding is coloured with configuration six or eight (but the cut pointed out is less than size m), and the set where at least one vertex encoding is not coloured with configuration six or eight. For convenience, let U_0^+ be the former set of colourings, and U_0^- be the latter set. Our strategy is to demonstrate that, with an appropriate choice of s and t, each of these two sets comes up no more than $Z/8$ times, thus automatically satisfying the $Z/4$ requirement.

Firstly, we study the relative contribution of U_0^+. The most a cut of less than size m can come up (as a colouring of G') is $(\nu(p,2)\nu(q,5))^n 2^{t(m-1)}$, and there are at most 2^n such cuts, so we need[10]

$$\frac{2^n(\nu(p,2)\nu(q,5))^n 2^{t(m-1)}}{(\nu(p,2)\nu(q,5))^n 2^{tm}} \leq 1/8$$

[10]We have omitted the factor of 2 because it appears on both the numerator and denominator and hence simply cancels out

This is easily satisfied; if we set $t = 2n$ then the LHS becomes 2^{-n} and this is less than or equal to $1/8$ for $n \geq 3$. Hence we have shown $|U_0^+| \leq Z/8$.

Proving the $Z/8$ bound for U_0^- is slightly more involved. Note that a pessimistic upper bound on the number of distinct configurations possible in an encoding of a vertex is $(2^3)^4 = 2^{12}$. We derive this figure by assuming that in each of $A[u], B[u], B'[u], A'[u]$ any of the 2^3 possible subsets of $V(H)$ are possible. Casting the net this wide automatically "hoovers up" every wasteful and non-dominant configuration. Thus, an upper bound on the contribution of U_0^- is:

$$2^{12n} X^{s(n-1)} (X-1)^s 9^{tn^2}$$

(The 9^{tn^2} term is a very crude upper bound on the number of ways that all the S, S' sets can be coloured; the n^2 arises because $|E(G)| \leq n^2$.) Hence, we require

$$\frac{2^{12n} X^{s(n-1)} (X-1)^s 9^{tn^2}}{(\nu(p,2)\nu(q,5))^n 2^{tm}} \leq 1/8 \tag{2.5}$$

We have foregone the use of the "onto" notation in the numerator because the numerator is an upper bound, and X^s is an upper bound on $\nu(p,2)\nu(q,5)$ for the designated choice of p and q. This follows from the useful Lemma 2.4 first expressed in [8]. (We detour briefly from the main proof to discuss Lemma 2.4, Corollary 2.5 amd Observation 2.6, and then return to the main proof.)

Lemma 2.4 (DGGJ) *If a and b are positive integers and $a \geq 2b\ln(b)$ then*

$$b^a(1 - \exp(-a/(2b))) \leq \nu(a,b) \leq b^a$$

(We omit the proof of this lemma.) This lemma essentially states that $\nu(a,b)$ is exponentially of the same order as b^a, which is why we can use them interchangeably in most circumstances. However, in this proof the tight lower bound given by Lemma 2.4 is more than we require; we know that X^s is exponentially bigger than the other exponentials in s, and as long as our lower bound retains the X^s part as the major term this dominance is preserved. For situations such as these a simpler, looser lower bound is adequate:

Corollary 2.5 *Let a and b be integers. If $b \geq 2$ and $a \geq 2b\ln(b)$ (or if $b = 1$ and $a \geq 1$),*

$$\frac{1}{2}b^a \leq \nu(a,b)$$

Proof. The case when $b = 1, a \geq 1$ is trivial. The case when $b \geq 2, a \geq 2b\ln(b)$ is immediate from solving the inequality $(1/2)b^a \leq b^a(1 - \exp(-a/(2b)))$.

Lemma 2.4 and Corollary 2.5 are used extensively throughout this thesis. Generally, we utilise these inequalities without worrying too much about the $a \geq 2b\ln(b)$ condition on their use. This is because b usually represents a number of colours from some graph H - and as a result is a constant - whereas, in contrast, a usually represents the size of some polynomially-large gadget. In such cases it is reasonable to argue that the inequality becomes true for n beyond some very small, fixed constant threshold. Even where we do not have the luxury of stating that n must lie beyond this threshold, it is generally fine to assume that, in addition to any other lower bounds that a must beat, a must be large enough to satisfy $a \geq 2b\ln(b)$.

Observation 2.6 *The exact value of $\nu(a,b)$ is easily computable in polynomial time*

To see this, note that an expression for $\nu(a,b)$ is:

$$\nu(a,b) = b^a - \sum_{i=1}^{b-1} \binom{b}{i} \nu(a,i)$$

and this can be computed iteratively.

(We now return to the main proof.) It transpires that, if s is chosen appropriately, we can even be as loose as to drop the 2^{tm} term from the denominator. That is, we can satisfy inequality (2.5) by applying Corollary 2.5 (twice) and satisfying the resulting inequality:

$$\frac{2^{12n}X^{s(n-1)}(X-1)^s 9^{tn^2}}{(1/4)^n X^{sn}} \leq 1/8$$

Re-arranging,

$$2^{14n}9^{tn^2}\left(\frac{X-1}{X}\right)^s \leq 1/8$$

If we take $t = 2n$ as derived above, then the only competitor to the exponentially small term in s is an (at most) cubic exponent in n (emerging from 9^{tn^2}), so in a similar manner to earlier proofs we can adequately satisfy the above equation by making s a polynomial of degree > 3, i.e. $s = n^4$. This finishes the proof that $|U_0^-| \leq Z/8$.

It remains only to show that the configurations listed in the table exhaust all non-wasteful configurations. We spend a little time demonstrating a general technique for proving exhaustiveness, before using it to put the final piece in the (admittedly very long!) proof for the graph *1-wrench*.

We say that a (possible) configuration $(C_1, C_2 : C_3, C_4)$ is wasteful iff for at least one of the C_i, there is a set D_i such that $C_i \subset D_i$ and D_i could replace C_i in the configuration without rendering the new configuration impossible. For example, $(rb, b : rbg, b)$ is wasteful because $C_1 = \{r, b\}$ could be upgraded to $\{r, b, g\}$. Alternatively, $C_2 = \{b\}$ could have been upgraded to $\{r, b\}$. To see that wasteful configurations are exponentially inferior, observe that if it is C_1 or C_4 which could be upgraded, the new configuration has a higher index in the exponential in p. If it is C_2 or C_3 that could be upgraded, then observe that as long as the new configuration is possible (which we require) then at least one extra edge of H is possible between $B[u]$ and $B'[u]$. This increases the exponential in q.

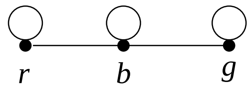
Now, recall the meaning of $H[C]$, where $C \subseteq V(H)$:- it is the subgraph of H induced by the mutual neighbours of C. Here we relax the notation ever so slightly so that $H[C]$ actually refers to the vertex set of $H[C]$. Secondly, let a $universal$ vertex[11] $c \in V(H)$ be one which is adjacent to every vertex in $V(H)$, itself included. We claim that, for H with at least one universal vertex (which we assume is b), a possible configuration $(C_1, C_2 : C_3, C_4)$ is wasteful iff at least one of the following conditions is *false*: $C_1 = H[C_2], C_2 = H[C_1], C_4 = H[C_3]$ and $C_3 = H[C_4]$. The direction $\rightarrow$ is easy to verify:- if the configuration is wasteful because (say) C_1 could have been

[11]Universal vertices appear frequently later on in the thesis.

upgraded, then $C_1 \subset H[C_2]$. Similarly, if it is wasteful because (say) C_2 could have been upgraded, then $C_2 \subset H[C_1]$. In the other direction, suppose (wlog) $C_1 = H[C_2]$ is false. Then we could have upgraded C_1 to $H[C_2]$, and the new configuration would by definition be possible, so the original configuration was wasteful. Alternatively, suppose (wlog) $C_2 = H[C_1]$ is false; there are two cases to consider here. If C_3 contains b then we could simply upgrade C_2 to $H[C_1]$ - the new configuration is definitely possible because the extra colours in $H[C_1]$ can if need be all link to b vertices in $B'[u]$. If C_3 does not contain b, then we could upgrade C_3 to include b:- the resulting configuration is possible because vertices in $B'[u]$ coloured b can connect to any colours of their choice from C_2. Either way, this shows that the original configuration was wasteful.

This is quite a lot of work, but it affords us a generally applicable technique. When H has a universal vertex, we can exhaustively list non-wasteful configurations (for vertex encodings as used in this proof) as follows. Construct the set $GoodPairs(H) = \{(C_i, C_j) | C_i, C_j \subseteq V(H) \wedge C_i = H[C_j] \wedge C_j = H[C_i]\}$. Now, the non-wasteful configurations are the $|GoodPairs(H)|^2$ configurations of the form $(C_1, C_2 : C_3, C_4)$ where $(C_1, C_2) \in GoodPairs(H)$ and $(C_3, C_4) \in GoodPairs(H)$.

The $GoodPairs$ of the *1-wrench* graph are $\{(\{b\}, \{r, b, g\}), (\{r, b\}, \{r, b\}), (\{r, b, g\}, \{b\})\}$, and hence the exhaustiveness of our table is assured. Thus, the proof that $\#H \equiv_{\mathsf{AP}} \#SAT$ (which began on page 45) is now finally complete. $\square$



(Graph 11) Status: $\equiv_{\mathsf{AP}} \#BIS$. This graph, known as the 2-particle Widom-Rowlinson graph (or *2-WR*) is the only non-trivial connected 3-vertex H that has not been proven to be $\equiv_{\mathsf{AP}} \#SAT$. In [8] it is shown that $\#$*2-WR*$\equiv_{\mathsf{AP}} \#BIS$; since $\equiv_{\mathsf{AP}} \#BIS$ is a class of apparently "intermediate" complexity it is necessary to show both $\#$*2-WR*$\leq_{\mathsf{AP}} \#BIS$ and $\#BIS \leq_{\mathsf{AP}} \#$*2-WR*. The relevant reductions we now show are essentially the same as those in [8]; they are extremely redundant and tailored to this graph but are simple and elegant and worthy of an airing. More generalised reductions appear later in this chapter.

Proving $\#$*2-WR*$\leq_{\mathsf{AP}}\#BIS$

To aid this result, we first introduce some preliminaries about bipartite H. The complexity of approximately counting H-colourings when H is a bipartite graph is not particularly well understood. However, there are a number of very important "complexity bridges" between the domains of bipartite H and non-bipartite H. Perhaps the most important is the technique of *bipartisation*, a construction technique used by Dyer and Greenhill in [10] and more recently in [8].

Bipartisation

Let $H = (V(H), E(H))$ be a non-bipartite graph[12] where $V(H) = \{c_0, c_1, ..., c_{|V(H)|-1}\}$. Let $H' = bi(H) = (V_L(H'), V_R(H'), E(H'))$ be the bipartite graph defined as follows:

$$V_L(H') = \{d_0, d_1, ..., d_{|V(H)|-1}\}$$

$$V_R(H') = \{d'_0, d'_1, ..., d'_{|V(H)|-1}\}$$

$$E(H') = \bigcup_{\{c_i,c_j\}\in V(H)} \{\{d_i, d'_j\}, \{d_j, d'_i\}\}$$

The significance of this is that $bi(H)$ (the *bipartisation* of H) is, to all intents and purposes, equivalent to the problem of counting H-colourings when the input G is *restricted to being bipartite*. Indeed, if G is bipartite, and we let $H' = bi(H)$,

$$\#H'(G) = 2\#H(G)$$

(This expression holds for disconnected H too.) Thus, $\#bi(H)\leq_{\mathsf{AP}}\#H$ for all H. The largely insignificant factor of two emerges because there is a choice of whether to map the left-hand side colours of $bi(H)$ to the left-hand side vertices of G or the right-hand side vertices of G. (This issue, of *orientations*, is considered in greater length in Section 2.4.2.)

[12] It is worth noting that the following transformation is also valid where H is already bipartite:- $bi(H)$ comprises two disjoint copies of H. However, apart from Section 4.5 in Chapter 4, we do not use $bi(H)$ in relation to bipartite H.

The fact that $\#bi(H) \leq_{\mathsf{AP}} \#H$ allows us to easily establish relationships such as those shown in Figure 2.6. We explain this bipartisation process here because it is di-

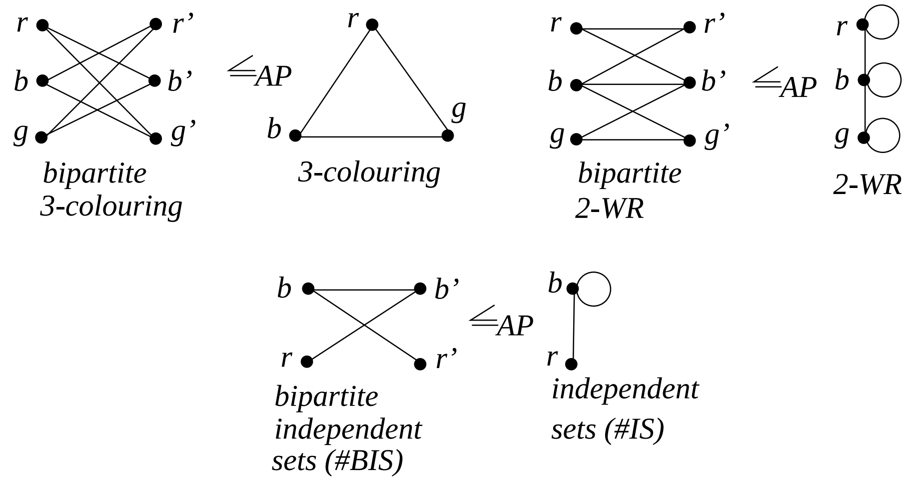


Figure 2.6: *Examples of bipartisation*

rectly relevant to $\equiv_{\mathsf{AP}}\#BIS$-easiness and in particular $\equiv_{\mathsf{AP}}\#BIS$-hardness reductions. More specifically, since $\#BIS$ is the problem of counting independent sets in a bipartite graph, it follows that the graphical representation of $\#BIS$ is (ignoring the factor of 2) the bipartisation of the graphical representation of $\#IS$. Thus, as Figure 2.6 shows, P_4 (the path on 4 vertices) is essentially the H-colouring equivalent of $\#BIS$. Hence, $\#P_4 \equiv_{\mathsf{AP}} \#BIS$, and as a result we treat the two problems as interchangeable.

(We now return to the proof that $\#$*2-WR*$\leq_{\mathsf{AP}}\#BIS$.) Thus, if we let $H' = P_4$, and can show $\#$*2-WR*$\leq_{\mathsf{AP}}\#H'$ then this is sufficient to show the $\equiv_{\mathsf{AP}}\#BIS$-easiness of $\#$*2-WR*. Let G be the input to $\#$*2-WR*; we proceed by constructing a graph G' for input to $\#H'$ that is bipartite even if G is not. For each vertex $u \in V(G)$, introduce two vertices $T[u], B[u]$ to G', and connect $T[u]$ to $B[u]$. For each edge $\{u, v\}$ in $E(G)$ connect $T[u]$ to $B[v]$ and $B[u]$ to $T[v]$. (To see that G' is bipartite, note that all the $T[.]$ vertices form one side of the bipartition, and all the $B[.]$ vertices form the other, with no edges between $T[.]$ vertices or between $B[.]$ vertices.) Now, observe that $\#$*2-WR*$(G) = (1/2)\#H'(G')$. This follows because each pair $(T[u], B[u])$ can be coloured with one from (assuming this orientation) $\{(b, b'), (b, r'), (r, b')\}$ and the only two such pairs that *cannot* be adjacent to each other are (b, r') and (r, b'). Hence, (b, b') acts like b from *2-WR*, and the other two colour pairs act like r and g from *2-WR*. The factor of 2 appears, like before, because we could have coloured G' with the opposite orientation i.e. $\{(b', b), (b', r), (r', b)\}$. Constant multipliers are not a problem so $\#$*2-WR*$\leq_{\mathsf{AP}}\#H'\leq_{\mathsf{AP}}\#BIS$. $\square$

Proving $\#BIS\leq_{\mathsf{AP}}\#$*2-WR*

(This proof is taken from [8].) Since G is bipartite we assume its LHS vertices are $V_L(G) = \{u_0, u_1, ..., u_{n_l-1}\}$ and its RHS vertices are $V_R(G) = \{v_0, v_1, ..., v_{n_r-1}\}$. As a standard convention for bipartite G, we let n represent the total number of vertices i.e. $n = n_l + n_r$. A graph G' is built as follows. (Note how the reduction exploits the fact that the input to $\#BIS$, G, is bipartite: in particular, the way vertices in $V_L(G)$ are coded up is different to the way vertices in $V_R(G)$ are coded up.) For each vertex $u_i \in V_L(G)$ we introduce U_i which is a copy of K_p (for p to be determined later). For each vertex $v_i \in V_R(G)$ simply add it without change. Finally, we introduce K which is another copy of K_p. We connect every vertex in K to every v_i. For each edge $\{u_i, v_j\}$ in G we connect v_j to every vertex in U_i. This construction is demonstrated in Figure 2.7. Notice that, because K and the various U_i are all complete graphs, the only colourings

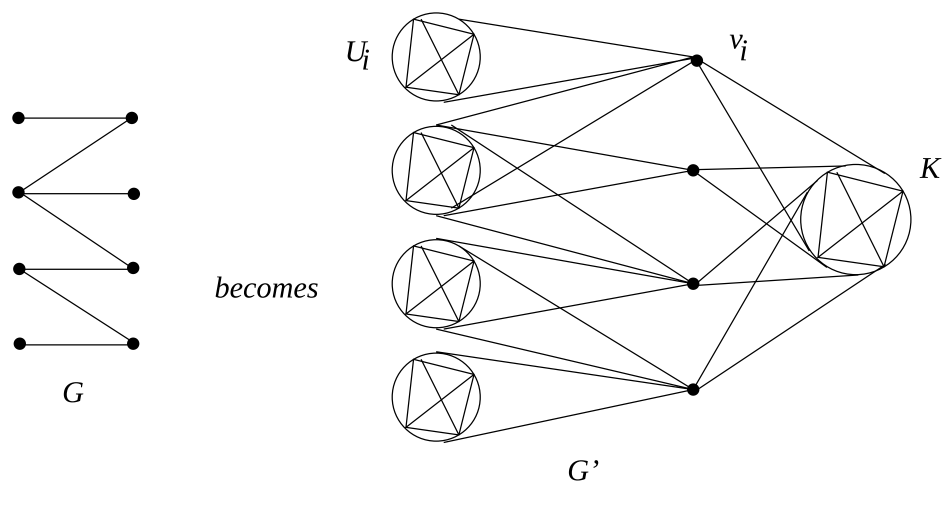


Figure 2.7: *Reducing $\#BIS$ to $\#$2-WR*

that are allowed within them are of the form $\{b\}$,$\{r\}$,$\{g\}$, $\{b, r\}$ or $\{b, g\}$. On this occasion we consider a full colouring to be one where each size-p set is coloured with either $\{r, b\}$ or $\{b, g\}$. We will show (as usual) how full colourings outstrip other types of colouring at the end of the proof. Focusing now on just full colourings, it is clear that if K is coloured (say) $\{b, g\}$ then each v_i is restricted to either being b or g. Note that a U_i coloured $\{b, g\}$ can be adjacent to v_i coloured b or g, but a U_i coloured $\{r, b\}$ can only be adjacent to v_i coloured b. Thus, the induced behaviour is that of bipartite independent sets (i.e. P_4): a vertex u_i is IN the independent set if U_i is coloured $\{r, b\}$ and OUT if U_i is coloured $\{b, g\}$. A vertex v_i is IN if it is coloured g, OUT if it is coloured b. (The IN/OUT definition switches symmetrically if K is coloured $\{r, b\}$ but the behaviour is exactly the same.)

Hence, each independent set comes up exactly

$$2\nu(p, 2)^{n_l+1} = 2(2^p - 2)^{n_l+1}$$

times as a full colouring, so we divide $\#\hat{H}(G')$ by this and round to attain an approximation to $\#BIS$. It only remains to demonstrate that non-full colourings are insignificant by showing (in the usual manner) that non-full colourings are less than a quarter of $2\nu(p, 2)^{n_l+1}$. Now, a non-full colouring is any in which at least one of $U_0, U_1, ..., U_{n_l-1}, K$ is not bicoloured. We can derive a generous although adequate upper bound on the number of non-full colourings as follows. An upper bound on the number of times any one non-full configuration comes up is $2^{pn_l}1^p1^{n_r}$, which is an "almost" full configuration with only one of the size-p sets coloured monochrome. A loose upper bound on the number of such configurations is $5^{n_l+1}3^{n_r}$: 5 is used because there are 5 possible colour combinations within each size-p set, $\{\{b\}, \{g\}, \{r\}, \{r, b\}, \{b, g\}\}$, and the 3^{n_r} term appears because we assume the v_i vertices can take any colour. Hence, a crude upper bound on the total number of non-full colourings is the product of these two values, which is $5^{n_l}3^{n_r}2^{pn_l}5$. Now, we need to choose p such that

$$\frac{5^{n_l}3^{n_r}2^{pn_l}5}{2\nu(p, 2)^{n_l+1}} \leq 1/4$$

Using the fact that $(1/2)b^a \leq \nu(a, b)$ (by Corollary 2.5) and simplifying shows that it is adequate, then, to prove

$$\frac{10^{n_l}3^{n_r}5}{2^p} \leq 1/4$$

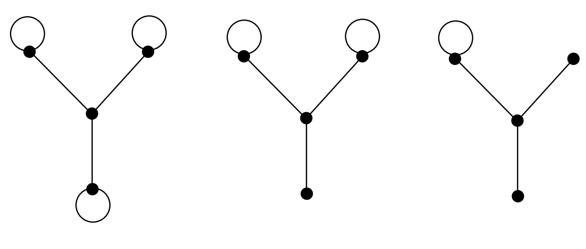
Given that $n_l \leq n$ and $n_r \leq n$ it is simpler to absorb the n_l, n_r terms and instead look to satisfy:-

$$\frac{30^n 5}{2^p} \leq 1/4$$

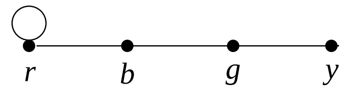
By this stage of the chapter it should be apparent that such inequalities are easy to cope with; taking $p = 5n$ suffices (for n beyond some constant threshold), but in general we needn't process the details because making p an adequately large superlinear polynomial is almost always sufficient, as long as it is strictly larger than the exponent characterising the contribution of the non-full colourings. It is in this spirit that we henceforth omit technical details of this nature. We choose instead to justify (in each case) the use of a particular gadget or gadgets by showing that the types of colouring we wish to count are the "best" at colouring the gadgets. For example, the core of the very first proof in Section 2.2 essentially rested on the fact that $H[r]$ can colour K_2 in 3 ways and $H[b]$ can colour K_2 in only 1 way. So there the fact that $3 > 1$ was the clinching factor, on the (correct) assumption that our freedom to choose the size of the gadget (and thus the size of the exponential in the counting equation) allowed us to make this the exponentially dominant feature. □

2.3 4-vertex H for which $\#H$ is interreducible with $\equiv_{\mathsf{AP}} \#SAT$

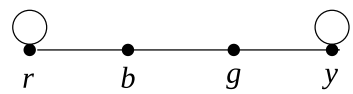
In this section we list all the 4-vertex H for which AP-interreducibility with $\#SAT$ has been ascertained. (This is all original work.) It transpires that a sizeable majority of 4-vertex H fall into this category.



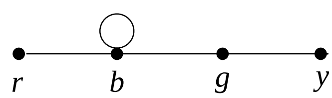
(Graphs 22, 20, 18) Status: $\equiv_{\mathsf{AP}}\#SAT$. Note that, in all of the above graphs, the subgraph pointed out by any looped colour (i.e. the H representing independent sets) can colour a copy of K_3 in 4 ways, contrasting with at most 3 ways for any of the subgraphs pointed out by unlooped vertices. Hence, for all the above graphs using a K_3-cliqueset makes it exponentially likely that independent sets are pointed out, thus categorising them all as $\equiv_{\mathsf{AP}}\#SAT$. (Hence, in this instance the $\#IS\leq_{\mathsf{AP}}\#H$ proof is very similar to that used with Graph 8 on page 32, except that copies of K_3 are used instead of copies of K_2.)



(Graph 27) Status: $\equiv_{\mathsf{AP}}\#SAT$. To show this graph is in $\#SAT$ we reduce $\#IS$ to it, using the same K_2-cliqueset reduction deployed in the very first 3-vertex H reduction in Section 2.2. Note that, as in that case, $H[r]$ can colour a copy of K_2 in 3 ways which is more than any of $H[b]$, $H[g]$ or $H[y]$ can individually manage. Thus, for appropriately large k the nexus x is exponentially likely to be coloured r, pointing out independent sets in G.

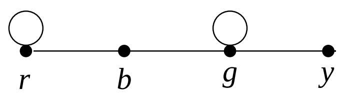


(Graph 31) Status: $\equiv_{\mathsf{AP}}\#SAT$. Again, reduction from $\#IS$, using an almost identical reduction to the above. The only difference is the factor of 2 introduced as a result of the fact that r and y are indistinguishable.

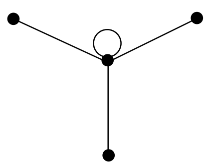


(Graph 25) Status: $\equiv_{\mathsf{AP}}\#SAT$. We demonstrate that this graph is $\equiv_{\mathsf{AP}}\#SAT$ by showing a very simple reduction from the 3-vertex graph the *compass*, itself shown to be $\equiv_{\mathsf{AP}}\#SAT$ by Lemma 2.3. (The compass is the right-hand graph in Figure 2.2, on page 36.) Note that the degree of b is 3 and the maximum degree of any of the other colours is 2. Also, we see that, for any colour c, $H[c]$ can colour a copy of K_1 exactly $deg(c)$ times. Hence, if we use a K_1-cliqueset rather than a K_2-cliqueset this has the property of making the nexus x (see Figure 2.1) exponentially likely to be coloured b, pointing out compass colourings in G. We could have continued using a K_2-cliqueset but we generally use the simplest gadgets that can perform the job adequately.

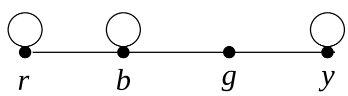
The use of the K_1-cliqueset is so fundamental that we refer to it as the *maxdeg* gadget, because of its property of picking out maximum degree colours in H.



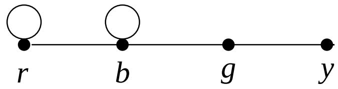
(Graph 28) Status: $\equiv_{\mathsf{AP}}\#SAT$. The same reduction as above also works here, because g is the maximum degree colour and $H[g]$ is the compass.



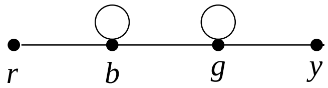
(Graph 17) Status: $\equiv_{\mathsf{AP}}\#SAT$. Note that this graph is the independent set problem with weight 3 on the unlooped vertex, so has already been classified by Lemma 2.3.



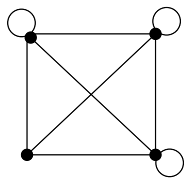
(Graph 32) Status: $\equiv_{\mathsf{AP}}\#SAT$. The colour b is the maximum degree colour in H, and $H[b]$ is the *1-wrench* which has already been shown to be $\equiv_{\mathsf{AP}}\#SAT$ in Section 2.2; see page 45. Hence, the maxdeg gadget also works here (as a means of picking out b, and thus pointing out *1-wrench* colourings.)



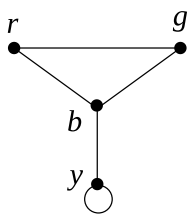
(Graph 29) Status: $\equiv_{\mathsf{AP}} \#SAT$. Given that b is again the maximum degree colour in H and $H[b]$ is the *1-wrench*, the previous reduction works here also.



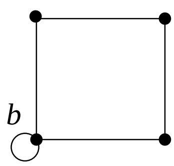
(Graph 26) Status: $\equiv_{\mathsf{AP}} \#SAT$. The colours b and g are both of maximum degree, and $H[b]$ and $H[g]$ are both the *1-wrench*, so the maxdeg gadget allows a reduction from *1-wrench*. (The factor of 2 is, as usual, easy to deal with.)



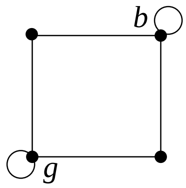
(**Graph 64**) Again, this is a weighted version of the independent set problem, this time with weight 3 on the looped vertex, so is $\equiv_{\mathsf{AP}} \#SAT$ by Lemma 2.3.



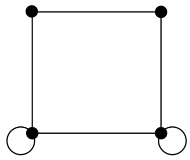
(Graph 40) Status: $\equiv_{\mathsf{AP}} \#SAT$. We use a K_3-cliqueset with a view to picking out y and thus pointing out independent sets. Note that, in this instance, both $H[b]$ and $H[y]$ can colour a copy of K_2 3 times, so a K_2-cliqueset would not be adequate. However, $H[b]$ can colour K_3 in only 1 way, whilst $H[y]$ can colour it in 4 ways.



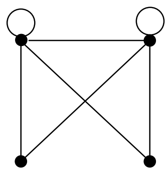
(Graph 47) Status: $\equiv_{\mathsf{AP}}\#SAT$. The colour b has the maximum degree in H, so using the maxdeg gadget gives us a reduction from the compass.



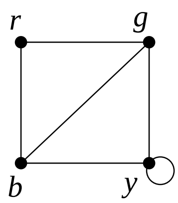
(Graph 48) Status: $\equiv_{\mathsf{AP}}\#SAT$. As the previous reduction, the only difference being a factor of two stemming from the fact that b and g are indistinguishable.



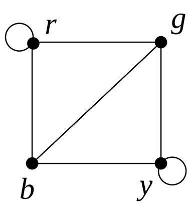
(Graph 49) Status: $\equiv_{\mathsf{AP}}\#SAT$. Using the maxdeg gadget picks out the looped colours, and both point out $\equiv_{\mathsf{AP}}\#SAT$ *1-wrench* colourings in G.



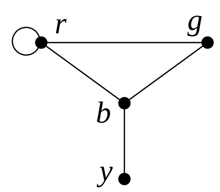
(Graph 54) Status: $\equiv_{\mathsf{AP}}\#SAT$. This is the independent set with weight 2 on both vertices. Hence, following Lemma 2.3 we know immediately that it is $\equiv_{\mathsf{AP}}\#SAT$.



(Graph 55) Status: $\equiv_{\mathsf{AP}}\#SAT$. If a K_4-cliqueset is used then y is picked out, and *pyramid* (i.e. Graph 13) colourings pointed out in G. The colour y "wins" because $H[y]$ can colour K_4 in 21 ways ($yyyy$, 4 x $yyyb$, 4 x $yyyg$, 12 x $yygb$), but both $H[g]$ and $H[b]$ can only colour a copy of K_4 in 5 ways.



(Graph 58) Status: $\equiv_{\mathsf{AP}}\#SAT$. Using the K_4-cliqueset also works in this instance, also allowing a reduction from pyramid colourings. Note that both $H[r]$ and $H[y]$ can colour a copy of K_4 in 21 ways, as above. Both $H[g]$ and $H[b]$ can individually only colour a copy of K_4 in 10 ways. The factor of two introduced as a result of r and y being indistinguishable is, as usual, inconsequential.



(Graph 36) Status: $\equiv_{\mathsf{AP}}\#SAT$. Here it would suffice to use a K_4 cliqueset to pick out r and thus point out pyramid colourings. However, this H allows us the opportunity to demonstrate another powerful reduction technique.

Firstly, we note that both r and b are maximum degree (3) colours. Secondly, observe that the subgraph of H *induced* (as opposed to pointed out) by $\{r, b\}$ is in fact the independent set problem. Thus, if all vertices of G could be made exponentially likely to be coloured with a maximum degree colour then we would have a reduction from $\#IS$ to $\#H$. This is indeed possible, and is the spirit in which we construct G'.

Let G be the input to $\#IS$. G' is a copy of G with extra vertices and edges added as follows. For each vertex $u \in V(G)$, we introduce an independent set $I[u]$ of size k, and connect u to every vertex in $I[u]$. Figure 2.8 should serve to clarify this.

There is a natural mapping from $H(G')$ colourings to $H(G)$ colourings, derived simply by inspecting the colours on the "original" vertices of G'. Hence, a colouring of G that solely uses the colours $\{r, b\}$ comes up 3^{kn} times as a colouring of G'. Any other colouring of G comes up at most $3^{k(n-1)}2^k$ times, and a loose upper bound on the number of such colourings is 4^n. In the usual manner, then, we argue that dividing $\#\hat{H}(G')$ by 3^{kn} and rounding yields a good approximation to $\#IS$. To cement this, we

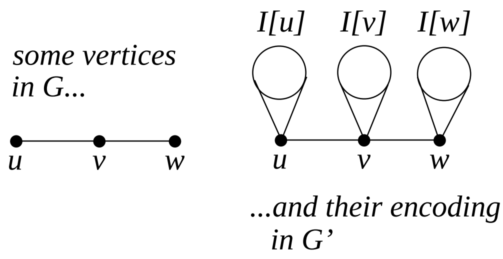


Figure 2.8: *Picking out maximum degree colours at the vertex level*

need to choose k such that

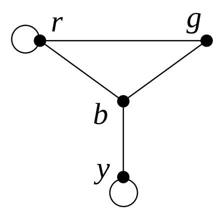
$$\frac{4^n 3^{k(n-1)} 2^k}{3^{kn}} \leq 1/4$$

Choosing $k = 7n$ satisfies this for all n.

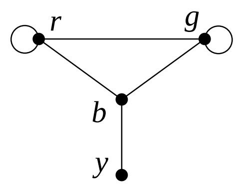
The important point to note about this reduction compared to, say, cliqueset, is that it operates at the level of the individual vertex. Had we picked out one of the maximum degree vertices using one overarching maxdeg gadget and used it to point out colourings in G, we would have to argue that the complexity of calculating $\#H'(G) + \#H''(G)$ (where $H' = H[r]$ and $H'' = H[b]$) is $\equiv_{\mathsf{AP}} \#SAT$. (Later on we are indeed able to develop arguments that address this "sum of colourings" type of equation, but at this stage it is a far from ideal situation to be in.)

Factoring the gadget "in", however, as we did in this case, makes a virtue of this flaw and induces a graph rather than pointing one out. The lesson emerging from this is that it is sometimes beneficial to factor gadgetry "out", as with cliqueset-style reductions, and sometimes beneficial to factor it "in" so that it operates at the level of each individual vertex.

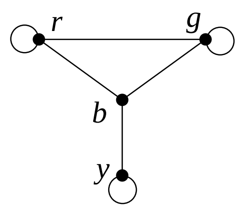
(The observant reader will notice that this technique could have been applied to earlier graphs; the choice of this graph as a vehicle for demonstration is arbitrary. For example, repeating this reduction but utilising K_2-cliquesets rather than maxdeg gadgets would have provided an alternative reduction from $\#IS$ to graph 40.)



(Graph 42) Status: $\equiv_{\mathsf{AP}}\#SAT$. Exactly the same reduction as just described for graph 36 also works here.



(Graph 38) Status: $\equiv_{\mathsf{AP}}\#SAT$. In this graph, the maximum degree colours are $\{r, b, g\}$, and these colours induce ear (i.e. graph 14) colourings. Hence, if we use the same technique as demonstrated for graph 36 (i.e. using gadgetry to pick out the graph induced by maximum-degree colours) we have a reduction from the $\equiv_{\mathsf{AP}}\#SAT$ problem of counting ear colourings to the problem $\#H$, thereby demonstrating the $\equiv_{\mathsf{AP}}\#SAT$ status of this graph.



(Graph 44) Status: $\equiv_{\mathsf{AP}}\#SAT$. We prove that this graph is $\equiv_{\mathsf{AP}}\#SAT$ by using a reduction which we call a *looped clique grabbing* reduction. Ultimately (in Sections 5.4 and 5.5) we refine this idea of "grabbing looped cliques" (which will become clear in due course) into a pair of generalised lemmas. This generalisation yields a fairly powerful and general classification method, but for the purpose of this chapter it is instructive to demonstrate the technique as tailored to individual H graphs.

We reduce, once again, from ear colourings. The idea is to build a gadget which is exponentially likely to be coloured $\{r, g\}$, because we can then use this to point out the graph $H[\{r,g\}]$ i.e. ear colourings. To isolate $\{r,g\}$ we seek to exploit the fact that $\{r,g\}$ constitute the *largest* looped clique in H.

Let G be the graph in which we wish to count ear colourings. G' is constructed as follows. In addition to a copy of G, we introduce K, a complete graph on p vertices, and I, an independent set containing q vertices. Every vertex in K is connected to every vertex in G, and every vertex in K is connected to every vertex in I.

If K is coloured with $\{r,g\}$ exactly the gadgetry (excluding G) can be coloured in $\nu(p,2)3^q$ ways; we let these be the full colourings. In this proof it makes life slightly easier to separate the non-full colourings into two groups, and show that in each case the ratio of non-full to full is less than $1/8$. The first group contains the colourings where K is coloured either just $\{r\}$ or just $\{g\}$; these are both wasteful because K may as well contain both r and g, since this doesn't change the colourings possible in I or G. Both $\{r\}$ and $\{g\}$ can colour the gadgetry 3^q times, so it is immediate that if p is (say) at least linear in n (or in fact any value greater than or equal to 5) the value $3^q 2/\nu(p,2)3^q$ quickly drops below $1/8$.

The combined contribution of configurations $\{r,b\}$, $\{g,b\}$ and $\{r,g,b\}$ can be bound above by the single term[13] $p2^{p-1}2^q$. Continuing, if K is coloured solely $\{y\}$ this comes up 2^q times, and if K is coloured $\{y,b\}$ this comes up p times. Clearly $p2^{p-1}2^q$ is an upper bound on both p and 2^q so we can account for the combined contribution of $\{r,b\}$, $\{g,b\}$, $\{r,g,b\}$, $\{y\}$ and $\{y,b\}$ by introducing a factor of 3 in front of $p2^{p-1}2^q$ to yield the term $3p2^{p-1}2^q$. Therefore, if we generously assume these configurations can colour G in $|V(H)|^n$ ways, and apply Corollary 2.5, p and q must be chosen such that:

$$\frac{4^n 3p2^{p-1}2^q}{(1/2)2^p 3^q} \leq 1/8$$

Choosing $p = n^2$ and $q = 10n$ is adequate for this (for n above a certain constant threshold.[14])

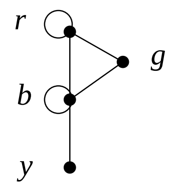
[13] Only one vertex in K can be coloured b, and that can be any of the p vertices in K.

[14] Because the index of the resulting exponential in n is less than 1.

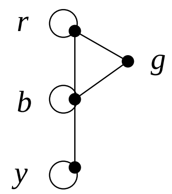
As mentioned, there is formally speaking no need to make p an increasing function in n; making K a single copy of K_5 would also be sufficient to ensure domination over both groups of non-full configurations. Similarly, the reader may be wondering why we define full colourings to require the presence of *both* r and g, when $\{r\}$ and $\{g\}$ have the same overall effect. In both cases this is because we wish to start as we mean to go on; in this instance it just so happens that we can get away with a constant-sized K and allowing $\{r\}$ and $\{g\}$ as full colourings, but this is not generally the case, and we wish to make extensive use of this gadget later in the thesis. More specifically, we want to show that K is exponentially likely to be coloured with *all* the colours from a maximum looped clique, because some reductions will only behave correctly if all colours are present. This is also why we set q to be smaller than p, even though in this instance setting $q = p = n^2$ would have been fine. This is to prevent I, rather than K, being the primary determinant of which colourings dominate.

The next four graphs are all $\equiv_{\mathsf{AP}} \#SAT$ by virtue of reductions from ear colourings. The technique in the previous reduction works for each of these graphs; taking $p = n^3$ and $q = n^2$ crudely but effectively has the desired effect. When this technique is formalised and generalised in Section 5.4 we provide much tighter expressions for p, q, but at this stage the main point to appreciate is the overall behaviour of the reduction. Having K as the most significant gadget ensures that K contains those colours that constitute the maximum looped clique in H. Where there is more than one such looped clique, the addition of I ensures that the looped cliques that point out the *largest subgraphs* (in terms of the number of colours) extend their dominance even further.

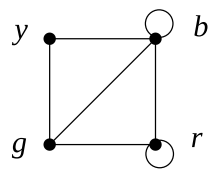
The addition of I also ensures that only colours from *within* maximum looped cliques appear in K, and not colours from the larger subgraph that maximum looped cliques point out e.g. in the previous reduction it is I that enables us to make $\{r, g\}$ in K exponentially dominant over $\{r, b, g\}$. Where none of the maximum looped cliques point out non-trivial subgraphs, I is not necessary to enforce this property, but seeing as the pointing out of non-trivial subgraphs is the reason for the existence of this gadget it is obviously required whenever such a reduction is intended.



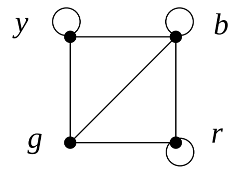
(Graph 37) Status: $\equiv_{\mathsf{AP}} \#SAT$. (We use the same "looped clique grabbing" technique as for Graph 44.) As before the primary determinant of dominance is the number of colours in K. If K contains $\{r, b\}$ and I contains $\{r, b, g\}$ there are approximately $2^p 3^q$ ways to achieve this; this is the maximum configuration. To see this, observe that to compete with this the colours appearing in K must at the very least have K_2^* as a subgraph, because of p being so much larger than q. Therefore the only possible competitor is where K contains $\{r, b, g\}$ and I contains $\{r, b\}$, but this is not maximal because it comes up at most $p2^{p-1}2^q$ times, which is exponentially less than $2^p 3^q$. Hence, K is exponentially likely to be coloured $\{r, b\}$ and this points out ear-colourings in G.



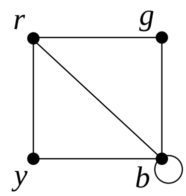
(Graph 43) Status: $\equiv_{\mathsf{AP}} \#SAT$. (We use the same "looped clique grabbing" technique as for Graph 44.) This time the only possible competitors for maximality are (K, I) coloured (rb, rbg), (by, by) or (rbg, rb), and these come up (approximately) $2^p 3^q$, $2^p 2^q$ and $p2^{p-1}2^q$ times respectively. Hence (K, I) is exponentially likely to be coloured (rb, rbg) and ear colourings are therefore pointed out.



(Graph 56) Status: $\equiv_{\mathsf{AP}} \#SAT$. (We use the same "looped clique grabbing" technique as for Graph 44.) Using the same argumentation as in the proof of Graph 37, the only configurations we need to consider are (br, brg) and (brg, br). (This is as usual because $\{b, r\}$ and $\{b, r, g\}$ are the only sets of colours that are possible in K *and* have K_2^* as a subgraph.) They come up $2^p 3^q$ and $p2^{p-1}2^q$ times respectively, so (br, brg) dominates and hence ear colourings are pointed out.

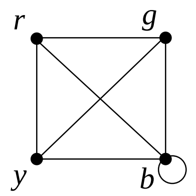


(Graph 59) Status: $\equiv_{\mathsf{AP}} \#SAT$. (We use the same "looped clique grabbing" technique as for Graph 44.) The only configurations we need to consider are (br, brg), (brg, br), (by, byg) and (byg, by). By the usual observations we see that (br, brg) and (by, byg) are both maximal configurations. They both point out ear colourings so a factor of 2 is introduced, but as we know this is easily dealt with.

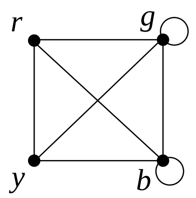


(Graph 53) Status: $\equiv_{\mathsf{AP}} \#SAT$. We show this graph is $\equiv_{\mathsf{AP}} \#SAT$ by reducing from the independent set problem, and (assuming G is the input to $\#IS$) build G' as follows. Introduce an independent set I_1 of size p and an independent set I_2 of size q, setting $q = 2p$. Every vertex in I_1 is connected to every vertex in I_2, and every vertex in I_2 is connected to every vertex in a copy of G. The configurations possible in (I_1, I_2) are as follows:- $(b, rbgy)$ comes up approximately $1^p 4^{2p} = 16^p$ times, (br, bgy) comes up $2^p 3^{2p} = 18^p$ times, (bgy, br) comes up $3^p 2^{2p} = 12^p$ times, and $(rbgy, b)$ comes up 4^p times. Thus, if p is chosen adequately large, (br, bgy) dominates and $H[\{bgy\}]$ is the independent set problem. To see that we have exhausted all non-wasteful configurations, firstly observe that if C is the set of colours appearing in I_2, we should always colour I_1 with all the colours from $H[C]$. Now, focusing on I_2, note that C may as well always contain b, and that if C contains either g or y it may as well contain both of them. Thus, non-wasteful configurations are when we colour I_2 either $\{b\}$, $\{b, r\}$, $\{b, g, y\}$ or

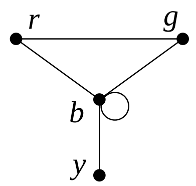
$\{b, r, g, y\}$ and we have covered all these possibilities.



(Graph 62) Status: $\equiv_{\mathsf{AP}}\#SAT$. We reduce from $\#IS$; let G be the input to $\#IS$. We build G' exactly as in the previous reduction. This time, note that - owing to the fact that the unlooped colours in H are indistinguishable - the four non-wasteful configurations (modulo symmetries) on (I_1, I_2) can be characterised as when I_2 contains b plus $\{0, 1, 2, 3\}$ colours. If $q = 2p$, these come up 4^p, $3^p 2^{2p} = 12^p$, $2^p 3^{2p} = 18^p$ and $4^{2p} = 16^p$ times respectively, so the dominant configurations are (br, bgy), (bg, bry) and (by, brg), all of which point out independent sets in G.

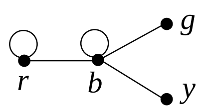


(Graph 63) Status: $\equiv_{\mathsf{AP}}\#SAT$. We reduce from ear colourings, constructing G' as above except for the fact that $p = q$. This makes both (bgr, bgy) and (bgy, bgr) dominate, because they each contribute $3^p 3^p = 9^p$, whereas $(bg, bgry)$ and $(bgry, bg)$ each only contribute $2^p 4^p = 8^p$. Both the dominating configurations point out ear colourings. We have exhausted all non-wasteful colourings because (again looking at I_2) we may as well always use both b and g, plus $\{0, 1, 2\}$ extra colours.



(Graph 35) Status: $\equiv_{\mathsf{AP}}\#SAT$. This graph is interreducible with $\#IS$ but for a slightly unusual reason. In particular, for all G, $\#H(G) = (\#IS(G))^2$. The origin of this somewhat surprising fact is explained in Appendix A.3, but observe that this clean relationship between $\#H(G)$ and $\#IS(G)$ allows us to develop a simple reduction (in both directions) between the two. To reduce $\#IS(G)$ to $\#H(G)$, we need only obtain our approximation to $\#H(G)$ and take the square root; using ϵ (the accuracy we have to estimate $\#IS(G)$ to) as the accuracy parameter in the oracle call is sufficient. To reduce $\#H(G)$ to $\#IS$, we call the $\#IS(G)$ oracle with accuracy $\epsilon/2$ and square the result. (The asymmetry in the accuracy parameters used stems from the fact that squaring an approximation slightly diminishes accuracy, but taking the square root actually *improves* it.)

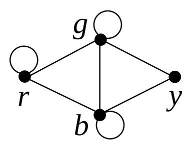
The strong relationship between $\#IS(G)$ and $\#H(G)$ can be attributed to a particular structural relationship between the graph representing the independent set problem (i.e. K_2^* minus one loop) and H. H is in fact the "square" of the graph representing the independent set problem. Of course, the use of the term "square" presupposes the existence of a *graph multiplication* operator, and the basic properties of this operator are explained in Appendix A.3. (Interestingly, two graph transformations we have already discussed can be expressed in terms of graph multiplication. The graph $bi(H)$ is obtained by "multiplying" H by $K_{1,1}$, and the act of increasing the weights on a graph H by a factor of $p \in \mathbb{N}^+$ can be achieved by "multiplying" H by K_p^*.)



(Graph 19) Status: $\equiv_{\mathsf{AP}}\#SAT$. The above graph can be shown to be $\equiv_{\mathsf{AP}}\#SAT$ using a $\#LargeCut$ reduction very similar to that used to classify the *1-wrench*; see page 45. This time, we set $q = 2p$, where q is the size of the B, B' sets and p is the size of the A, A' sets. The following table demonstrates that the $(A, B : B', A')$ configuration $(b, rbgy : rb, rb)$ and its symmetry $(rb, rb : rbgy, b)$ are exponentially dominant, contributing approximately 72^p. (Note that we have omitted listing symmetrical configurations to save space.) As before, we argue that cut edges weigh exponentially more than non-cut edges, because $(b, rbgy : rb, rb)$ (or the other symmetry) adjacent to itself forces the

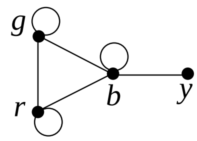
S, S' sets to be coloured b, but $(b, rbgy : rb, rb)$ adjacent to $(rb, rb : rbgy, b)$ enables the S, S' sets to be coloured in any of 2^t ways. We can prove the table is exhaustive by observing that, because H has a universal vertex, the $GoodPairs$ "exhaustiveness checking" technique from page 53 is valid. In this case $GoodPairs(H)$ consists of pairs $(\{b\}, \{r, b, g, y\})$, $(\{r, b\}, \{r, b\})$ and $(\{r, b, g, y\}, \{b\})$, and (after deleting symmetrical cases) the table lists all the various $GoodPairs$ combinations.

$A[u]$	$B[u] : B'[u]$	$A'[u]$	$edgespan$	approx	$q = 2p$
$\{r, b, g, y\}$	$\{b\} : \{b\}$	$\{r, b, g, y\}$	1	$16^p 1^q$	16^p
$\{r, b\}$	$\{r, b\} : \{b\}$	$\{r, b, g, y\}$	2	$8^p 2^q$	32^p
$\{r, b\}$	$\{r, b\} : \{r, b\}$	$\{r, b\}$	4	$4^p 4^q$	64^p
$\{b\}$	$\{r, b, g, y\} : \{b\}$	$\{r, b, g, y\}$	4	$4^p 4^q$	64^p
$\{b\}$	$\{r, b, g, y\} : \{r, b\}$	$\{r, b\}$	6	$2^p 6^q$	72^p
$\{b\}$	$\{r, b, g, y\} : \{r, b, g, y\}$	$\{b\}$	8	$1^p 8^q$	64^p



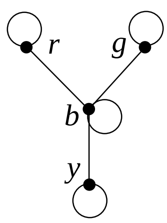
(Graph 57) Status: $\equiv_{\mathsf{AP}} \#SAT$. Again, we reduce from $\#LargeCut$, this time setting $q = 2.1p$. The table shows that the configuration $(bg, bgry : bgr, bgr)$ (and its symmetry) dominates, coming up approximately 923^p times. When the same symmetry is adjacent to itself, S, S' are forced to be coloured $\{b, g\}$, which can be done $2^t 2^t = 4^t$ ways. When opposite symmetries are adjacent, one of S, S' is coloured $\{b, g\}$ and the other $\{r, b, g\}$, thus contributing a superior 6^t. Accounting for all non-wasteful configurations is similar to the previous proof: $GoodPairs$ is also valid here. The $GoodPairs$ are $(\{b, g\}, \{b, g, r, y\})$, $(\{b, g, r, y\}, \{b, g\})$ and $(\{b, g, r\}, \{b, g, r\})$ and (modulo symmetries) we have listed all 6 cases.

$A[u]$	$B[u] : B'[u]$	$A'[u]$	$edgespan$	approx	$q = 2.1p$ (3sf)
$\{b, g, r, y\}$	$\{b, g\} : \{b, g\}$	$\{b, g, r, y\}$	4	$16^p 4^q$	294^p
$\{b, g, r\}$	$\{b, g, r\} : \{b, g\}$	$\{b, g, r, y\}$	6	$12^p 6^q$	517^p
$\{b, g, r\}$	$\{b, g, r\} : \{b, g, r\}$	$\{b, g, r\}$	9	$9^p 9^q$	908^p
$\{b, g\}$	$\{b, g, r, y\} : \{b, g\}$	$\{b, g, r, y\}$	8	$8^p 8^q$	630^p
$\{b, g\}$	$\{b, g, r, y\} : \{b, g, r\}$	$\{b, g, r\}$	11	$6^p 11^q$	923^p
$\{b, g\}$	$\{b, g, r, y\} : \{b, g, r, y\}$	$\{b, g\}$	13	$4^p 13^q$	874^p



(Graph 39) Status: $\equiv_{\mathsf{AP}} \#SAT$. Same reduction again, this time taking $q = 11p$. The dominant configuration is $(b, brgy : brg, brg)$ and its symmetrical equivalent. When the same symmetry is adjacent to itself the S, S' sets are forcibly coloured b but when opposite symmetries are adjacent, the S, S' sets can be coloured in 3^t ways. Exhaustiveness is guaranteed by the $GoodPairs$ technique; the $GoodPairs$ are $(\{b\}, \{b, rg, y\})$, $(\{b, r, g, y\}, \{b\})$ and $(\{b, r, g\}, \{b, r, g\})$ and (modulo symmetries) we have listed all possibilities.

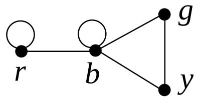
$A[u]$	$B[u] : B'[u]$	$A'[u]$	$edgespan$	approx	$q = 11p$ (3sf)
$\{b, r, g, y\}$	$\{b\} : \{b\}$	$\{b, r, g, y\}$	1	$16^p 1^q$	16.0^p
$\{b, r, g\}$	$\{b, r, g\} : \{b\}$	$\{b, r, g, y\}$	3	$12^p 3^q$	2.13×10^6
$\{b, r, g\}$	$\{b, r, g\} : \{b, r, g\}$	$\{b, r, g\}$	9	$9^p 9^q$	2.82×10^{11}
$\{b\}$	$\{b, r, g, y\} : \{b\}$	$\{b, r, g, y\}$	4	$4^p 4^q$	1.68×10^7
$\{b\}$	$\{b, r, g, y\} : \{b, r, g\}$	$\{b, r, g\}$	10	$3^p 10^q$	3.00×10^{11}
$\{b\}$	$\{b, r, g, y\} : \{b, r, g, y\}$	$\{b\}$	11	$1^p 11^q$	2.85×10^{11}



(Graph 23) Status: $\equiv_{\mathsf{AP}}\#SAT$. In [8] it was established that the above graph, the 3-particle Widom-Rowlinson graph (*3-WR*), was $\equiv_{\mathsf{AP}}\#BIS$-hard. However, it's exact complexity was not known. Here we show that it is in fact $\equiv_{\mathsf{AP}}\#SAT$ by virtue of a slightly modified $\#LargeCut$ reduction. G' is connected up in a very similar fashion to the previous three reductions, taking $q = p$, but there is an important difference in the way each $u \in V(G)$ is coded up. In previous reductions, for each vertex u we connected every vertex in $A[u]$ to every vertex in $B[u]$, every vertex in $A'[u]$ to every vertex in $B'[u]$, and introduced a "perfect matching" between $B[u]$ and $B'[u]$. In this reduction, we "invert" these connections: we connect every vertex in $B[u]$ to every vertex in $B'[u]$, and introduce perfect matchings between $A[u]$ and $B[u]$ and also between $A'[u]$ and $B'[u]$. The S, S' sets are connected up as before. We argue in due course that the only non-wasteful configurations are when $(A : B, B' : A')$ is coloured (modulo symmetry) with $(rbgy : b, rbgy : rbgy)$, or alternatively (wlog) $(rbgy : rb, rb : rbgy)$. The former configuration comes up $\nu(p, 4)\nu(p, 10) \approx 40^p$ times, because between the sets $\{r, b, g, y\}$ and $\{b\}$ there are 4 possible edges and between $\{r, b, g, y\}$ and $\{r, b, g, y\}$ there are 10 possible edges. In the same way, the configuration $(rbgy : rb, rb : rbgy)$ comes up $\nu(p, 6)\nu(p, 6) \approx 36^p$ times. Thus, all vertices are exponentially likely to have (B, B') coloured either $(b, rbgy)$ or $(rbgy, b)$. When the same symmetry is adjacent to itself the S, S' sets are forced to be b, but when different symmetries are adjacent S, S' can be coloured in 4^t ways, which is superior.

We now show that the two configurations discussed are the only non-wasteful configuration. For $X \subseteq V(H)$ let $reach(X)$ be the union of the adjacency sets of all the colours in X. In this context we say a configuration $(C_1 : C_2, C_3 : C_4)$ is wasteful iff at least one of the following conditions is *false*: $C_1 = reach(C_2)$, $C_4 = reach(C_3)$, $C_2 = H[C_3]$ and $C_3 = H[C_2]$. To see this, observe that if (say) $C_1 \subset reach(C_2)$ then we could upgrade C_1 to $reach(C_2)$, and this is guaranteed to give us more coloured edges to play with by the definition of $reach$. Alternatively, suppose $C_2 \subset H[C_3]$. Then we could upgrade C_2 to $H[C_3]$ and C_1 to $reach(H[C_3])$; this also gives us a larger choice of coloured edges, because (as long as there are no degree zero colours in H)

$reach(C_2) \subset reach(H[C_3])$. So, what transpires is that we can automatically disregard all configurations except those of the form $(reach(C_2) : C_2, C_3 : reach(C_3))$ where $(C_2, C_3) \in GoodPairs(H)$. The $GoodPairs$ are $(\{b\}, \{b, r, g, y\})$ and its symmetry, and the three structurally identical pairs of the form $(\{r, b\}, \{r, b\})$. Hence we are exhaustive.



(Graph 41) Status: $\equiv_{\mathsf{AP}}\#SAT$. *Exactly* the same reduction as above works here. The $GoodPairs$ are $(\{r, b\}, \{r, b\})$, $(\{b\}, \{r, b, g, y\})$, $(\{r, b, g, y\}, \{b\})$, $(\{b, g\}, \{b, y\})$ and $(\{b, y\}, \{b, g\})$. The configuration $(rbgy : rb, rb : rbgy)$ comes up (approximately) 36^p times, $(rbgy : bg, by : rbgy)$ also comes up 36^p times, but $(rbgy : b, rbgy : rbgy)$ and its symmetry are exponentially dominant, both coming up 40^p times. $\square$

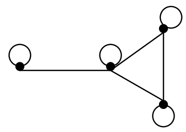
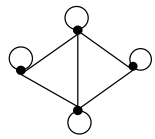
2.4 4-vertex H interreducible with $\equiv_{\mathsf{AP}}\#BIS$

This section looks at the four connected, 4-vertex H that are known to be $\equiv_{\mathsf{AP}}\#BIS$. These are the two weighted variants of *2-WR*, shown in the following diagram, and also P_4 and P_4^*. (Recall that *2-WR* and P_3^* are the same graph.) We generalise these results so that they apply to all weighted variants of the following graphs: *2-WR*, P_4, P_k^* $(k > 3)$, P_k $(k > 4)$. Most of this work is original, although in [8] reductions were demonstrated to prove the $\equiv_{\mathsf{AP}}\#BIS$ status of unweighted P_k^* $(k \geq 3)$, and earlier research used the "crossing" result - discussed in Chapter 7 - to show that weighted P_k $(k \geq 4)$ are $\equiv_{\mathsf{AP}}\#BIS$-easy[15]. Lemma 2.14 was proven in conjunction with Dyer and Goldberg.

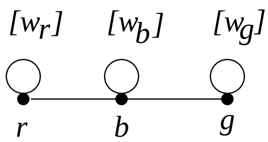
Before proceeding, it is important to note that the $\equiv_{\mathsf{AP}}\#BIS$-hardness reductions in this section, as with all reductions of the form $\#BIS\leq_{\mathsf{AP}}\#H$, are in some

[15] To clarify the meaning of $\equiv_{\mathsf{AP}}\#BIS$-easy, recall from Section 1.2.4 that - because we are dealing solely in terms of AP-reducibility, saying that a problem is $\equiv_{\mathsf{AP}}\#BIS$-easy has the same meaning as saying that it is $\#BIS$-easy.

sense superseded by Theorem 4.1 from Chapter 4. However, as is noted in both Chapters 3 and 4, it is not known whether this result - which essentially states that, if you can approximately sample H-colourings[16], you can both approximately sample and count bipartite independent sets - transfers wholly to the approximate counting domain. Hence, while this is not apparent, there is value in demonstrating explicit $\equiv_{\mathsf{AP}}\#BIS$-hardness reductions.

Both the above graphs (which are graphs 45 and 60 in the graph index) are $\equiv_{\mathsf{AP}}\#BIS$. Rather than classify each graph individually we prove instead that all weighted versions of *2-WR* are $\equiv_{\mathsf{AP}}\#BIS$. That is, for fixed $w_r, w_b, w_g \in \mathbb{Q}^+$ the graph given (in compact form) by $V(H) = \{r, b, g\}, E(H) = \{\{r,r\},\{b,b\},\{g,g\},\{r,b\},\{b,g\}\}$ and $w(r) = w_r$, $w(b) = w_b$, $w(g) = w_g$ is $\equiv_{\mathsf{AP}}\#BIS$.



To put one technicality aside, we assume that (without loss of generality) w_r, w_b, w_g are all in $\mathbb{N}^+$; Observation 2.2 permits us to do this.

Owing to the fairly math-heavy nature of the next few proofs, we introduce a mild, temporary abuse of notation and (where there is no confusion) use r, b, g interchangeably with w_r, w_b, w_g. We acknowledge that this is suspect practice but it has the major benefit of making the fairly dense maths coming up more intuitive and readable.

2.4.1 Weighted versions of 2-WR and P_k^* $(k > 3)$ are $\equiv_{\mathsf{AP}}\#BIS$

We first prove that $\#H \leq_{\mathsf{AP}} \#BIS$ for all H that are weighted versions of *2-WR*, by reducing $\#H$ to the $\equiv_{\mathsf{AP}}\#BIS$-easy problem $\#DownSets$, a reduction technique first applied in [8]. To recap, the input to $\#DownSets$ is a partial order $(X, \prec)$, and the

[16] Where H has no trivial components

solution is the number of "downsets" in that partial order. A downset is any $Y \subseteq X$ such that, for all $g, h \in X$, $((g \prec h) \wedge (h \in Y) \Rightarrow (g \in Y))$. The principle manner in which $\#H$ is reduced to an instance of $\#DownSets$ is by uniformly encoding each vertex $u \in V(H)$ as a constant-size partial order $D[u]$, and each edge $\{u, v\}$ as additional relational $\prec$ constraints between $D[u]$ and $D[v]$. The encoding of each vertex u as a constant-size partial order means that, within $D[u]$, there are a finite number of downsets possible[17]. Furthermore, the encoding of edges means that not all downsets can be adjacent to each other. Thus, each possible downset within a vertex encoding defines a "colour", and the edge encoding defines which "colours" are allowed to be adjacent to each other. (There is a clarifying example in Appendix A.7 which shows how *#2-WR* can be coded up using $\#DownSets$.)

Note that Lemmas 2.7 and 2.8 (and subsequently 2.10) generalise a proof technique from [8] which was used to establish the $\equiv_{\mathsf{AP}}\#BIS$ status of the graph *2-WR*.

Lemma 2.7 *Weighted versions of 2-WR are* $\equiv_{\mathsf{AP}}\#BIS$*-easy*

Proof. We encode an input graph G (as an input $(X, \prec)$ to $\#DownSets$) as follows. For each vertex $u \in V(G)$, encode $D[u]$ as partial order with elements $\{u_i | 1 \leq i \leq r + b + g - 1\}$, and relational constraints $\{u_i \prec u_j | i < j\}$. For each edge $\{u, v\} \in E(G)$, we introduce relational constraints $u_g \prec v_{b+g}$ and $v_g \prec u_{b+g}$. This encodes a 1:1 correlation between $H(G)$ and $DownSets(X, \prec)$, which immediately gives us $\#H \leq_{\mathsf{AP}} \#DownSets$. To see this, note that within each $D[u]$ downsets can be characterised by the index of the topmost element "in" the downset. Thus, the w_r downsets designated by $\{u_{b+g}, u_{b+g+1}, ..., u_{b+g+r-1}\}$ act like the colour r, the w_b downsets designated by $\{u_g, u_{g+1}, ..., u_{g+b-1}\}$ act like b, and the remaining w_g downsets $\{\emptyset, u_1, u_2, ..., u_{g-1}\}$ act like g. $\Box$

[17] Technically it is incorrect to speak of those downsets possible *within* a vertex encoding, because by definition a downset is defined in terms of the *whole* partial order, not a subset of it. However, we tolerate this abuse of terminology because it is useful for communicating the intuitive behaviour of the proof.

Lemma 2.8 *Weighted versions of 2-WR are* $\equiv_{\mathsf{AP}}\#BIS$*-hard*

Proof. The $\equiv_{\mathsf{AP}}\#BIS$-hardness of weighted *2-WR* graphs is now proven. We do this by reducing the problem $\#MaxBIS$ to $\#H$; $\#MaxBIS$ is the problem of counting all the *maximum*-size independent sets in bipartite G, and is shown to be interreducible with $\#BIS$ in [8]. Unlike $\#LargeIS$, its non-bipartite counterpart, there is no need to encode the size of the maximum-size independent set as part of the input. This is because the maximum size of an independent set in a bipartite graph G can be computed in polynomial time using a *network flow* algorithm.

Now, if $r = b = g$ then the result is immediate, because by Lemma 2.1 this would mean $\#H\equiv_{\mathsf{AP}}\#$*2-WR*, and we know already that $\#$*2-WR*$\equiv_{\mathsf{AP}}\#BIS$. Otherwise, the three possibilities are $r + b < b + g$, $b + g < r + b$, or $r + b = b + g$. (The first two cases are symmetries of each other and hence equivalent.) We assume for the time being that $r + b < b + g$. The input to $\#MaxBIS$ is bipartite so we can assume $G = (V_L(G), V_R(G), E(G))$ where $V_L(G) = \{u_0, u_1, ..., u_{n_l-1}\}$ and $V_R(G) = \{v_0, v_1, ..., v_{n_r-1}\}$. (Let $n = n_l + n_r$.) We encode G', the input to $\#H$, as follows, where s, t and q are to be determined. For each $u_i \in V_L(G)$ we introduce an independent set U_i of size s. For each vertex $v_i \in V_R(G)$ we introduce an independent set V_i of size t. For each edge $\{u_i, v_j\}$ we connect every vertex in U_i to every vertex in V_j. Finally, a clique K of size p is added (where p is to be determined), and for each U_i every vertex in U_i is connected to every vertex in K. The idea is that, for large enough p, K is exponentially likely to be coloured with $\{b, g\}$ because (as a result of their weighting) these colours form the largest looped clique in H. Hence, colours allowed to appear in each U_i are $\{b, g\}$, whilst each V_i can take colours from $\{r, b, g\}$. Note that a U_i that contains g cannot be adjacent to a V_i that contains r. So, we say each u_i is IN the independent set if U_i contains g, and OUT if it does not, and that each v_i is IN the independent set if V_i contains r, and OUT if it does not. Intuitively we see that (wlog) a U_i coloured in such a way that it represents a vertex IN the independent set comes up more frequently than when it represents a vertex OUT of the independent set i.e. because it can be coloured with two colours as opposed to one. Informally, that is why this reduction is from $\#MaxBIS$ i.e. because maximum-size independent sets in G come up exponentially most often as H-colourings of G'.

To make this reduction work, it is necessary (but not sufficient) to show that p can be chosen large enough such that K contains, in most cases, at least one g vertex. This is not difficult and is left until later. Of more importance, however, is resolving the problem that if (say) s is left simply as being equal to t, the contribution of an independent set (in terms of the number of times it comes up as a colouring of G') is sensitive to the way its vertices are divided between $V_L(G)$ and $V_R(G)$; in this instance, setting $s = t$ would mean there was an exponential bias towards independent sets with most of their IN vertices being from $V_R(G)$. This bias can be neutralised through an appropriate choice of s and t. However, actually choosing s and t is technically quite tricky and this explains the length of the analysis that follows.

First, however, observe that an independent set with k vertices from $V_L(G)$ and l vertices from $V_R(G)$ comes up the following number of times (using the model we described above):-

$$((b+g)^p - b^p)((b+g)^s - b^s)^k b^{s(n_l-k)}((r+b+g)^t - (b+g)^t)^l (b+g)^{t(n_r-l)}$$

Rearranging,

$$\Big((b+g)^p - b^p\Big)\left(\left(\frac{b+g}{b}\right)^s - 1\right)^k \left(\left(\frac{r+b+g}{b+g}\right)^t - 1\right)^l b^{sn_l}(b+g)^{tn_r}$$

Now, if we define constants $A = \frac{b+g}{b}$ and $B = \frac{r+b+g}{b+g}$, and let $m = k + l$ (i.e. m is the total size of the independent set), and multiply through by

$$\frac{(B^t-1)^k}{(B^t-1)^k}$$

this expression becomes

$$\Big((b+g)^p - b^p\Big)\left(\frac{A^s-1}{B^t-1}\right)^k (B^t-1)^m b^{sn_l}(b+g)^{tn_r} \tag{2.6}$$

Since G is bipartite we can pre-compute the maximum size of an independent set in G, let this value be M. We define the full colourings of G' to be those that point out size-M independent sets. Ignoring, for a moment, the term exponential in k (which we shortly demonstrate is *very* close to 1), the idea is that dividing $\#\hat{H}(G')$ by

$$Z = \Big((b+g)^p - b^p\Big)(B^t - 1)^M b^{sn_l}(b+g)^{tn_r}$$

and rounding gives a good approximation to the number of maximum-size independent sets in G.

Let ϵ be the accuracy that we have to approximate $\#MaxBIS(G)$ to. Now, suppose we can choose p, s and t such that simultaneously we have $p = n^2(s+t)$, $t \geq n^2$ and, for all $k \in [0, n]$,

$$e^{-\epsilon/42} \leq \left(\frac{A^s - 1}{B^t - 1}\right)^k \leq e^{\epsilon/42} \tag{2.7}$$

(We show how to meet these conditions at the end of the proof, but for now let's assume they have been met.) Consider the set $S \subseteq H(G')$ of non-full colourings in $H(G')$. S can be partitioned into disjoint sets S^- and S^+ where S^- contains colourings in which K does not contain g and S^+ contains colourings in which K does contain g but the independent set of G pointed out is of size less than M. We can show that $|S|/Z \leq 1/4$ by showing $|S^-|/Z \leq 1/8$ and $|S^+|/Z \leq 1/8$. First, consider S^-; it is sufficient that

$$\frac{(r+b)^p(r+b+g)^{n_l s + n_r t}}{(b+g)^p - b^p} \leq 1/8 \tag{2.8}$$

Since $b+g > r+b$, we can easily satisfy this (for n bigger than some fixed constant) by choosing p to be a bigger polynomial than $n_l s + n_r t$. An upper bound on $n_l s + n_r t$ is $n(s+t)$ so our choice of $p = n^2(s+t)$ is crude but entirely adequate to satisfy this. Now, we have to show $|S^+|/Z \leq 1/8$. A crude upper bound on $|S^+|$ can be obtained by assuming that there are 2^n valid independent sets in G, each pointing out an independent set of size $M-1$, and that for each such independent set the stray term $((A^s-1)/(B^t-1))^k$ is as large as can be i.e. $e^{1/42}$. (The $e^{1/42}$ value emerges from combining the upper bound given by (2.7) with the observation that $\epsilon \leq 1$.) Given that

$e^{1/42} < 2$, the core inequality to satisfy is:

$$\frac{2^n(B^t-1)^{M-1}2}{(B^t-1)^M} \le 1/8$$

This can be easily satisfied by choosing t to be a superlinear polynomial in n. Thus, for n beyond a fixed constant threshold our choice of $t \ge n^2$ is fine. So we know that $|S|/Z \le 1/4$.

Ideally s and t would be such that $A^s - 1 = B^t - 1$, because it would then follow that the number of times an independent set comes up is independent of k and l, and dependent solely on the overall size of the independent set. Unfortunately, however, the solution to $A^s = B^t$ may generally require either s or t to take an irrational value, which is of course not possible, because s and t must be discrete. So we now have to show that, while respecting the requirement that s and t be no larger than a polynomial in n and ϵ^{-1}, we can achieve an approximate solution to $A^s = B^t$ which is "good enough" for the purposes required. We claim that, along with $p = n^2(s+t)$ and $t \ge n^2$, the fact that (2.7) holds (for all $k \in [0,n]$) is adequate for a "good enough" solution.

At this stage it is important to clarify exactly what is required. Referring back to the definition of the "rounding" technique described in Section 2.1.3, this essentially shows how to obtain an approximation to an answer N in the range $[e^{-\epsilon}N, e^{\epsilon}N]$ from a value in the range $[e^{-\epsilon/21}(N-\frac{1}{4}), (N+\frac{1}{4})e^{\epsilon/21}]$. In most reductions we do not refer to this explicitly because $\lfloor \#H(G)/Z \rfloor$ is exactly equal to N. However, owing to the fact that $((A^s-1)/(B^t-1))$ is unlikely to be exactly 1 we have no choice in this instance but to get our hands dirty with technical details. To use the rounding technique, we need the following condition to hold:

$$e^{-\epsilon/21}(N-\frac{1}{4}) \le \frac{\#\hat{H}(G')}{Z} \le e^{\epsilon/21}(N+\frac{1}{4}) \qquad (2.9)$$

(Where N is the number of maximum-size independent sets in G and $\#\hat{H}(G')$ is the value returned by our approximation oracle.) Now, at present we know that, because $p = n^2(s+t)$, $t \ge n^2$ and (2.7) holds for all $k \in [0,n]$, the following inequality is true:

$$Ne^{-\epsilon/42} \le \frac{\#H(G')}{Z} \le Ne^{\epsilon/42} + \frac{1}{4} \qquad (2.10)$$

The $e^{\epsilon/42}$ and $e^{-\epsilon/42}$ terms appear because they are upper and lower bounds (respectively) on the term $((A^s-1)/(B^t-1))^k$, as given by (2.7). A second source of inaccuracy creeps in because our $\#H$ oracle is only approximate. However, if we use $\delta = \epsilon/42$ as the accuracy parameter we pass to our $\#H$ oracle it follows from (2.10) that:

$$Ne^{-\epsilon/42}e^{-\epsilon/42} \leq \frac{\#\hat{H(G')}}{Z} \leq e^{\epsilon/42}\left(Ne^{\epsilon/42} + \frac{1}{4}\right)$$

This implies that (2.9) holds so we are done. It remains only to show that we can choose s and t such that $t \geq n^2$ and that (2.7) holds for $k \in [0, n]$. (Once we have chosen s and t we can just set $p = n^2(s+t)$ so the constraint on p is easy to satisfy.) First, we observe that proving (2.7) holds for $k = n$ automatically proves that it holds for all $k \in [0, n]$. To see why this is, observe that $A^s - 1$ can either be equal to $B^t - 1$, less than it, or greater than it. If $A^s - 1 = B^t - 1$ then (2.7) trivially holds for all $k \in [0, n]$. However, suppose (wlog) $A^s - 1 < B^t - 1$. This means $((A^s-1)/(B^t-1))^k \leq 1$ for all $k \in [0, n]$, and as a consequence the $e^{1/42}$ bound is trivially satisfied for all $k \in [0, n]$. (Recall that $1 < e^{1/42}$.) Regarding the lower bound we know that, because $((A^s-1)/(B^t-1)) < 1$, the term $((A^s-1)/(B^t-1))$ is smallest for $k = n$, so satisfying the lower bound of (2.7) for $k = n$ automatically satisfies it for $k \leq n$. The case when $A^s - 1 > B^t - 1$ is symmetrical; this time the lower bound is trivially satisfied, and the upper bound is satisfied for $k \leq n$ because the term $((A^s-1)/(B^t-1))^k$ is largest for $k = n$.

Hence, we can satisfy (2.7) by proving that:

$$e^{-\epsilon/42} \leq \left(\frac{A^s - 1}{B^t - 1}\right)^n \leq e^{\epsilon/42}$$

Furthermore, observe that an alternative way of writing this is

$$e^{-\epsilon/42} \leq \left(\frac{B^{zs} - 1}{B^t - 1}\right)^n \leq e^{\epsilon/42} \tag{2.11}$$

where $z = \log_B(A)$. So let W be a positive integer such that $\lfloor zW \rfloor \geq n^2$. Let $R = \lceil 16(\ln(B))Wn/(7\delta)\rceil$. A Corollary from [8] provides the useful result:

Corollary 2.9 (DGGJ) *For any positive integer W and any positive integer R, there is an $x \in [1, ..., R]$ such that*

$$\mathit{min}(zWx - \lfloor zWx \rfloor, \lceil zWx \rceil - zWx) \leq W/R$$

(We do not reproduce the proof of this corollary here.) We set $s = Wx$ where x is the value yielded by the above Corollary. Now, t can take one of two possible values. If it is $\lfloor zWx \rfloor$ that is closer to zWx set $t = \lfloor zWx \rfloor$, otherwise set $t = \lceil zWx \rceil$. (Note that in either case $t \geq n^2$.) We must now show that either choice of t is good enough to satisfy (2.11). First, we define $\gamma = \epsilon/42n$.

For the case where $t = \lfloor zWx \rfloor$, $((B^{zs} - 1)/(B^t - 1)) > 1$, so the lower bound of (2.11) is trivially satisfied, but for the upper bound:

$$\ln(B)(zs - \lfloor zs \rfloor) \leq \ln(B)W/R \leq 7\gamma/16 \leq \ln(1 + \gamma/2)$$

(The rightmost inequality relies on the fact that $\gamma < 1/2$.) Exponentiating both sides,

$$B^{zs} \leq B^{\lfloor zs \rfloor}(1 + \gamma/2) \leq B^{\lfloor zs \rfloor} + \gamma(B^{\lfloor zs \rfloor} - 1)$$

(Technically speaking, the RHS inequality in the above requires $B^{\lfloor zs \rfloor} \geq 2$. This is true for sufficiently large n because $B > 1$ and $\lfloor zs \rfloor = \lfloor zWx \rfloor \geq \lfloor zW \rfloor \geq n^2$.) Thus,

$$\frac{B^{zs} - B^{\lfloor zs \rfloor}}{B^{\lfloor zs \rfloor} - 1} \leq \gamma$$

Adding 1 to both sides,

$$\frac{B^{zs} - 1}{B^{\lfloor zs \rfloor} - 1} \leq 1 + \gamma \leq e^{\gamma}$$

When $t = \lceil zWx \rceil$, $((B^{zs} - 1)/(B^t - 1)) < 1$, so in this instance it is the lower bound of (2.11) which is non-trivial to satisfy. We omit the details of this because it is directly analogous to the above. This completes the proof for the case $r + b < b + g$ (and by symmetry the case when $r + b > b + g$.)

To complete the overall proof, we need to consider the case when $r + b = b + g$. This is actually slightly easier to deal with; in this case we let full colourings be any in which K is coloured with more colours than just b (and size-M independent sets are pointed out.) Observing that $((b + g)^p - b^p) + ((r + b)^p - b^p) = 2((b + g)^p - b^p)$, (2.8) therefore becomes

$$\frac{b^p(r + b + g)^{n_l s + n_r t}}{2((b + g)^p - b^p)} \leq 1/8$$

Again, a choice of $p = n^2(s + t)$ is more than adequate for this purpose. The proof thereafter is virtually identical to the proof for $r + b < b + g$ except for the fact that the value of Z used is exactly twice as large as previously. This stems from the fact that a full colouring can either have K coloured $\{r, b\}$ or $\{b, g\}$, both of which behave identically. □

Lemma 2.10 *All weighted versions of 2-WR are* $\equiv_{\mathsf{AP}}\#BIS$

Proof. Immediate by combining Lemma 2.7 and Lemma 2.8. □

In fact, Lemma 2.10 can be generalised to apply to to any weighted version of P_k^* for[18] $k \geq 3$. First we show that such graphs are $\equiv_{\mathsf{AP}}\#BIS$-easy.

Lemma 2.11 *Weighted versions of P_k^* for $k > 3$ are $\equiv_{\mathsf{AP}}\#BIS$-easy*

Proof. We use H in compact form so (from left to right) let the colours of H be $\{c_1, c_2, ..., c_k\}$ and let w_i be the value of $w(c_i)$. As in the proof of Lemma 2.7 we reduce to $\#DownSets$. Each vertex in G is encoded in the same "chain" style as in Lemma 2.7, but this time with $w_1 + w_2 + ... + w_k - 1$ elements. Now, for $1 \leq j < (k-1)$, define $p[j]$ and $q[j]$ as follows:

$$p[j] = \sum_{1 \leq l \leq j} w_l$$

$$q[j] = p[j] + w_{j+1}$$

For each edge $\{u, v\}$ we add the relational constraints

$$\bigcup_{1 \leq j < (k-1)} \left\{ (u_{p[j]} \prec v_{q[j]}), (v_{p[j]} \prec u_{q[j]}) \right\}$$

There is a 1:1 correlation between $DownSets(X, \prec)$ and the weighted P_k^* colourings we wish to count. □

Lemma 2.12 *Weighted versions of P_k^* for $k > 3$ are $\equiv_{\mathsf{AP}}\#BIS$-hard*

Proof. Let H be any weighted version of P_k^* (for $k > 3$). We show that $\#BIS \leq_{\mathsf{AP}} \#H$ by proving $\#H' \leq_{\mathsf{AP}} \#H$, where H' is some uniquely specified, weighted version of

[18] This covers the $\equiv_{\mathsf{AP}}\#BIS$-easiness of P_4^*, the other 4-vertex graph interreducible with $\#BIS$. All P_q^* $(q \geq 3)$ were originally shown to be interreducible with $\#BIS$ in [8].

2-WR. We know from Lemma 2.8 that all such H' are $\equiv_{\mathsf{AP}}\#BIS$-hard, so (by transitivity) this gives us the result we require. The exact form of the graph H' is discussed in due course.

First, we assume that H is in compact form, so from left-to-right the colours of H are $c_1, c_2, ..., c_k$, and the corresponding weights on these colours are $w_1, w_2, ..., w_k$. (The total number of colours in the expanded graph is therefore $|V'(H)| = \sum w_i$.) Informally, we know that if we can pick out any c_i (where $2 \le i \le k-1$) then we are close to a result because we can then point out $H[c_i]$, a graph we already know to be $\equiv_{\mathsf{AP}}\#BIS$-hard (owing to the fact that it is a weighted variant of *2-WR*.) However, as we explain in the remark at the end of the proof, we need to know a priori *exactly* what this pointed-out graph will look like i.e. we need to ensure that all the various $H[c_i]$ pointed out by our gadgetry are isomorphic. This isomorphism requirement explains why this proof is quite long, and why it uses such (relatively) complex gadgetry. Essentially, we are going to build a gadget that firstly prioritises vertex degree, then vertex weight, and then the adjacency of the vertex to looped cliques.

Let G be the input to $\#H'$; we construct G' as follows. A new vertex x is introduced and attached to every vertex in a copy of G. Two independent sets I_1 and I_2, of size p, q respectively are introduced, and a complete graph on t vertices, K, is also added. We connect x to every vertex in I_1, to every vertex in I_2, and to every vertex in K. Finally, we connect every vertex in I_1 to every vertex in I_2. We specify exact values for p, q and t later in this proof, but at this stage it is sufficient to recognise that they are chosen such that $p \gg q \gg t \gg n$. Recalling the weighting-related definitions from Section 2.2.1[19], we argue that x is exponentially likely to be coloured with any colour $c \in V(H)$ for which property 3 (from the following list of properties) holds:-

1. No colour $c' \in V(H)$ has an effective degree higher than c;

2. Property 1 holds, plus no colour c' of the same effective degree as c has weight greater than c;

[19] Recall in particular that, for H given in compact form, the *effective* degree of $c \in V(H)$ is defined to be equal to $deg'(c)$, which is itself equal to the sum of the weights of all the colours in $adj(c)$.

3. Property 2 holds, plus no colour c' of the same effective degree and weight as c is adjacent to a weightier looped clique than c is adjacent to.

(Note that the above properties are closed downwards.) It is necessary to clarify the meaning of the terminology used in property 3. For H in compact form, a vertex $c \in V(H)$ is adjacent to a looped clique of weight at least W iff $S \subseteq H[c]$ where S is a looped clique and the sum of the weights of the colours in S is at least W. So, the weightiest looped clique adjacent to c_1 is of weight $w_1 + w_2$, the weightiest looped clique adjacent to c_k has weight $w_{k-1} + w_k$, and for $i \in \{2, 3, ..., k-1\}$ the weightiest looped clique adjacent to c_i has weight $\max(w_{i-1} + w_i, w_i + w_{i+1})$.

To show how the reduction works, we must show that, for all c, c' that satisfy the above criteria, $H[c]$ is isomorphic to $H[c']$ - thus unambiguously defining H' - and secondly that the construction of G' does indeed make x exponentially likely to be coloured with one of the colours satisfying this criteria. (This makes it exponentially likely that H' colourings are pointed out in G.) First, note that the colours in H with maximum effective degree cannot be either c_1 or c_k, so as a result it is not possible to point out a trivial subgraph of H. This follows because the effective degree of a colour $c_i \in V(H)$ is equal to the sum of its weight and the sum of its neighbours' weights. Hence, c_1 is of effective degree $w_1 + w_2$ and c_k is of effective degree $w_{k-1} + w_k$. However, the effective degree of c_2 is $w_1 + w_2 + w_3$, and the effective degree of c_{k-1} is $w_{k-2} + w_{k-1} + w_k$, so neither c_1 nor c_k can be of maximum effective degree. Consider, then, two colours c_i and c_j ($i \neq j$, $i, j \in \{2, ..., k-1\}$) that both satisfy the above three properties. Since c_i and c_j both have the same effective degree, we know that $w_{i-1} + w_i + w_{i+1} = w_{j-1} + w_j + w_{j+1}$. Furthermore, by property 2 we know that $w_i = w_j$, so $w_{i-1} + w_{i+1} = w_{j-1} + w_{j+1}$. By property 3 and the definition of "looped clique adjacency" just given, it must be the case that (wlog) $w_{i-1} + w_i = w_j + w_{j+1}$, which means that $w_{i-1} = w_{j+1}$, and applying this to the earlier expression gives $w_{i+1} = w_{j-1}$. This proves that $H[c]$ and $H[c']$ are isomorphic.

We now show that x is exponentially likely to be coloured with a colour satisfying the above properties. We set $p = n^4$, $q = n^3$ and $t = n^2$. Let α be the maximum effective degree of a colour in H; let β be the maximum weight of colours with effective degree α, and let γ be the weight of the weightiest looped clique that any colour with effective degree α and weight β is adjacent to. We partition $V(H)$ into 4 categories: Max_0 contains those colours that do not satisfy property 1 and, (for $i \in \{1,2,3\}$), Max_i contains those colours for which the highest property satisfied is property i. Let C_i represent the sum of the weights of the colours in Max_i; clearly $C_i \leq |V'(H)|$. Finally, we partition $H(G')$ into sets $H_i(G')$ where $H_i(G')$ are those colourings with x mapped to some colour from Max_i. It follows, therefore, that $\#H_3(G') = \#H'(G)Z$, where $Z = C_3 Z'$ and Z' is the number of colourings possible in I_1, I_2 and K if we assume x is mapped to a colour from Max_3. Note that it is not difficult to compute the exact value of Z in polynomial time, and indeed that it is necessary to do so because later in the reduction we divide $\#\hat{H}(G')$ by Z. To see that it is easy to compute Z exactly, observe that the contribution of any one individual configuration is easy to enumerate exactly, and even if we assume every subset of $V'(H)$ is possible in each of I_1, I_2 and K this still leaves us with at most a constant number of distinct configurations i.e. $2^{3|V'(H)|}$. (It is worth mentioning that there is no complexity-theoretic reason why Z has to be computed exactly as part of the reduction; however, it is useful to do so because it drastically cuts down on the amount of analysis required to demonstrate the correctness of the reduction.)

For the purpose of *analysis* the main point to note is that

$$Z' \geq \alpha^{n^4} \beta^{n^3} \gamma^{n^2}$$

It is important to remember that (wlog) we assume all the w_i are in $\mathbb{N}^+$, and hence it follows that α, β, γ are also in $\mathbb{N}^+$. Also, we observe that $\alpha > 1$, $\beta \geq 1$ and $\gamma > 1$. We know $\alpha > 1$ because otherwise H would be trivial. Likewise, we know $\gamma > 1$ because all $c_i \in V(H)$ are next to a looped clique of at least weight 2. However, there are H for which β may legitimately be 1, and this leads to a very mild technical problem which we tackle later on.

Now, if we can show that, for $i \in \{0, 1, 2\}$, $\#H_i(G')/Z \leq (1/12)$, it follows that

$$\#H'(G) \leq \frac{\#H(G')}{Z} \leq \#H'(G) + (1/4) \tag{2.12}$$

which is necessary for the usual rounding technique to work. First we consider $\#H_0(G')/Z$.

$$\frac{\#H_0(G')}{Z} \leq \frac{C_0(\alpha-1)^{n^4}|V'(H)|^{n^3+n^2+n}}{\alpha^{n^4}} \leq \left(\frac{\alpha-1}{\alpha}\right)^{n^4} |V'(H)|^{n^3+n^2+n+1}$$

(The $|V'(H)|^n$ term appears because we assume any colouring is possible in G. Additionally, we know $\alpha > 1$ because otherwise H would be trivial, so the terms above are non-zero.) Clearly the RHS of the above inequality drops below 1/12 for large enough n. The situation with $\#H_1(G')$ and $\#H_2(G')$ is slightly different. Consider $H_1(G')$ colourings, for example: these can be partitioned into those colourings where I_1 contains α colours, and those where I_1 contains fewer than α colours. Thus,

$$\#H_1(G') \leq C_1\alpha^{n^4}(\beta-1)^{n^3}|V'(H)|^{n^2+n} + C_1 2^{|V'(H)|}(\alpha-1)^{n^4}|V'(H)|^{n^3+n^2+n}$$

The right-hand term emerges because we generously assume that any subset of $V'(H)$ is possible in I_1, and that each such colouring comes up $(\alpha-1)^{n^4}|V'(H)|^{n^3+n^2+n}$ times. Note, however, that the right-hand term in the above expression is a vanishing fraction of the whole expression, because the index of its biggest exponent is only $(\alpha-1)$. In other words, there is never any "benefit" to be gained by holding colours back in a larger gadget with a view to making more colours available in a later, smaller gadget...the term with the leading exponential devours all. We call configurations that fail to use all the colours available to them in a larger gadget *misguided* configurations. Note that, when attempting to enumerate the contribution of misguided configurations, a constant multiplicative factor (such as $2^{|V'(H)|}$ in this instance) is introduced. However, this can clearly not compete against a quantity exponentially small in the size of the input, such as $((\alpha-1)/\alpha)^{n^4}$.

When developing upper bounds, therefore, we often absorb the contribution of misguided configurations by (somewhat lazily) putting a factor of 2 in front of the leading term, c.f

$$\frac{\#H_1(G')}{Z} \leq \frac{2C_1\alpha^{n^4}(\beta-1)^{n^3}|V'(H)|^{n^2+n}}{\alpha^{n^4}\beta^{n^3}} \leq \left(\frac{\beta-1}{\beta}\right)^{n^3}|V'(H)|^{n^2+n+2}$$

Again, this clearly drops below[20] 1/12 for sufficiently large n.

A similar argument applies to $\#H_2(G')$:-

$$\frac{\#H_2(G')}{Z} \leq \frac{2C_2\alpha^{n^4}\beta^{n^3}(\gamma-1)^{n^2}|V'(H)|^{n}}{\alpha^{n^4}\beta^{n^3}\gamma^{n^2}} \leq \left(\frac{\gamma-1}{\gamma}\right)^{n^2}|V'(H)|^{n+2}$$

Finally, we see that once n is larger than the largest of the three threshold values (i.e. the constant value beyond which the RHS of the inequality drops below 1/12) (2.12) holds and $\#H'(G)$ can be calculated in the usual manner, using the "rounding" technique. □

The above reduction is a classic example of the *descending polynomial* principle:- by combining gadgets of decreasing size, we can progressively refine the behaviour of the overall gadgetry. The one crucial point to take away from this reduction is that, because of the dominance of larger gadgets over smaller gadgets, maximal configurations are (in some sense) *forced* to colour their larger gadgets with as many colours as possible. This has the interesting effect of forcing certain conditions on smaller gadgets, and often allows us to remove surplus gadgetry. For example, in the above reduction there was technically no need to connect x to every vertex in I_2; the original reason for doing this was to ensure I_2 used colours only in the neighbourhood of the colour c_i on x. However, we needn't have done this, because we know that any maximal configuration must colour I_1 with α colours, which in turn forces I_2 to be restricted to the same colour as that used on x. Finally, the structure comprising I_1 and I_2 is of general use and we henceforth name it a *maxdegweight* gadget, because (as we have just shown) it has the property of picking out those maximum degree colours which, amongst all the maximum degree colours, have the highest weight[21].

[20] The reader may have noticed that if $\beta = 1$ the above terms drop to zero. To see that this is not a problem observe that if $\beta = 1$, $Max_1 = \emptyset$ - because every c_i satisfying property 1 automatically also satisfies property 2 - and hence $\#H_1(G') = 0$.

[21] Note that removing the redundant edges between x and I_2 ensures that the gadget functions correctly when H contains unlooped colours.

Remark. It is important to explain why we go to such trouble to point out a *unique* weighted version of *2-WR*, when we know already that all weighted versions of *2-WR* are $\equiv_{\mathsf{AP}}\#BIS$-hard. If we revisit Lemma 2.8 then we see that the values of s,t,q used in the proof of that lemma are conditioned by the exact structure of the graph we are reducing to. If we can't say the exact structure of H' *a priori* (where H' is the subgraph pointed out by the above reduction), then how are we supposed to choose s,t and q appropriately when looking to prove $\#BIS \leq_{\mathsf{AP}} \#H'$? This is a major (apparent) limitation of AP-reductions. In Chapter 3 we introduce the much more general SP-reduction, which has been designed with situations such as this directly in mind.

Lemma 2.13 *Weighted versions of P_k^* for $k > 3$ are $\equiv_{\mathsf{AP}}\#BIS$*

Proof. Immediate from the combination of Lemmas 2.11 and 2.12. □

2.4.2 All weighted versions of P_4 and P_k $(k > 4)$ are $\equiv_{\mathsf{AP}}\#BIS$

As in Figure 2.6 on page 55, label the two centre colours of P_4 b, b' and the end colours r, r'. As before, we introduce a temporary abuse of notation and, when referring to weighted versions of P_4, let these labels also refer to the weight of the corresponding colour. (As usual for weighting-related results, we assume compact form is being used.) Applying the same observation as in the proof for weighted *2-WR*, we can assume $r, b, b', r \in \mathbb{N}^+$.

We first prove the $\equiv_{\mathsf{AP}}\#BIS$ status of all weighted versions of P_4, and then show how this generalises to weighted versions of P_k for $k > 4$. (Lemma 2.14 was formulated in conjunction with Dyer and Goldberg.)

Lemma 2.14 *[Dyer, Goldberg, Kelk] Weighted versions of P_4 are $\equiv_{\mathsf{AP}}\#BIS$-easy*

Again, we reduce to $\#DownSets$. We exploit the fact that the input to $\#H$, G, is bipartite. That is, we encode vertices from $V_L(G)$ in a different manner to vertices from

$V_R(G)$. For each vertex $u_i \in V_L(G)$, we introduce $r+b-1$ partial order elements, $\{p_{i,0}, p_{i,1}, ..., p_{i,r+b-2}\}$, with $p_{i,s} \prec p_{i,t}$ for $s < t$. For each vertex $v_i \in V_R(G)$, we introduce $r'+b'-1$ partial order elements, $\{q_{i,0}, q_{i,1}, ..., q_{i,r'+b'-2}\}$, with $q_{i,s} \prec q_{i,t}$ for $s < t$. Finally, for each edge $\{i,j\} \in V(G)$, we add the constraint $q_{j,r'-1} \prec p_{i,b-1}$ Thus, bar the technicality discussed below, there is a 1:1 correspondence between $DownSets(X, \prec)$ and $H(G)$. Figure 2.9 should clarify this.

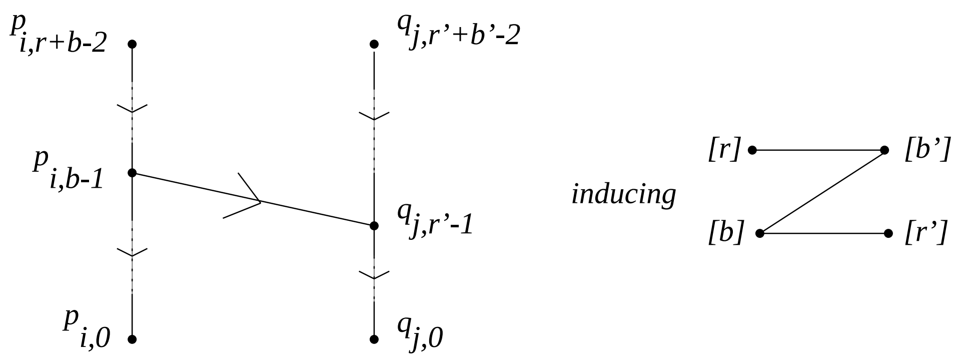


Figure 2.9: *Using $\#DownSets$ to prove weighted versions of P_4 are $\equiv_{\mathsf{AP}}\#BIS$-easy*

The technicality concerns the fact that every bipartite colouring problem has two distinct *orientations*: the above reduction technique actually counts only the *left orientation* of P_4. The following section explains. (We return to the proof of Lemma 2.14 after this section.)

Bipartite orientations

For bipartite $H = (V_L(H), V_R(H), E(H))$, H-colourings of a bipartite graph $G = (V_L(G), V_R(G), E(G))$ can be naturally classified into one of two categories. When colours from $V_L(H)$ are used to colour $V_L(G)$, this is a *left orientation* colouring, and when colours from $V_L(H)$ are used to colour $V_R(G)$, this is a *right orientation* colouring. If we represent the problem of counting left orientation colourings as $\overleftarrow{\#H}$, and right orientation colourings as $\overrightarrow{\#H}$, it is clear that:

$$\#H(G) = \overleftarrow{\#H}(G) + \overrightarrow{\#H}(G) \tag{2.13}$$

There are a number of simple results in this area:

$$\overleftarrow{\#H} \equiv_{\mathsf{AP}} \overrightarrow{\#H}$$

To see this, note that we can use (wlog) an $\overleftarrow{\#H}$ oracle to compute $\overrightarrow{\#H}(G)$ simply by swapping $V_L(G)$ and $V_R(G)$. One consequence of this is that $\overleftarrow{\#H}$ and $\overrightarrow{\#H}$ are truly interchangeable.

Lemma 2.15 *$\#H \leq_{AP} \overleftarrow{\#H}$*

Proof. $\#H(G)$ can be computed by adding $\overleftarrow{\#H}(G)$ to $\overleftarrow{\#H}(G')$, where G' is G with $V_L(G)$ and $V_R(G)$ swapped round. Summations such as these tend to preserve accuracy so it is fine to use ϵ, the input accuracy parameter, as the accuracy parameter for the two[22] oracle calls. □

Certain bipartite H have the property that they are *symmetric*. That is, there is an automorphism on H which maps every colour to the other side of the bipartition. As an example, the graph $bi(H)$ (discussed towards the end of Section 2.2) is by definition symmetric for all H. For symmetric H, $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#H$, and to see this observe that for symmetric H, $\overleftarrow{\#H}(G) = (1/2)\#H(G)$. However, it is not known, for general bipartite H, whether $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#H$. There are certain non-symmetric bipartite H that we do, however, know to have this property.

Example: Let $H = (V_L(H), V_R(H), E(H))$ be a bipartite graph such that there is a unique maximum degree vertex c such that $c \in V_L(H)$ and c is connected to every vertex in $V_R(H)$. (The maximum degree condition ensures that there is no distinct vertex d in either $V_L(H)$ or $V_R(H)$ with degree as high as c.) Then $\#H \equiv_{\mathsf{AP}} \overleftarrow{\#H}$.

Proof. $\#H \leq_{\mathsf{AP}} \overleftarrow{\#H}$ is immediate. Let G be an input to $\overleftarrow{\#H}$. To reduce $\overleftarrow{\#H}$ to $\#H$, build G' as follows. Take a copy of G, a large independent set I of size p ($p \gg n$) and a new vertex x. Connect x to every vertex in I, and connect x to every vertex in

[22] Note that this is one of the few reductions in this thesis that use more than one oracle call.

$V_R(G)$. If I is large enough then x is exponentially likely to be coloured c, thus forcing $V_R(G)$ to be coloured with $V_R(H)$ as required. $\square$

The reason the interreducibility result holds in this very specific example is because of two fundamental characteristics of H. Firstly, it has a feature (i.e. the maximum degree vertex, c) which allows us to "zone in" on one side of H's bipartition. Secondly, the fact that c is connected to *every* vertex in $V_R(H)$ (i.e. c is, in the bipartite sense, universal) means the orientation can be fixed without dropping any colours. That is, $\overleftarrow{H}$-colourings rather than colourings pertaining to some proper *subgraph* of $\overleftarrow{H}$ can be counted. For general H, this is the bigger problem of the two: there is plenty of gadgetry to enable us to zone in one side of a bipartition for a given graph H, but how to use that to fix the orientation without eliminating at least one colour is not generally known.

(We now complete the proof of Lemma 2.14.) The above results show that, though we have only counted left orientation P_4 colourings, $\#H \leq_{\mathsf{AP}} \overleftarrow{\#H} \leq_{\mathsf{AP}} \#BIS$ so by transitivity $\#H \leq_{\mathsf{AP}} \#BIS$. $\square$

Lemma 2.16 *Weighted versions of P_4 are $\equiv_{\mathsf{AP}}\#BIS$-hard*

Proof. In demonstrating that all weighted versions of P_4 are $\equiv_{\mathsf{AP}}\#BIS$-hard, we first demonstrate that (for all H in this category), $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#H$. We then show $\#BIS \leq_{\mathsf{AP}} \overleftarrow{\#H}$. The reason for proceeding in this manner is that it is in general much easier to develop a reduction (for bipartite H) when we can say *a priori* which side of H maps to which side of the input, G.

Proving $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#H$

Without loss of generality, suppose (in compact form) $V_L(H) = \{r, b\}$ and $V_R(H) = \{r', b'\}$. (We discuss the weights on these colours shortly.) By the definition of the $\overleftarrow{\#H}$ problem, we thus want to count only those colourings of G in which $V_L(G)$ takes colours from $V_L(H)$. Firstly, note that if b and b' are both of the same effective degree *and* both of the same weight then H is symmetric and the result follows immediately, because (as explained in Section 2.4.2) $\#H \equiv_{\mathsf{AP}} \overleftarrow{\#H}$ for symmetric H. Otherwise, b and b' must differ in effective degree and/or weight. In this case, the goal is to introduce a gadget which has one vertex x exponentially likely to be coloured either b or b', and furthermore in which we are able to say *a priori* which of the two it will be. Suppose x is exponentially likely to be coloured b; then connecting x to every vertex in $V_R(G)$ forces the orientation to be that which we desire. Conversely, if b' is more likely, we connect x to every vertex in $V_L(G)$. (Note that it is important that both b and b' are adjacent to every vertex on the other side of H's bipartition; otherwise colours would be lost and a proper *subgraph* of $\overleftarrow{\#H}$ would be pointed out, rather than $\overleftarrow{\#H}$ itself.) The question, then, is how to build a gadget with this property. It is tempting to argue that, in order to make x exponentially likely to be coloured with a unique colour from $\{b, b'\}$, all we have to do is attach a sufficiently large maxdeg gadget to x. However, this is insufficient because b, b' may be of the same effective degree but nevertheless point out different graphs. For example, where the weights of r, b', b, r' are 1,3,4,2 respectively, b and b' are both of effective degree 5 but would clearly force different colourings on G. To avoid this problem, we don't use a maxdeg gadget but use a maxdegweight gadget instead. (See Lemma 2.12.) That is, in addition to x we introduce an independent set I_1 of size p, and an independent set I_2 of size q. We connect x to every vertex in I_1, and every vertex in I_1 to every vertex in I_2. If we choose p and q to be, say, n^3 and n^2 respectively, this makes x exponentially likely to be coloured with the colour from $\{b, b'\}$ with higher effective degree, and if this is insufficient to separate them, the one with higher weight. A proof that the maxdegweight gadget behaves as described can be found in Lemma 2.12, where it was first introduced. Hence, we can pre-calculate which of $\{b, b'\}$ is picked out, which is what we require. This completes the proof that $\overleftarrow{\#H}$ $\leq_{\mathsf{AP}} \#H$.

Proving $\#BIS \leq_{\mathsf{AP}} \overleftarrow{\#H}$

This part of the proof is similar in nature to that shown in Lemma 2.8 which shows that all weighted versions of *2-WR* are $\equiv_{\mathsf{AP}}\#BIS$-hard. We assume that H is in compact form where $V_L(H) = \{r, b\}$ and $V_R(H) = \{r', b'\}$ and the colour labels double-up as the weights of those colours. The reduction proceeds by constructing G' as follows. For each $u_i \in V_L(G)$, an independent set U_i of size s is introduced. For each $v_i \in V_R(G)$, an independent set V_i of size t is introduced. For each edge $\{u_i, v_j\} \in E(G)$, every vertex in U_i is connected to every vertex in V_j. A vertex $u_i \in V_L(G)$ is considered IN the independent set if U_i contains r, and OUT otherwise. A vertex $v_i \in V_R(G)$ is considered IN the independent set if V_i contains r', and OUT otherwise. Thus, an independent set with k vertices in $V_L(G)$ and l in $V_R(G)$ comes up the following number of times:

$$\left((r+b)^s - b^s\right)^k b^{s(n_l-k)} \left((r'+b')^t - (b')^t\right)^l (b')^{t(n_r-l)}$$

As before, we aim to reduce from $\#MaxBIS$, so we again need to choose s and t such that the contribution of independent set in G is dependent only on its overall size and not the number of its vertices on one side of G or the other. This time, we need to choose s and t so they satisfy, as best as possible,

$$\left(\frac{r+b}{b}\right)^s = \left(\frac{r'+b'}{b'}\right)^t$$

If we let $A = (r+b)/b$, $B = (r'+b')/b'$, then $sz = t$ where $z = \log_B A$. Hence, to achieve the desired accuracy we have to show (as in the proof of Lemma 2.8) that

$$e^{-\epsilon/42} \leq \left(\frac{B^{zs}-1}{B^t-1}\right)^n \leq e^{\epsilon/42} \tag{2.14}$$

In fact, the process by which we choose s and t is virtually identical to that described in Lemma 2.8, so we omit most of the details.[23] To ensure that maximum-size independent sets exponentially dominate over non-maximal independent sets, ensuring (say) $t \geq n^2$ is again adequate. Let W be the smallest positive integer such that $\lfloor zW \rfloor \geq n^2$. To satisfy (2.14) a choice of $R = \lceil 16\ln(B)Wn/(7\delta)\rceil$ is adequate in the application of Corollary 2.9. □

Lemma 2.17 *Weighted versions of* P_4 *are* $\equiv_{\mathsf{AP}}\#BIS$

[23] To see why we have used n as the exponent in (2.14) rather than k, recall the justification we gave for the derivation of (2.11) on page 82.

Proof. Immediate by combining Lemmas 2.14 and 2.16. □

We now exhibit a proof that weighted paths longer than length 4 are also $\equiv_{\mathsf{AP}}\#BIS$. First, we prove the $\equiv_{\mathsf{AP}}\#BIS$-easiness of this category.

Lemma 2.18 *Weighted versions of P_k for $k > 4$ are $\equiv_{\mathsf{AP}}\#BIS$-easy*

Proof. This is a generalisation of the reduction used in the proof of Lemma 2.14. Rather than formally detail the construction, Figure 2.10 shows two illustrative examples of how longer paths are coded up. Note how one example is for odd-length H, and the other for even-length H. □

Just as we provided a $\equiv_{\mathsf{AP}}\#BIS$-hardness proof for weighted P_k^* where $k > 3$, we

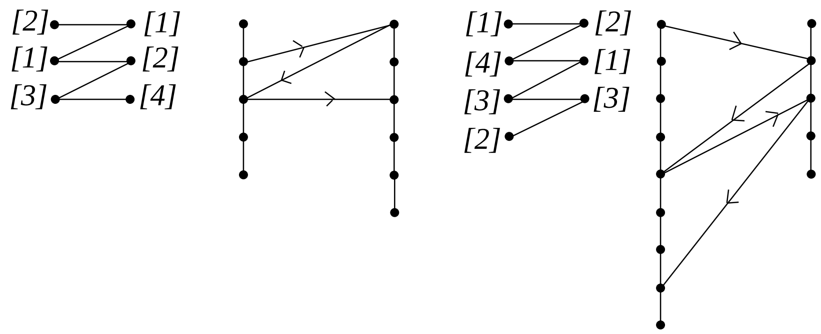


Figure 2.10: *Two examples of using $\#DownSets$ to show weighted paths of length greater than 4 are $\equiv_{\mathsf{AP}}\#BIS$-easy*

now detail a $\equiv_{\mathsf{AP}}\#BIS$-hardness proof for weighted P_k where $k > 4$. In actual fact, we provide the proof for a much stronger result. Firstly, we say that a graph H has a *non-repeating m-cycle* if it contains some cycle on m vertices such that the only vertex visited twice on the cycle is the start/finish vertex.

Lemma 2.19 *Let H be any non-trivial bipartite graph such that, when visualised in its* compact *form, contains no non-repeating 4-cycles.*[24] *Then $\#BIS \leq_{\mathsf{AP}} \#H$.*

[24] *For example, P_4 with weights 1,2,1,1 is one such graph, because even though its expanded form contains a non-repeating 4-cycle, its compact form does not.*

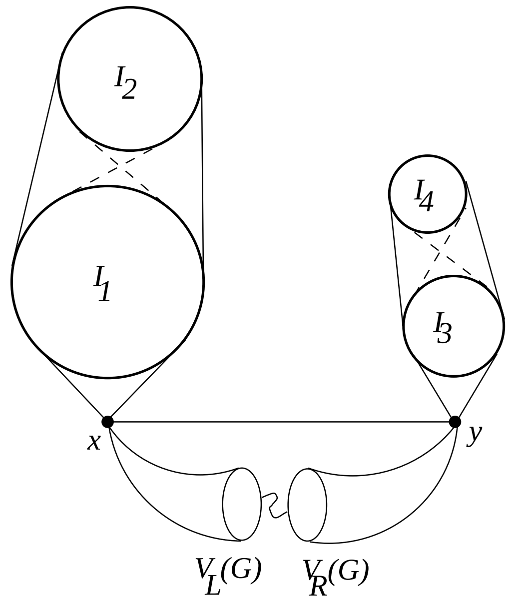


Figure 2.11: *Proving the $\equiv_{\mathsf{AP}}\#BIS$-hardness of bipartite graphs that have no non-repeating 4-cycles*

Proof. Note first that H also contains no non-repeating 3-cycles, because this would contradict the assertion that H is bipartite. We work with H in compact form, so let $V(H) = \{c_0, c_1, ..., c_{|V(H)|-1}\}$ and let w_i represent $w(c_i)$, the weight on colour c_i. (Whilst recognising that H is bipartite, in this instance it makes for a cleaner proof to treat $V_L(H)$ and $V_R(H)$ as one merged set, $V(H)$.) Let Δ_1 be the maximum effective degree of any colour in $V(H)$. We make the following sequential definitions.

$$
\begin{aligned}
L_0 &= \{c_i \in V(H) | deg'(c_i) = \Delta_1\} \\
W_1 &= \max\{w_i | c_i \in L_0\} \\
L_{max} &= \{c_i \in L_0 | w_i = W_1\} \\
R_0 &= \{c_i \in V(H) | (\exists c_j \in L_{max} | \{c_i, c_j\} \in E(H))\} \\
\Delta_2 &= \max\{deg'(c_i) | c_i \in R_0\} \\
R_1 &= \{c_i \in R_0 | deg'(c_i) = \Delta_2\} \\
W_2 &= \max\{w_i | c_i \in R_1\} \\
R_{max} &= \{c_i \in R_1 | w_i = W_2\} \\
Top &= \{(c_i, c_j) | c_i \in L_{max} \wedge c_j \in R_{max} \wedge \{c_i, c_j\} \in E(H)\}
\end{aligned}
$$

Note that Top comprises *ordered* pairs. Suppose we construct a graph G', based on the input graph $G = (V_L(G), V_R(G), E(G))$, as follows. Take a copy of G, add new vertices x and y, connect x to y, connect x to every vertex in $V_L(G)$ and y to every vertex in $V_R(G)$. Observe that, if $(c_i, c_j) \in Top$ and x is coloured c_i and y is coloured c_j, colourings of the form shown by Figure 2.12 are induced. In this figure, we have shown the induced graph in compact form, and introduced the new labels b, r, b', r' to represent the four different equivalence classes of colours in this graph. (So, b represents the W_2 colours in the equivalence class c_j, b' represents the W_1 colours in the equivalence class c_i, r the $\Delta_1 - W_2$ colours which are adjacent to c_i but not equal to c_j, and r' the $\Delta_2 - W_1$ colours which are adjacent to c_j but not equal to c_i.) The non-adjacency of the colours r, r' in this figure can be accounted for by the fact that H contains no non-repeating 4-cycles or 3-cycles. Furthermore, if we can prove that Top is non-empty, that $\Delta_1 > W_2$ and that $\Delta_2 > W_1$, the graph shown in Figure 2.12 is a fixed orientation, weighted version of P_4, which we know to be $\equiv_{\mathsf{AP}}\#BIS$-hard by combining Lemmas 2.16 and 2.15. So, if we can do this and in addition make x and y exponentially likely to be coloured with a pair from Top, counting colourings of the type shown in Figure 2.12 can be reduced to $\#H$. By transitivity this will give us $\#BIS \leq_{\mathsf{AP}} \#H$. Hence, our first task is to show Top is non-empty. This follows because Top is empty iff $E(H)$ is empty; clearly $E(H)$ cannot be empty, otherwise H would be trivial. Since Top is non-empty, it follows that $\Delta_1 \geq W_2$ and $\Delta_2 \geq W_1$. This is because there is at least one $(c_i, c_j) \in Top$, $deg'(c_i) = \Delta_1$, $deg'(c_i) \geq w_j$, $deg'(c_j) = \Delta_2$, $deg'(c_j) \geq w_i$. Thus, if we can show that c_i has at least one neighbour other than c_j, and vice-versa, it follows that $\Delta_1 > W_2$ and $\Delta_2 > W_1$, which is what we need.

Suppose c_i has no neighbour distinct from c_j. Since c_i is of effective degree Δ_1, and c_j is of weight W_2, it follows that $\Delta_1 = W_2$. Continuing, suppose c_j has some neighbour c_k distinct from c_i. Then c_k cannot have any neighbour other than c_j, because if it did $deg'(c_k) > W_2$, and since $W_2 = \Delta_1$ this would mean c_k had effective degree greater than Δ_1:- contradiction. So if c_j has a neighbour c_k other than c_i, c_k has no neighbours other than c_j. But in that case c_k is equivalent to c_i, and should not therefore exist because it would have been accounted for in the weight of c_i. So, it transpires that c_j has no neighbour other than c_i. However, if the only neighbour of c_i is c_j and vice-versa, H is a complete bipartite graph and thus trivial:- contradiction. So this proves that c_i must have at least one neighbour distinct from c_j.

Suppose c_j has no neighbour distinct from c_i. For starters, then, we know that $\Delta_2 = W_1$. We know from the above that c_i must have some neighbour distinct from c_j, call it c_m, so the effective degree of c_m is at least W_1. If c_m has a neighbour other than c_i then c_m must have effective degree greater than W_1, but that is not possible because that would mean there was a colour adjacent to c_i with effective degree greater than Δ_2:- contradiction. So c_m has no neighbours other than c_i. However, this means c_m is equivalent to c_j, which contradicts our precondition that c_m is distinct from c_j. So in conclusion c_j must have some neighbour other than c_i.

The above two results therefore show that $\Delta_1 > W_2$ and $\Delta_2 > W_1$. It remains to show how x and y can be made exponentially likely to be coloured with pairs from Top. To do this we complete the construction of G' (which we began earlier) by adding independent sets I_1, I_2, I_3, I_4 and connecting them as depicted in Figure 2.11. We set the size of these independent sets to be n^5, n^4, n^3, n^2 respectively. The backbone of our argument is similar to that given in the proof of Lemma 2.12; specifically, since I_1 is the polynomially biggest gadget a configuration cannot be maximal unless it colours I_1 with as many colours as possible, and then I_2 takes priority, and so on down to I_4. All other configurations are, in the language introduced in the proof of Lemma 2.12, *misguided*. It follows that, if (x, y) is coloured with some pair $(c_i, c_j) \in Top$, I_2 is exponentially likely to be restricted to c_i and I_4 is exponentially likely to be restricted to c_j. Why? Observe that, to avoid being misguided, I_1 must be coloured with $deg'(c_i)$ colours, and the only colour that can be adjacent to all neighbours of c_i is c_i itself. (A similar argument applies for c_j.) Now, if we let H' be the graph from Figure 2.12, and define full colourings to be those with Top colourings on x and y, the number of full colourings of G' can be expressed exactly as $\#H'(G)Z$ where $Z = W_1 W_2 |Top| Z'$ and

Z' is exactly the number of colourings possible in the gadgetry when a Top colouring appears on x and y. (Note that the leading $W_1W_2|Top|$ term comes from the fact that, when considering the expanded form of H, each colour from L_{max} represents W_1 different colours and each colour from R_{max} represents W_2 different colours.) As in the proof of Lemma 2.12 we need to compute Z exactly, and as in that proof we observe that this is achievable in polynomial time. Nonetheless, for the purpose of analysis we only need reason that

$$Z' \geq \Delta_1^{n^5} W_1^{n^4} \Delta_2^{n^3} W_2^{n^2}$$

Now, we want to show that dividing $\#\hat{H}(G')$ by Z and rounding gives an adequate approximation to $\#H'(G)$. Hence, in the usual fashion, we have to show that the ratio of non-full colourings to Z is less than 1/4. A crude upper bound on the total contribution of non-full colourings is

$$|V'(H)|^2 |V'(H)|^n (2^{|V(H)|})^4 \Delta_1^{n^5} W_1^{n^4} \Delta_2^{n^3} (W_2 - 1)^{n^2}$$

The $|V'(H)|^2$ term arises because we assume any combination of colours is possible on x and y, and the $|V'(H)|^n$ term arises because we assume any colourings are possible in G. The $(2^{|V(H)|})^4$ term represents an upper bound on the number of different configurations possible in I_1,I_2,I_3,I_4. It remains for us to show that the above quantity divided by Z is less than $1/4$, i.e.

$$\frac{|V'(H)|^2 |V'(H)|^n 16^{|V(H)|} \Delta_1^{n^5} W_1^{n^4} \Delta_2^{n^3} (W_2 - 1)^{n^2}}{W_1 W_2 |Top| \Delta_1^{n^5} W_1^{n^4} \Delta_2^{n^3} W_2^{n^2}} \leq 1/4$$

(We used our lower bound on Z in the above inequality.) Cancelling and writing $|V'(H)|^{2n}$ as an upper bound of $|V'(H)|^{n+2}$,

$$\frac{|V'(H)|^{2n} 16^{|V(H)|}}{W_1 W_2 |Top|} \left(\frac{W_2 - 1}{W_2} \right)^{n^2} \leq 1/4$$

An exponent of $2n$ cannot compete with an exponent of n^2 so this inequality clearly holds for n larger than some threshold constant. Hence, the rounding technique can be applied and this completes the proof, bar one minor technicality. In some circumstances $W_2 = 1$, causing $(W_2 - 1)^{n^2}$ to be equal to zero and therefore rendering the above inequalities meaningless. This can be fixed easily by rewriting the upper bound as:

$$|V'(H)|^2 |V'(H)|^n 16^{|V(H)|} \Delta_1^{n^5} W_1^{n^4} (\Delta_2 - 1)^{n^3}$$

We have already shown that $\Delta_2 > 1$. □

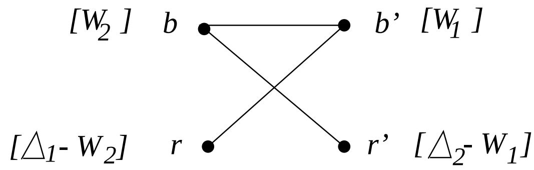


Figure 2.12: *The graph induced in G by the gadgetry from Figure* 2.11

Lemma 2.20 *Weighted versions of P_k (for $k \geq 4$) are $\equiv_{\mathsf{AP}}\#BIS$*

Proof. Immediate from combining Lemmas 2.18 and 2.19.

2.5 4-vertex H not yet classified

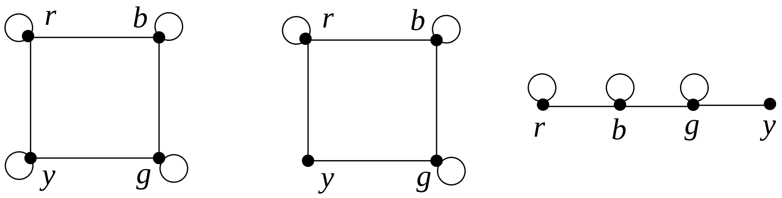


Figure 2.13: *H at least as hard as $\equiv_{\mathsf{AP}}\#BIS$, but for which we don't know whether $\#SAT \leq_{\mathsf{AP}} \#H$*

Figure 2.13 shows the only (connected) H on 4-vertices or less which have not been classified. (From left to right, graph indices are 51, 50, 30.) This is discussed at length in Chapter 7. What is easy to show, however, is that they are all $\equiv_{\mathsf{AP}}\#BIS$-hard i.e. $\#BIS$ is AP-reducible to each of them. To see this, note that in all three cases b (or another colour indistinguishable from it, as in the case of the graph on the left-hand side) can be made exponentially dominant using a K_2-cliqueset, thus pointing out the $\equiv_{\mathsf{AP}}\#BIS$ graph *2-WR*. □

Chapter 3

Approximately count - or approximately sample?

3.1 Introduction

The task of determining the relative complexity (in the approximate counting sense) of an arbitrary graph H can, broadly speaking, be accomplished in two ways. The first is to examine the structure of H and see whether this puts it in a family of graphs for which general results already exist. Failing that, there is always the recourse to hand-crafted, ad-hoc reductions (such as those used in Chapter 2.) Quite often the more general classifications only become apparent following the insights gained from repeatedly considering graphs on a case-by-case, ad-hoc basis; for example, many of the results in Chapter 5 were inspired by the work in Chapter 2. Similarly, the work in this chapter was undertaken following the discovery of what appear to be structural limitations of AP-reductions.

In the following section we demonstrate a graph H for which the "obvious" AP-reduction seems intuitively powerful enough, but is frustrated by apparent structural constraints of counting reductions. This toy example provides the motivation for diversifying into *sampling-preserving* reductions (SP-reductions) and the remaining sections further explore this sampling-counting relationship. (The sampling-preserving reduction and related definitions in Sections 3.4 and 3.5 were originally formulated jointly with Goldberg and Paterson in [14].)

3.2 A motivating example

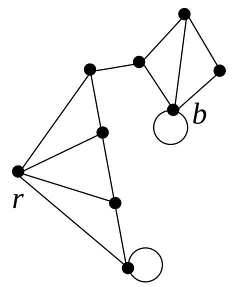


Figure 3.1: *A graph with two distinct maximum degree colours*

If confronted with a graph H and asked to determine its complexity in the same ad-hoc manner as much of Chapter 2, a valuable heuristic is to first examine the subgraphs of H pointed out by its maximum degree colours. If there is only one maximum degree colour, and it points out a subgraph H', or if there are multiple maximum degree colours and they all point out the same isomorphic subgraph H', then we can show $\#H' \leq_{\mathsf{AP}} \#H$ simply by using a maxdeg gadget (as defined on page 60.) This tactic is particularly effective when $\#H' \equiv_{\mathsf{AP}} \#SAT$, because it immediately follows that $\#H \equiv_{\mathsf{AP}} \#SAT$. Now, consider the graph H in Figure 3.1 above. Suppose for one moment we don't know any other reduction techniques and look, as our first port of call, at the maximum degree vertices of H, r and b. Significantly, both $\#H[r]$ and $\#H[b]$ are $\equiv_{\mathsf{AP}} \#SAT$, as proven on pages 59 and 69 respectively. However, even though we know we can pick out r and b using a maxdeg gadget, and that both $\#H[r]$ and $\#H[b]$ are $\equiv_{\mathsf{AP}} \#SAT$, it is not apparent how this can be capitalised on to prove $\#H \equiv_{\mathsf{AP}} \#SAT$. In practice we could (in this instance) avoid this issue by using a K_2-cliqueset to point out just $H[b]$. However, the option of finding a "better" gadget is neither always possible nor always desirable; in many ways it would be preferable to be able to make arguments of the kind, "*both the pointed-out subgraphs are hard so surely it is hard itself*". This is the issue we now explore.

If we let $H_r = H[r]$ and $H_b = H[b]$, observe that the quantity approximated if the maxdeg gadget (and rounding) is used is $\#H_b(G) + \#H_r(G)$. Before proceeding any further, it is worth pointing out at this juncture the combinatorial significance of the "+" operator in this context. The value $\#H_1(G) + \#H_2(G)$ (for arbitrary graphs H_1, H_2 and *connected* G) is *exactly* the same value as $\#H'(G)$ if H' consists of two disjoint components, H_1 and H_2. So the issue of querying the complexity of a graph H in terms of its pointed-out subgraphs is extremely closely related to the question of determining the complexity of disconnected H. (Clearly this generalises to more than just two graphs H_1 and H_2.)

In a small number of instances the problem of approximating $\#H_1(G) + \#H_2(G)$ can be reduced to the problem of approximating just $\#H_1(G)$ or $\#H_2(G)$. One instance when this happens, as explained in Chapter 6, is when (wlog) H_1 has a subgraph K_m^* such that $m > |V(H_2)|$. The idea here is that the presence of the large K_m^* component allows us to argue, without inspecting the input G, that $\#H_1(G)$ will always exponentially dominate over $\#H_2(G)$. However, such cases - where we can decide *a priori* which, if any, out of H_1 and H_2 exponentially dominate - are fairly rare. In general, therefore, a more flexible approach is desired.

Consider, then, our example. We know that both $\#IS \leq_{\mathsf{AP}} \#H_b$ and $\#IS \leq_{\mathsf{AP}} \#H_r$, via separate reductions. In each case, the reduction involves constructing a new graph G' (based on the input G, to $\#IS$), calling the appropriate oracle, dividing by a particular value and then rounding. For a graph G, let G_b refer to the graph G' constructed for our $\#IS \leq_{\mathsf{AP}} \#H_b$ reduction, and let G_r refer to the graph constructed for the $\#IS \leq_{\mathsf{AP}} \#H_r$ reduction. Now, if we wish to exploit these reductions to show $\#IS \leq_{\mathsf{AP}} \#H$, we have to decide whether to connect our maxdeg gadget to a copy of G_b or alternatively to a copy of G_r. This is the crux of the matter:- how can we tell, in general, which one to use? If we use (say) G_b, the value approximated will be $\#H_b(G_b) + \#H_r(G_b)$, rather than the $\#H_b(G_b)$ which is required. This scuppers the conventional "rounding" technique, because the extraneous term $\#H_r(G_b)$ may well be significant enough to distort the final approximation. More fundamentally, in most cases it is not practical to reason about the relative contribution of additive terms like $\#H_r(G_b)$, because this in itself presupposes some ability to approximate $\#H_r(G_b)$. To compound matters, the following example shows how in an expression of the form $\#H_1(G) + \#H_2(G)$ there is no guarantee that either the $\#H_1(G)$ or the $\#H_2(G)$ term is consistently dominant over the other[1] for all G. Consider the two H graphs in Figure 3.2. If $G = K_n$, then a lower

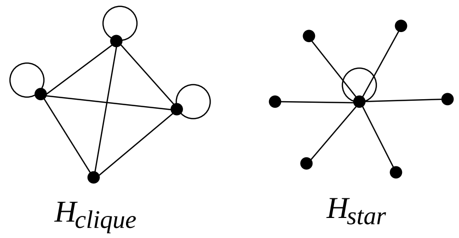


Figure 3.2: *Two graphs that, depending on G, alternate their dominance*

bound on $\#H_{clique}(G)$ is 3^n whilst $\#H_{star}(G) = 1 + 6n$. If G is the $(n-1)$-star (i.e. the star with one centre vertex and $n-1$ prongs) then an upper bound on $\#H_{clique}(G)$ is 4^n, whilst a lower bound on $\#H_{star}(G)$ is 7^{n-1}. Clearly in the first case H_{clique} is exponentially dominant, whilst in the second H_{star} is exponentially dominant. Though this example is somewhat contrived, it serves to highlight how exponential dominance is not just a function of H but also of G, the input graph.

All this points to some severe limitations for AP-reductions, or more accurately the particular flavour of AP-reduction that we have used predominantly throughout this thesis. The existence of spurious terms (such as $\#H_r(G_b)$ in our example), our inability to reason in advance about their relative contribution and their variability with changing G renders the divide-then-round technique impotent when confronted with more than one (distinct) maximal configuration. In particular, what value should we divide by? Will this dwarf the contribution of the extraneous terms, as is hoped, or instead be dwarfed by them? Even in reductions that do not involve a division after the oracle call,

[1]In addition, there may be G for which neither term is exponentially dominant over the other.

there is still the danger of exponential dominance alternating depending on G, as is the case with H_{clique} and H_{star}. Collectively this explains why the proof for Lemma 2.12 is so long, despite the fact that a simple maxdeg gadget would guarantee *some* weighted version of *2-WR* pointed out. Crucially, this would not tell us *which* weighted version would be pointed out, and without this knowledge we would have insufficient parameters to follow through the construction described in the proof of Lemma 2.8. We cannot simply assume arbitrary parameters, because colouring a $\equiv_{\mathsf{AP}}\#BIS$-hardness construction with a weighted version of *2-WR* other than that which was intended might lead to dangerously large inaccuracies, as discussed above.

In a nutshell, then, AP-reductions seem to break down completely when confronted with “choice”.

3.3 Enter sampling:- an informal introduction

Suppose, however, that we switch to the world of *sampling*, or to be more precise, *approximate sampling*. Consider, first, the reductions for $\#IS\leq_{\mathsf{AP}}\#H_r$ and $\#IS\leq_{\mathsf{AP}}\#H_b$. These reductions are typical of most AP-reductions in the sense that the structures being counted, independent sets, are actually constructed; as we have seen from Chapter 2, a standard strategy is to build G' so that each item being counted appears an exponentially large number of times as a colouring from $H(G')$. (For example, in the reduction for H_b, each independent set comes up approximately 18^p times.) So, if we take a sample from $H(G')$, and it is a full colouring - which it is exponentially likely to be - we can produce an approximately uniform independent set sample simply by “reading off” the independent set sample from the encoding of G within G'.

Now suppose, for a given graph G, we construct the graph G' as in Figure 3.3, as input to an almost uniform sampler for H. (We will formalise terms like this in due course but for now it is adequate to understand it as a sampler that produces colourings with a distribution “not too far” from uniform.)

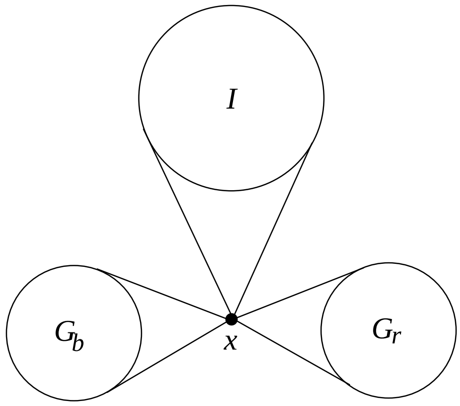


Figure 3.3: *An example of "glueing" sampling problems together*

Now, if the maxdeg gadget is big enough (in terms of $|V(G_r)|$ and $|V(G_b)|$) then, as we know, the vertex x is exponentially likely to be coloured r or b, and thus a sample of $H(G')$ is exponentially likely to have x coloured r or b. Observe that, if we take a sample of $H(G')$, we can almost always (i.e. with exponentially small failure probability) return an independent set sample. This is because, having obtained our sample, we can *inspect* the colour of x. If it's r, we know that H_r colourings are pointed out in G_r and G_b, so we can read off an independent set by "zooming in" on G_r. If x is coloured b, we can read off an independent set by zooming in on G_b. (If x is coloured neither r or b, we return the "error" symbol $\perp$, but this will not happen very often.) The point is, the ability to *inspect* and then *decide* from which subreduction to pull a sample gives sampling a significant edge over counting, at least in the way we have built counting reductions so far. In this example, it doesn't matter that we don't know whether x is more or less likely to be coloured r or b: we get a sample either way. Counting reductions do not *seem* to allow us this flexibility. The only apparent way this reduction could work as a "rounding"-based AP-reduction would be under the extremely unrealistic conditions that (having studied G_r and G_b) we know that one of r and b is exponentially more likely, we know which one, and (say it is b) we are able to exactly compute $\#H_b(G_r)$. (The final condition is there because each $H_b(G_b)$ colouring comes up $4^{|I|}\#H_b(G_r)$ times as a colouring of G', and therefore we would want to compute and round the quantity $\#\hat{H}(G')/4^{|I|}\#H_b(G_r)$ to obtain our approximation.)

So this is an informal illustration of where the motivation for what we call *sampling preserving reductions* came from. We now look to formalise this concept. We define a framework for reductions between approximate sampling problems, and then explore the relationship that such reduction have with AP-reductions, taking into account known results about the relative complexity of approximately counting H-colourings as compared to approximately sampling H-colourings.

3.4 Definitions

The total variation distance between two distributions π and π' on a countable set Ω is given by

$$\mathrm{d}_{\mathrm{TV}}(\pi, \pi') = \frac{1}{2} \sum_{\omega \in \Omega} |\pi(\omega) - \pi'(\omega)| = \max_{A \subseteq \Omega} |\pi(A) - \pi'(A)|.$$

A sampling problem X maps each instance σ to a set of structures $X(\sigma)$. The goal is to produce a member of $X(\sigma)$ uniformly at random. The size of each structure in $X(\sigma)$ is at most a polynomial in $|\sigma|$. For a given graph H, the sampling problem *Sample-H* will be defined as follows.

Name: Sample-H

Instance: A graph G

Output: An H-colouring of G chosen uniformly at random.

An *almost uniform sampler* [7, 20, 16] for X is a randomised algorithm that takes input σ and accuracy parameter $\epsilon \in (0, 1]$ and gives an output such that the variation distance between the output distribution of the algorithm and the uniform distribution on $X(\sigma)$ is at most ϵ. We will say that such an algorithm is a *polynomial almost uniform sampler (PAUS)* if its running time is bounded from above by a polynomial in the size of the instance $|\sigma|$ and $1/\epsilon$. (Note that, throughout this thesis, a $PAUS$ on a distribution $X(\sigma)$ is not restricted to definitely returning an element from $X(\sigma)$; it is permitted to return a “failure” symbol $\perp$ if necessary. We always take this into account when handling $PAUS$es.)

A *sampling-preserving reduction* (SP-reduction) from a sampling problem X to a sampling problem Y (which is denoted $X \leq_{\mathsf{SP}} Y$) consists of

1. A function f which takes an input (σ, ϵ), in which σ is an instance of X and $\epsilon \in (0,1]$ is an accuracy parameter, and produces an instance $f(\sigma, \epsilon)$ of Y. If $X(\sigma)$ is non-empty then $Y(f(\sigma, \epsilon))$ must be non-empty.

2. A function g which maps each tuple (σ, ϵ, y) with $y \in Y(f(\sigma, \epsilon))$ to a member of $X(\sigma) \cup \{\perp\}$ where "$\perp$" is an error symbol and for every (σ, ϵ) and every $x \in X(\sigma)$,

$$\begin{aligned} e^{-\epsilon} \frac{|Y(f(\sigma,\epsilon))|}{|X(\sigma)|} &\leq |\{y \in Y(f(\sigma,\epsilon)) \mid g(\sigma,\epsilon,y) = x\}| \\ &\leq e^{\epsilon} \frac{|Y(f(\sigma,\epsilon))|}{|X(\sigma)|}. \end{aligned} \tag{3.1}$$

Equation (3.1) says that for every $x \in X(\sigma)$, the number of $y \in Y(f(\sigma, \epsilon))$ which are mapped to x by g is roughly $\frac{|Y(f(\sigma,\epsilon))|}{|X(\sigma)|}$. Thus, each $x \in X(\sigma)$ is roughly equally represented and the error symbol $\perp$ is represented by only about an ϵ-fraction of $Y(f(\sigma, \epsilon))$.

The functions f and g must be computable in time which is bounded by a polynomial in $|\sigma|$ and $1/\epsilon$.

A comment on notation

In earlier chapters, we use the notation $\#H(G)$ to refer to the problem of counting H-colourings of a graph G. With respect to sampling, we tend to drop the fairly cumbersome *Sample-H* notation and simply refer to the sampling problem as H itself, where the meaning of this is unambiguous from the context. For example, whereas in the AP world we write $\#H_1 \leq_{\mathsf{AP}} \#H_2$, in the sampling world we write $H_1 \leq_{\mathsf{SP}} H_2$ because it is clear that the placeholders H_1 and H_2 refer to the sampling problems associated with H_1 and H_2 respectively.

Also, as would be expected, the expression $X \equiv_{\mathsf{SP}} Y$ can be used iff $X \leq_{\mathsf{SP}} Y$ and $Y \leq_{\mathsf{SP}} X$.

Lemma 3.1 *If $X \leq_{SP} Y$ and Y has a $PAUS$, then X has a $PAUS$.*

Proof. Let (f, g) be the reduction from X to Y and let $\mathcal{A}$ be a $PAUS$ for Y. Here is a $PAUS$ for X: On input (σ, ϵ), let y be the output of $\mathcal{A}$ when it is run with inputs $f(\sigma, \epsilon/4)$ and $\epsilon/2$; return $g(\sigma, \epsilon/4, y)$.

We must show that the variation distance between the output distribution of this algorithm and the uniform distribution on $X(\sigma)$ is at most ϵ. Let σ be an input with $|X(\sigma)| \geq 1$. Consider any subset A_x of $X(\sigma)$. Let

$$A_y = \{y \in Y(f(\sigma, \epsilon/4)) \mid g(\sigma, \epsilon/4, y) \in A_x\}.$$

Then the probability that $\mathcal{A}$ gives an output in A_y is at most

$$\begin{aligned}
\frac{|A_y|}{|Y(f(\sigma, \epsilon/4))|} + \frac{\epsilon}{2} &\leq \frac{e^{\epsilon/4}|A_x|}{|X(\sigma)|} + \frac{\epsilon}{2} \\
&\leq \frac{(1 + \epsilon/2)|A_x|}{|X(\sigma)|} + \frac{\epsilon}{2} \\
&\leq \frac{|A_x|}{|X(\sigma)|} + \frac{(\epsilon/2)|A_x|}{|X(\sigma)|} + \frac{\epsilon}{2} \\
&\leq \frac{|A_x|}{|X(\sigma)|} + \epsilon.
\end{aligned}$$

Also, the probability that $\mathcal{A}$ gives an output in A_y is at least

$$\begin{aligned}
\frac{|A_y|}{|Y(f(\sigma, \epsilon/4))|} - \frac{\epsilon}{2} &\geq \frac{e^{-\epsilon/4}|A_x|}{|X(\sigma)|} - \frac{\epsilon}{2} \\
&\geq \frac{(1 - \epsilon/2)|A_x|}{|X(\sigma)|} - \frac{\epsilon}{2} \\
&\geq \frac{|A_x|}{|X(\sigma)|} - \frac{|A_x|(\epsilon/2)}{|X(\sigma)|} - \frac{\epsilon}{2} \\
&\geq \frac{|A_x|}{|X(\sigma)|} - \epsilon.
\end{aligned}$$

3.5 A proof technique

Figure 3.3 demonstrated pictorially the principle of simultaneously combining several reductions into one overall reduction. This can be formalised as follows. When we introduce an SP-reduction from a sampling problem X to a sampling problem Y, we will need to show that Equation (3.1) is satisfied. We will typically do this by partitioning $Y(f(\sigma, \epsilon))$ into disjoint sets $Y_0, \ldots, Y_k$. We will show that each of $Y_1, \ldots, Y_k$ is fairly representative of $X(\sigma)$. In particular, for every $x \in X(\sigma)$ and every $i \in [1, k]$,

$$e^{-\epsilon/2} \frac{|Y_i|}{|X(\sigma)|} \leq |\{y \in Y_i \mid g(\sigma, \epsilon, y) = x\}| \leq e^{\epsilon/2} \frac{|Y_i|}{|X(\sigma)|}. \tag{3.2}$$

For every $y \in Y_0$, we will have $g(\sigma, \epsilon, y) = \perp$ but we will show that Y_0 is a small part of $Y(f(\sigma, \epsilon))$. In particular,

$$\sum_{i=1}^{k} |Y_i| \geq e^{-\epsilon/2} |Y(f(\sigma, \epsilon))|. \tag{3.3}$$

Together, (3.2) and (3.3) imply (3.1). Note that (3.3) follows from

$$|Y_0| \leq (\epsilon/4)|Y(f(\sigma, \epsilon))|, \tag{3.4}$$

since (3.4) implies $|Y| - |Y_0| \geq (1 - \epsilon/4)|Y(f(\sigma, \epsilon))|$ and in turn $(1 - \epsilon/4)|Y(f(\sigma, \epsilon))| \geq e^{-\epsilon/2}|Y(f(\sigma, \epsilon))|$.

Quite often the reduction $X \leq_{\mathsf{SP}} Y$ will involve several subproblems $Z_1, Z_2, \ldots$ such that, for each of these, an SP-reduction (f_i, g_i) from X to Z_i is already known. The instance $f(\sigma, \epsilon)$ of Y is then formed by "glueing" together instances $f_1(\sigma, \epsilon/2)$ of Z_1, $f_2(\sigma, \epsilon/2)$ of Z_2, and so on. Y_i is (roughly) the portion of $Y(f(\sigma, \epsilon))$ for which, within each $y \in Y_i$, we can "zoom in" on a structure $z \in Z_i(f_i(\sigma, \epsilon/2))$. Each structure in $Z_i(f_i(\sigma, \epsilon/2))$ is represented by an equal number of $y \in Y_i$ so we can get (3.2) by referring to the SP-reduction from X to Z_i. Establishing (3.4) is essentially showing that, although $Y(f(\sigma, \epsilon))$ has some structures which don't allow us to "zoom in" on an appropriate sub-problem to find our sample, these aren't so numerous.

Finally, let $Y_i(x) = \{y \in Y_i \mid g(\sigma, \epsilon, y) = x\}$. Suppose that no $y \in Y_i$ has $g(\sigma, \epsilon, y) = \perp$. In this case we can show (3.2) by showing that for all $x, x' \in X(\sigma)$,

$$|Y_i(x)| \leq e^{\epsilon/2} |Y_i(x')|. \tag{3.5}$$

To see this, note that

$$\begin{aligned} \frac{|Y_i|}{|X(\sigma)|} = \frac{\sum_{x' \in X(\sigma)} |Y_i(x')|}{|X(\sigma)|} &\geq e^{-\epsilon/2} \frac{\sum_{x' \in X(\sigma)} |Y_i(x)|}{|X(\sigma)|} \\ &= e^{-\epsilon/2} |Y_i(x)|. \end{aligned}$$

It should be stressed that demonstrating an SP-reduction $X \leq_{\mathsf{SP}} Y$ for two sampling problems X and Y is just one possible way of showing that a $PAUS$ for Y yields a $PAUS$ for X. In other words, we are not claiming that the SP-reduction is a "definitive" sampling-preserving reduction technique. Indeed, the current SP-reduction definition is a good balance between generality and clarity; the SP-reduction is sufficiently general to be of use throughout this thesis, but as we discuss later a trade-off for keeping the definition relatively simple is that there are some situations where a $PAUS$ for Y can clearly be used to build a $PAUS$ for X, but the natural reduction is not an SP-reduction. In such instances we demonstrate alternative $PAUS$-to-$PAUS$ reductions, which is adequate as long as we bear in mind that the SP-reduction is simply a means to an end and not the "last word" on the relative complexity of sampling problems.

3.6 The proof technique in action

We return once again to our running example. First, we demonstrate (separately) that $IS \leq_{\mathsf{SP}} H_r$ and $IS \leq_{\mathsf{SP}} H_b$, and then show formally how the argument put forward visually in Figure 3.3 yields $IS \leq_{\mathsf{SP}} H$.

Proving $IS \leq_{\mathsf{SP}} H_r$

The reduction $\#IS \leq_{\mathsf{AP}} \#H_r$ is shown on page 59. We can convert this very easily to an SP-reduction:- Section 3.8.1 shows how many AP-reductions map easily to an SP counterpart, but here we show the SP-reduction explicitly. The type of gadgetry used is identical (i.e. the K_2-cliqueset, see Figure 2.1), although the size of the K_2-cliqueset used is derived in a different way. Let (G, ϵ) be the input to IS. Following a similar line to the AP-reduction, let $G' = f(G, \epsilon)$ be the graph built by introducing a new vertex x and a K_2-cliqueset K of size k, and connecting x to every vertex in a copy of G and every vertex in K. The idea is simple: we take an H_r sample of G', and inspect the colour of x. If x is coloured r we know independent sets are pointed out in G so we can sample $IS(G)$ just by reading off the colouring in G.[2] If x is not coloured r then we throw the sample away and return $\perp$. Casting this in the framework we have just described, Y_0 is the subset of $H_r(G')$ where x is not coloured r and Y_1 is the subset

[2]Note that in this instance r refers to the "red" from Graph 27 on page 59, rather than the "red" of our running example.

where x is coloured r. (So Y_1 essentially represents the set of full colourings.) Clearly (3.2) is immediately satisfied because every independent set comes up exactly the same number of times as a colouring of Y_1. It remains, then, to satisfy (3.4), which in this context can be expressed as $\#H(G'|x \twoheadrightarrow r) \leq (\epsilon/4)\#H(G')$. We know that a loose upper bound on $\#H(G'|x \twoheadrightarrow r)$ is $1^k 4^n$, and a lower bound on $\#H(G')$ is 3^k, so we need to choose k such that $4^n \leq (\epsilon/4)3^k$. This is satisfied by setting:

$$k = \left\lceil \frac{(n+1)\ln(4) + \ln(1/\epsilon)}{\ln(3)} \right\rceil$$

The ceiling function is just to ensure that k is discrete. Note that here our derivation of k is a function of n *and* ϵ, whereas in the AP reduction we were somewhat more crude and didn't bother incorporating ϵ, preferring just to use a sufficiently big polynomial in n. (Section 3.8.1 shows how we can still make use of the derivation solely in n without having to explicitly re-derive in terms of n and ϵ.)

Proving $IS \leq_{\mathsf{SP}} H_b$

Again exploiting the convertability of many AP-reductions, we use the same gadgetry and argument as for the AP reduction on page 69. If (I_1, I_2) is coloured (br, bgy) - again noting that colour labels used in this proof are not from our running example, but from Graph 53 on page 69 - then we can sample independent sets by reading off the independent set colouring pointed out in G. If (I_1, I_2) is not coloured this way then we return $\perp$. So, let Y_1 be those colourings where (I_1, I_2) are coloured (br, bgy) and Y_0 be all the rest. Again, it is easy to satisfy (3.2) because every independent set comes up the same number of times as a colouring of Y_1. Therefore we need to make p big enough so that (3.4) is satisfied. Recall that (br, bgy) comes up $\nu(p, 2)\nu(2p, 3) \approx 18^p$ times, and that the closest to this is $(b, rbgy)$ which comes up an inferior 16^p times; all other configurations are smaller again. A pessimistic upper bound on the number of non-maximal configurations on (I_1, I_2) is $2^4 2^4$ - derived by assuming that both I_1 and I_2 can be coloured with any subset of colours - so coupling this with the assumption that all non-maximal configurations on (I_1, I_2) permit unrestrained colourings in G gives

$16^p 4^n 256$ as an upper bound on $|Y_0|$. A lower bound on (br, bgy) is $(1/4)18^p$ so[3] we need to choose p such that $16^p 4^n 256 \le (\epsilon/4)(1/4)18^p$. Hence, a choice of

$$\left\lceil \frac{(n+6)\ln(4) + \ln(1/\epsilon)}{\ln(18/16)} \right\rceil$$

is adequate.

Putting $IS \le_{\mathsf{SP}} H_r$ and $IS \le_{\mathsf{SP}} H_b$ together: proving $IS \le_{\mathsf{SP}} H$

Having established that $IS \le_{\mathsf{SP}} H_r$ and $IS \le_{\mathsf{SP}} H_b$, we now show how these results can be combined to prove $IS \le_{\mathsf{SP}} H$. That is, we now define (f, g) as the SP-reduction between IS and H. Given an input (G, ϵ), $f(G, \epsilon)$ is built as follows; for convenience we refer to $f(G, \epsilon)$ as G'. Let (f_r, g_r) be the SP-reduction from IS to H_r and (f_b, g_b) be the SP-reduction from IS to H_b. Let $G_r = f_r(G, \epsilon/2)$ and let $G_b = f_b(G, \epsilon/2)$. We introduce a new vertex x and connect x to every vertex in G_r, and to every vertex in G_b. Finally we introduce a maxdeg gadget I of size k (to be determined) and connect x to every vertex in I. (This is as Figure 3.3 depicted.) Now, we let Y_1 be those colourings of G' where x is coloured b, Y_2 be those where x is coloured r, and Y_0 be all other colourings. We consider a sample $y \in H(G')$. If $y \in Y_0$ we let $g(G, \epsilon, y) = \perp$. If $y \in Y_1$ we let $g(G, \epsilon, y) = g_b(G, \epsilon/2, y')$ where y' is simply y restricted to G_b. Finally, if $y \in Y_2$ we let $g(G, \epsilon, y) = g_r(G, \epsilon/2, y')$ where y' is y restricted to G_r. In other words, if x is coloured b we "zoom in" on G_b and use the reduction $IS \le_{\mathsf{SP}} H_b$ to gain an independent set sample, and if x is coloured r we zoom in on G_r using $IS \le_{\mathsf{SP}} H_r$ to obtain a sample. To complete the reduction we must prove (3.2) and (3.4). The former is not difficult to satisfy; observe that (for Y_1) we need to prove that for every independent set $\gamma \in IS(G)$,

$$e^{-\epsilon/2} \frac{\#H(G'|x \to b)}{\#IS(G)} \le |\{y \in Y_1 | g(G, \epsilon, y) = \gamma\}| \le e^{\epsilon/2} \frac{\#H(G'|x \to b)}{\#IS(G)} \tag{3.6}$$

Now, we know that $\#H(G'|x \to b) = \#H_b(G_b)\psi$ where $\psi = 4^k \#H_b(G_r)$, although the important fact about ψ is that its value remains the same irrespective of γ. Since

[3] The 1/4 factor emerges because we apply Corollary 2.5 to both $\nu(p, 2)$ and $\nu(2p, 3)$.

$G_b = f_b(G, \epsilon/2)$ we know from (3.1) that for all $\gamma \in IS(G)$

$$e^{-\epsilon/2}\frac{\#H_b(G_b)}{\#IS(G)} \leq |\{y \in H_b(G_b)|g_b(G, \epsilon/2, y) = \gamma\}| \leq e^{\epsilon/2}\frac{\#H_b(G_b)}{\#IS(G)} \tag{3.7}$$

Given that, for all $\gamma \in IS(G)$,

$$|\{y \in Y_1|g(G, \epsilon, y) = \gamma\}| = \psi|\{y \in H_b(G_b)|g_b(G, \epsilon/2, y) = \gamma\}|$$

we see that multiplying (3.7) through by ψ yields (3.6), and hence (3.2) clearly holds. The case is analogous for Y_2. It remains to show that $|Y_0|/\#H(G')$ is less than $(\epsilon/4)$. Aside from b and r all $c \in V(H)$ are at most of degree 3 so, if $q = |V(G_b)| + |V(G_r)|$ then we know that

$$|Y_0| \leq |V(H)|3^k|V(H)|^q$$

A lower bound on $\#H(G')$ is 4^k so we can satisfy (3.4) by setting:

$$k = \left\lceil \frac{\ln(4) + (q+1)\ln(|V(H)|) + \ln(1/\epsilon)}{\ln(4/3)} \right\rceil$$

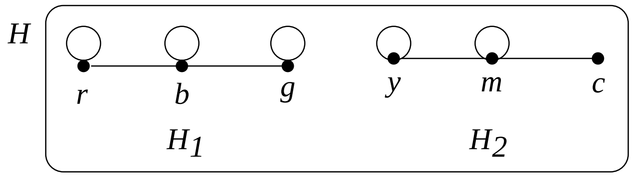
Note that this value is not too large because $q = |V(G_r)| + |V(G_b)|$ and both $|V(G_r)|$ and $|V(G_b)|$ are bounded above by polynomials in $|V(G)|$ and $1/\epsilon$. This completes the SP-reduction for $IS \leq_{\mathsf{SP}} H$. □

3.7 An important observation about sampling

The proof technique described in Section 3.5 was developed to establish a framework within which Theorem 4.1 of Chapter 4 could be proven. The challenge was to remedy the now-familiar weakness of counting reductions i.e. where we can engineer G' such that the "fail space" of $H(G')$ is exponentially small, but we are unable to disaggregate the two or more subproblems encoded in the useful remainder of the solution space. As a result, it is sometimes tempting to view the proof technique simply as some kind of enhanced AP-reduction[4] with "plural" capability. Whilst this is indeed a significant characteristic of the technique, it is not the only area in which sampling seems to offer

[4]Acknowledging, as we discuss shortly, that though the existence of an AP-reduction in many cases implies the existence of an analogous SP-reduction, this is not always known to be the case.

significantly greater flexibility than counting. In particular, the technique allows sampling reductions to succeed in situations where the colourings to be sampled constitute perhaps as little as a polynomially-small fraction of the total colouring space. Consider the following disconnected graph, H



We could show $\#H_1 \leq_{\mathsf{AP}} \#H$ by (for example) using a K_2-cliqueset to make b exponentially likely, thus enabling us to point out H_1 colourings. However, we can demonstrate $H_1 \leq_{\mathsf{SP}} H$ with a simple observation and no "custom" gadgetry such as a cliqueset. Here is the SP-reduction. We define the function f first. Given a problem (G, ϵ), we let $f(G, \epsilon) = G'$ be the graph consisting of k disjoint copies of G, which we label $G_1, ..., G_k$. (We will determine k shortly.) We let Y_1 be the set of colourings in $H(G')$ in which at least one of the G_i takes a colouring from H_1. We let Y_0 be all remaining colourings, i.e. where all the G_i are coloured with H_2 colourings. Now, for a colouring y from $H(G')$, we define $g(G, \epsilon, y)$ as follows. If $y \in Y_1$ we return the H_1 sample in G_j, where j is the index of the first graph to be coloured with H_1. Otherwise we return $\perp$. Now, observe that (3.2) is immediately satisfied because no H_1 colouring is any more or less likely to come up than any other. The crux of the matter becomes apparent when we establish (3.4). Observe that for any graph G, $\#H_2(G) \leq \#H_1(G)$ because H_2 is a subgraph of H_1. As a result, $\#H_1(G)/\#H(G) \geq 1/2$ for all connected G. It follows that $|Y_0|/\#H(G') \leq 1/2^k$, and as a consequence all we have to do to establish (3.4) is set k as

$$\left\lceil \frac{\ln(4) + \ln(1/\epsilon)}{\ln(2)} \right\rceil$$

This generalises easily to the situation where the fraction of $H(G)$ occupied by desirable (i.e. non-Y_0) colourings is as small as n^{-c}. In such an instance we have to satisfy

$$\left(\frac{n^c - 1}{n^c}\right)^k \leq \epsilon/4$$

and we can do this by setting

$$k = \left\lceil \frac{\ln(4) + \ln(1/\epsilon)}{\ln(n^c/(n^c-1))} \right\rceil$$

(It is not difficult to show that the multiplicative factor introduced by dividing by $\ln(n^c/(n^c-1))$ is polynomially bound.[5]) In other words, this "powering" technique allows us to amplify a subset of the solution space as long as it occupies at least a polynomially-small fraction of it. Clearly this is a very close relative of the gadget-based boosting techniques that form the backbone of Chapter 2. However, as we have seen, powering in this manner it is considerably more general and flexible because no surplus gadgetry (other than copies of G) is required and the ease with which the solution space can be inspected and disaggregated.

It is interesting to note that the "powering" technique described has a natural equivalent that lies outside the SP-framework. Observe that, in the example we just studied, we could build a $PAUS$ for H_1 out of a $PAUS$ for H by repeated sampling and rejecting. In other words, make up to k separate calls to the H oracle (with a single copy of G as input) and if no H_1 sample has been seen by the kth oracle call we return $\perp$.

3.8 Manual conversions between AP and SP-reductions

In this section we observe that many of the AP-reductions described throughout this thesis can be converted easily into SP-reductions, but that conversion in the opposite direction (i.e. converting SP-reductions to AP-reductions) appears to be the exception rather than the rule. This is not surprising given that in an informal sense SP-reductions were developed to "extend" the capabilities of the AP-reduction. These observations contribute to the work in Section 3.10, in which the the relationship between the $FPRAS$ and the $PAUS$ is explored, alongside a study of how AP-reductions and SP-reductions are one way of linking them together.

[5]To see this, note that $\ln(n^c/(n^c-1)) = \ln(1+1/(n^c-1))$ and, for $x < 1$, $x-(1/2)x^2 \leq \ln(1+x)$.

3.8.1 Converting AP-reductions to SP-reductions

Intuitively, the "conversion" of an AP-reduction $\#X_1 \leq_{\mathsf{AP}} \#X_2$ is the modification of the gadgetry and computational processes involved to produce the SP-reduction $X_1 \leq_{\mathsf{SP}} X_2$. Many of the AP-reductions in Chapter 2 convert relatively easily and directly, and we look at several examples of these, although we do not attempt to offer a general conversion process. This is for several reasons. Firstly, we have no way of knowing the whole universe of reductions possible within the AP-reduction framework; this is why we speak of *apparent* rather than *actual* limitations of AP-reductions. (In particular, it must be remembered that most of our AP-reductions involve H-colouring, and make heavy use of gadgetry, when of course there is no requirement that an AP-reduction involves either H-colouring or gadgetry. This is why we have elected to use the problem placeholders X_1 and X_2 in this section, rather than the H-colouring specific H_1 and H_2.) Secondly, as mentioned earlier there are AP-reductions $\#X_1 \leq_{\mathsf{AP}} \#X_2$ for which there exist natural and direct sampling equivalents, but (mainly for stylistic reasons i.e. related to the way we have defined the SP-reduction) they do not fit cleanly inside the SP-framework. Finally, and more fundamentally, we know concrete examples of AP-reductions for which it is difficult to envisage a clean, direct conversion to the sampling world, either in or outside the SP-framework. (That is, it is difficult to envisage how the reduction might be moved to the sampling world without significantly deviating from the method and mechanics of the original AP-reduction.)

We now look at various different reduction techniques introduced in Chapter 2 and examine how well they map over to the sampling world, and in particular whether they fit into the SP-reduction framework.

Rounding reductions

Most of the reductions in Chapter 2 use the "rounding" technique. That is, each of the N items to be counted come up Z times as full-colourings, which altogether comprise

NZ elements in the solution space, and non-full colourings account for P elements in the solution space where $P/Z \leq 1/4$. Hence, it is natural to let Y_1 be the set of full-colourings and Y_0 be the set of non-full colourings, so $|Y_1| = NZ$ and $|Y_0| = P$. Now, let us assume the reduction in question is $\#H_1 \leq_{\mathsf{AP}} \#H_2$. The basic principle underpinning the corresponding sampling reduction is simple:- take an H_2 sample, return $\perp$ if the sample is non-full, and read off the relevant H_1 sample otherwise. H_1 solutions are (in most cases) distributed uniformly throughout the full-colourings of H_2 so there is no problem with satisfying (3.2). However, the fact that the AP-reduction may only show $P/Z \leq 1/4$ means we may have to adopt a more sophisticated function $f(G, \epsilon)$ than simply $f(G, \epsilon) = G'$ (where G' is the input to $\#H_2$ that the AP-reduction codes up the input to $\#H_1$ as) if we are to ensure $|Y_0|/|Y_1| \leq \epsilon/4$. So what we do is use the *powering* technique, as described in Section 3.7. That is, we let $f(G, \epsilon)$ equal a polynomial number of disconnected copies of G', and let Y_0 comprise those colourings where every single copy of G' is coloured badly (i.e. with a non-full colouring.) (All other colourings are in Y_1.) We use as many copies of G' as is required[6] to force $|Y_0|/|Y_1| \leq \epsilon/4$. The only downside of this technique is that the input to H_2 is potentially disconnected, although this is not a problem in this thesis.

If disconnection is not desirable there may be alternative conversion strategies possible, depending on the exact nature of the AP-reduction in question. However, these may not (strictly speaking) be SP-reductions. For example, the following cannot be converted to an SP-reduction because it potentially requires randomization (and in SP-reductions g must be deterministic.) Nonetheless, it is worth repeating because it highlights an important general principle about small ϵ. It proceeds thus; assume again that the AP-reduction in question is $\#H_1 \leq_{\mathsf{AP}} \#H_2$. In most cases throughout Chapter 2 we are fairly relaxed about choosing the size of gadgets, generally choosing them to be bigger than necessary, so in these cases P/Z is not just less than $1/4$ but in fact exponentially small in n (or some higher polynomial). That is, in such cases it

[6] Given that $P/Z \leq 1/4$ we know full-colourings trivially constitute at least a polynomial fraction of the solution space, so we are guaranteed to need no more than a polynomial number of copies of G'.

is reasonable to assume $P/Z \leq c^{-n}$ for some constant c. Now, we can build a $PAUS$ for H_1 from a $PAUS$ for H_2 as follows; let (G, ϵ) be the input to the H_1 sampler. If $c^{-n} \leq \epsilon/2$ then we pass $(G', \epsilon/2)$ as input to the H_2 sampler (where G' is the graph we turn G into in the AP-reduction), reading off the appropriate H_1 colouring if the sample returned is a full colouring, and returning $\perp$ otherwise. If $\epsilon/2 \leq c^{-n}$ then we can build an *exactly* uniform sampler (which is obviously a $PAUS$) without even calling the $PAUS$ for H_2, but instead deploying the following "brute force" sampling algorithm. We exhaustively consider in turn all $|V(H_1)|^n$ possible mappings $G \to V(H_1)$, discard those which are not valid $H_1(G)$ colourings, and pick one of the remaining colourings uniformly at random. This is an *exactly* uniform sampler because it makes no use of the H_2 oracle; the running time is approximately $|V(H_1)|^n$, but this is not too large because it is clearly no more than polynomial in $1/\epsilon$.

Both of the above two techniques assume that $H_1(G)$ colourings are distributed uniformly throughout the full-colourings of $H_2(G')$. As seen in the proof of Lemma 2.8 (page 77), this is not always the case. However, it is not too difficult to convert this to an SP-reduction; the main point is that, in the AP-reduction, we have shown that a maximum-size independent set can come up at most $e^{\epsilon/42}$ times more than any other. So, if we port the reduction straight across, we see that (3.2) is satisfied because (3.5) is immediately satisfied. However, there is some work to do to ensure that (3.4) is satisfied. It is inadvisable to use the powering technique here because it skews the distribution of H_1 samples amongst the H_2 samples. So, the best thing to do is to modify the original reduction so as to explicitly engineer ϵ into the gadget building process i.e. to ensure that it beats a $\epsilon/4$ bound rather than a $1/4$ bound[7].

Other common types of reductions

Amongst other types of reductions, the most straightforward reductions to convert are

[7] We don't go into any more detail because in this thesis we do not need to convert such kinds of AP-reductions.

the parsimonious AP-reductions i.e. reductions $\#X_1 \leq_{\mathsf{AP}} \#X_2$ which map an $\#X_1$ input σ_1 to the $\#X_2$ input σ_2 such that $\#X_1(\sigma_1) = \#X_2(\sigma_2)$. If all the solutions in $X_1(\sigma_1)$ are actually constructed (as elements of $X_2(\sigma_2)$) then conversion to the SP-reduction domain is generally easy, because there is an easily computable bijection from $X_2(\sigma_2)$ to $X_1(\sigma_1)$. In terms of Chapter 2, examples of such reductions are those of the form $\#H \leq_{\mathsf{AP}} \#DownSets$, because every downset in the constructed partial order structure corresponds to a unique H-colouring.

In many instances "nearly parsimonious" reductions also convert fairly easily, although as we shall see there are caveats to this assertion. These are reductions $\#X_1 \leq_{\mathsf{AP}} \#X_2$ where each element from $X_1(\sigma_1)$ comes up $h(\sigma_1)$ times as an element of $X_2(\sigma_2)$, for some easily computable function h. (Crucially, as with parsimonious reductions, none of the elements in $X_2(\sigma_2)$ should be "bad" i.e. fail to map to an element of $X_1(\sigma_1)$.) At the simplest level, $h(\sigma_1)$ is a constant, such as in the observation that $\#BIS \leq_{\mathsf{AP}} \#P_4$ (on page 54) in which $h(G) = 2$. Assuming the colours of P_4 are labelled as in Figure 2.6 on page 55, it is clear that a P_4 sample can be turned into a BIS sample by mapping both b and b' to OUT and r, r' to IN. So, in essence, one BIS sample corresponds to two different P_4 samples. We call this process - where multiple elements from the solution space of X_2 are collapsed into one element from the solution space for X_1 - *flattening*. As a second example, consider the reduction for graph 35 on page 70, in which we observe that $\#H(G) = (\#IS(G))^2$. This very pure algebraic relationship has a combinatorial underpinning so the reduction $IS \leq_{\mathsf{SP}} H$ can be achieved simply by flattening H samples:- each independent set comes up $\#IS(G)$ times as a colouring of $H(G)$. (For more details of the exact relationship, see Appendix A.3)

The opposite of flattening is *elevating*, and this is where SP-reductions struggle ever so slightly. Whereas flattening reductions generally reflect a one-to-many mapping from $X(\sigma)$ to $Y(f(\sigma, \epsilon))$, elevating reductions tend to represent many-to-one relationships between the same sets. For example, consider the reduction $\#P_4 \leq_{\mathsf{AP}} \#BIS$. Each BIS sample corresponds to 2 P_4 colourings, because we have a choice as to which orientation of P_4 to map to. Clearly a $PAUS$ for BIS can be used to build a $PAUS$ for P_4; simply take a BIS sample and map it u.a.r. to one of the two P_4 colourings that correspond to it. However, the fact that some kind of randomisation is required in the mapping from $Y(f(\sigma,\epsilon))$ to $X(\sigma)$ can be problematic because, as the definition currently stands, the "mapping back" function g is deterministic. This is a notable shortcoming of the SP-reduction, but it is not serious either from a more general sampling reduction viewpoint or from the viewpoint of this thesis. For example, one possibility (in the case of converting $\#P_4 \leq_{\mathsf{AP}} \#BIS$ to an SP-reduction) is to pass as input to the P_4-sampler a copy of G (the graph input to P_4) alongside an extra, disconnected bit. Given that the bit can be IN or OUT independent of what happens to G, it acts as an ad-hoc random bit generator, and we can use its IN/OUT status as the determinant of which P_4 orientation we map to. (If more sophisticated gadgetry is used we can produce a wider array of random numbers.) However, if we step outside the SP-framework we can solve most of these "random mapping" problems elegantly with the Observation below. (Thanks to Leslie Goldberg for this.)

Observation 3.2 *Let P be a set of elements, partitioned into l disjoint sets, $P_1,...P_l$. Let Q be a set of l elements $(q_1,...,q_l)$ and let π be a distribution on Q defined by $\pi(q_i) = |P_i|/|P|$. Now, suppose we have an approximate sampler on π, which produces a distribution π' on Q which deviates no more than ϵ' from π, and runs in time at most poly$(|Q|, 1/\epsilon')$. If for all $q_i \in Q$ we can efficiently choose u.a.r. a sample from P_i when presented with q_i, we can build a $PAUS$ for P.*

Proof. We have put the proof in the Appendix A.4; the proof is not particularly complicated but to present it here would break up this section with unnecessary technical details. The central idea is that a $PAUS$ for P can be built by taking a sample from Q (using our approximate sampler for π) and, if q_i is the sample obtained, choosing an element u.a.r. from P_i. $\square$

This observation allows us to easily build (say) a $PAUS$ for P_4 out of a $PAUS$ for BIS. We know $\#P_4(G) = 2\#BIS(G)$, so partition $P_4(G)$ into $P_1,...,P_l$ where each

P_i contains the two P_4 colourings corresponding to a particular independent set colouring of G. Since all the P_i are the same size, π (as defined in the above Observation) is uniform on $BIS(G)$ and hence our $PAUS$ for BIS takes the role of the approximate sampler (for $BIS(G)$) mentioned in the Observation.

The P_4-to-BIS example does not fully demonstrate the flexibility that the Observation permits. A more general application of the Observation can be seen in reducing an "unweighted" sampler to a "weighted" sampler. Consider, for example, the graph H_1 in Figure 3.4. As discussed in Section 2.2.1, one way of representing the graph H_1 when it has (say) weight 2 on its b vertex is by a new graph with 2 b vertices e.g. H_2 in Figure 3.4. So, if we want a sampler for H_1 with weight 2 on its b vertex, a sampler for H_2 is (in effect) sufficient. An alternative way of "visualising" a sampler for H_1 with weight 2 on its b vertex would be to have a sampler which returns H_1 samples but from a non-uniform distribution i.e. the more b vertices a colouring from $H_1(G)$ has, the higher the probability that it comes up. We can formalise this as follows. In particular, let π be the distribution on $H_1(G)$ such that, if $c_1 \in H_1(G)$ is a colouring with k_1 vertices coloured b and $c_2 \in H_1(G)$ is a colouring of G with k_2 vertices coloured b, $\pi(c_1)/\pi(c_2) = 2^{k_1-k_2}$. (To see where this distribution comes from, consider that an $H_1(G)$ colouring with k vertices coloured b comes up 2^k times as an $H_2(G)$ colouring.) Now, if we call the H_2 sampler the "unweighted" sampler and the π distribution the "weighted" sampler, the Observation allows us to build a $PAUS$ for the "unweighted" sampler from the approximate "weighted" sampler. To see this, suppose $H_1(G)$ contains l colourings $c_1, ..., c_l$. We partition $H_2(G)$ into $P_1, ..., P_l$ such that P_i contains all the colourings from $H_2(G)$ which correspond to c_i (i.e. if c_i contains k vertices coloured b, $|P_i| = 2^k$.) It follows from our definition of π that $\pi(c_i) = |P_i|/|P|$ and hence the precondition of the Observation is satisfied.

The reason reducing "unweighted" samplers to "weighted" samplers is useful is that, occasionally, directly converting an AP-reduction to the sampling world essentially creates a weighted sampler, which is not a uniform distribution, and it is important to convert the sampler to an unweighted sampler so we can continue to work with uniform

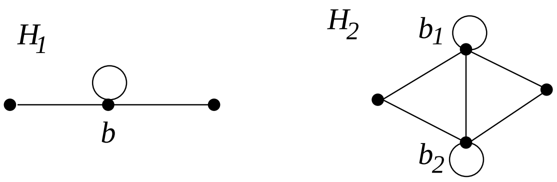


Figure 3.4: *Sampling H_2 is one possible way of obtaining samples of H_1 with weight 2 on its b vertex*

distributions.

While on the topic of "artificial" constraints imposed by our definition of the SP-reduction, it is important to note that an approximate H-colouring sampler that assumes its input G is connected (H_{con}) is *not* automatically SP-interreducible with the same sampler operating on general G (H_{gen}.) This compares unfavourably with our observation in Chapter 2 that, in the counting world, AP-interreducibility between connected G/general G versions of a problem does hold. (This interreducibility is why, in terms of AP-reductions, we are allowed to assume the input G is connected.) To see why SP-interreducibility does not automatically hold, note that an SP-reduction $X \leq_{\mathsf{SP}} Y$ only allows one call of the Y oracle. Now, the natural way to reduce H_{gen} to H_{con} for disconnected input G is to call the H_{con} sampler once for each component of G, but the SP-reduction structurally prevents this[8]. This is annoying because a $PAUS$ for sampling H-colourings from general G can very easily be built from a $PAUS$ for sampling H-colourings from connected G: if we need samples to accuracy ϵ we simply use the connected sampler at accuracy ϵ/k for each of the k components in G. Clearly, therefore, this is not a serious structural limitation, and indeed it does not cause us problems in this thesis. It is worth being aware of, however.

Problems and subtleties

As mentioned earlier, the shortcomings of the SP-reduction that Observation 3.2 addresses are relatively cosmetic. That is, the difficulties do not arise because there does

[8] $H_{con} \leq_{\mathsf{SP}} H_{gen}$ follows immediately, of course.

not exist a natural sampling analogue of the counting reduction, but because this natural sampling analogue does not fit well into our SP-reduction world. However, there are some AP-reductions for which it is not apparent how a *direct* sampling equivalent might be built, irrespective of definitions. Consider Lemma 2.15. This Lemma states that for all bipartite H, $\#H \leq_{\mathsf{AP}} \overleftarrow{\#H}$. It proves this by showing that $\#H(G)$ can be computed simply by computing $\overleftarrow{\#H}(G)$ and then adding this to $\overleftarrow{\#H}(G')$, where G' is G with its bipartition reversed.

However, it is not apparent how this might be converted to an SP-reduction (or, indeed, any sampling reduction) without significantly deviating from the simplicity of the AP-reduction. The problem is that we don't in general know (unless H is symmetric) whether to return a sample from $\overleftarrow{H}(G)$ or $\overleftarrow{H}(G')$, because we don't know how much of the solution space of $H(G)$ each occupies. Now, though we have not explored this issue, it is admittedly possible to imagine a situation where we try and use our sampler (in conjunction with the fact that approximately counting H-colourings can be reduced to the problem of approximately sampling H-colourings - see Section 3.10) to generate estimates of $\overleftarrow{\#H}(G)$ and $\overleftarrow{\#H}(G')$ and then (contingent on the ratio of these two values) return a sample from either $\overleftarrow{H}(G)$ or $\overleftarrow{H}(G')$. However, as is easy to appreciate, such a sampling reduction would not really be a "direct" mapping of the original AP-reduction it sought to mimic. Indeed, this would seem to be a recurring problem whenever the AP-reduction to be converted is not "one shot", e.g. makes multiple oracle calls and adds the results together to provide an overall solution. Curiously, this problem seems to be the trade-off for the flexibility that sampling affords us in most other circumstances.

3.8.2 Converting SP-reductions to AP-reductions

The original reason for defining SP-reductions was to escape some of the apparent limitations posed by AP-reductions. It is no surprise therefore that most SP-reductions cannot obviously be converted to AP-reductions; we have already seen how SP-reductions seem to have an advantage in the areas of interleaving multiple subproblems together and powering polynomially small fractions of the solution space. In limited circumstances, however, such a conversion is possible. This is when only one subproblem is encoded - or several subproblems are encoded that are identical modulo labelling - and no powering is involved. (As we demonstrate shortly, it is not clear how the powering technique maps across into the counting world.) In such instances the reduction gadgetry tends to map fairly easily across.

To see this, note that in most SP-reductions we show that undesirable colourings are inconsequential by proving $|Y_0|/|Y_1| \leq \epsilon/4$. Where only one subproblem is encoded, $|Y_1|$ is (as mentioned earlier) usually equal to NZ, where Z is the value we would divide by in a corresponding rounding reduction. However, for the rounding reduction to work, we actually need $|Y_0|/Z \leq 1/4$. In practice, however, our SP-reductions do tend to take Z as a lower bound on $|Y_1|$ and thus in many cases prove the slightly stronger result of $|Y_0|/Z \leq \epsilon/4$. Combining this with the fact that $\epsilon \leq 1$ thus confirms that these SP-reductions can be mapped to AP-reductions.

As a quick example of how powering does not appear to work too well in the counting world, consider the example where (for all G) $\#H(G) = \#H_1(G) + \#H_2(G)$, both $\#H_1(G)$ and $\#H_2(G)$ are non-zero and $\#H_1(G) > \#H_2(G)$. At first sight it looks like powering up the input to $\#H$ (i.e. by producing k copies of G) might be a way to build the reduction $\#H_1 \leq_{\mathsf{AP}} \#H$, because the difference between $\#H_1(G)$ and $\#H_2(G)$ will be blown up to exponential levels. Yet the result from the $\#H$ oracle will be (approximately) $(\#H_1(G) + \#H_2(G))^k$, and this is an exponentially larger value than $\#H_1(G)^k + \#H_2(G)^k$, so it is difficult to see how an approximation to $\#H_1(G)^k$ could be extracted from this quantity.

Given that many of the AP-reductions we have used can be converted to SP-reductions, it is (where possible) preferable to demonstrate AP-reductions rather than SP-reductions. As we show in the following section it is probable that, at least in terms of "one-shot" reductions, showing an AP-reduction $\#X \leq_{\mathsf{AP}} \#Y$ is a stronger statement than showing $X \leq_{\mathsf{SP}} Y$. This is why in several of the subsequent proofs throughout this thesis we use an SP-reduction for the general case but provide an auxiliary AP-reduction for the special cases that we know the AP-reduction can cope with. (In most cases, this is where only one subproblem - or isomorphic multiple subproblems - are necessary to seal the proof.)

3.9 SP-reductions and SAT

In Chapter 1 we repeated the result (taken from [8]) that for all counting problems $\#X \in \#P$ - where $\#P$ is the complexity class "number P" - $\#X \leq_{\mathsf{AP}} \#SAT$. (This of course includes the whole domain of H-colouring problems.) There is an analogous result in the sampling domain. In particular, if $\#X \in \#P$, then $X \leq_{\mathsf{SP}} SAT$. This follows because the $\#X \leq_{\mathsf{AP}} \#SAT$ reduction rests on the existence of proofs of Cook's theorem that use *parsimonious* reductions. In other words, these proofs show that for any problem $\#X \in \#P$ and its input, there exists a formula (used as input to $\#SAT$) such that there is a one-to-one correspondence between satisfying assignments of the formula and accepting computations of the Turing machine/input pair for $\#X$. We know parsimonious counting reductions map directly into the SP-domain, so $X \leq_{\mathsf{SP}} SAT$ flows directly from this fact.

Though we know that (for all $\#X \in \#P$) $X \leq_{\mathsf{SP}} SAT$, we do not know whether (in general) $\#X \equiv_{\mathsf{AP}} \#SAT$ implies $X \equiv_{\mathsf{SP}} SAT$. This is because (unlike the direction $\#X \leq_{\mathsf{AP}} \#SAT$) there is no "generic form" for reductions $\#SAT \leq_{\mathsf{AP}} \#X$, and as we have seen from Section 3.8.1 (specifically page 125), there are AP-reductions which in any practical sense do not seem to map to SP-reductions. Fortunately, we do know that, for *most* of the problems $\#X$ encountered in this thesis (and in [8]) which have the property that $\#X \equiv_{\mathsf{AP}} \#SAT$, it is also the case that $X \equiv_{\mathsf{SP}} SAT$. To see this, observe that the "root" $\equiv_{\mathsf{AP}} \#SAT$ problems used in this thesis and [8] are $\#LargeCut$, $\#LargeIS$ and $\#LargeIS\text{-}Cubic$ (where the last problem in the list is $\#LargeIS$ with its input restricted to cubic graphs.) Crucially, there are parsimonious (or near-parsimonious) reductions possible between $\#SAT$ and each of these problems; Appendix A of [8] demonstrates a near-parsimonious[9] reduction $\#SAT$ to $\#LargeIS$-

[9] Each solution to the SAT input comes up a constant number of times in the solution space of $LargeIS\text{-}Cubic$, so not parsimonious but so close that that conversion to an SP-reduction is still

$Cubic$ (which also takes care of $\#LargeIS$), and elsewhere in [8] it is noted that a parsimonious reduction from $\#SAT$ to $\#LargeCut$ can be found in [17]. Hence, any AP-reduction from one of these problems to a problem $\#X$, satisfying the requirement that the AP-reduction maps successfully to an SP-reduction, also demonstrates that $SAT \leq_{\mathsf{SP}} X$. Virtually all of the $\equiv_{\mathsf{AP}} \#SAT$ classifications encountered both in this thesis and [8] have this property, because if we trace back down the chain of AP-reductions used in each such classification we observe that in most cases all the AP-reductions are transferable to SP-reductions and that $\#LargeCut$ or $\#LargeIS$ lie at the bottom of the reduction chain.

The only $\equiv_{\mathsf{AP}} \#SAT$ problems from [8] that we are not sure about (in terms of whether $\equiv_{\mathsf{AP}} \#SAT$ automatically yields $\equiv_{\mathsf{SP}} SAT$) are $\#H$ where H is a "Hell and Nešetřil" graph i.e. loopless and non-bipartite. These are the graphs for which the *decision* problem is NP-complete; unfortunately, the NP-hardness reduction (found in [15]) is far from parsimonious, so an open question is whether the "Hell and Nešetřil" graphs are $\equiv_{\mathsf{SP}} SAT$. It so happens that, beyond the immediateness of their $\equiv_{\mathsf{AP}} \#SAT$ classification, these graphs do not figure prominently in this thesis, so this grey area of our understanding does not cause us significant problems.

3.10 Approximate counting and approximate sampling

It is well-known that, for the class of *self-reducible* problems in $\#P$, approximate counting and approximate sampling are of similar complexity. Informally, if you can approximately count the number of elements in the solution space then you can approximately sample from it, and vice-versa; the seminal paper on this topic is by Jerrum, Valiant and Vazirani [16]. To further explore this relationship we have to introduce a new piece of terminology - used in (amongst others) [12] - the *fully polynomial approximate sampler* ($FPAS$). The $FPAS$ is similar to the $PAUS$ but with two differences, one purely stylistic and one somewhat more fundamental. The stylistic difference is that, as its

trivially possible.

name suggests, the $PAUS$ is explicitly dedicated to approximately sampling from *uniform* distributions. The more significant difference is that if (σ, ϵ) is the input, a $PAUS$ is allowed running time poly($|\sigma|$, ϵ^{-1}), whereas an $FPAS$ is only allowed running time poly($|\sigma|, \log(\epsilon^{-1})$). (Unsurprisingly we use the title $FPAUS$ for an $FPAS$ that is explicitly dedicated to approximately sampling from the uniform distribution.) Hence, because of this difference in running times, demonstrating the existence of an $FPAUS$ for a problem is a stronger result than demonstrating a $PAUS$.

The $\log(\epsilon^{-1})$ bound is more commonplace for a sampler, so the reader may be wondering why this thesis (and [14], where the $PAUS$ was first defined) uses the $PAUS$ and not the $FPAUS$. This is because the $PAUS$ was defined to facilitate the proof of Theorem 4.1 from Chapter 4, which also appears in [14]. As part of this proof it is necessary to use gadgets which are polynomially large in ϵ^{-1}, and hence the more liberal sampler has to be used[10].

(The reader will observe that, with the exception of the result in Chapter 4, quite a few of the SP-reductions we build actually meet the stronger poly($|\sigma|, \log(\epsilon^{-1})$) running-time bound. The proof in Section 3.6 is one such SP-reduction. So, if desired a stronger (i.e. faster) version of the SP-reduction could perhaps be defined to take advantage of these situations, leading to reductions between $FPAUS$es instead of reductions between $PAUS$es, as we have at present. However, we have not explored this issue in any detail, and only use the standard $PAUS$-to-$PAUS$ SP-reduction in this thesis.)

Interestingly, we know that in certain circumstances a $PAUS$ for a problem can be used to *automatically* build an $FPAUS$ for that problem. It transpires that, owing to a result from [16], the $PAUS$ can be "powered up"[11] to an $FPAUS$ as long as the domain of elements to be sampled is not larger than exponential in the size of the input,

[10] A useful artefact of this restriction, however, is that the resulting Theorem is actually *stronger* than if we dealt solely in terms of the $FPAUS$.

[11] In this context "powered up" does not refer to the process of powering used throughout this section, although in spirit it is a very similar idea.

and the problem in question is self-reducible.

Given that for H-colouring problems the size of the solution set is bound above by an exponential in n the existence of a $PAUS$ for self-reducible H problems appearing in this thesis could be used to build an $FPAUS$ for those problems. This allows us to exploit the result from [12], in which the following result[12] concerning (specifically) H-colouring appears:- if the sampling problem for a given H has an $FPAUS$, the counting problem has an $FPRAS$. In other words, if we can efficiently approximately sample H-colourings then we can efficiently approximately count them. (So, for self-reducible H, a $PAUS$ for H yields an $FPRAS$ for $\#H$. For H not known to be self-reducible, we presently have the slightly weaker result that an $FPAUS$ for H yields an $FPRAS$ for $\#H$. It may be, however, that the "$PAUS$ implies $FPRAS$" result does actually apply to all H-colouring problems.)

The truth or otherwise of the other direction - reducing sampling to counting - is open. The problem is that we do not know whether, in general, H-colouring problems are self-reducible. The sticking point seems to be that, working within the kind of framework described in [16], approximately sampling H-colourings can be reduced fairly naturally to the problem of approximately counting *list* H-colourings, but it is not obvious at all how we might reduce approximately sampling H-colourings to approximately counting normal H-colourings. Certain well known problems, such as $\#IS$, $\#BIS$ and $\#SAT$ are self-reducible, but for the majority of graphs it is not clear whether the JVV technique applies directly. Thus, we do not know whether in general the existence of an $FPRAS$ for an H-colouring problem implies the existence of an $FPAUS$ (or a $PAUS$.)

It may in fact be the case that, for certain H-colouring problems, approximate counting is actually *easier* then approximate sampling. (Hence it is not yet clear to what extent negative sampling results - such as Theorem 4.1 in the following chapter - imply the existence of analogous negative approximate counting results.)

[12] The actual result proven is more general than this.

3.10.1 The relevance of AP-reductions and SP-reductions

This uncertainty over the true nature of H-colouring with respect to self-reducibility means that AP-reductions and SP-reductions have continued relevance as separate entities in their own right. It is instructive to examine the differences between the statement $\#H_1 \leq_{\mathsf{AP}} \#H_2$ and $H_1 \leq_{\mathsf{SP}} H_2$ and in particular what they tell us about approximately counting/sampling H_1-colourings once we know the complexity of approximately counting/sampling H_2-colourings.

Suppose neither H_1 nor H_2 are self-reducible and we have the result $H_1 \leq_{\mathsf{SP}} H_2$, but not $\#H_1 \leq_{\mathsf{AP}} \#H_2$. (Various different permutations are possible if one or both of H_1 and H_2 are self-reducible but we prefer to deal with the most restrictive case.) Then a $PAUS$ for H_2 can be used to build a $PAUS$ for H_1. However, the discovery of an $FPRAS$ for $\#H_2$ does not tell us anything (in itself) about H_1 (or $\#H_1$) because most SP-reductions don't seem to convert to AP-reductions.

However, suppose instead (while observing the same restrictions on H_1 and H_2) that we know $\#H_1 \leq_{\mathsf{AP}} \#H_2$. An $FPRAS$ for $\#H_2$ lets us build an $FPRAS$ for $\#H_1$. Furthermore, if the AP-reduction converts to an SP-reduction - which, crucially, it will do in many cases - we know that a $PAUS$ for H_2 will yield a $PAUS$ for H_1.

The point of this is that when we have a result $\#H_1 \leq_{\mathsf{AP}} \#H_2$ we (probably) learn something new about the H_1 approximate sampling/counting problem if an $FPRAS$ is found for $\#H_2$ *or* a $PAUS$ is found for H_2. However, if we only have the result $H_1 \leq_{\mathsf{SP}} H_2$ then we *only* learn something new about approximately sampling/counting H_1 if we find a $PAUS$ for H_2. This is why, in general, we prefer to demonstrate AP-reductions where possible - they "tell us more about the world" than SP-reductions. This is also why we tend to (where possible) demonstrate restricted AP-reduction corollaries to SP-reduction results.

Closing comment

As a final comment, it is important to not get confused between comparing reductions (i.e. comparing AP-reductions and SP-reductions) and comparing absolute complexity (i.e. comparing the complexity of approximate sampling and approximate counting.) For example, if we prove that $H_1 \leq_{\mathsf{SP}} H_2$ and also that $\#H_1 \leq_{\mathsf{AP}} \#H_2$ this does not *in itself* make a statement on how difficult approximately sampling H_2 colourings is compared to approximately counting H_2 colourings. On a related note, we have explained that most SP-reductions do not seem to "map" to AP-reductions. While this is certainly true given our current (lack of) understanding about utilising the full power of AP-reductions, we do on a number of occasions throughout this thesis argue that, in certain circumstances, the existence of a "pluralistic"[13] SP-reduction does constitute some evidence that an analogous AP-reduction exists. This may sound a touch contradictory but it seems to be empirically supported by our experience of *hardness* reductions. Recall that the SP-reduction is particularly useful for H-colouring hardness results, because it enables us to skirt around the problem of not being able to delineate between multiple sub-problems occurring simultaneously. (Often these subproblems are subgraphs of H.) Yet, if we are given a specific graph H and asked to manually develop whatever gadgetry is required to ensure that a single, unique subgraph is pointed out, we are usually successful. In other words, there do seem to be AP-reductions lurking hidden away that we can manually prise out using hand-crafted gadgetry, but we are not very good at *generalising* this process. This is why, for SP-reductions that effectively bolt together multiple AP-reductions to achieve a hardness result, we tend to view the SP-reduction as limited evidence that an AP-reduction does (somewhere!) exist.

[13] That is, one that interleaves multiple sub-problems together

Chapter 4

Approximately sampling H-colourings where H has no trivial components is at least as hard as approximately sampling BIS

4.1 Introduction

In this chapter, we prove that for *any* graph H with no trivial components, the problem of nearly-uniformly sampling H-colourings is intractable in a complexity-theoretic sense, which constitutes weak provisional evidence that approximately sampling such H-colourings is intractable in a more general sense. In particular, we show that for any graph H with no trivial components, $BIS \leq_{\mathsf{SP}} H$ i.e. that the problem of approximately sampling independent sets in bipartite graphs is SP-reducible to the problem of approximately sampling H-colourings. Thus, if there were a $PAUS$ for any such H-colouring problem, there would also be a $PAUS$ for BIS (and, by the self-reducibility of the independent set problem, an $FPRAS$ for $\#BIS$). In [8], DGGJ have shown that $\#BIS$

is complete in a certain logically-defined subclass of $\#P$ which includes problems such as counting downsets in partial orders and counting satisfying assignments in "restricted Horn" CNF Boolean formulas. Thus, a $PAUS$ for sampling H-colourings would give an $FPRAS$ for the entire complexity class. (This logically-defined subclass is called $\#RH\Pi_1$ and is discussed further in Chapter 7.) In fact, the result holds even if the input G is restricted to being a connected bipartite graph. Additionally, in Section 4.6 we show that the $BIS \leq_{\mathsf{SP}} H$ result is "best possible" in the sense that there exists a graph H with one non-trivial component and one trivial component for which there exists a $PAUS$.

Recall from Sections 3.10 and 3.10.1 that, formally speaking, the result $BIS \leq_{\mathsf{SP}} H$ does not tell us much about the relationship between $\#BIS$ and $\#H$, because for most H we do not know whether H is self-reducible. However, in light of the comments at the very end of Section 3.10.1[1], we think that the $BIS \leq_{\mathsf{SP}} H$ result does constitute *some* informal evidence that a $\#BIS \leq_{\mathsf{AP}} \#H$ result does also exist.

This chapter is an adaptation of work (co-authored by this author alongside Goldberg and Paterson) that first appeared in [14]. (Comments from Martin Dyer, Paul Goldberg and Mark Jerrum were also useful in the development of this work.)

4.2 Definitions/conventions

One of the most cast-iron conventions of this thesis is that inputs to H-colouring problems are assumed to be connected. This is founded on the observation (from Chapter 2) that, in terms of AP-reducibility, the version of a problem $\#H$ where the input G is assumed to be connected is interreducible with the version where the input G may be disconnected. Recall also that our definition of the SP-reduction is such that we cannot assume that connected and disconnected versions of a sampling problem are

[1]Where we observe that, because we seem quite good at "hand-crafting" gadgets to point out a unique subgraph in a given graph H, induction-based SP-reductions designed to cope with the simultaneous pointing out of multiple subgraphs may indicate the existence - somewhere! - of an AP-reduction analogue to the SP-reduction result.

SP-interreducible, even if in practice a $PAUS$ for one is easily turned into a $PAUS$ for the other. On this basis we have decided to continue the "G is connected" convention through this chapter, with one mild exception, as we now explain.

In all the *sampling* problems we define in this chapter, we assume the input graph G is connected. Thus, up to and including the proof of Lemma 4.5 in Section 4.5, we assume (and ensure) that input graphs are connected. However, in Lemma 4.6, where $\#BIS$ makes its first appearance in this chapter, we assume that inputs to $\#BIS$ may be disconnected. This mild deviation from convention is simply to ensure consistency with the original paper that Theorem 4.1 came from; following the close of this chapter we revert back fully to the assumption that all inputs to H-colouring problems (whether sampling or counting) are connected.

Let us define the sampling problem biH as follows. Given a bipartite graph G, biH returns an H-coloring of G chosen uniformly at random. So biH is simply the restriction of H to bipartite input. In this regard, it is important to note the distinction between biH and the $bi(H)$ transformation we discussed earlier in Chapter 2 (see page 54.). If H is non-bipartite, biH is not *quite* the same problem as uniformly returning a $bi(H)$ sample of the input G. To see this, consider for example the H graph representing the IS problem, where the looped colour is b and the unlooped colour is r. The sample returned by biH would still be in terms of b and r but, because $bi(H)$ is itself a bipartite graph and hence has two orientations, a $bi(H)$ sample of G would be in terms of (for example) b, b', r and r'. (Note also that it is trivially the case that $biH \leq_{\mathsf{SP}} H$.)

The main result of this chapter is as follows:

Theorem 4.1 *Suppose that H is a graph with no trivial components. If the sampling problem biH has a $PAUS$ then the sampling problem BIS has a $PAUS$ and $\#BIS$ has an $FPRAS$. Thus, every problem which is AP-interreducible with $\#BIS$ has an $FPRAS$.*

To aid the reader, we now provide a brief intuitive explanation of the proof structure. The proof has three main components; we consider them in reverse order.

The *third* component (Lemma 4.6) shows that a $PAUS$ for BIS can be used to create an $FPRAS$ for $\#BIS$.

The *second* component (Section 4.5) proves that, for any H with no non-trivial components, we can use gadgetry to find one or more bipartite graphs with certain properties - connected, non-trivial, with at least one vertex on each side of its bipartition connected to every vertex on the other side of its bipartition - that are reducible to it. We do this by (essentially) using two $maxdeg$ gadgets to "grab" one or more edges of H, thus ensuring that all such grabbed edges have high-degree colours at either end. Then, if (c, d) is such an edge, we note that the bipartite graph induced by bipartition $adj(c) \cup adj(d)$ has the properties we require. By Lemma 4.3 - which we discuss in a moment - we know that BIS is reducible to each of these bipartite graphs, so we can "bolt together" these reductions into one amalgamated graph. Then, when we take an H-sample of this amalgamated graph, we can use whichever reduction is appropriate (as determined by the high-degree edge that the sample picks out) to grab a BIS sample.

The *first*, and core, component shows how to reduce BIS to a bipartite graph with the properties listed above. The proof is inductive, based on the idea that the result is true for graphs smaller than H. Essentially, we combine two different reduction strategies. Suppose G is the input to BIS. The main idea is that we code up a representation of G with very special vertex encodings. Each vertex encoding has numerous other reductions "hanging off" it:- the reductions that we know exist (by induction) for subgraphs of H that are strictly smaller than H. So, when we take an H-sample of our G-encoding, we first examine all of our vertex encodings. If at least one of them is coloured in such a way that it points out a proper, non-trivial subgraph of itself in the reductions hanging off it, we can zone in on the appropriate reduction and (by induction) pull out a BIS sample from it. If none of the vertex encodings are coloured in this way, then they must each be in one of two very special states, which we can think of as being IN and OUT of the independent set. In other words, we can in this eventuality obtain an independent set sample by looking at the whole G-encoding.

The rest of the work in this proof component is showing how to avoid a skewed distribution amongst independent set samples (when read off from the wider G-encoding) and showing that unusable colourings of the G-encoding occupy only a tiny, irrelevant fraction of the solution space.

4.3 Technical lemmas

Recall that $\nu(a,b)$ denotes the number of onto functions from a set of size a to a set of size b. We need to use the following lemma, which we first used in Chapter 2.

Lemma 2.4 *(DGGJ) If a and b are positive integers and $a \geq 2b \ln b$ then*

$$b^a\,(1-\exp(-a/(2b))) \leq \nu(a,b) \leq b^a.$$

We also need the following technical lemma.

Lemma 4.2 *Suppose c_1 and c_2 are fixed positive reals with $c_1 < c_2$. For any $\delta > 0$ and any non-negative integers q and a_0, there are non-negative integers a and b with $a \geq a_0$ which are in $O((a_0+q)/\delta)$ and satisfy*

$$e^{-\delta} c_2^{a+q} \leq c_1^{b+q} \leq e^{\delta} c_2^{a+q}.$$

Proof. First, note that it would suffice to find non-negative integers a' and b' which are in $O(q'/\delta)$ and satisfy

$$e^{-\delta} c_2^{a'+q'} \leq c_1^{b'+q'} \leq e^{\delta} c_2^{a'+q'},$$

where $q' = q + a_0$ because we could simply set $a = a' + a_0$ and $b = b' + a_0$ which would imply $a' + q' = a + q$ and $b' + q' = b + q$.

Taking logarithms, what we need is

$$\left| b' - \frac{a' \log c_2 + q' \log(c_2/c_1)}{\log c_1} \right| \leq \frac{\delta}{\log c_1}. \tag{4.1}$$

Now let ρ be defined by $c_2 = c_1^{1+\rho}$. Then we want

$$\left| b' - (a'(1+\rho) + q'\rho) \right| \leq \frac{\delta}{\log c_1}. \tag{4.2}$$

For a positive integer r, we will choose $a' = q'r$, so we want

$$\left| b' - a' - \rho q'(r+1) \right| \leq \frac{\delta}{\log c_1}. \tag{4.3}$$

Let $R = \lceil 2 \log c_1 / \delta \rceil$. Lemma 19 of [8] says: *For any real $z > 0$ and any positive integer R there is an $x \in [1, \ldots, R]$ such that*

$$\min(zx - \lfloor zx \rfloor, \lceil zx \rceil - zx) \leq 1/R.$$

Thus, there is an $x \in [1, \ldots, R]$ such that $\rho q' x$ is within $1/R$ of a non-negative integer. If $x > 1$ we will set $r + 1 = x$. If $x = 1$ then note that $\rho q' 2$ is within $2/R$ of a non-negative integer, so we will set $r = 1$.

Now recall that $a' = q'r$, so $a' \in O(q'/\delta)$ as required. □

4.4 Sampling "left orientation" colourings

Suppose that H is a connected, bipartite graph, with vertex partition $(V_L(H), V_R(H))$. Recall from our discussion of "left orientation" and "right orientation" H-colouring on page 91 that the problem $\overleftarrow{\#H}$ is (for bipartite input G) the problem of counting the number of "left orientation" colourings of G i.e. those H-colourings of G where the vertices of $V_L(G)$ receive colours from $V_L(H)$. So, in terms of sampling, $\overleftarrow{H}$ is the problem of sampling from that domain.

Proceeding, recall from Chapter 2 that, in a connected non-bipartite H, a *universal* colour is one that is adjacent to every colour in $V(H)$, itself included. If a connected, non-bipartite H has a universal colour then we say that H is a universal graph. The notion of universality has a natural counterpart in the bipartite domain, as we now show.

A vertex in $V_L(H)$ is said to be *universal* if it is adjacent to every vertex in $V_R(H)$. Similarly, a vertex in $V_R(H)$ is said to be *universal* if it is adjacent to every vertex in $V_L(H)$. The graph H is said to be *universal* if both $V_L(H)$ and $V_R(H)$ contain at least one universal vertex.

Lemma 4.3 *Suppose that H is a connected non-trivial universal bipartite graph. Then $BIS \leq_{SP} \overleftarrow{H}$.*

Proof. We'll prove the lemma by induction on the number of vertices in H. For the base case, suppose that H has at most 4 vertices. The only connected non-trivial universal bipartite graph H with at most 4 vertices is the path on 4 vertices i.e. P_4. Let G be an input to BIS. There is a one-to-one correspondence between independent sets of G and $\overleftarrow{H}$-colourings of G: The endpoints of H point out the vertices which are in the independent set.

We will now move on to the inductive step. The high-level idea is the following. By considering the graph H, we will construct several graphs $H_{S_1}, \ldots, H_{S_{j+k}}$, each of which is smaller than H and satisfies certain conditions. By induction, for each i, there is an SP-reduction from BIS to $\overleftarrow{H_{S_i}}$. If we apply this reduction to our instance G of BIS, we get an instance G_i of $\overleftarrow{H_{S_i}}$. Our goal is to construct an instance $f(G, \epsilon)$ of $\overleftarrow{H}$. We do this by "glueing together" the various G_i's. Now consider the constructed instance $f(G, \epsilon)$ of $\overleftarrow{H}$. When we sample from the output distribution $\overleftarrow{H}(f(G, \epsilon))$, we would like to recover the output distribution of $BIS(G)$. Curiously, we can not determine during the reduction itself the relative weights of the sub-instances $G_1, G_2, \ldots$. Nevertheless, once we have an output to $\overleftarrow{H}(f(G, \epsilon))$, the output itself tells us which H_i is relevant. From this, we can recover an output to $\overleftarrow{H_{S_i}}(G_i)$ and from this we can recover an output to $BIS(G)$. The main technical difficulty lies in showing that the distributions are correct. In particular, since the sub-reductions are SP-reductions (i.e., the equations in Section 3.5 are satisfied in the construction of $G_1, G_2, \ldots$), the combined reduction is also an SP-reduction.

We now describe the details. Let $F_L(H)$ be the set of universal vertices in $V_L(H)$ and let $F_R(H)$ be the set of universal vertices in $V_R(H)$. Let $f_L = |F_L(H)|$

and $f_R = |F_R(H)|$ and $v_L = |V_L(H)|$ and $v_R = |V_R(H)|$. For a subset S of $V_R(H)$, let $N(S)$ be the set of mutual neighbours of S:

$$N(S) = \{v \in V_L(H) \mid \forall u \in S, \{u, v\} \in E(H)\}.$$

Note that $F_L(H) \subseteq N(S) \subseteq V_L(H)$. S is said to be *left-reducing* if $F_L(H) \subset N(S) \subset V_L(H)$. If S is left-reducing, let H_S be the subgraph of H induced by vertex partition $(N(S), V_R(H))$. Note that H_S has fewer vertices than H. Also, it is connected, universal and non-trivial: The set of universal vertices in $V_L(H_S)$ is $F_L(H)$; the set of universal vertices in $V_R(H_S)$ includes all of $F_R(H)$ but it does not equal $V_R(H)$ since $N(S) \supset F_L(H)$.

Similarly, a subset S of $V_L(H)$ is *right-reducing* if $F_R(H) \subset N(S) \subset V_R(H)$. If S is right-reducing, let H_S be the subgraph of H induced by vertex partition $(V_L(H), N(S))$. H_S has fewer vertices than H and is connected, universal and non-trivial.

Now, let $S_1, \ldots, S_k$ be the left-reducing subsets of $V_R(H)$ and let $S_{k+1}, \ldots, S_{k+j}$ be the right-reducing subsets of $V_L(H)$. (Either or both of k and j could be zero.) For every $i \in \{1, \ldots, k+j\}$, let (f_i, g_i) be an SP-reduction from BIS to $\overleftarrow{H_{S_i}}$. Take the input (G, ϵ) to BIS and let $G_i = f_i(G, \epsilon/2)$. Let $a_i = |V_L(G_i)|$ and let $b_i = |V_R(G_i)|$. Let $q = \sum_{i=1}^{k+j}(a_i + b_i)$ and let $n = |V_L(G)| + |V_R(G)|$.

Let $f(G, \epsilon)$ be the graph which is constructed as follows, where a and b will be chosen later to satisfy

$$a \geq 2v_L \lceil q \ln(v_R/f_R) + \ln(16n/\epsilon) \rceil, \tag{4.4}$$

and

$$b \geq 2v_R \lceil q \ln(v_L/f_L) + \ln(16n/\epsilon) \rceil. \tag{4.5}$$

See Figure 4.1. For every vertex u of G, put a size-a set $L[u]$ into $V_L(f(G, \epsilon))$ and a size-b set $R[u]$ into $V_R(f(G, \epsilon))$. Also, add edges $L[u] \times R[u]$ to $E(f(G, \epsilon))$. If $u \in V_L(G)$ is connected to $v \in V_R(G)$ by an edge of G then add edges $R[u] \times L[v]$ to $E(f(G, \epsilon))$.

Also, for every vertex u of G and every $i \in [1, \ldots, k+j]$, let $A_{i,u}$ and $B_{i,u}$ be copies of G_i and let $A'_{i,u}$ and $B'_{i,u}$ be copies of G_i in which left-vertices and right-vertices

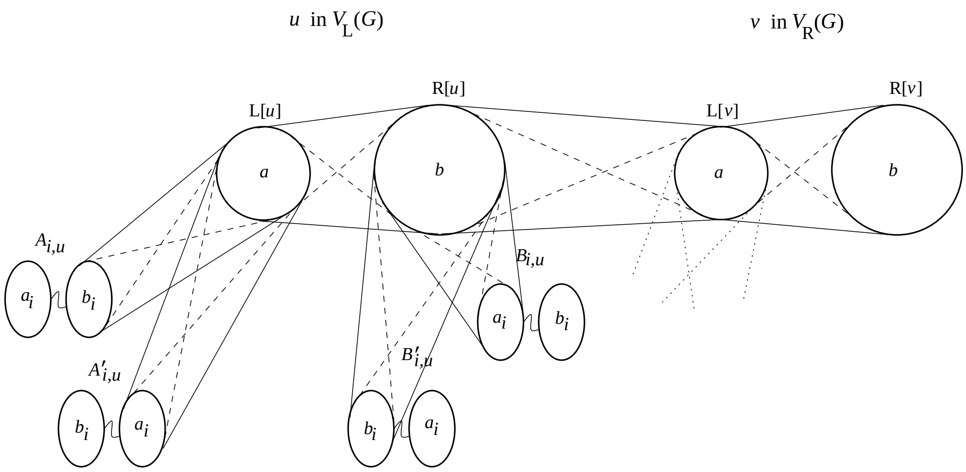


Figure 4.1: *The construction of $f(G, \epsilon)$ in the proof of Lemma 4.3.*

are switched (so the vertices in $V_L(A'_{i,u})$ correspond to the vertices in $V_R(G_i)$ and the vertices in $V_R(A'_{i,u})$ correspond to the vertices in $V_L(G_i)$). Add edges $L[u] \times V_R(A_{i,u})$ and $L[u] \times V_R(A'_{i,u})$ and $R[u] \times V_L(B_{i,u})$ and $R[u] \times V_L(B'_{i,u})$ to $E(f(G, \epsilon))$.

Let

$$V_L(f(G,\epsilon)) = \bigcup_u L[u] \cup \bigcup_{u,i} \{V_L(A_{i,u}) \cup V_L(A'_{i,u}) \cup V_L(B_{i,u}) \cup V_L(B'_{i,u})\}$$

and let Y be the set of $\overleftarrow{H}$-colourings of $f(G, \epsilon)$. We will partition Y into sets $Y_0, \ldots, Y_{k+j+1}$. First we show which elements from Y go into Y_1,...Y_{k+j} and then, a little later, show how the remaining elements from Y are split between Y_0 and Y_{k+j+1}.

For $i \in [1, \ldots, k]$, Y_i is the set of colourings which are not in $Y_1, \ldots, Y_{i-1}$ but in which some $u \in V_L(G)$ has $R[u]$ coloured with (exactly) the colours in S_i. For $i \in [k+1, \ldots, k+j]$, Y_i is the set of colourings which are not in $Y_1, \ldots, Y_{i-1}$ but in which some $v \in V_R(G)$ has $L[v]$ coloured with S_i.

Now note that for any remaining colourings (any colourings in Y_0 or Y_{k+j+1}), every vertex $u \in V_L(G)$ has $R[u]$ coloured with a set S of colours such that $N(S)$ is either $F_L(H)$ or $V_L(H)$. Similarly, every vertex $v \in V_R(G)$ has $L[v]$ coloured with a set S of colours such that $N(S)$ is either $F_R(H)$ or $V_R(H)$.

Consider a colouring y. Vertex $u \in V_L(G)$ satisfies *Condition (A)* if $R[u]$ is coloured with a set S of colours with $N(S) = F_L(H)$ but $S \subset V_R(H)$. It satisfies *Condition (B)* if $R[u]$ is coloured with a set S of colours with $N(S) = V_L(H)$ but $L[u]$

is coloured with a proper subset of $V_L(H)$. Vertex $v \in V_R(G)$ satisfies *Condition (C)* if $L[v]$ coloured with a set S of colours with $N(S) = F_R(H)$ but $S \subset V_L(H)$. It satisfies *Condition (D)* if $L[v]$ is coloured with a set S of colours with $N(S) = V_R(H)$ but $R[v]$ is coloured with a proper subset of $V_R(H)$.

We now define

$$Y_0 = \{y \in Y - \{Y_1 \cup \cdots \cup Y_{k+j}\} \mid \text{some vertex satisfies Condition A or B or C or D}\}.$$

Now note that colourings in Y_{k+j+1} have the following property. Every vertex u of G either has $R[u]$ coloured $V_R(H)$ or has $L[u]$ coloured $V_L(H)$.

We will first work on establishing (3.4). Let $Y_{u,A}$ denote the subset of Y in which u satisfies (A). Define $Y_{u,B}$, $Y_{u,C}$ and $Y_{u,D}$ similarly. We will show that the size of each of $Y_{u,A}$, $Y_{u,B}$, $Y_{u,C}$ and $Y_{u,D}$ is at most $(\epsilon/(16n))|Y|$. (3.4) follows since

$$|Y_0| \leq \sum_{u \in V(G)} |Y_{u,A}| + |Y_{u,B}| + |Y_{u,C}| + |Y_{u,D}|.$$

First, let's show that $|Y_{u,A}| \leq (\epsilon/(16n))|Y|$. Consider the set of colourings in Y in which all neighbours of vertices in $R[u]$ have colours from $F_L(H)$ and let ψ be the number of induced colourings on vertices other than the vertices of $R[u]$. If $\psi = 0$ then $|Y_{u,A}| = 0$, so the claim is trivial. Otherwise, $|Y_{u,A}| \leq \psi(v_R^b - \nu(b, v_R))$ which is at most $\psi v_R^b \exp(-b/(2v_R))$ by Lemma 2.4. On the other hand, $|Y| \geq \psi v_R^b$, so the claim follows from (4.5). The proof that $|Y_{u,C}|$ is sufficiently small is similar.

Next, let's show that $|Y_{u,B}| \leq (\epsilon/(16n))|Y|$. Consider the set of colourings in Y in which $R[u]$ is coloured with a subset of $F_R(H)$ and let ψ be the number of induced colourings on all vertices except those in $L[u]$ and $A_{i,u}$ and $A'_{i,u}$ (for $i \in [1, \ldots, j+k]$). If $\psi = 0$ then $|Y_{u,B}| = 0$, so the claim is trivial. Otherwise, $|Y_{u,B}| \leq \psi(v_L^a - \nu(a, v_L))v_R^q v_L^q$ which is at most $\psi v_L^a \exp(-a/(2v_L))v_R^q v_L^q$ by Lemma 2.4. On the other hand, $|Y| \geq \psi v_L^a f_R^q v_L^q$, so the claim follows from (4.4). The proof that $|Y_{u,D}|$ is sufficiently small is similar.

We will now work on establishing (3.2). First consider $i \in [1, \ldots, k]$. Let $Y_{u,i}$ be the set of colourings in Y_i for which $u \in V_L(G)$ is the first vertex in $V_L(G)$ with $R[u]$ coloured S_i. Let Γ be the set of induced colourings on $B_{i,u}$. Note that Γ is the set of $\overleftarrow{H_{S_i}}$-colourings of $G_i = f_i(G, \epsilon/2)$. Also, each colouring in Γ comes up ψ times in $Y_{u,i}$ for some ψ. (In particular, ψ is the number of colourings of vertices other than $B_{i,u}$ which are induced by colourings in $Y_{u,i}$.) For colouring $y \in Y_{u,i}$ we will let $g(G, \epsilon, y) = g_i(G, \epsilon/2, y')$ where y' is the induced colouring on $B_{i,u}$. Then for every independent set x in the set $IS(G)$ of independent sets of G,

$$|\{y \in Y_{u,i} \mid g(G, \epsilon, y) = x\}| = \psi\, |\{y' \in \Gamma \mid g_i(G, \epsilon/2, y') = x\}|. \tag{4.6}$$

Since (f_i, g_i) is an SP-reduction, (3.1) gives

$$e^{-\epsilon/2}\frac{|\Gamma|}{|IS(G)|} \le |\{y' \in \Gamma \mid g_i(G_i, \epsilon/2, y') = x\}| \le e^{\epsilon/2}\frac{|\Gamma|}{|IS(G)|} \tag{4.7}$$

and (3.2) follows for $Y_{u,i}$ from (4.6) and (4.7) since $|Y_{u,i}| = \psi\,|\Gamma|$. Colourings in $Y_{k+1}, \ldots, Y_{k+j}$ are handled similarly except that we look at induced colourings of $A_{i,u}$ rather than $B_{i,u}$.

It remains to satisfy (3.2) for $i = k + j + 1$. Note that any colouring y in Y_{k+j+1} points out an independent set of G. A vertex $u \in V_L(G)$ is in the independent set if $R[u]$ is coloured $V_R(H)$. A vertex $v \in V_R(G)$ is in the independent set if $L[u]$ is coloured $V_L(H)$. We will define $g(G, \epsilon, y)$ to be this independent set. Let us focus attention on a given independent set containing w_L vertices in $V_L(G)$ and w_R vertices in $V_R(G)$. We will now calculate how many colourings in Y_{k+j+1} correspond to this independent set.

For any bipartite graph G' with vertex partition $(V_L(G'), V_R(G'))$, let (in the usual fashion) $\overleftarrow{\#H}(G')$ denote the number of $\overleftarrow{H}$-colourings of G'. Then the number of times that this independent set comes up as a colouring in Y_{k+j+1} is the product of the following two quantities.

$$\left(\nu(b, v_R) f_L^a \prod_{i=1}^{k+j} \overleftarrow{\#H}(A_{i,u})\overleftarrow{\#H}(A'_{i,u}) f_L^{a_i+b_i} v_R^{a_i+b_i}\right)^{w_L+v_R-w_R},$$

$$\left(f_R^b \nu(a, v_L) \prod_{i=1}^{k+j} \overleftarrow{\#H}(B_{i,u})\overleftarrow{\#H}(B'_{i,u}) v_L^{a_i+b_i} f_R^{a_i+b_i}\right)^{v_L-w_L+w_R}.$$

Now note that $\overleftarrow{\#H}(A_{i,u}) = \overleftarrow{\#H}(B_{i,u})$ and $\overleftarrow{\#H}(A'_{i,u}) = \overleftarrow{\#H}(B'_{i,u})$. So if we let

$$Z = \left(\prod_{i=1}^{k+j} \overleftarrow{\#H}(A_{i,u})\overleftarrow{\#H}(A'_{i,u})\right)^{v_L+v_R} \left(f_L^a \nu(b, v_R) f_L^q v_R^q\right)^{v_R} \left(f_R^b \nu(a, v_L) v_L^q f_R^q\right)^{v_L},$$

the contribution of the independent set becomes

$$Z\left(\nu(b, v_R) f_L^a f_L^q v_R^q\right)^{w_L - w_R} \left(f_R^b \nu(a, v_L) v_L^q f_R^q\right)^{w_R - w_L},$$

which is

$$Z\left(\frac{\nu(b, v_R) v_L^a}{v_R^b \nu(a, v_L)}\right)^{w_L - w_R} \left(\left(\frac{v_R}{f_R}\right)^{b+q} \left(\frac{f_L}{v_L}\right)^{a+q}\right)^{w_L - w_R}.$$

To get (3.5) we will show that a and b can be chosen so that

$$e^{-\epsilon/(8n)} \leq \left(\frac{\nu(b, v_R) v_L^a}{v_R^b \nu(a, v_L)}\right) \leq e^{\epsilon/(8n)}, \text{ and} \tag{4.8}$$

and

$$e^{-\epsilon/(8n)} \leq \left(\frac{v_R}{f_R}\right)^{b+q} \left(\frac{f_L}{v_L}\right)^{a+q} \leq e^{\epsilon/(8n)}. \tag{4.9}$$

This guarantees that the contribution of this independent set is in the range $[e^{-\epsilon/4}Z, e^{\epsilon/4}Z]$, and (3.5) follows for Y_{k+j+1}. To establish (4.8), use Lemma 2.4 to observe that

$$\left(\frac{\nu(b, v_R) v_L^a}{v_R^b \nu(a, v_L)}\right) \leq \frac{1}{1 - \exp(-a/(2v_L))}.$$

Since (4.4) gives $1 - \exp(-a/(2v_L)) \geq 1 - \epsilon/(16n) \geq e^{-\epsilon/(8n)}$, the right-hand inequality of (4.8) follows. The left-hand inequality is similar.

We will now show how to choose the values of a and b to satisfy (4.9). If $v_R/f_R = v_L/f_L$ then simply choose $a = b$ and make them large enough to satisfy (4.4) and (4.5). Suppose that $v_R/f_R < v_L/f_L$. Then use Lemma 4.2 with $c_1 = v_R/f_R$, $c_2 = v_L/f_L$, $\delta = \epsilon/(8n)$, and

$$a_0 = 2v_L \left\lceil q \ln(v_R/f_R) + \ln(16n/\epsilon)\right\rceil + 2v_R \left\lceil q \ln(v_L/f_L) + \ln(16n/\epsilon)\right\rceil.$$

The lemma gives values of a and b which are in $O((a_0 + q)/\delta)$, which is not too large. Thus, our reduction is sampling-preserving. Note that the reduction can be done in polynomial time — the calculation of a and b does not involve computing Z. The case where $v_L/f_L < v_R/f_R$ is similar. $\square$

4.5 The proof of Theorem 4.1

Recall from Section 2.2.1 (page 54) the process of "bipartisation" i.e. the construction of a bipartite graph $bi(H)$ from a potentially non-bipartite graph H. This construction, used earlier in [10], is pivotal in completing the proof of Theorem 4.1. Here we recap on how the graph $bi(H)$ is constructed from H; note that the construction is valid even if H is already bipartite. (If H is already bipartite, $bi(H)$ comprises two disjoint copies of H.)

Let the vertices of H be $c_0, \ldots, c_h$, where $h = |V(H)| - 1$. The vertex set of $bi(H)$ consists of $\{d_0, \ldots, d_h\}$ on one side of the bipartition and $\{d'_0, \ldots, d'_h\}$ on the other. The edge set of $bi(H)$ is:

$$\bigcup_{\{c_i,c_j\} \in E(H)} \Big\{ \{d_i, d'_j\}, \{d_j, d'_i\} \Big\}$$

Thus, a loop $\{c_i, c_i\}$ in H corresponds to the edge $\{d_i, d'_i\}$ in $bi(H)$ and a non-loop $\{c_i, c_j\}$ in H (for which $i \neq j$) corresponds to two edges $\{d_i, d'_j\}$ and $\{d_j, d'_i\}$ in $bi(H)$. For every edge $\{d_i, d'_j\}$ of $bi(H)$, let

$$V_L(H_{i,j}) = \left\{ d_\ell \,\middle|\, \{d_\ell, d'_j\} \in E(bi(H)) \right\}$$

and

$$V_R(H_{i,j}) = \left\{ d'_\ell \,\middle|\, \{d_i, d'_\ell\} \in E(bi(H)) \right\}$$

and let $H_{i,j}$ be the subgraph of $bi(H)$ induced by vertex set $V_L(H_{i,j}) \cup V_R(H_{i,j})$. Note that $d_i \in V_L(H_{i,j})$ and $d'_j \in V_R(H_{i,j})$ and d_i is adjacent to all of $V_R(H_{i,j})$ in $H_{i,j}$ and d'_j is adjacent to all of $V_L(H_{i,j})$. Thus, $H_{i,j}$ is connected and universal. Let $\Delta_1(H)$ be the degree of H. That is,

$$\Delta_1(H) = \max\{\deg(c_s) \mid c_s \in V(H)\}.$$

Similarly, let $\Delta_2(H)$ be the maximum degree amongst neighbours of vertices with degree $\Delta_1(H)$:

$$\Delta_2(H) = \max\left\{\deg(c_t)\middle| \text{ for some } c_s \in V(H) \text{ with } \deg(c_s) = \Delta_1(H),\ \{c_s, c_t\} \in E(H)\right\}.$$

Let

$$R(H) = \left\{(c_i, c_j)\middle|\{c_i, c_j\} \in E(H) \text{ and } \deg(c_i) = \Delta_1(H) \text{ and } \deg(c_j) = \Delta_2(H)\right\}.$$

It is important to note that $R(H)$ consists of *ordered* pairs, so if (say) c_1 and c_2 are adjacent and both have degree $\Delta_1(H)$, then both the pairs (c_1, c_2) and (c_2, c_1) will be in $R(H)$.

We will start with the following lemma.

Lemma 4.4 *Let H be any graph with no trivial components. Then $R(H)$ is non-empty and $\Delta_1(H) > 1$ and $\Delta_2(H) > 1$. Also, for all $(c_i, c_j) \in R(H)$, $H_{i,j}$ is connected, bipartite, universal and non-trivial.*

Proof. Since H has no trivial components, $R(H)$ is non-empty and $\Delta_1(H) > 1$ and $\Delta_2(H) > 1$. Suppose $(c_i, c_j) \in R(H)$. Recall that $H_{i,j}$ is connected, bipartite and universal. Suppose for contradiction that $H_{i,j}$ is a complete bipartite graph (so vertices in $V_L(H_{i,j})$ have degree $\Delta_1(H)$ in $H_{i,j}$ and vertices in $V_R(H_{i,j})$ have degree $\Delta_2(H)$ in $H_{i,j}$).

This assumption guarantees that $H_{i,j}$ is a connected component of $bi(H)$: $bi(H)$ cannot have an edge with exactly one endpoint in $V_L(H_{i,j})$ — the endpoint would then have degree exceeding $\Delta_1(H)$ in $bi(H)$, which is a contradiction; similarly, $bi(H)$ cannot have an edge with exactly one endpoint in $V_R(H_{i,j})$.

Thus, for any $d_\ell \in V_L(H_{i,j})$,

$$\{c_r \mid \{c_\ell, c_r\} \in E(H)\} = \{c_r \mid d'_r \in V_R(H_{i,j})\}. \tag{4.10}$$

Similarly, for any $d'_\ell \in V_R(H_{i,j})$,

$$\{c_r \mid \{c_\ell, c_r\} \in E(H)\} = \{c_r \mid d_r \in V_L(H_{i,j})\}. \tag{4.11}$$

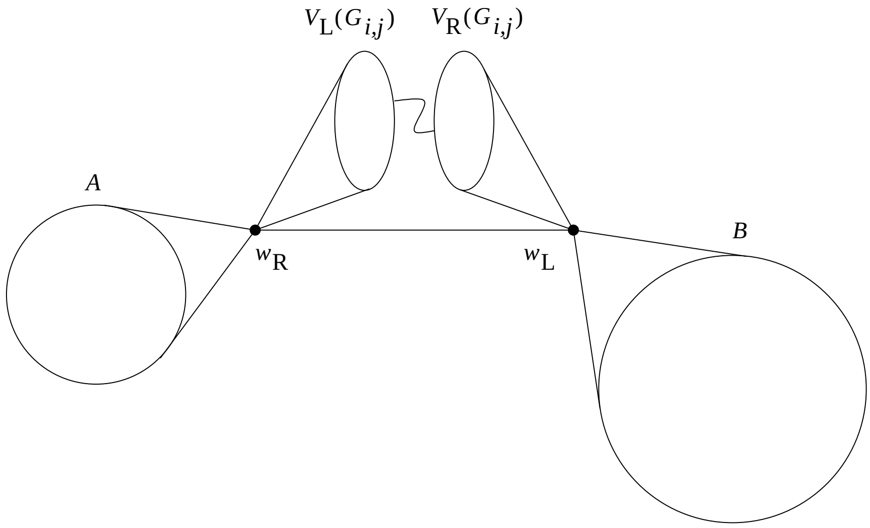


Figure 4.2: *The construction of $f(G,\epsilon)$ in the proof of Lemma 4.5.*

Now if H has a vertex c_ℓ such that $\{c_i, c_\ell\} \in E(H)$ and $\{c_j, c_\ell\} \in E(H)$ then $d_\ell \in V_L(H_{i,j})$ and $d'_\ell \in V_R(H_{i,j})$ so (4.10) and (4.11) imply that

$$\{c_r \mid d'_r \in V_R(H_{i,j})\} = \{c_r \mid d_r \in V_L(H_{i,j})\}.$$

Thus, $H_{i,j}$ corresponds to a component of H and that component is a looped clique, which contradicts the fact that H has no trivial component.

On the other hand, if there is no c_ℓ with $\{c_i, c_\ell\} \in E(H)$ and $\{c_j, c_\ell\} \in E(H)$ then $H_{i,j}$ corresponds to a connected component of H which is a complete bipartite graph, again giving a contradiction. □

We can now prove the main lemma.

Lemma 4.5 *Suppose that H is a graph with no trivial components. Then $BIS \leq_{SP} biH$.*

Proof. Let (G, ϵ) be an input to BIS. For each $(c_i, c_j) \in R(H)$, Lemma 4.4 and Lemma 4.3 guarantee that there is a sampling-preserving reduction $(f_{i,j}, g_{i,j})$ from BIS to $\overleftarrow{H_{i,j}}$. Let $G_{i,j} = f_{i,j}(G, \epsilon/2)$. Let $f(G, \epsilon)$ be the graph which is constructed as follows. See Figure 4.2. Let $q = \sum_{(c_i,c_j)\in R(H)} |V_L(G_{i,j})| + |V_R(G_{i,j})|$. Let

$$V_L(f(G,\epsilon)) = A \cup \{w_L\} \cup \bigcup_{(c_i,c_j)\in R(H)} V_L(G_{i,j}),$$

and

$$V_R(f(G,\epsilon)) = B \cup \{w_R\} \cup \bigcup_{(c_i,c_j)\in R(H)} V_R(G_{i,j}),$$

where A and B are sets of vertices with

$$|A| = \left\lceil \frac{q\ln(|V(H)|) + \ln(8|E(H)|/\epsilon)}{\ln(\Delta_2(H)/(\Delta_2(H)-1))} \right\rceil$$

and

$$|B| = \left\lceil \frac{(q+|A|+1)\ln(|V(H)|) + \ln(8|V(H)|/\epsilon)}{\ln(\Delta_1(H)/(\Delta_1(H)-1))} \right\rceil.$$

Note that there is no division by zero, since $\Delta_1(H)$ and $\Delta_2(H)$ are bigger than one (by Lemma 4.4). In addition to the edges in the graphs $G_{i,j}$, we add edge (w_L, w_R) and $w_L \times B$ and $w_R \times A$ and, for all $(c_i, c_j) \in R(H)$, we add edges $w_L \times V_R(G_{i,j})$ and $w_R \times V_L(G_{i,j})$.

Let Y be the set of H-colourings of $f(G,\epsilon)$. Y_0 will be the set of colourings in Y in which the pair (w_L, w_R) is not coloured with a pair (c_i, c_j) from $R(H)$. We will now establish (3.4). For every $c_i \in V(H)$ with $\deg(v) < \Delta_1(H)$ let $Y_0(v)$ be the set of colourings in Y in which w_L is coloured c_i. Now

$$|Y_0(v)| \leq (\Delta_1(H)-1)^{|B|}|V(H)|^{q+|A|+1}.$$

Now consider any $(c_i, c_j) \in R(H)$. There are at least $\Delta_2(H)^{|A|}\Delta_1(H)^{|B|}$ colourings of $f(G,\epsilon)$ with (w_L, w_R) coloured (c_i, c_j) (for example, the colourings in which all of the vertices of the graphs $G_{i,j}$ are coloured with either c_i or c_j). Thus, $|Y| \geq \Delta_2(H)^{|A|}\Delta_1(H)^{|B|} \geq \Delta_1(H)^{|B|}$. We conclude that

$$|Y_0(v)| \leq (\epsilon/(8|V(H)|))|Y|. \tag{4.12}$$

Now consider any ordered pair $(c_i, c_k) \in V(H) \times V(H)$ such that $\{c_i, c_k\} \in E(H)$ and $\deg(c_i) = \Delta_1(H)$ but $\deg(c_k) < \Delta_2(H)$. Let $Y_0(v_i, v_k)$ be the set of colourings in Y in which (w_L, w_R) is coloured (c_i, c_k). Now

$$|Y_0(c_i, c_k)| \leq \Delta_1(H)^{|B|}(\Delta_2(H)-1)^{|A|}|V(H)|^q.$$

Also, from before $|Y| \geq \Delta_2(H)^{|A|}\Delta_1(H)^{|B|}$ so

$$|Y_0(c_i, c_k)| \leq (\epsilon/(8|E(H)|))|Y|. \tag{4.13}$$

(4.12) and (4.13) imply (3.4) since $|Y_0| \leq \sum_{c\in V(H)} Y_0(c) + \sum_{(c_i,c_k)} |Y_0(c_i,c_k)|$.

For a pair $(c_i,c_j) \in R(H)$, let $Y_{i,j}$ be the set of colourings of $f(G,\epsilon)$ with the pair (w_L,w_R) coloured (c_i,c_j). Let Γ be the set of induced colourings on $G_{i,j}$. Note that Γ is the set of $\overleftarrow{H_{i,j}}$-colourings of $G_{i,j}$. Also, each colouring in Γ comes up ψ times in $Y_{i,j}$ where ψ is the number of induced colourings on the vertices other than $G_{i,j}$. For a colouring $y \in Y_{i,j}$ we will set $g(G,\epsilon,y) = g_{i,j}(G,\epsilon/2,y')$ where y' is the induced colouring on $G_{i,j}$. Then (3.2) follows from the fact that $(f_{i,j},g_{i,j})$ is an SP-reduction. □

Theorem 4.1 follows from Lemma 4.5 and from Lemmas 3.1 (from Chapter 3) and 4.6 below.

Lemma 3.1 *If $X \leq_{\mathsf{SP}} Y$ and Y has a $PAUS$, X has a $PAUS$.*

Proof. Proved on page 110. □

Recall the following definitions. A *randomised approximation scheme* (RAS) for a counting problem F is a randomised algorithm that takes input σ and accuracy parameter $\epsilon \in (0,1)$ and produces as output an integer random variable Y satisfying the condition $\Pr(e^{-\epsilon}F(\sigma) \leq Y \leq e^{\epsilon}F(\sigma)) \geq 3/4$. It is a "fully polynomial" randomised approximation scheme ($FPRAS$) if it runs in time $\mathrm{poly}(|\sigma|,\epsilon^{-1})$. The problem $\#BIS$ is self-reducible so the following lemma follows from [16].

Lemma 4.6 *(JVV) If BIS has a $PAUS$ then $\#BIS$ has an $FPRAS$.*

Proof. The lemma is a special case of Theorem 6.4 of [16]. In order to apply Theorem 6.4 directly we would need to define "self-reducible" formally and to deal with some easy (though annoying) issues:

(i) Inputs to $\#BIS$ may be disconnected but inputs to BIS may not. (Remember our comment from Section 4.2 that, to ensure consistency with the original paper from which this result is taken, we have allowed the input to $\#BIS$ to be disconnected.)

(ii) In order to apply Theorem 6.4 we technically need a *fully* polynomial almost uniform sampler ($FPAUS$) for BIS[2]. This can be obtained from a $PAUS$ as [16] explains.

Rather than dealing with these issues, we prefer to simply provide a proof for the lemma. The details given here are from the proof of Proposition 3.4 of [20]. Technically, Jerrum's proof from [20] is about counting *matchings* but the few changes that are needed to yield our lemma are completely routine.

Let (G, ϵ) be an input to $\#BIS$. Suppose that G has components $G_1, \ldots, G_k$. For each i, let the two parts of the vertex set be $V_L(G_i)$ and $V_R(G_i)$ and let the sizes of these parts be ℓ_i and r_i, respectively. Let $N_i = \ell_i r_i$ and let $E(G_i) = \{e_i(1), \ldots, e_i(m_i)\}$. Denote the non-edges of G_i by $\{e_i(m_i+1), \ldots, e_i(N_i)\}$. For $j \in \{1, \ldots, N_i\}$, let $G_i(j)$ be the graph $(V(G_i), \{e_i(1), \ldots, e_i(j)\})$. For any graph G', let $IS(G')$ denote the set of independent sets of G'. Let

$$\rho_i(j) = \frac{|IS(G_i(j+1))|}{|IS(G_i(j))|}.$$

Note that

$$|IS(G_i)| = (\rho_i(m_i)\rho_i(m_i+1)\cdots\rho_i(N_i-1))^{-1}|IS(G_i(N_i))|.$$

Also, the number of independent sets of the complete bipartite graph $G_i(N_i)$ is $2^{\ell_i} + 2^{r_i} - 1$, so

$$|IS(G_i)| = (2^{\ell_i} + 2^{r_i} - 1) \prod_{j=m_i}^{N_i-1} \rho_i(j)^{-1}. \tag{4.14}$$

Furthermore,

$$|IS(G)| = \prod_{i=1}^{k} |IS(G_i)| = \prod_{i=1}^{k} (2^{\ell_i} + 2^{r_i} - 1) \prod_{j=m_i}^{N_i-1} \rho_i(j)^{-1}. \tag{4.15}$$

Now let $z = \sum_{i=1}^{k}(N_i - m_i)$. In order to estimate $|IS(G)|$, we need to estimate the z ratios $\rho_i(j)$.

For each ratio $\rho_i(j)$ we can make some observations.

(i) $\rho_i(j) \le 1$, since $IS(G_i(j+1)) \subseteq IS(G_i(j))$

(ii) $\rho_i(j) \ge 1/2$, since $IS(G_i(j)) \backslash IS(G_i(j+1))$ can be mapped injectively into $IS(G_i(j+1))$ by removing the "lexicographically" least[3] endpoint of $e_i(j+1)$.

[2]Recall that an $FPAUS$ is simply an $FPAS$ that is explicitly dedicated to sampling from the uniform distribution.

[3]Any natural, arbitrary ordering is of course suitable here

(iii) Let $\mathcal{A}$ be a $PAUS$ for BIS. For $i \in [1, \ldots, k]$ and $j \in [m_i, \ldots, N_i - 1]$, let $Z_i(j)$ be the indicator variable for the event that, when we run $\mathcal{A}$ with input $G_i(j)$ and accuracy parameter δ, the output is an independent set of $G_i(j+1)$. Note that $\rho_i(j) - \delta \leq E[Z_i(j)] \leq \rho_i(j) + \delta$. This follows immediately from the definition of $PAUS$, but it is important to note that the input to $\mathcal{A}$, $G_i(j)$, is connected[4] (since all inputs to BIS must be connected).

Let $\overline{Z_i(j)}$ be the result obtained by calling $\mathcal{A}$ $\lceil 74\epsilon^{-2}z \rceil$ times with input $G_i(j)$ and accuracy parameter $\delta = \epsilon/(6z)$ and averaging the value of $Z_i(j)$ which occurs each time. Jerrum shows in his proof that with probability at least $3/4$,

$$e^{-\epsilon} \prod_{i=1}^{k} \prod_{j=m_i}^{N_i-1} \rho_i(j) \leq \prod_{i=1}^{k} \prod_{j=m_i}^{N_i-1} \overline{Z_i(j)} \leq e^{\epsilon} \prod_{i=1}^{k} \prod_{j=m_i}^{N_i-1} \rho_i(j).$$

Thus, the quantity

$$\prod_{i=1}^{k} (2^{\ell_i} + 2^{r_i} - 1) \prod_{j=m_i}^{N_i-1} \overline{Z_i(j)}^{-1}$$

is a sufficiently accurate estimate of $|IS(G)|$.

For each of the z pairs (i,j), $O(\epsilon^{-2}z)$ samples were needed, each of which is produced in time $\mathrm{poly}(|G|, z/\epsilon)$. Since $z \leq |V(G)|^2$, the total running time is $\mathrm{poly}(|G|, \epsilon^{-1})$ and thus we have an $FPRAS$. $\square$

4.6 The presence of trivial components

Theorem 4.1 shows that sampling H-colourings is difficult (in a complexity-theoretic sense) if every component of H is non-trivial. Recall from [10] that exactly counting H-colourings is $\#P$-complete if H has even one non-trivial component. Thus, it might appear that Theorem 4.1 can be improved. In this section, we show that the existence of a single non-trivial component is not enough to make sampling difficult. In particular, we give an example of a graph H with a non-trivial component, for which H has a $PAUS$. Specifically, let H be the graph depicted in Figure 4.3.

Observation 4.7 *Suppose that H is the graph depicted in Figure 4.3. H has a $PAUS$.*

[4] This is because G_i is connected, and (for $j \geq m_i$) the edgeset of G_i is a subset of the edgeset of $G_i(j)$.

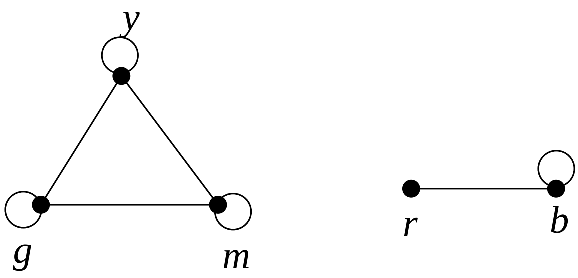


Figure 4.3: *An H with a non-trivial component for which H has a $PAUS$.*

Proof. Here is a $PAUS$[5] for H. The input is an instance (G, ϵ) where G has n vertices and, without loss of generality[6], is connected. If $\epsilon < 2^n/(2^n + 3^n)$ then the algorithm simply runs for 5^n steps, constructs all of the H-colourings of G (and counts them) and selects one uniformly at random. Note that the running time is at most poly$(1/\epsilon)$ in this case. Otherwise, the algorithm chooses i uniformly at random from $1, \ldots, 3^n + 2^n$. If $i \leq 3^n$, then the algorithm outputs the i'th colouring from the 3^n colourings with colours "g", "y", and "m". Otherwise, let C be the $(i - 3^n)$th of the 2^n (proper and improper) colourings with colours "r" and "b". If C is a legal H-colouring of G, then the algorithm outputs it. Otherwise, it outputs the error symbol $\perp$. The variation distance between the output distribution of the algorithm and the uniform distribution on H-colourings of G is at most the probability that the algorithm outputs $\perp$, which is at most $2^n/(2^n + 3^n) \leq \epsilon$. □

Later, in our discussion of disconnected H (i.e. Chapter 6), we generalise the above observation and prove, in Lemmas 6.1 and 6.2, that the complexity of a disconnected H is closely related to the complexity of its exponentially dominant components. Finally, it is also worth observing that, though this graph verifies that there exist non-trivial H which permit an $FPRAS/PAUS$, it is still not completely clear whether there exists a

[5]Note that it would actually not be difficult to build an $FPAUS$ for this graph. We do not go into details, but the basic idea would be to use a simple Monte Carlo sampling algorithm in the same style as Jerrum's observation in Chapter 6. The main difference would be that we would use a probability of $3^n/(2^n + 3^n)$ in step (a) of that algorithm. This $FPAUS$ could then be used to build an $FPRAS$ using the result from [12]. Alternatively an $FPRAS$ could be built for this graph by using Lemma 6.1 of Chapter 6.

[6]We can assume that the input is a connected graph without losing generality because we can obtain an H-colouring of a k-component graph G by independently calling our $PAUS$ for each component, specifying accuracy parameter ϵ/k. The final variation distance (between the output distribution and the uniform distribution on H-colourings of G) is at most ϵ.

non-trivial *connected* H with such a property, although (as Theorem 4.1 shows) there is now some complexity-theoretic evidence against this outcome. (Albeit predicated on the as-yet unknown absolute complexity of $\#BIS$.)

Chapter 5

The class $\equiv_{\mathsf{AP}}\#SAT$

5.1 Introduction

The significance of the complexity class $\equiv_{\mathsf{AP}}\#SAT$ has already been discussed on a number of occasions throughout this thesis. As the "hardest" approximation complexity class for problems in $\#P$ it is natural to try and determine which (connected) H-colouring counting problems are AP-interreducible with $\#SAT$, and - just as importantly - why. Chapter 2 demonstrated an ad-hoc approach to graph classification and, even though the classification of connected 4-vertex H is not yet complete, showed that a large majority of connected 4-vertex H are $\equiv_{\mathsf{AP}}\#SAT$. In this chapter we graduate from ad-hoc $\equiv_{\mathsf{AP}}\#SAT$-hardness results to more generalized lemmas and observations. In addition to providing a formal suite of $\equiv_{\mathsf{AP}}\#SAT$-hardness lemmas and corollaries, we use this chapter to discuss some of the more qualitative and apparent characteristics of H graphs interreducible with $\equiv_{\mathsf{AP}}\#SAT$, which (as we see in due course) takes us into a discussion of how $\equiv_{\mathsf{AP}}\#SAT$ relates to bipartite H and also how it relates to the variant of H-colouring known as "partial" H-colouring.

To be more specific, approximately half of this chapter - Sections 5.2 through 5.5 - is dedicated to the presentation and proof of five $\equiv_{\mathsf{AP}}\#SAT$-hardness / $\equiv_{\mathsf{SP}}SAT$-hardness lemmas (and, where relevant, corollaries of those lemmas.) Section 5.2 (*Two "symmetrical" $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions*) describes Lemmas 5.1 and 5.4 which demonstrate $\equiv_{\mathsf{AP}}\#SAT$-hardness via reductions from $\#IS$ and $\#LargeCut$ respectively. Both reductions exploit gadgetry and characteristics of the graph H to encode two equally maximal, "symmetrical" states that can then be used to represent vertices being IN or OUT of an independent set, or on the $LEFT$ or $RIGHT$ side of a large cut, respectively. Both these lemmas yield results with respect to AP-reducibility (as opposed to SP-reducibility). In Section 5.3 (*H with at least one universal loop, but no other loops besides, are $\equiv_{SP}SAT$*) we introduce our third lemma, Lemma 5.5, which is probably the most elegant and general result in this chapter. The lemma shows that for H with at least one universal loop, but no other loops, $H\equiv_{\mathsf{SP}}SAT$. (There is no known counting version of this result.) In Section 5.4 (*Grabbing looped cliques*) we present Lemma 5.7 which shows (using a new piece of gadgetry called a $MaxCliqueGrab$) how to exploit the situation where maximum-size looped cliques point out non-trivial subgraphs to show that $SAT\leq_{\mathsf{SP}}H$ i.e. hard in the sampling sense. (Corollary 5.8 shows that under restricted circumstances such graphs are also $\equiv_{\mathsf{AP}}\#SAT$. Recall from Section 3.10 the discussion that AP-reductions are preferable but where SP-reductions are more general and powerful it is nonetheless helpful to demonstrate an AP-reduction corollary even if it works under more restricted circumstances.) Section 5.5 (*Grabbing loops*) introduces a further piece of new gadgetry ($MaxLoopGrab$) and in Lemma 5.10 uses it to show that, where all the loops in H are "isolated", $H\equiv_{\mathsf{SP}}SAT$. (Corollary 5.11 is the restricted, AP-reducibility version of this result which in a different sense is slightly more general than Lemma 5.10.)

In the second half of the chapter, we shift from rigorous proofs (required to formalise the above lemmas) to a slightly more "conversational" style, in which we mix observations and smaller-scale results with open questions and conjectures. Section 5.6 (*Bipartite H and $\equiv_{AP}\#SAT$*) briefly considers why we have struggled to find a connected bipartite H which is $\equiv_{\mathsf{AP}}\#SAT$[1], and articulates our suspicion that this is because no such graph exists. Finally, in Section 5.7 (*Partial H-colouring*) we bring the chapter to a close by looking at the complexity of approximate "partial" H-colouring. Though

[1] Or, indeed, $\equiv_{\mathsf{SP}}$ SAT-hard

partial H-colouring is not exclusively restricted to $\equiv_{\mathsf{AP}}\#SAT$ H-colouring problems, we have included this section in this chapter because studying how partial H-colouring ties in with the five $\equiv_{\mathsf{AP}}\#SAT$-hardness lemmas in the first half of this chapter yields some interesting extensions to those lemmas.

5.1.1 Notational recap

Before continuing with this chapter, it is useful to set out some frequently used definitions.

In Chapter 2 we introduced the definition $GoodPairs(H)$, which we now repeat here.

$$GoodPairs(H) = \left\{(S,T)\middle|\emptyset \subset S,T \subseteq V(H) \wedge S = H[T] \wedge T = H[S]\right\}$$

Strictly speaking, the $H[.]$ operator defines a graph rather than set of colours, but it is convenient to relax the notation so that expressions such as $S = H[T]$ have the expected meaning i.e. "S is the maximal set of colours mutually adjacent to T." A frequently encountered extension of $GoodPairs(H)$ is $MaxPairs(H)$, which we define as follows.

$$mp(H) = max\left(|S||T|\middle|(S,T) \in GoodPairs(H)\right)$$
$$MaxPairs(H) = \left\{(S,T)\middle|(S,T) \in GoodPairs(H) \wedge |S||T| = mp(H)\right\}$$

Recall that, in the context of non-bipartite H, a colour $c \in V(H)$ is *universal* iff $adj(c) = V(H)$. As a convention, we let $F(H)$ be the set of universal colours in H.

Finally, let $Loops(H) = \{c \in V(H)|\text{c has a loop}\}$. Furthermore, if Δ_l is the maximum degree of colours in $Loops(H)$, we let:

$$MaxLoops(H) = \{c \in Loops(H)|deg(c) = \Delta_l\}$$

5.2 Two "symmetrical" $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions

Lemma 5.1 *(The "$(F(H),V(H)) - (V(H),F(H))$ dominance" lemma.) Let H be a graph where $F(H) \neq \emptyset$ and $F(H) \subset V(H)$. If*

$$MaxPairs(H) = \{(F(H),V(H)),(V(H),F(H)\}$$

then $\#SAT \leq_{\mathsf{AP}} \#H$.

Proof. We demonstrate that $\#SAT \leq_{\mathsf{AP}} \#H$ by showing that $\#IS \leq_{\mathsf{AP}} \#H$. First, note that H is non-bipartite (because $F(H) \neq \emptyset$) and non-trivial; non-triviality follows because, given that H is non-bipartite, it can only be trivial if it is a fully looped clique, and $F(H) \subset V(H)$ precludes this. Now, let G be an input to $\#IS$; we construct G' as follows. For each vertex $u \in V(G)$, introduce disjoint sets $L[u]$ and $R[u]$ (each of size p, to be determined), and connect every vertex in $L[u]$ to every vertex in $R[u]$. For each edge $\{u,v\} \in E(G)$, connect every vertex in $L[u]$ to every vertex in $L[v]$. (See Figure 5.1.) Now, we let full colourings of G' be those where, for every vertex $u \in V(G)$, $(L[u],R[u])$ is coloured either $(F(H),V(H))$ or $(V(H),F(H))$. We claim that every full colouring corresponds to a specific member of $IS(G)$, and in particular that every independent set in G comes up $(\nu(p,|F(H)|)\nu(p,|V(H)|))^n$ times as a full colouring of G'. To see this, first note that an $(L[.],R[.])$ pair can always be coloured $(F(H),V(H))$ irrespective of whether its neighbour pairs in G' are coloured $(F(H),V(H))$ or $(V(H),F(H))$. This is because, by definition, $F(H)$ can be adjacent to any subset of $V(H)$. On the other hand, a pair coloured $(V(H),F(H))$ can only be adjacent to pairs coloured $(F(H),V(H))$; observe that the only conditions under which $V(H)$ can be adjacent to $V(H)$ is when H is a looped clique, and this contradicts our assertion that $F(H) \subset V(H)$. Hence, a vertex $u \in V(G)$ is IN the independent set if $(L[u],R[u])$ is coloured $(V(H),F(H))$, and OUT if $(L[u],R[u])$ is coloured $(F(H),V(H))$. Given that $(F(H),V(H))$ and $(V(H),F(H))$ pairs both come up the same number of times, we reason that dividing the result of our approximation to $\#H(G')$ by $(\nu(p,|F(H)|)\nu(p,|V(H)|))^n$ and rounding gives a good approximation to $\#IS(G)$. It remains to show that full colourings dwarf all other possible colourings.

The contribution of full colourings is at least $(\nu(p,|F(H)|)\nu(p,|V(H)|))^n$ because as a minimum the colouring where every pair is coloured $(F(H),V(H))$ (i.e. every vertex is OUT of the independent set) is guaranteed to be valid. Now, let $m = mp(H)$. (We know that $m > 1$ because $F(H) \subset V(H)$.) An upper bound on the number of non-full colourings is $(2^{|V(H)|}2^{|V(H)|})^n m^{p(n-1)}(m-1)^p$, which we derive by assuming that all configurations on the n $(L[.], R[.])$ pairs are possible and come up as many times as possible. We know (from Corollary 2.5) that $(1/2)b^a \leq \nu(a,b)$ so it follows that $(1/4)^n m^{pn} \leq (\nu(p,|F(H)|)\nu(p,|V(H)|))^n$. Hence, we need to choose p such that

$$\frac{4^{|V(H)|n} m^{p(n-1)}(m-1)^p}{(1/4)^n m^{pn}} \leq 1/4$$

We re-arrange this to give:

$$4^{(|V(H)|+1)n}((m-1)/m)^p \leq 1/4$$

Clearly, choosing $p = n^2$ is adequate for this purpose, for n above a particular constant threshold. □

Note that (following the comments in Section 3.8.1 of Chapter 3) the above reduction can easily be mapped to an SP-reduction, yielding $SAT \leq_{\mathsf{SP}} H$ rather than $\#SAT \leq_{\mathsf{SP}} \#H$ in the text of Lemma 5.1. This fact will come in use later, in the proof of Lemma 5.5.

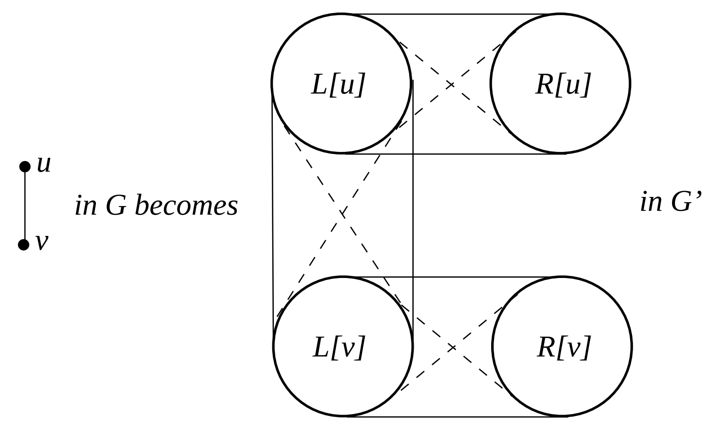


Figure 5.1: *The construction used in the proof of Lemma 5.1*

Comments

Lemma 5.1 is an important result and is often the "first stop" in trying to determine whether a graph is $\equiv_{\mathsf{AP}}\#SAT$ or not. Though this is a very loose way of describing the lemma, it suggests that, as a rule of thumb, sparse H graphs which have heavy (i.e. numerous) universal loops and a high number of colours are often $\equiv_{\mathsf{AP}}\#SAT$. The result also provides a much cleaner proof of Lemmas 25 and 26 from [8]. Lemma 25 proved that, for $q \geq 3$, $\#$*q-wrench*$\equiv_{\mathsf{AP}}\#SAT$. Lemma 5.1 also proves this because if H is the *q-wrench* ($q \geq 3$), $|F(H)||V(H)| \geq 5$ whilst the nearest competitor configuration (C, C) (where C comprises the centre loop plus one of the other looped colours) is subordinate because $|C||C| = 4$. Lemma 5.1 also updates Lemma 26 (which proved that, for $q \geq 4$, $\#$*q-WR*$\equiv_{\mathsf{AP}}\#SAT$) in exactly the same way.

It is also a sensible point to briefly demonstrate a generic technique we call "tipping". Observe, for example, that the graph *3-WR* (i.e. graph 23, which we showed to be $\equiv_{\mathsf{AP}}\#SAT$ with an ad-hoc reduction in Chapter 2) does not *quite* fit into the domain of Lemma 5.1. This is because, though $(F(H), V(H))$ and $(V(H), F(H))$ are maximal, they are not uniquely so. In this instance we can effectively "tip" *3-WR* into the domain of the lemma by observing that, if appropriate modifications are made to a graph G to yield G', there exists a graph H' (which *does* fall into the domain of Lemma 5.1) such that $\#H(G') = \#H'(G)$ i.e. $\#SAT\leq_{\mathsf{AP}}\#H'\leq_{\mathsf{AP}}\#H$. This is how we construct G' from G. For each vertex $u \in V(G)$, we introduce vertices $T[u]$ and $B[u]$ in G', and connect $T[u]$ to $B[u]$. For each edge $\{u, v\} \in E(G)$, we connect $T[u]$ to $T[v]$. That completes the construction. Now, it follows that $\#H(G') = \#H'(G)$ where H' is the weighted version of *3-WR* with weight 4 on the centre colour and weight 2 on all other colours. (This is because the addition of the $B[.]$ vertices means that each colour in *3-WR* becomes weighted by a factor equal to its degree.) As expected, H' does fall into the domain of the lemma, because $|F(H')||V(H')| = 4 \times 10 = 40$ and this beats the only competing configuration i.e. $|C||C| = 6 \times 6 = 36$ where C is the centre loop plus one non-centre colour.[2]

[2] If this seems familiar, there is a good reason for this. Note that the original ad hoc reduction for

The important point about the above is not the complexity of the graph *3-WR* or the specifics of the transformation on G, but the fact that if we take a graph G, transform it in some way and work out which graph H' is such that $\#H'(G) = \#H(G')$, we have derived a new graph H' which we can use as a proxy for H in hardness lemmas. This is what we mean when we talk about "tipping" a graph H into a lemma domain. We mention this process simply because it can be useful when we have otherwise exhausted our options with a graph H; the stylistic focus on the transformation of G as a "generator" of a new graph H' is deliberate, because in practice transformations on G are often applied speculatively i.e. we transform the graph without any particular knowledge of what graph H' will emerge.

The next lemma we prove is similar to Lemma 5.1, but is slightly more general in that there is no requirement that $F(H) \neq \emptyset$. It also differs in that the reduction is from $\#LargeCut$, rather than $\#IS$. We discuss this in more detail later, but it is perhaps significant to note that in both lemmas, our reductions rely on being able to extract "symmetrical" behaviour from H. That is, as a minimum we have to build a structure which is dominated by two configurations, each a symmetry of the other (e.g. $(F(H), V(H))$ and $(V(H), F(H))$) and each as likely as the other. This is so we can encode a uniform distribution on OUT/IN vertices (in the case of $\#IS$) and $LEFT/RIGHT$ cut vertices (in the case of $\#LargeCut$.)

Before describing the next lemma, it is helpful to make two key observations.

Observation 5.2 *(The axiom of mutual adjacency.) Let H be any graph. Let S and T be any sets such that $\emptyset \subset S, T \subseteq V(H)$. Then $H[S \cup T] \subseteq H[S]$ and $H[S \cup T] \subseteq H[T]$.*

Proof. To see that this observation holds, note that adding a colour to a set S can only decrease (or at best maintain) the mutual adjacency set of S. It further follows that $|H[S \cup T]| \leq \min(|H[S]|, |H[T]|)$.

3-WR is essentially combining the two steps just described (i.e. the transformation of G and then the application of Lemma 5.1) into a single step.

Observation 5.3 *Let H be any graph. Let $S, T \subseteq V(H)$ be any sets such that $S \neq T$ and $\{(S,T),(T,S)\} \subseteq MaxPairs(H)$. Then $|H[S \cup T]|^2 < |H[S]||H[T]|$.*

Proof. We obtain $|H[S \cup T]|^2 \leq |H[S]||H[T]|$ immediately, simply by applying Observation 5.2:- if (wlog) $|H[S]| \leq |H[T]|$, then $|H[S \cup T]|^2 \leq |H[S]|^2 \leq |H[S]||H[T]|$. To obtain the stronger result $|H[S \cup T]|^2 < |H[S]||H[T]|$ requires a little more work. First, if $|H[S]| \neq |H[T]|$ then the result must be true, because if (wlog) $|H[S]| < |H[T]|$ then $|H[S \cup T]|^2 \leq |H[S]|^2 < |H[S]||H[T]|$. So the only situation in which the result might not hold is when $|H[S]| = |H[T]|$. We show that the result holds even then. Noting that $S = H[T]$ and $T = H[S]$, it follows that $|S| = |T|$. Combining this with the fact that $S \neq T$, there must exist some colour c such that (wlog) $c \in T$ but $c \notin S$. Now, suppose $|H[S \cup T]| = |H[S]|$. Then $H[S] \subseteq H[\{c\}]$. (To see this, consider that if some colour c' is in $H[S]$ but not in $H[\{c\}]$, then c' is not in $H[S \cup T]$ and hence $|H[S \cup T]| < |H[S]|$.) However, if $H[S] \subseteq H[\{c\}]$, then we could construct a new valid pair (S', T') where $T' = T$ and $S' = S \cup \{c\}$. This would give us a contradiction, however, because $|S'||T'| > |S||T|$ and hence (S,T) would never have been in $MaxPairs(H)$ in the first place. So $|H[S \cup T]| < |H[S]|$, proving that $|H[S \cup T]|^2 < |H[S]||H[T]|$. $\Box$

Lemma 5.4 *(The "$(S,T) - (T,S)$ dominance" lemma.) Let H be a graph such that $MaxPairs(H) = \{(S,T),(T,S)\}$ where $S \neq T$ and $H[S \cup T] \neq \emptyset$. Then $\#SAT \leq_{\mathsf{AP}} \#H$.*

Proof. We reduce from the $\equiv_{\mathsf{AP}}\#SAT$ problem $\#LargeCut$ to $\#H$. (This proof is a simple variant of the proof for the graph *1-wrench*, i.e. graph 9.) First, note H is non-bipartite and non-trivial. The non-bipartite property follows because $H[S \cup T] \neq \emptyset$. To see that H is non-trivial, suppose by contradiction it *is* trivial; in which case, H is a fully looped clique, and $MaxPairs(H)$ should comprise solely $(V(H), V(H))$, but this is not possible because $S \neq T$. Also, S and T are both non-empty because, owing to its non-trivial status, H must contain at least one edge, which in turn ensures that any pair $(S,T) \in MaxPairs(H)$ contains at least one vertex in S

and at least one vertex in T.

Now, let (G, m) be an input to $\#LargeCut$ where m is the size of the largest cut in G; we build G' as follows. For each $u \in V(G)$ we introduce disjoint sets $L[u]$ and $R[u]$, both of size p (to be determined), and connect every vertex in $L[u]$ to every vertex in $R[u]$. For every edge $\{u, v\} \in E(H)$ we introduce two disjoint sets $S[uv]$ and $S'[uv]$, both of size t (to be determined.) Furthermore, we connect every vertex in $L[u]$ to every vertex in $S[uv]$, every vertex in $S[uv]$ to every vertex in $R[v]$, every vertex in $R[u]$ to every vertex in $S'[uv]$ and finally every vertex in $S'[uv]$ to every vertex in $L[v]$. We argue that that (as long as p is sufficiently large with respect to t) it is exponentially likely that, for every $u \in V(G)$, $(L[u], R[u])$ is coloured either (S, T) or (T, S). Furthermore, we observe that there is an exponential hierarchy within this set of colourings too. If $\{u, v\} \in E(G)$ and both $(L[u], R[u])$ and $(L[v], R[v])$ are coloured (S, T), $S[uv]$ and $S'[uv]$ are both restricted to $H[S \cup T]$. The same is true if they are both coloured (T, S). However, if (wlog) $(L[u], R[u])$ is coloured (S, T) and $(L[v], R[v])$ is coloured (T, S), $S[uv]$ is restricted to $H[S]$ and $S'[uv]$ is restricted to $H[T]$. Now, if we say a vertex $u \in V(G)$ is on the left side of the cut when $(L[u], R[u])$ is coloured (S, T), and on the right side of the cut when coloured (T, S), it follows that cut edges in G - those bridging the left/right sides of the cut - come up exponentially more (as colourings of G') than non-cut edges. This follows because a non-cut edge comes up $|H[S \cup T]|^{2t}$ times whilst a cut edge comes up $(|H[S]||H[T]|)^t$ times, and we know from Observation 5.3 that $|H[S \cup T]|^2 < |H[S]||H[T]|$. (Both non-cut edges and cut edges are definitely possible as colourings of G', because $H[S \cup T] \neq \emptyset$, and given that $H[S] = T$ and $H[T] = S$, the fact that both S and T are non-empty implies both $H[S]$ and $H[T]$ are non-empty.)

Hence, we claim that size-m cuts of G come up exponentially most as colourings of G', and we now formalise this. (We say that the full colourings of G' are those that correspond to size-m cuts in G.)

A size-i cut of G comes up the following number of times:

$$Z_i = 2(\nu(p, |S|)\nu(p, |T|))^n (|H[S]||H[T]|)^{ti} |H[S \cup T]|^{2t(|E(G)|-i)}$$

(The factor of 2 accounts for the fact that our choice of which of (S,T) and (T,S) denotes the left side of the cut is arbitrary.) Let $Z = Z_m$. We want to show that dividing our approximation of $\#H(G')$ by Z and rounding gives a good approximation to the number of size-m cuts in G. As we did in the $\#LargeCut$ reduction for *1-wrench*, we complete the proof by partitioning the set of non-full colourings into two sets, and showing that each set comes up at most $Z/8$ times. So, we let Y_0^+ be those colourings that do have every $(L[.], R[.])$ pair coloured either (S,T) or (T,S), but which correspond to cuts smaller than size m. We let Y_0^- be those colourings where at least one $(L[.], R[.])$ pair is *not* coloured either (S,T) or (T,S). We deal with Y_0^+ first. An upper bound on the number of cuts is 2^n, and if we assume each cut is of size $m-1$, we can account for Y_0^+ by proving

$$\frac{2^n(\nu(p,|S|)\nu(p,|T|))^n(|H[S]||H[T]|)^{t(m-1)}|H[S\cup T]|^{2t(|E(G)|-m+1)}}{(\nu(p,|S|)\nu(p,|T|))^n(|H[S]||H[T]|)^{tm}|H[S\cup T]|^{2t(|E(G)|-m)}} \leq 1/8$$

(The factor of 2 cancels.) Simplifying, we need

$$2^n\left(\frac{|H[S\cup T]|^2}{|H[S]||H[T]|}\right)^t \leq 1/8$$

Since $|H[S\cup T]|^2 < |H[S]||H[T]$, choosing $t = n^2$ is crude but adequate to satisfy this inequality for n beyond a threshold constant. It remains to show that Y_0^- colourings also come up at most $Z/8$ times. If we let $k = mp(H) = |S||T|$, an upper bound on the contribution of Y_0^- is at most $(2^{|V(H)|}2^{|V(H)|})^n k^{p(n-1)}(k-1)^p|V(H)|^{2t|E(G)|}$, where the $|V(H)|^{2t|E(G)|}$ term can be explained by generously assuming that all the $S[.]$ and $S'[.]$ sets are unrestrained in the colours they can take. A (very) crude lower bound on the contribution of full colourings is $(1/4)^n k^{pn}$, which we derive from Corollary 2.5. Hence, we need to show that

$$4^{(|V(H)|+1)n}((k-1)/k)^p|V(H)|^{2t|E(G)|} \leq 1/8$$

Assuming $t = n^2$, and noting that $|E(G)| \leq n^2$, it follows that the exponent of the $|V(H)|^{2t|E(G)|}$ term is (at most) of the order of n^4. Hence setting $p = n^5$ adequately satisfies the inequality for n beyond a constant threshold. $\square$

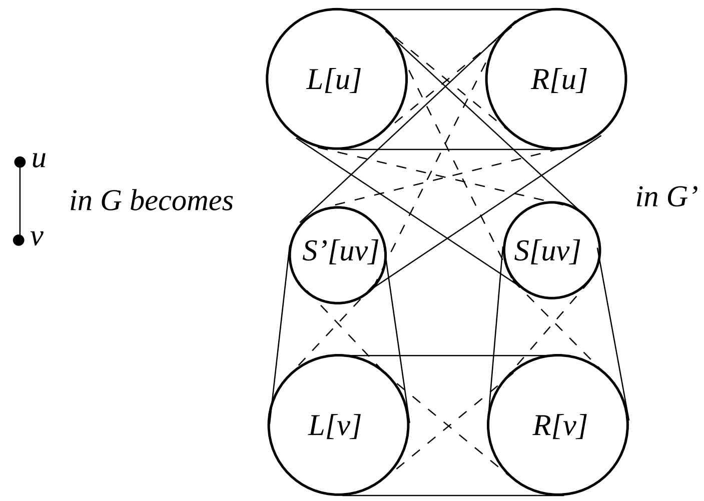


Figure 5.2: *The construction used in the proof of Lemma 5.4*

5.3 H with at least one universal loop, but no other loops besides, are $\equiv_{\mathsf{SP}} SAT$

Lemma 5.5 *(The "only loops are universal loops" lemma.) Let H be a non-trivial graph such that H has at least one universal colour, and there are no looped colours in H apart from universal colours. Then $IS \leq_{SP} H$ and by implication $SAT \leq_{SP} H$.*

The proof of this lemma can be broken down into a number of cases. The principle at the core of the most general case is elegant and demonstrates the flexibility of SP-reductions.

We recall the definitions of $mp(H)$, $MaxPairs(H)$, $F(H)$ and $Loops(H)$. The precondition set out by the lemma can be more formally expressed as the conjunction of the clauses $F(H) \neq \emptyset$, $F(H) \subset V(H)$ and $Loops(H) \setminus F(H) = \emptyset$. Let $UniversalUnlooped$ be the set of graphs fulfilling these requirements. Now, the proof is a mixture of case analysis and induction. The inductive core of the proof is to assume that $IS \leq_{\mathsf{SP}} H'$ for all graphs H' which are also in $UniversalUnlooped$ but smaller than H (in terms of number of vertices). As in the proof of Theorem 4.1 from Chapter 4 therefore, the idea is to build a graph G' by "glueing" together the SP-reductions $IS \leq_{\mathsf{SP}} H'$ for various $UniversalUnlooped$ subgraphs H' of H, where the SP-reductions are already known by induction. Then we show that most colourings in $H(G')$ allow us to "zoom in" on one of these reductions, thus allowing us to read off an independent set sample and hence proving that $IS \leq_{\mathsf{SP}} H$.

The easy part of the induction is establishing the base case: the smallest graph in $UniversalUnlooped$ is IS itself. Otherwise, there are three cases to consider, which we now describe.

Case 1: $MaxPairs(H) = \{(F(H), V(H)), (V(H), F(H))\}$.

In this case we can terminate the induction by observing that $IS \leq_{\mathsf{SP}} H$ follows immediately from the sampling version of Lemma 5.1.

Case 2: $\{(F(H), V(H)), (V(H), F(H))\} \not\subseteq MaxPairs(H)$.

Firstly, observe that, even though $(F(H), V(H))$ and $(V(H), F(H))$ do not occur in $MaxPairs(H)$, the fact that they exist means $MaxPairs(H)$ cannot be empty. To capitalise on this fact we first need to make an important observation:

Observation 5.6 *Let H' be any graph in $UniversalUnlooped$. Let $(S,T) \in MaxPairs(H')$ be such that $(S,T) \neq (F(H'), V(H'))$ and $(S,T) \neq (V(H'), F(H'))$. Then $H'[S]$ and $H'[T]$ are both in $UniversalUnlooped$, and both $H'[S]$ and $H'[T]$ have fewer vertices than H'.*

Proof. We need to first inspect the structure of (S,T). Given the definition of $MaxPairs(H')$, we know immediately that $F(H') \subseteq S$ and $F(H') \subseteq T$. The key to proving the observation is showing that $F(H') \subset S, T \subset V(H')$. (We explain why a little further on.) $F(H') \subset S, T$ is *almost* immediate because, by the definition of $MaxPairs(H')$, $F(H') \subseteq S, T$. To strengthen $\subseteq$ to $\subset$ observe that both S and T must contain at least one unlooped colour. If neither S nor T contained an unlooped colour, than $(S,T) = (F(H'), F(H'))$; this is not possible because $(F(H'), F(H'))$ cannot be in $MaxPairs(H')$ for any $H' \in UniversalUnlooped$. Suppose (wlog) only S contains an unlooped colour. Then $T = F(H')$. Since $(V(H'), F(H')) \notin MaxPairs(H')$, it follows that $S \subset V(H')$. But that means $|V(H')||F(H')| > |S||T|$, so $(S,T) \notin MaxPairs(H')$. Hence both S and T contain unlooped colours. We can show that $S,T \subset V(H')$ in a similar way. We know $(S,T) = (V(H'), V(H'))$ is not possible because H' contains at least one unlooped colour. So suppose (wlog) $S = V(H')$ but $T \subset V(H')$. But if $H'[T] = V(H')$ then $T = F(H')$:- T could not contain an unlooped colour and simultaneously be adjacent to $V(H')$. So that would mean $(V(H'), F(H')) \in MaxPairs(H')$ which we know is not the case. So we have shown that $F(H') \subset S,T \subset V(H')$. Recall that for every $(S,T) \in MaxPairs(H')$, we know (by definition) that $S = H'[T]$ and $T = H'[S]$. Consider (wlog) $H'[S]$. We have shown that $F(H') \subset H'[S] \subset V(H')$, so $H'[S]$ is connected (owing to $F(H') \subset H[S']$), smaller than H' (because $H'[S] \subset V(H')$) and in $UniversalUnlooped$ because $F(H') \subset V(H')$ and $H'[S]$ contains at least one unlooped colour. The case for $H'[T]$ is completely analogous. This completes the proof of Observation 5.6.

We return to the proof of Case 2. We enumerate the members of $MaxPairs(H)$ as (S_1,T_1), (S_2,T_2),..., (S_k,T_k) where $k = |MaxPairs(H)|$. Let $H_i = H[S_i]$. Given that all the H_i are in $UniversalUnlooped$ and smaller than H, we know by induction that $IS \leq_{\mathsf{SP}} H_i$ so let (f_i, g_i) be the SP-reduction from IS to H_i. Now, consider an input (G,ϵ) to IS. For each H_i, let $G_i = f_i(G, \epsilon/2)$. Here is (f,g), the SP-reduction from IS to H. We define $f(G,\epsilon) = G'$ as follows. Let I_1 and I_2 be two large sets of disjoint vertices, each of size p. (We will determine p in due course.) Connect every vertex in I_1 to every vertex in I_2. Now, for each H_i we attach every vertex in G_i to every vertex in I_1. Let Y_i be those colourings from $H(G')$ where (I_1, I_2) is coloured exactly with (S_i, T_i), and let Y_0 be all remaining colourings in $H(G')$. Consider any colouring $y \in H(G')$. If $y \in Y_0$ then we define $g(G,\epsilon,y) = \perp$. Otherwise, if $y \in Y_i$ then we know that H_i is pointed out in all the graphs hanging off I_1, so we can zoom in on G_i and pull an independent set sample out of there: we set $g(G,\epsilon,y) = g_i(G, \epsilon/2, y')$ where y' is y restricted to G_i.

As usual, we seek to demonstrate the correctness of this by first proving (3.4) i.e. showing that Y_0 is insignificant. Let $t = \sum_{(S_i,T_i)\in MaxPairs(H)} |V(G_i)|$. If we let $m = mp(H)$, a crude upper bound on $|Y_0|$ is

$$2^{|V(H)|}2^{|V(H)|}(m-1)^p|V(H)|^t$$

On the other hand, we know that (arbitrarily) $\#H(G') \geq |Y_1|$, so a very crude lower bound on $\#H(G')$ is the size of $|Y_1|$, which is itself bound below by $(1/4)m^p$. (We arrive at this quantity by noting that, for all (S_i, T_i), the contribution is at least $\nu(p, |S_i|)\nu(p, |T_i|)$ - where $|S_i||T_i| = m$ - so applying Corollary 2.5 to both onto functions gives the lower bound of $(1/4)m^p$.) [3] Hence, we need:

$$\frac{4^{|V(H)|}(m-1)^p|V(H)|^t}{(1/4)m^p} \leq \epsilon/4$$

This is easily satisfied by taking

$$p = \left\lceil \frac{\ln(4/\epsilon) + t\ln(|V(H)|) + (|V(H)|+1)\ln(4)}{\ln(m/(m-1))} \right\rceil$$

All we have to do now is prove (3.2). In reductions such as these, this poses no problem whatsoever. Each (f_i, g_i) is an SP-reduction, so we know that for all $x \in IS(G)$,

$$e^{-\epsilon/2}\frac{\#H_i(G_i)}{\#IS(G)} \leq |\{y' \in H_i(G_i)|g_i(G, \epsilon/2, y') = x\}| \leq e^{\epsilon/2}\frac{\#H_i(G_i)}{\#IS(G)} \qquad (5.1)$$

We also know $|Y_i| = \#H_i(G_i)\nu(p, |S_i|)\nu(p, |T_i|)\psi_i$ where ψ_i is the product of all $\#H_i(G_j)$ values for $j \neq i$, but more importantly is fixed for changing x. Given that for all $x \in IS$,

$$|\{y \in Y_i|g(G, \epsilon, y) = x\}| = \nu(p, |S_i|)\nu(p, |T_i|)\psi_i|\{y' \in H_i(G_i)|g_i(G, \epsilon/2, y') = x\}|$$

it therefore follows that we can obtain (3.2) by multiplying inequality (5.1) through by $\nu(p, |S_i|)\nu(p, |T_i|)\psi_i$. $\square$

Case 3: $\{(F(H), V(H)), (V(H), F(H))\} \subset MaxPairs(H)$

This case is significant because it falls "halfway" between Cases 1 and 2. Case 1

[3] One technicality that should be clarified is that all the (S_i, T_i) are definitely valid i.e. it is never the case that $Y_i = \emptyset$ for $i > 0$. This is because the G_i attached to I_1 can, if all other colourings are invalid, always be coloured just with $F(H)$.

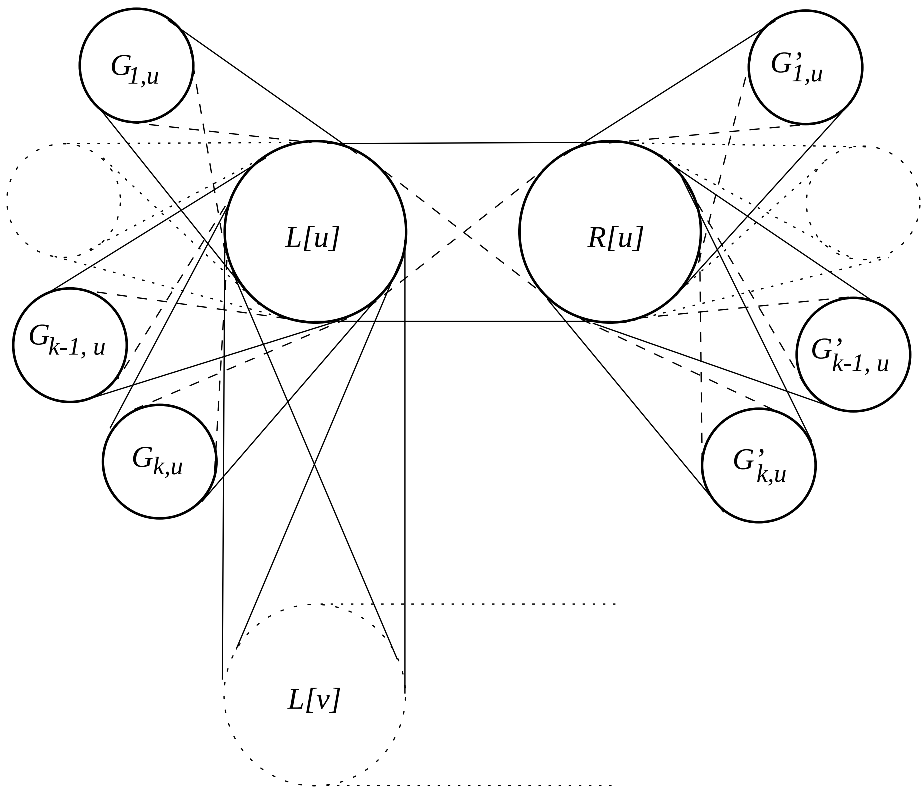


Figure 5.3: *The construction used in Case 3 of the proof for Lemma 5.5*

can't be used because the (S,T) pairs not equal to $(F(H),V(H))$ or $(V(H),F(H))$ might accidentally dominate the graph constructed in Lemma 5.1, and Case 2 can't be used because it may be exponentially likely that (I_1,I_2) is coloured $(F(H),V(H))$. Our solution, therefore, is to "merge" both cases into one reduction; in this respect it mirrors certain techniques used in the proof of Theorem 4.1 in Chapter 4. As a consequence, the reader will notice a high degree of similarity between the two respective proofs. Much of this similarity stems from the fact that here we are trying to do for non-bipartite H what Theorem 4.1 does for bipartite H i.e. code up independent sets. The proof of Theorem 4.1 is more sophisticated, however, because no "equalising" is required here; we explain this further when we come to the relevant part of the proof.

First, a few definitions. For $H \in UniversalUnlooped$, we define $ShrinkPairs(H)$ as follows:

$$ShrinkPairs(H) = MaxPairs(H) \setminus \{(F(H),V(H)),(V(H),F(H))\}$$

As the name suggests, $ShrinkPairs(H)$ are all those pairs $(S,T) \in MaxPairs(H)$

that point out smaller graphs (i.e. $H[S]$ and $H[T]$ are smaller than H) and so are useful in the context of the inductive step. We enumerate $ShrinkPairs(H)$ as $\{(S_1, T_1),$ (S_2, T_2), ..., $(S_k, T_k)\}$ [4]. For convenience we note that (for some k') $k = 2k'$ - because every distinct S, T combination comes up as both (S, T) and (T, S) - and say that, for $i \in [k'+1, 2k']$, $(S_i, T_i) = (T_{i-k'}, S_{i-k'})$. (In other words we enumerate $ShrinkPairs(H)$ such that the upper half of the entries are just the symmetries of the lower half.) For every $(S_i, T_i) \in ShrinkPairs(H)$, we define $H_i = H[S_i]$ and $H'_i = H[T_i]$. Now, we know by induction that for every $(S_i, T_i) \in ShrinkPairs(H)$, $IS \leq_{\mathsf{SP}} H_i$ and $IS \leq_{\mathsf{SP}} H'_i$. Hence, for each such (S_i, T_i) let (f_i, g_i) be the SP reduction from IS to H_i and let (f'_i, g'_i) be the SP reduction from IS to H'_i. We now show how to build an SP-reduction from IS to H; let (G, ϵ) be an input to IS. Firstly, for $i \in [1, k]$ we let $G_i = f_i(G, \epsilon/2)$ and $G'_i = f'_i(G, \epsilon/2)$. To proceed, we code up $f(G, \epsilon) = G'$ as follows. For every vertex $u \in V(G)$, we introduce two disjoint sets (both of size p, to be determined) $L[u]$ and $R[u]$ and connect every vertex in $L[u]$ to every vertex in $R[u]$. For $i \in [1, k]$ we introduce a copy of G_i (which we call $G_{i,u}$) and a copy of G'_i (which we call $G'_{i,u}$), connecting every vertex in $G_{i,u}$ to every vertex in $L[u]$ and every vertex in $G'_{i,u}$ to every vertex in $R[u]$. (Note that for $i > k'$, G_i is the same graph as $G'_{i-k'}$.) That completes the encoding of the vertex u. It is important to note that (unlike the construction in Case 2) the encoding of vertices from G is entirely symmetrical i.e. we hang the same graphs off $L[.]$ as we do off $R[.]$. The significance of this will be explained later. To complete the construction of G', we encode each *edge* $\{u, v\} \in E(H)$ by connecting every vertex in $L[u]$ to every vertex in $L[v]$.

We now partition $H(G')$ into several different categories. We define Y_{k+1} to be those colourings in $H(G')$ where, for all vertices $u \in V(G)$, $(L[u], R[u])$ is either coloured $(F(H), V(H))$ or $(V(H), F(H))$. We define Y_0 to be those colourings that are not in Y_{k+1}, and for which there does *not* exist a vertex $u \in V(G)$ and $i \in [1, k]$ such that $(L[u], R[u])$ is coloured exactly (S_i, T_i). To classify all remaining colourings, we first arbitrarily order $V(G)$ as $\{u_1, u_2, ..., u_{|V(G)|}\}$. Now, for a colouring $y \in H(G')$

[4] Note that, for all i, $S_i \neq T_i$, because $S_i = T_i$ could only happen if $(S_i, T_i) = (F(H), F(H))$ and this is not possible.

which is neither in Y_0 nor Y_{k+1}, let u_j be the first vertex for which $(L[u_j], R[u_j])$ is coloured exactly with some pair from $ShrinkPairs(H)$. If $(L[u_j], R[u_j])$ is coloured (S_i, T_i), then we say that $y \in Y_i$.

We sample independent sets as follows.

Let y be our sample from $H(G')$. If y is in Y_0, we return $\perp$. If $y \in Y_i$ (where $i \in [1, k]$) then observe that we can read off an independent set by zooming in on one of the subreductions hanging off the $(L[.], R[.])$ pairs. In particular, if $(L[u_j], R[u_j])$ is the pair coloured (S_i, T_i) we return the independent set given by $g_i(G, \epsilon/2, y')$ where y' is y restricted to G_{i,u_j} (In essence, therefore, we have pointed out H_i colourings in G_{i,u_j}.) Finally, suppose $y \in Y_{k+1}$. With this kind of colouring, the encoding of every vertex $u \in V(G)$ is either coloured $(F(H), V(H))$ or $(V(H), F(H))$. We can exploit this fact to read an independent set off directly, without the need to use the inductive step. Specifically, we say that a vertex u is IN the independent set if $(L[u], R[u])$ is coloured $(V(H), F(H))$, and OUT if it is coloured $(F(H), V(H))$.

(We attempt to do something very similar in the proof of Theorem 4.1, but reading off bipartite independent sets instead. On that occasion, we don't have the luxury of being able to set $L[.]$ and $R[.]$ to be the same size, because in general $|F_L(H)||V_R(H)| \neq |V_L(H)||F_R(H)|$ and if we can't "equalise" the contribution of $(F_L(H), V_R(H))$ and $(V_L(H), F_R(H))$ we get a non-uniform distribution on bipartite independent sets.)

This completes the informal explanation of how the proof works.

Before formally demonstrating the correctness of the reduction, the reader may be wondering why we have to hang all the various copies of G_i' from the $R[.]$ sets when they are not used to read off an independent set. This is to ensure that the encoding of each vertex in G' is symmetrical, which (in colourings from Y_{k+1}) ensures there is no exponential bias in favour of IN vertices over OUT vertices, or vice-versa.

The easiest part to formalise is (3.4) i.e. showing that $|Y_0|/\#H(G') \leq \epsilon/4$. To do this, observe that $\#H(G') \geq |Y_{k+1}|$. Furthermore, we know $|Y_{k+1}| \neq 0$ because if all other colourings fail we can always colour every $(L[.], R[.])$ with $(F(H), V(H))$, and colour all the copies of G_i and G'_i with $F(H)$. So if we can show $|Y_0|/|Y_{k+1}| \leq \epsilon/4$ we are done. Now, a lower bound on $|Y_{k+1}|$ is $(\nu(p, |F(H)|)\nu(p, |V(H)|))^n$. Using Corollary 2.5 a further lower bound on this is $(1/4)^n m^{pn}$. A very crude upper bound on $|Y_0|$ can be derived if we assume every one of the $(2^{|V(H)|}2^{|V(H)|})^n$ possible configurations on the $(L[.], R[.])$ pairs comes up $m^{p(n-1)}(m-1)^p|V(H)|^{tn}$ times, where

$$t = \sum_{i \in [1,k]} \Big(|V(G_i)| + |V(G'_i)| \Big)$$

Hence, we need to show

$$\frac{4^{n|V(H)|} m^{p(n-1)}(m-1)^p |V(H)|^{tn}}{(1/4)^n m^{pn}} \leq \epsilon/4$$

Gathering like terms, this becomes

$$4^{n(|V(H)+1)} \left(\frac{m-1}{m} \right)^p |V(H)|^{tn} \leq \epsilon/4$$

We can satisfy this by setting p as follows:

$$\left\lceil \frac{\ln(4/\epsilon) + n(|V(H)+1)\ln(4) + tn\ln(|V(H)|)}{\ln(m/(m-1))} \right\rceil$$

(Note that this value is not too large because t is at most polynomial in $|V(G)|$ and $1/\epsilon$. This follows because $|ShrinkPairs(H)|$ is bounded above by a constant (i.e. $2^{|V(H)|}2^{|V(H)|}$) and each G_i is itself no larger than polynomial in $|V(G)|$ and $1/\epsilon$, by the definition of SP-reduction.)

It remains to prove (3.2) for each Y_i ($i > 0$). We deal with Y_{k+1} first. Owing to the symmetry of the $(L[.], R[.])$ gadgets, every independent set $x \in IS(G)$ comes up *exactly* the same number of times as a colouring in Y_{k+1}. If we let $\alpha = |F(H)|^{t/2}$ and $\beta = \prod_{i \in [1,k]} \#H(G_i)$ we see that every independent set x comes up exactly $(\nu(p, |F(H)|)\nu(p, |V(H)|)\alpha\beta)^n$ times as a colouring of Y_{k+1}. (The fact underpinning this is that, for $i \in [k'+1, k]$, G_i is isomorphic to $G'_{i-k'}$.) Hence, for Y_{k+1} (3.2) is instantly satisfied.

Now we need to prove (3.2) for $Y_1, ..., Y_k$. Let $Y_{u,i}$ be the set of colourings in Y_i for which $u \in V(G)$ is the first vertex whose $(L[.], R[.])$ encoding is coloured (S_i, T_i). For these colourings we (as described above) exploit the fact that H_i colourings are pointed out in graph $G_{i,u}$ to read an independent set from the subreduction G_i encodes. Now, if Γ is the set of H_i colourings induced in $G_{i,u}$, it follows that for every independent set $x \in IS(G)$,

$$|\{y \in Y_{u,i} | g(G, \epsilon, y) = x\}| = \psi_{u,i} |\{y' \in \Gamma | g_i(G, \epsilon/2, y') = x\}| \tag{5.2}$$

where y' is y restricted to just $G_{i,u}$ and $\psi_{i,u}$ is the number of colourings of vertices other than $G_{i,u}$ which are induced by colourings in $Y_{u,i}$. (Crucially, $\psi_{u,i}$ does not change for changing x.) Since (f_i, g_i) is an SP-reduction, (3.1) gives

$$e^{-\epsilon/2} \frac{|\Gamma|}{\#IS(G)} \leq |\{y' \in \Gamma | g_i(G, \epsilon/2, y') = x\}| \leq e^{\epsilon/2} \frac{|\Gamma|}{\#IS(G)} \tag{5.3}$$

Given that $|Y_{u,i}| = |\Gamma| \psi_{u,i}$, multiplying (5.3) through by $\psi_{u,i}$ gives

$$e^{-\epsilon/2} \frac{|Y_{u,i}|}{\#IS(G)} \leq |\{y \in Y_{u,i} | g(G, \epsilon, y) = x\}| \leq e^{\epsilon/2} \frac{|Y_{u,i}|}{\#IS(G)}$$

Now, (3.2) follows because $|Y_i|$ is equal to the sum of $|Y_{u,i}|$ over all $u \in V(G)$, and the number of Y_i colourings that map to a particular independent set x is equal to the sum of the number of colourings that map to x in each of the $|Y_{u,i}|$ for $u \in V(G)$. $\square$

Comment

Lemma 5.5 shows that, for graphs $H \in UniversalUnlooped$, $SAT \leq_{\mathsf{SP}} H$. Hence, by the self-reducibility of the SAT problem, the discovery of a $PAUS$ for H would allow us to build a $PAUS$ for SAT and hence an $FPRAS$ for $\#SAT$[5]. Given that there is no $FPRAS$ for $\#SAT$ unless $RP = NP$, this shows that approximate sampling for $UniversalUnlooped$ is intractable.

We suspect that approximately counting $\#H$ is similarly intractable (i.e. has no $FPRAS$), but cannot yet prove this owing to the uncertain relationship between

[5] Recall from Section 3.10 that, because SAT is self-reducible, a $PAUS$ for SAT yields an $FPAUS$ for SAT which in turn yields an $FPRAS$ for $\#SAT$.

AP-reductions and SP-reductions and (separately) the unclear relationship that H-colourings have with regard to self-reducibility. As we have mentioned on a couple of occasions we tend to think that the existence of SP-reduction hardness results which utilise the "delineating between multiple subgraphs" method constitute some evidence that an AP-reduction analogue does, somewhere, exist. Additionally, given that $UniversalUnlooped$ represents a rigidly-defined subset of H-colouring problems it could be a fruitful area of work to try and show that (for graphs in $UniversalUnlooped$) approximate sampling is reducible to approximate counting, either by demonstrating self-reducibility of those graphs or through some other method.

5.4 Grabbing looped cliques

Recall that a looped clique is a set of colours that are all mutually adjacent. Hence, a graph H has a looped clique of size m iff K_m^* is a subgraph of H. (We identify a looped clique by the set of vertices that comprise it.) The following lemma seeks to exploit the fact that we can build a general gadget which allows us to pick out subgraphs which are large, looped cliques. In particular, we note that where such subgraphs themselves point out non-trivial subgraphs, the pointed out subgraphs have a structural property which makes them $\equiv_{\mathsf{SP}} SAT$. (Note that Lemma 5.7 is a "sampling" lemma i.e. it uses SP-reductions rather than AP-reductions. A less general version, based on the AP-reduction, is given by Lemma 5.8 in due course.)

Informally, Lemma 5.7 operates by introducing a new piece of gadgetry ($MaxCliqueGrab$ - which by definition is exponentially likely to be coloured with large cliques from H) and hanging lots of $IS \leq_{\mathsf{SP}} H'$ reductions off it, for those H' pointed out by such cliques. Thus, when the $MaxCliqueGrab$ gadget is coloured with such a clique, we can obtain an IS sample by "zooming in" on the appropriate reduction. (This shows $IS \leq_{\mathsf{SP}} H$ and thus $SAT \leq_{\mathsf{SP}} H$.)

Lemma 5.7 *(The "clique-grabbing" lemma.) Let H be a graph and let m be the size of the largest looped clique in H, where $m \geq 2$. If there exists at least one size-m*

looped clique $C \subset V(H)$ such that $|H[C]| > m$, then $SAT \leq_{SP} H$.

We actually show that $IS \leq_{SP} H$, with a little help from Lemma 5.5 on page 167. Let l be the largest value of $|H[C]|$ ranging over all size-m looped cliques C i.e. l is the size of the largest subgraph pointed out by any size-m looped clique. (Note that $l > m$ by the precondition of the lemma.) Now, we let $C_1, ..., C_k \subset V(H)$ be the k size-m looped cliques that point out subgraphs of size l; from our preconditions we know that $k \geq 1$. We let $H_i = H[C_i]$ for $i \in [1, ..k]$. The following observation is crucial:- all of the H_i are non-trivial, of size l, and have m universal colours, but *no looped colours other than those*. First we show that each H_i has m universal colours. This follows because $C_i \subset H[C_i]$ and by definition every colour in $H[C_i]$ must be adjacent to all the colours in C_i. To see that there are no loops in $H[C_i]$ other than its m universal colours, suppose $H[C_i]$ contained (say) a looped colour c in addition to its m universal colours. This would make K^*_{m+1} a subgraph of $H[C_i]$, and by implication also a subgraph of H, but we know that the largest looped clique in H is of size-m, so this is not possible. The non-triviality of $H[C_i]$ (and by implication of H also) therefore follows because $H[C_i]$ has at least one universal colour (ensuring it is connected) and contains both looped and unlooped colours. Hence we know that, by Lemma 5.5, $IS \leq_{SP} H_i$ for each H_i. This fact lies at the heart of the following proof, which employs gadgetry to make the C_i exponentially dominant, and (in the usual SP-fashion) then "glues" together the $IS \leq_{SP} H_i$ reductions for all the C_i. To operationalise this we use a gadget called a $MaxCliqueGrab$, which has already been used in a simpler, more ad-hoc form throughout Chapter 2. The $MaxCliqueGrab$ gadget is of general importance so it is helpful to study it now in its own right, rather than "hard-wiring" it irretrievably into the rest of the proof. Though it would be nice to produce a completely self-contained gadget that can be formally parameterised and just "plugged in" where needed, it unnecessarily complicates matters to do so. Therefore, in the following description we opt for a compromise solution and (as we explain shortly) assume the t and ϵ' values used are defined elsewhere in the proof that the $MaxCliqueGrab$ gadget is used in. (We return to the proof of Lemma 5.7 after discussing the construction and behaviour of the

$MaxCliqueGrab$ gadget.)

The MaxCliqueGrab gadget

The gadget consists of K (a complete graph on p vertices) and I (a set of q disjoint vertices.) We connect every vertex in K to every vertex in I. We claim that it is possible to choose values of p and q such that, irrespective of how the gadget is connected into the rest of the graph, the sum of the number of colourings (with i ranging from 1 to k) where K is coloured with exactly the set of colours C_i, is exponentially large. In other words, it is exponentially likely that a colouring of the gadget will have K coloured with exactly the set of colours from some maximum clique.

For the sake of generality, we assume that the only edges connecting the gadget to the rest of the graph[6] stem from K; this is a reasonable assumption because the intention is usually to point out $H[C_i]$ in some other part of the graph. We further ensure generality by simply specifying that the graph beyond the gadget contains t vertices; this, along with the value ϵ' (which we assume is defined elsewhere), is enough information from which to derive adequate values for p and q. We should state that, as depicted in Figure 5.4, the assumption is that the $MaxCliqueGrab$ gadget is used only once, as the dominant feature of the constructed graph. As discussed in the ad-hoc proof for graph 36 there are situations where it is sometimes desirable to use a gadget at the vertex level i.e. introducing a new copy of the gadget for each vertex in the input graph, rather than introducing a single, overarching copy of the gadget. This derivation does not hold in these situations (because t is no longer meaningfully defined) but the analysis is very similar and where necessary could be easily adapted from this analysis.

We show the dominance of C_i colourings in K by considering all the different configurations possible in K. Firstly, we say $c \in V(H)$ is a *maxclique colour* iff $c \in C_i$ for some C_i. (Note that a non-maxclique colour may be looped, and a non-maxclique colour may be a member of a size-m looped clique C, as long as $|H[C]| < l$.) For

[6]By "rest of the graph" we mean whatever graph the $MaxCliqueGrab$ gadget is being used in i.e. probably some graph G' being constructed on the basis of the input graph G.

each configuration, we are interested in the number of colours that appear in K, and in particular the number of maxclique colours that appear in K. Therefore, for a given configuration, we let x be the number of maxclique colours appearing in K, and z be the number of non-maxclique colours appearing in K.

In this context, the configuration we wish to dominate is characterised by the constraint set $\{x + z = m, z = 0\}$ where all the x maxclique colours are drawn from the *same* C_i. Such a configuration comes up at least $\nu(p,m)l^q$ times: we show that no other configuration can beat this.

For ease of reference we enumerate the other types of configuration possible within K. In each case we denote the set of colours appearing in K as S. After analysing the contribution that a configuration from each case makes, we show how it is possible to choose p and q to render them all inferior. Note that the case $x > m$ is not possible because then the colours in S would induce[7] a subgraph of H containing a looped clique of at least size-$(m + 1)$, but we know H does not contain any looped cliques of size larger than m.

[Rival Case 1] First, we consider the situation where $\{x + z = m, z = 0\}$ but the x maxclique colours from S are sourced from more than one C_i. More formally, suppose $c, d \in S$ are both maxclique colours, with $c \neq d$, but that there is no C_i such that c and d are both in C_i.

Now, since $|S| = m$, it follows that the colours in S induce a subgraph of H with underlying graph K_m. (The *underlying* graph is the graph obtained by erasing all loops.) Furthermore, since all colours in S come from various C_i, all the colours in S are looped. That means the graph induced by S is actually a size-m looped clique. Now, $|H[S]| < l$ because if $|H[S]| = l$ then S would actually be equal to some C_i, and this would contradict our assertion that there is no C_i which simultaneously contains c and d. Therefore we can assume that $|H[S]| \leq (l - 1)$. This means that an upper bound on the number of times this configuration can come up is $m^p(l - 1)^q$.

[7] Recall that, throughout this thesis, the subgraph "induced by" a set S is a distinct concept to the subgraph "pointed out" by a set S i.e. $H[S]$. We use the phrase "induced by" in the conventional graph-theoretic sense i.e. the subgraph remaining if all vertices but S are removed.

The remaining cases we have to consider are $\{x < m, x+z < m\}$, $\{x < m, x+z \geq m\}$ and $\{x = m, z > 0\}$.

[Rival Case 2] If $x < m, x+z < m$ then a (crude) upper bound on the number of times this configuration comes up is $(m-1)^p|V(H)|^q$.

[Rival Case 3] We now consider $\{x < m, x+z \geq m\}$. Observe that the colours in S induce a subgraph that has underlying structure $K_{|S|}$. As a consequence there must be less than or equal to $m - x$ looped colours amongst the z non-maxclique colours, because otherwise S would induce a subgraph containing a looped clique at least of size $(m+1)$ which is not possible. Now, we have to split this case into two subcases.

1. In **[Rival Case 3.1]**, we assume that there are fewer than m looped colours in S overall. Proceeding, we let $Perm(i,j)$ equal the number of ways of picking i elements from j where order is important i.e. $Perm(i,j) = \frac{j!}{(j-i)!}$. Observe that if S contains α looped colours and β unlooped colours, K can be coloured in exactly $\nu(p-\beta, \alpha)Perm(\beta, p)$ ways. (This is because each unlooped colour can only occur once.) Given that $Perm(i,j) \leq j^i$, it follows that a crude upper bound on the number of times a configuration within this case can come up is $(m-1)^p p^{|V(H)|}|V(H)|^q$.

2. In **[Rival Case 3.2]**, we assume S contains exactly m looped colours in total. These m looped colours cannot all be maxclique colours because this would contravene the precondition that $x < m$. However, the m looped colours do induce a size m looped clique in H. So, by definition, I must be coloured with fewer than l colours, because otherwise H would contain a size m looped clique *not equal to any* C_i which nonetheless points out a subgraph of size greater than or equal to l - contradiction! Hence, an upper bound on this case is $m^p p^{|V(H)|}(l-1)^q$.

[Rival Case 4] The remaining case is $\{x = m, z > 0\}$. At the core of this case is the need to show that $|H[S]| < l$. We know $|H[S]| \leq l$ because, were it the case that $|H[S]| > l$, the m maxclique colours in S would constitute a size-m looped clique that points out a subgraph larger than size l, and by definition this is not possible. So let's assume that $|H[S]| = l$. Since $x = m$ and $z > 0$ there is at least one unlooped colour $c \in S$, so we know $c \notin H[S]$. Now suppose S' is $S \setminus \{c\}$. We know $H[S']$ is similar to $H[S]$ except that c can appear in $H[S']$, making $H[S']$ a subgraph of size $l+1$. But then the size-m looped clique induced by the m maxclique colours in S points out a subgraph larger than size l, contradiction! So $|H[S]| < l$. Hence, a crude upper bound on this configuration is $m^p p^{|V(H)|}(l-1)^q$.

Recall that the configuration we wish to be dominant comes up at least $\nu(p,m)l^q$ times; a lower bound on this is $(1/2)m^p l^q$. Here is a summary of the rival cases we have identified:

[Rival Case 1]: $m^p(l-1)^q$
[Rival Case 2]: $(m-1)^p|V(H)|^q$
[Rival Case 3.1]: $(m-1)^p p^{|V(H)|}|V(H)|^q$
[Rival Case 3.2]: $m^p p^{|V(H)|}(l-1)^q$
[Rival Case 4]: $m^p p^{|V(H)|}(l-1)^q$

We note that there can be no more than $2^{|V(H)|}$ different configurations corresponding to each Rival Case. (The $2^{|V(H)|}$ value emerges if we assume that all subsets of colours are possible in K and fit into that case.) We also pessimistically assume that the rival cases allow the t vertices that comprise the rest of the graph to be coloured freely with all $|V(H)|$ colours, explaining the presence of the $|V(H)|^t$ term in the analysis below.

Now, if for each Rival Case 1 through 4 (pessimistically assuming, in each case, that there are $2^{|V(H)|}$ configurations in each case) we can show that the ratio of its contribution to $(1/2)m^p l^q$ is less than or equal to $\epsilon'/20$, this proves that the overall ratio of full to non-full colourings is less than or equal to $\epsilon'/4$ (Here full and non-full refer to the desirable configuration and Rival Cases 1-4 respectively.) Conveniently, we actually need only satisfy an $\epsilon'/8$ bound for Cases (3.1) and (4), because proving it for Case (3.1) automatically covers Case (2), and proving Case (4) automatically covers Cases (1) and Case (3.2).[8] So first we consider Case (3.1). The relevant inequality is:

$$\frac{|V(H)|^t 2^{|V(H)|}(m-1)^p |V(H)|^q p^{|V(H)|}}{(1/2)m^p l^q} \leq \epsilon'/8$$

If we tidy this up we get

$$|V(H)|^t 2^{|V(H)|+1}\left(\frac{m-1}{m}\right)^p \left(\frac{|V(H)|}{l}\right)^q p^{|V(H)|} \leq \epsilon'/8 \tag{5.4}$$

The relevant inequality for Case (4) is

$$|V(H)|^t 2^{|V(H)|+1}\left(\frac{l-1}{l}\right)^q p^{|V(H)|} \leq \epsilon'/8 \tag{5.5}$$

(To recap, we know that $l-1 \geq 2$ because, by the precondition of the lemma, we know that $l > m$ and $m \geq 2$.) We have to choose p and q to simultaneously satisfy these two inequalities. To help facilitate this we set $p = Mq$ where M is some positive integer constant with the property that

$$\left(\frac{m-1}{m}\right)^M \left(\frac{|V(H)|}{l}\right) \leq \left(\frac{l-1}{l}\right)$$

This is achieved by taking $M \geq \ln(|V(H)|/(l-1))/\ln(m/(m-1))$ so we simply take M to be the lowest positive integer that satisfies this inequality. To explain, observe that choosing M in this fashion means that if we then satisfy (5.5) we automatically satisfy (5.4) i.e. it leaves us with just a single inequality to solve. Continuing with (5.5) then, it becomes

$$|V(H)|^t 2^{|V(H)|+1} M^{|V(H)|}\left(\frac{l-1}{l}\right)^q q^{|V(H)|} \leq \epsilon'/8$$

The $q^{|V(H)|}$ term can be absorbed by noting that

$$\left(\frac{l-1}{l}\right)^q q^{|V(H)|} \leq \left(\frac{l-(1/2)}{l}\right)^q$$

as long as $\ln(q)/q \leq \ln((l-(1/2))/(l-1))|V(H)|^{-1}$. Assuming q is chosen large enough to satisfy this condition (which is not difficult because the RHS of this inequality

[8] This is because, in total, there can be no more than $2^{|V(H)|}$ rival configurations to consider.

is constant) q also needs to satisfy

$$|V(H)|^t 2^{|V(H)|+1} M^{|V(H)|} \left(\frac{l-(1/2)}{l}\right)^q \leq \epsilon'/8$$

and this is satisfied by setting q to

$$\left\lceil \frac{\ln(8/\epsilon') + t\ln(|V(H)|) + (|V(H)|+1)\ln(2) + |V(H)|\ln(M)}{\ln(l/(l-(1/2)))} \right\rceil$$

or whatever value is necessary to satisfy the $\ln(q)/q$ bound just mentioned, whichever is the larger. (The larger is almost certainly the value in the above inequality.) □

This completes our analysis of the $MaxCliqueGrab$ gadget; we have shown that in "most" cases the set of colours appearing in K is exactly equal to C_i for some C_i. Having described the gadget we now return to our proof of Lemma 5.7 where we demonstrate the gadget in use.

Continuing with the proof of Lemma 5.7

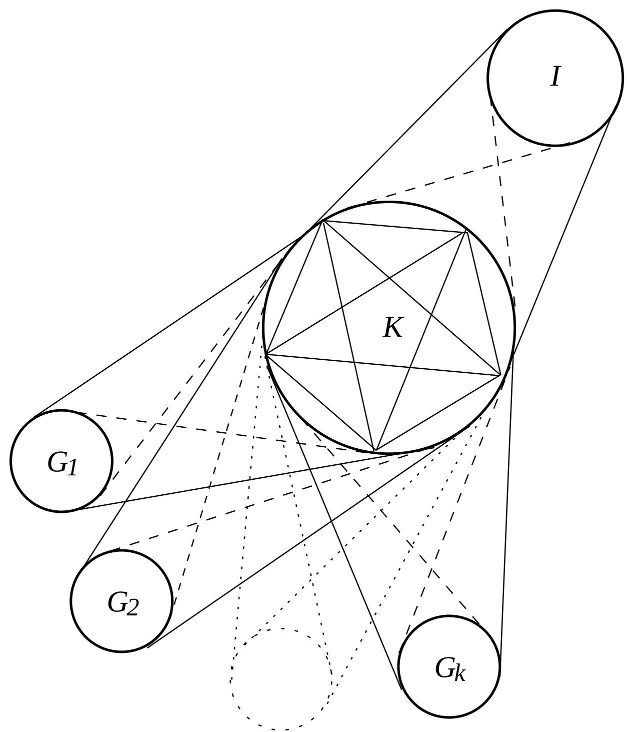


Figure 5.4: *The construction of $f(G,\epsilon)$ in Lemma 5.7*

We let (G, ϵ) be an input to the IS sampling problem. Here is an SP-reduction (f, g) from IS to H. For $i \in [1, k]$, let (f_i, g_i) be the SP-reduction from IS to H_i; we know that these reductions exist because as discussed earlier $IS \leq_{\mathsf{SP}} H_i$ by Lemma 5.5. For each i, we let $G_i = f_i(G, \epsilon/2)$. Now, $f(G, \epsilon)$ (which we refer to as G') is constructed as follows. Let $t = \sum_{i \in [1,k]} |V(G_i)|$. Now, we introduce a $MaxCliqueGrab$ gadget, taking t as just defined and $\epsilon' = \epsilon$. For each G_i, we connect every vertex in G_i to every vertex in the K set of the $MaxCliqueGrab$ gadget. This completes the construction of $f(G, \epsilon)$. Now, we let Y_i (for $i \in [1, k]$) be the set of H-colourings of G' where K is coloured precisely with colours from C_i. We let Y_0 be the set of all remaining colourings. To complete our SP-reduction, we need to define $g(G, \epsilon, y)$ for $y \in H(G')$. So let $y \in H(G')$; if $y \in Y_0$ we set $g(G, \epsilon, y) = \perp$. Otherwise, we inspect K and let C_i be the set of colours appearing there; since $H[C_i] = H_i$, we can pull an independent set sample from the SP-reduction (f_i, g_i), so we set $g(G, \epsilon, y) = g_i(G, \epsilon/2, y')$ where y' is just y restricted to G_i. To make our proof tight we must show that $|Y_0|$ is not too large; the MaxCliqueGrab derivations shows that $|Y_0|/\#H(G') \leq \epsilon/4$, so (3.4) is automatically covered. Finally, we need to show that (3.2) holds for every independent set x in $IS(G)$ and $i \in [1, k]$. This is immediate; we already know that

$$e^{-\epsilon/2} \frac{\#H_i(G_i)}{\#IS(G)} \leq |\{y \in H_i(G_i) | g_i(G, \epsilon/2, y) = x\}| \leq e^{\epsilon/2} \frac{\#H_i(G_i)}{\#IS(G)}$$

because (f_i, g_i) is an SP-reduction. We know that $|Y_i| = \#H_i(G_i)\psi(i)$ where $\psi(i)$ is the number of colourings possible in $G' \setminus G_i$ when K is coloured with C_i. In particular, each colouring in $H_i(G_i)$ comes up exactly $\psi(i)$ times as a colouring in $|Y_i|$, so (3.2) is assured. $\square$

Under certain restricted circumstances we can demonstrate an AP-reduction rather than an SP-reduction, as the following corollary demonstrates.

Corollary 5.8 *Let H be a non-bipartite, non-trivial graph. Let m be the size of the largest looped clique in H, where $m \geq 2$. We let l be the maximum value of $|H[C]|$ ranging over all size-m looped cliques C; we require $l > m$. Let $C_1, ..., C_k$ be the set of size-m looped cliques that point out subgraphs of size l. If $k = 1$ or if all the $H[C_i]$*

are isomorphic then $\#H' \leq_{\mathsf{AP}} \#H$, where $H' = H[C_1]$.

Proof. Before continuing, the reader may be wondering why $\#IS \leq_{\mathsf{AP}} \#H$ is not the outcome of this corollary, when this seems like the natural AP analogue to Lemma 5.7. This can be explained by recalling that Lemma 5.7 is heavily dependent on Lemma 5.5, and together they show that $IS \leq_{\mathsf{SP}} H[C_i]$ for all C_i. However, we don't in general know whether $\#IS \leq_{\mathsf{AP}} \#H[C_i]$ for all C_i, so we cannot claim that $\#IS \leq_{\mathsf{AP}} \#H$ unless we know $\#IS \leq_{\mathsf{AP}} \#H'$.

We let G be the input to $\#H'$. We let (K, I) be the MaxCliqueGrab gadget constructed with "parameters" $t = |V(G)|$ and ϵ' equal to 1.[9] We let G' consist of (K, I) together with G, and we connect every vertex in G to every vertex in K. Now, we know that when C_1 appears in K the $MaxCliqueGrab$ gadget can be coloured in $\nu(p, m)l^q$ ways. So, dividing the result of our $\#H(G')$ approximation by $k\nu(p, m)l^q$ and rounding gives us a good approximation to $\#H'(G)$. (It is important to point out, however, that this seamless "mapping over" of the gadgetry to the AP-world only works in this instance because our $MaxCliqueGrab$ derivation actually did more than was required and proved $|Y_0|/\nu(p, m)l^q \leq \epsilon/4$, when all it technically had to do was prove $|Y_0|/|Y_1| \leq \epsilon/4$. See Section 3.8.2 for a more detailed discussion of this.) □

Before introducing the next result, we have to introduce some new definitions.

5.5 Grabbing loops

As mentioned at the start of the section, $Loops(H) = \{c \in V(H) | \text{c has a loop}\}$ and - if Δ_l is the maximum degree of colours in $Loops(H)$ - $MaxLoops(H) = \{c \in Loops(H) | deg(c) = \Delta_l\}$. With these definitions at our disposal we can now introduce the $MaxLoopGrab$ gadget, which is a close relative of the $MaxCliqueGrab$ gadget.

The MaxLoopGrab gadget

[9] We can use $\epsilon' = 1$ in this instance because the standard rounding technique used in many of our AP-reductions only requires that we beat a $1/4$ threshold rather than the stricter $\epsilon/4$ threshold.

This is another general-purpose gadget so, as with the $MaxCliqueGrab$ gadget, we choose to separate its description from the result in which it is first used (i.e. Lemma 5.10, below.) Informally, the gadget makes it exponentially likely that some colour from $MaxLoops(H)$ is picked out. It is not possible to tell precisely which member or members of $MaxLoops(H)$ is/are picked out, because this is dependent on the way the gadget is connected to the rest of the graph and the structure of the subgraph pointed out by each $MaxLoops(H)$ colour, but we know at the very least that the colouring space in which the gadget does *not* pick out a member of $MaxLoops(H)$ can be made exponentially small. This is a significant characteristic because it means that, in essence, it is always possible to pick out a looped colour *irrespective of the properties of the unlooped colours in the graph*. This kind of "hard" delineation between different types of colour is generally difficult to come by so this is a welcome result.[10]

In building and analysing the gadget, we assume that H has at least one looped colour and that H is in *compact form*. Recall that this is where equivalent colours are grouped into equivalence classes and the weight of a colour (in compact form) is equal to the number of colours in the equivalence class. (There is a full explanation of compact form in Section 2.2.1.) Hence $H = (V(H), E(H), w)$ where $V(H) = \{c_1, ..., c_{|V(H)|}\}$ and we let $w_i = w(c_i)$ for $c_i \in V(H)$. To avoid ambiguity we state that, when H is given in compact form, $Loops(H)$ is given in compact form (i.e. $Loops(H) \subseteq V(H)$ rather than $Loops(H) \subseteq V'(H)$, where $V'(H)$ is the expanded vertex set of H) and $\Delta_l = \max(deg'(c_i) | c_i \in Loops(H))$ where (as introduced in Section 2.2.1) $deg'(c_i) = \sum_{c_j \in adj(c_i)} w_j$. Our assumption of compact form representation increases the generality of the results in this section.

The gadget consists of K, which is a copy of $K_{|V'(H)|+1}$, and I, which is a disjoint set of k vertices, where k is to be determined. We connect every vertex in K to every vertex in I. Again we generalise about the connection of the gadget to the "outside world", stipulating only that any edges leaving the gadget must extend from K only

[10]It should be stressed, however, that this gadget is not in general useful for (say) showing $\#H' \leq_{\mathsf{AP}} \#H$ where H' is the subgraph of H induced by $Loops(H)$, because the gadget introduces an exponential hierarchy amongst $Loops(H)$ based on a colour's degree.

(i.e. I is unconnected to the rest of the graph) and that the rest of the graph contains t vertices. (We define G' to be the whole graph i.e. the gadget plus the rest of the graph it is connected into.) We let full colourings of $H(G')$ be those where K is coloured in monochrome with some $c_i \in MaxLoops(H)$. Since c_i represents an equivalence class of colours, "monochrome" in this context means any of the $w_i^{|V'(H)|+1}$ colourings of K possible using colours from that equivalence class. Our goal is to show that the ratio of non-full to full colourings is less than or equal to $\epsilon'/4$, where ϵ' is specified by the context the gadget is used in.

First, observe that a lower bound on the number of colourings of G' in which K is coloured monochrome c_i (where $c_i \in MaxLoops(H)$) is $w_i^{|V'(H)|+1}\Delta_l^k$. This is the benchmark we show cannot be beaten. Now, consider that if K can be coloured exactly with a set of colours $S \subseteq V(H)$ then $S \cap Loops(H) \neq \emptyset$. This is because the only way that K could be coloured without a looped colour is if H (in its expanded form) has a $K_{|V'(H)|+1}$ subgraph, and obviously this is not the case. So let $c_i \in Loops(H)$ be some looped colour in S.

We know by Observation 5.2 that the number of colours possible in I when K is coloured with S (where S is *not* coloured monochrome with some colour from $MaxLoops(H)$) is less than or equal to the number of colours possible in I when K is just coloured with c_i. The question remaining is whether the relationship is always in fact "strictly less than" - which we want - rather than "less than or equal to". On this note, suppose c_j is also in S, where $c_j \neq c_i$. Since we are using compact form representation, $adj(c_i) \neq adj(c_j)$. Now, it is not too difficult to show that $|adj(c_i) \cap adj(c_j)| < |adj(c_i)|$. (Given that $adj(c_i) \neq adj(c_j)$, this is immediate if in addition $adj(c_i) \not\subset adj(c_j)$. To see that $adj(c_i) \not\subset adj(c_j)$, suppose by way of contradiction it *is* true that $adj(c_i) \subset adj(c_j)$. We know c_j cannot be looped, because then c_j would have higher effective degree than c_i and thus c_i would never have been in $MaxLoops(H)$ in the first place. But we also know that c_j cannot be unlooped, because the fact that c_i and c_j are in S simultaneously means that $\{c_i, c_j\} \in E(H)$, and that would mean $c_j \in adj(c_i)$ but $c_j \notin adj(c_j)$. Contradiction!)

Hence, if S contains c_i and some other distinct colour(s), that configuration on K can come up at most $|V'(H)|^t|V'(H)|^{|V'(H)|+1}(\Delta_l - 1)^k$ times. So, as long as we choose k large enough, this shows that it's always best to colour K monochrome c_i for some $c_i \in MaxLoops(H)$. We now formalise our choice of k; we want to show that the ratio of colourings in which K is not coloured monochrome c_i (for some $c_i \in MaxLoop(H)$) to colourings in which K *is* monochrome c_i is less than or equal to $\epsilon'/4$. Satisfying the following inequality will suffice:

$$\frac{|V'(H)|^t|V'(H)|^{|V'(H)|+1}(\Delta_l - 1)^k}{\Delta_l^k} \leq \epsilon'/4$$

To satisfy this, we can set k as

$$k = \left\lceil \frac{\ln(4/\epsilon') + \ln(|V'(H)|(t + |V'(H)| + 1))}{\ln(\Delta_l/(\Delta_l - 1))} \right\rceil$$

This satisfies the $\epsilon'/4$ bound. □

Before the next result, the following observation is useful:

Observation 5.9 *Let H be a non-bipartite, non-trivial graph. (Though all other results in this Section - i.e. Section 5.5 - assume H is presented in compact form, it is easier for this Observation to assume H is presented in expanded form.) Let c be some colour in $MaxLoops(H)$. Then $H[c]$ is non-trivial.*

Proof. Suppose $H[c]$ was trivial. Then $H[c] = K_m^*$, where $m = deg(c)$, because $c \in H[c]$ and the only trivial graphs containing loops are looped cliques. It follows that c and all its neighbours are of degree m. We know H is non-trivial, so K_m^* must be a strict subgraph of H. Given that H is connected, there must therefore be some colour $d \notin adj(c)$ which connects to one or more colours in $adj(c)$; suppose c' is one of the colours in $adj(c)$ that d connects to. But that would mean $deg(c') > m$ - contradiction! So $H[c]$ is non-trivial. □

Lemma 5.10 *(The "no adjacent loops" lemma.) Let $H = (V(H), E(H), w)$ be a non-trivial graph presented in compact form, where $V(H) = \{c_1, ..., c_{|V(H)|}\}$ and*

$Loops(H) \neq \emptyset$. Let $MaxLoops(H)$ and Δ_l be as defined at the beginning of Section 5.5, on page 185. If, for all $c_i \in MaxLoops(H)$, $((adj(c_i) \setminus \{c_i\}) \cap Loops(H)) = \emptyset$ then $SAT \leq_{SP} H$.

Proof. This lemma basically says that if all the $MaxLoops$ are "isolated" (i.e. are not adjacent to any other loops) then $SAT \leq_{SP} H$; we actually show $IS \leq_{SP} H$. If the only looped colours in H are universal colours, then the result follows immediately from Lemma 5.5. Otherwise, note that Observation 5.9 tells us that, for all $c_i \in MaxLoops(H)$, $H[c_i]$ is non-trivial. By definition c_i becomes a universal colour when it appears in $H[c_i]$, and given that $H[c_i]$ is non-trivial and has no other loops besides c_i - because all loops are "isolated" - it follows that $H[c_i]$ must contain at least one unlooped colour. As a result, we know by Lemma 5.5 that for all $c_i \in MaxLoops(H)$, $IS \leq_{SP} H[c_i]$. We let $r = |MaxLoops(H)|$ and (wlog) assume the colours in $MaxLoops(H)$ are c_1, c_2, ..., c_r. Now, for each $c_i \in MaxLoops(H)$ we let $H_i = H[c_i]$, let (f_i, g_i) be the SP-reduction - which we know exists - from IS to H_i, and let $G_i = f_i(G, \epsilon/2)$. Here is an SP-reduction from IS to H; assume (G, ϵ) is the input to IS. The graph $G' = f(G, \epsilon)$ is created by taking a $MaxLoopGrab$ gadget (K, I) - with "parameters" $\epsilon' = \epsilon$ and $t = \sum_{c_i \in MaxLoops(H)} |V(G_i)|$ - and, for each $c_i \in MaxLoops(H)$, connecting every vertex in G_i to every vertex in K. This completes the construction of G'. Now, for $c_i \in MaxLoops(H)$, we let Y_i be the set of colourings from $H(G')$ where K is coloured monochrome c_i, and let Y_0 be all other colourings. We can map a sample $y \in H(G')$ to an IS sample as follows. If $y \in Y_0$ we return $\perp$. Otherwise, if $y \in Y_i$ then (in standard fashion) we return the independent set given by $g_i(G, \epsilon/2, y')$ where y' is y restricted to G_i. We now have to show formally that this reduction works. First, note that (3.4) is automatically satisfied because we constructed the $MaxLoopGrab$ gadget to ensure that $|Y_0|/\#H(G') \leq \epsilon'/4$. It remains to show that the distribution of IS samples within each Y_i are approximately uniform i.e. show (3.2). This is easy because, firstly, we know that for $i \in [1, r]$,

$$e^{-\epsilon/2}\frac{\#H_i(G_i)}{\#IS(G)} \leq |\{y' \in H_i(G_i) | g_i(G, \epsilon/2, y') = x\}| \leq e^{\epsilon/2}\frac{\#H_i(G_i)}{\#IS(G)}$$

Additionally, we know that $|Y_i| = \#H_i(G_i)\psi_i$ where

$$\psi_i = w_i^{|V'(H)|+1} \Delta_l^k \prod_{j \neq i} \#H_i(G_j)$$

so multiplying the above inequality by ψ_i transforms it into (3.2), because for all $x \in IS(G)$ and $i \in [1, r]$,

$$|\{y \in Y_i | g(G, \epsilon, y) = x\}| = \psi_i |\{y' \in H_i(G_i) | g_i(G, \epsilon/2, y') = x\}|$$

□

The following is the restricted, AP-reducibility version of the above result. (It is interesting to note that many of the 4-vertex H classified in Chapter 2 can automatically be shown to be $\equiv_{\mathsf{AP}}\#SAT$ using this corollary, by piggy-backing off the fact that most connected H on 3 or fewer vertices are already known to be $\equiv_{\mathsf{AP}}\#SAT$. See Appendix A.11 for more details.) In actual fact, this corollary may be restricted (in the sense that it can't cope with multiple non-isomorphic subgraphs) but in another sense it is slightly more general, because the condition that loops must be mutually non-adjacent has been dropped. (The non-adjacency requirement was required in the original lemma because we were explicitly aiming for a $\equiv_{\mathsf{SP}}SAT$-hardness result.)

Corollary 5.11 *Let $H = (V(H), E(H), w)$ be a non-trivial, non-bipartite graph presented in compact form, where $V(H) = \{c_1, ..., c_{|V(H)|}\}$. Let $MaxLoops(H)$ and Δ_l be defined as before. Without loss of generality, let c_1 be some colour in $MaxLoops(H)$, and let $H' = H[c_1]$. If $|MaxLoops(H)| = 1$ or (alternatively) for all $c_i \in MaxLoops(H)$, $H[c_i]$ is isomorphic to H', then $\#H' \leq_{\mathsf{AP}} \#H$.*

Proof. Let G be an input to $\#H'$. We build G' as follows. Let (K, I) be a $MaxLoopGrab$ gadget conditioned by $t = |V(G)|$ and $\epsilon' = 1$. We take a copy of G and connect every vertex in the copy of G to every vertex in K. We let full colourings be those where K is coloured monochrome c_i for some $c_i \in MaxLoops(H)$; in such cases H' is pointed out in G. Now, note that if $H[c_i]$ and $H[c_j]$ are isomorphic for two distinct $MaxLoops(H)$ colours c_i and c_j then the weights on c_i and c_j must be identical. To see this, suppose by way of contradiction that $H[c_i]$ and $H[c_j]$ are isomorphic but (wlog) $w_i < w_j$. Now, since $c_j \in H[c_j]$ we know that $H[c_j]$ has a universal loop at least of weight w_j. So $adj(c_i)$ simply must contain some looped colours other than c_i that become universal colours in $H[c_i]$. A vertex $c_k \in adj(c_i)$ becomes universal in $H[c_i]$ iff $adj(c_i) \subseteq adj(c_k)$. But $adj(c_i) \neq adj(c_k)$ - because we are working in compact form - so the only hope of this working is if $adj(c_i) \subset adj(c_k)$. However, this would mean $deg'(c_i) < deg'(c_k)$ and hence c_i would never have been in $MaxLoops(H)$ in the first place - contradiction!

So, if $r = |MaxLoops(H)|$ and we let $R = rw_1^{|V'(H)|+1}$, it follows that each H' colouring comes up $Z = R\Delta_l^k$ times a full colouring of G'. If Y_0 is the set of non-full colourings in G', we already know (from the construction of the $MaxLoopGrab$ gadget) that $|Y_0|/\Delta_l^k \leq 1/4$, so all we need to do to obtain an adequate approximation to $\#H'(G)$ is take our approximation of $\#H(G')$, divide by Z and round. $\square$

5.6 Bipartite H and $\equiv_{\mathsf{AP}}\#SAT$

5.6.1 Introduction

It is notable that we have been unable to identify a bipartite H which is $\equiv_{\mathsf{AP}}\#SAT$ (or $\equiv_{\mathsf{SP}}SAT$.) In this section we describe what appears to be a structural limitation of bipartite H which suggests that it may not be possible to find a bipartite H for which $\#LargeCut \leq_{\mathsf{AP}} \#H$. This lends weight to the idea (which we suspect is true) that there does not exist a bipartite H for which $\#SAT \leq_{\mathsf{AP}} \#H$. (This idea feeds into the discussion in Chapter 7, where partly on this basis of this idea we conjecture that there exists a "complexity gap" between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$, in which certain bipartite H - and non-bipartite H - lie.)

A sensible starting point is to identify the graph characteristics that our $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions require to work. As pointed out in Section 3.9, the root $\equiv_{\mathsf{AP}}\#SAT$ problems we have used are $\#LargeCut$, variants of $\#IS$, and to a much lesser extent the "automatically hard" Hell and Nešetřil graphs. Given that the Hell and Nešetřil graphs have not figured heavily in our work, it follows that the vast majority of our $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions have been rooted in the problems $\#LargeCut$ and $\#IS$. Admittedly this is a somewhat narrow base upon which to make assertions about the graph characteristics required to put a graph in $\equiv_{\mathsf{AP}}\#SAT$; for example, there may well be other $\equiv_{\mathsf{AP}}\#SAT$ root problems, not yet utilised, which exploit different graph characteristics[11].

5.6.2 The apparent significance of "symmetry"

However, as it stands, there appears to be some correlation between a graph H being $\equiv_{\mathsf{AP}}\#SAT$ and H being such that it is possible to develop "symmetrical"[12] maximal configurations in some gadget. For example, if the $\#LargeCut$ reduction used in the proof of Lemma 5.4 is studied, it is typical in that it builds a gadget (used to encode each vertex of the input to $\#LargeCut$) which is then equally dominated by *two symmetrical configurations*, and encodes edges in such a way that it is exponentially more likely that opposite (as opposed to same) symmetries are adjacent. Indeed, this property is integral to all the $\#LargeCut$ reductions we have demonstrated thus far. Furthermore, we see that (with the possible exception of the Hell and Nešetřil graphs) this "symmetrical" property underpins all our $\equiv_{\mathsf{AP}}\#SAT$ graphs, because we have a direct reduction, exploiting this property, from $\#LargeCut$ to $\#IS$.[13] So, in the absence of a fuller understanding about what makes a graph $\equiv_{\mathsf{AP}}\#SAT$, it is interesting to proceed on the basis that this "symmetrical" property is a (provisional!) proxy for $\equiv_{\mathsf{AP}}\#SAT$-hardness.

A pertinent question, therefore, is whether we can ever engineer a $\#LargeCut$ reduction, and locate some bipartite H, such that we can demonstrate $\#LargeCut\leq_{\mathsf{AP}}\#H$. If we are to adapt the *specific model* of $\#LargeCut$ reduction deployed throughout *this thesis* so that it works for some bipartite H, we have to come up with some encoding for edges of G and, more importantly, for vertices of G, where G is the input

[11] Indeed, we have yet to fully explore using the Hell and Nešetřil graphs themselves as root problems.

[12] Note that this should not be confused with the distinct notion of a *symmetric* bipartite graph, which is defined in Section 2.4.2, on page 91.

[13] To see this note that if we set IS as the graph H in the text of Lemma 5.4, the lemma actually proves $\#LargeCut\leq_{\mathsf{AP}}\#IS$.

to $\#LargeCut$. We consider what properties the gadget used to encode vertices of G should have. As usual, it must minimally be the case that, when coloured with H, the gadget has two uniquely maximal, symmetrical configurations. However, it is furthermore crucial that the two parts of the gadget which connect to the edge encodings lie on the *same side* of the gadget's bipartition. (To clarify, consider the vertex and edge encodings used in Lemma 5.4. The two crucial parts of the vertex encoding in that reduction are $L[.]$ and $R[.]$ respectively:- these are the parts that connect to the edge encodings, $S[.]$ and $S'[.]$. Unfortunately $L[.]$ and $R[.]$ in this instance do not lie on the same side of the vertex encoding's bipartition, so such a construction would not seem to be useful in the bipartite world, as we now show.)

This "same side" condition is necessary because, to make our specific model of $\#LargeCut$ reduction work, we have to not only be able to encode cut edges but also non-cut edges. To elaborate, if we think of $(LEFT, RIGHT) \subseteq V(G) \times V(G)$ representing a cut of G, we have to be able to encode not just $LEFT - RIGHT$ edges but also $LEFT - LEFT$ and $RIGHT - RIGHT$ edges. This is not possible if the important parts of the gadget (i.e. the parts that connect to the edge encodings) lie on opposite sides of the bipartition. (Of course, even assuming we can satisfy the "same side" requirement, we also require that we can encode edges in such a way that it is exponentially preferable for opposite symmetries - as opposed to like symmetries - to be adjacent.)

Single-vertex and multiple-vertex switching gadgets

We are sceptical as to whether it is possible, for any bipartite H, to build a "same side" gadget boasting two uniquely maximal, symmetrical configurations. Before elaborating, it is useful to explain the difference between what, in the context of $\#LargeCut$ reductions, we describe as "single-vertex switching gadgets" and "multiple-vertex switching gadgets". A single-vertex switching gadget is a gadget which (for a particular H) has two uniquely maximal, symmetrical configurations and where the critical parts of the gadget (i.e. the parts that connect to the edge encodings) are *single vertices*. For example, consider the graph H in Figure 5.5. We sketch a proof of $\#LargeCut \leq_{\mathsf{AP}} \#H$ using a single-vertex switching gadget to encode vertices, as follows:

Let (G, m) be the input to $\#LargeCut$. We now construct G', the input to $\#H$. For each vertex $u \in V(G)$ we introduce a single vertex $L[u]$, a single vertex $R[u]$ and connect them together, attaching a maxdeg gadget $I_L[u]$ to $L[u]$ and a maxdeg gadget $I_R[u]$ to $R[u]$. (Both the maxdeg gadgets have size k.) For each edge $\{u, v\} \in E(G)$, we introduce disjoint sets $S[uv]$ and $S'[uv]$ both of size p. To encode the edge, we attach $L[u]$ to every vertex in $S'[uv]$, every vertex in $S'[uv]$ to $R[v]$, $R[u]$ to every vertex in $S[uv]$ and every vertex in $S[uv]$ to $L[v]$. The point is, assuming $k >> p >> n$ each $(L[.], R[.])$ pair is exponentially likely to be coloured (r, b) or (b, r) because $|adj(r)||adj(b)| = 6 \times 6 = 36$ and all other colours in H (besides r and b) have degree less than 6. Furthermore, note that when (wlog) (r, b) is adjacent to (r, b), the $S[.], S'[.]$ sets are coloured in approximately $5^p 5^p = 25^p$ ways whereas if (r, b) is adjacent to (b, r) the sets are coloured in approximately $6^p 6^p = 36^p$ ways. All the ingredients for a successful $\#LargeCut$ reduction are therefore in place. □

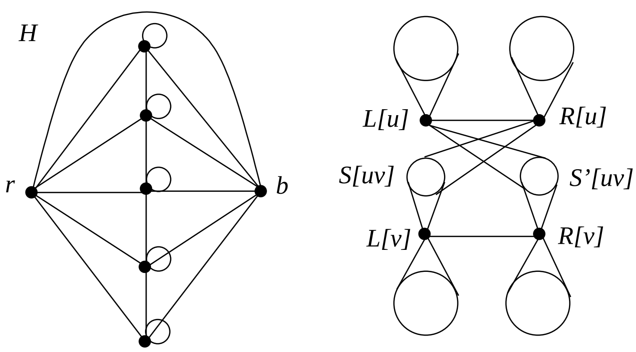


Figure 5.5: *On the right is how two vertices u, $v \in V(G)$ with an edge between them could be coded up in a reduction $\#LargeCut \leq_{\mathsf{AP}} \#H$*

We say that the above reduction used a single-vertex switching gadget because the $(L[.], R[.])$ pairs comprised *single* vertices. However, for many graphs H, single-vertex switching gadgets do not seem powerful enough to provide us with the two maximal, symmetrical configurations we need. Instead, we tend instead to use multiple-vertex switching gadgets. The most ubiquitous example of such a gadget (used, for example, in the proof of Lemma 5.4) is where we code up a vertex u as a pair of equal-size disjoint *sets* $L[u]$ and $R[u]$, and connect every vertex in $L[u]$ to every vertex in $R[u]$. Similarly, the elaborate gadget used to encode vertices in the original *1-wrench* (i.e. graph 9) proof is another example of a multiple-vertex switching gadget, because $B[.]$ and $B'[.]$ are both *sets* of vertices. In such cases, we are basing our reductions on the behaviour of *sets* of colours rather than single colours, where for many applications the "behaviour" of a set of colours is closely related to the mutual adjacency set of those colours.

We suspect that neither single-vertex *nor* multiple-vertex switching gadgets are ever possible in bipartite H, at least none that can be used in the context of a $\#LargeCut$ reduction. (Recall that, to be of use in such a reduction, both "sides" of the gadget need to occur on the same side of the gadget's bipartition.) In the context of bipartite H, we have not yet explored multiple-vertex switching gadgets as fully as we might, but we have looked at single-vertex switching gadgets in quite a bit more detail. In fact, we conjecture with some confidence that single-vertex switching gadgets are not possible in bipartite H.

More specifically, it seems that attempts to build single-vertex switching gadgets are thwarted by *monochrome* configurations. For example, suppose (for some unspecified graph H) we are trying to build a gadget where we want there to be two maximal, symmetrical configurations where the critical vertices (i.e. those that would connect to the edge encodings in a $\#LargeCut$ proof) are coloured (r, b) and (b, r) respectively. In practice, while we may be able to make (wlog) (r, b) maximal, this seems to prevent (b, r) from being maximal also. Alternatively, attempts at tweaking the gadgetry so that both configurations are equally dominant invariably seems to result in one of the monochrome configurations - (r, r) or (b, b) in this case - becoming maximal. And, clearly, monochrome configurations are no good whatsoever for $\#LargeCut$ reductions.

This apparent dominance of monochrome colourings seems particularly significant. The fact that symmetry seems to favour monochrome colourings would appear to be closely related to the fact that the mutual adjacency set of two distinct colours is never any larger (and often smaller) than the adjacency set of just one of those colours - see Observation 5.2 on page 163.

Continuing on this theme, it appears highly relevant that successful single-vertex switching gadgets seem to *structurally prohibit* significant monochrome colourings. For example, in Figure 5.5 observe that we have rendered it structurally impossible for a $(L[.], R[.])$ pair to be coloured either (r, r) or (b, b). Conversely, in a bipartite graph it is simply not possible to "outlaw" all colourings where two vertices are coloured with the same colour. (This is because you can always simply colour the relevant side of the bipartition monochrome in the desired colour, and the other side of the bipartition can be coloured monochrome in any colour that is adjacent to the desired colour.) This seems to be a critical fact. Curiously, a similar phenomenon may explain why we have had no luck finding a non-bipartite, fully looped H for which we can demonstrate $\#LargeCut \leq_{\mathsf{AP}} \#H$ using single-vertex switching gadgets. In particular, note that in this case you cannot outlaw all colourings where two vertices take the same colour because (owing to its loop) it is always legitimate to simply colour every vertex with that colour.

Multiple-vertex switching gadgets and closing comment

Even if the conjecture about single-vertex switching gadgets is true, it still remains a possibility that we might discover a multiple-vertex switching gadget for some bipartite H. Indeed, given that multiple-vertex switching gadgets seem more general and powerful than their single-vertex counterparts it seems reasonable to assume that, if there *is* some bipartite H for which $\#LargeCut \leq_{\mathsf{AP}} \#H$, the reduction will be via multiple-vertex switching gadgets. This is not an area that we have yet explored in great detail, but experiences thus far suggest that neither single nor multiple-vertex switching gadgets are possible in bipartite H.

On the basis of current knowledge, therefore, we remain sceptical as to whether there is any bipartite H which is $\equiv_{\mathsf{AP}}\#SAT$, acknowledging the various caveats listed throughout this section.

5.7 Partial H-colouring

5.7.1 Introduction

This section looks at some of the issues surrounding the variant of H-colouring known as "partial" H-colouring. We begin this section by defining what partial H-colouring is, and noting that every partial H-colouring problem is equivalent to some other standard H-colouring problem. In Section 5.7.3 we then move onto the main topic of this section and ask what the complexity - in terms of AP-reducibility - of $\#$*partial-H* (the partial version of $\#H$) is with respect to $\#H$. We conjecture that $\#$*partial-H* is always at least as hard as $\#H$, and this explains the inclusion of this section in this chapter: if we claim that there is some transformation which maintains or increases difficulty (from a complexity-theoretic viewpoint) then it is natural to particularly focus on those graphs that are already most difficult i.e. $\equiv_{\mathsf{AP}}\#SAT$ graphs. Finally, we look briefly at how the complexity of a problem $\#H$ is affected by applying the partial transformation multiple times.

5.7.2 Definitions

Essentially, a partial H-colouring of a graph G is similar to an H-colouring except that only a subset of $V(G)$ needs to be coloured. In other words, we are allowed to leave some vertices of G uncoloured. Formally, then, the problem is defined as follows.

Problem: $\#$*partial-H*

Instance: A graph G

Output: The number of functions $f : V(G) \to V(H) \cup \{x\}$ (where x is not in $V(H)$)

such that for all $\{u,v\} \in E(G)$, if $f(u) \neq x$ and $f(v) \neq x$, $\{f(u), f(v)\} \in E(H)$.

From the viewpoint of this thesis, an interesting aspect of partial H-colouring is that every partial H-colouring counting problem is equivalent to some other H-colouring counting problem. The relationship is as follows; we deal with non-bipartite H first. Assume $H = (V(H), E(H))$ is a connected, non-bipartite graph. Then $\#$*partial-H*(G) is equal to $\#H'(G)$ (over all inputs G) where $V(H') = V(H) \cup \{x\}$ (where x is a new colour not in $V(H)$) and

$$E(H') = E(H) \cup \Big\{\{x,x\}\Big\} \cup \bigcup_{c \in V(H)} \{c,x\}$$

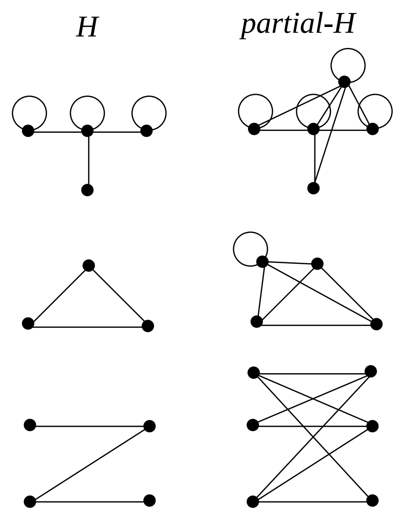


Figure 5.6: *Three examples of H and* partial-H

So, the partial version of a graph H is obtained by adding a universal colour to H (if H has no universal colour) or adding one to the weight of the existing universal colour, if H already has one. For bipartite $H = (V_L(H), V_R(H), E(H))$, $\#$*partial-H*$(G) = \#H'(G)$ (over all inputs G) where H' is defined as follows.

$$V_L(H') = V_L(H) \cup \{x_L\}$$

$$V_R(H') = V_R(H) \cup \{x_R\}$$

(where x_L and x_R are new colours) and

$$E(H') = E(H) \cup \Big\{\{x_L, x_R\}\Big\} \cup \bigcup_{c \in V_R(H)} \{x_L, c\} \cup \bigcup_{c \in V_L(H)} \{c, x_R\}$$

We henceforth use the term *partial transformation* quite a lot; its meaning is amply demonstrated by noting that *partial-H* is the graph obtained by applying the *partial transformation* to H.

(Though we do not look at this issue any further, the natural analogue of the partial transformation when applied to disconnected H is to apply the partial transformation separately to each of its components, rather than - for example - simply adding one universal colour and thus creating a connected graph. So the partial transformation preserves connected/disconnected status.)

Given that a significant part of this section is devoted to the question of how the partial transformation ties in with $\equiv_{\mathsf{AP}}\#SAT$-hardness, we do not discuss any further the partial transformation as pertaining to bipartite H. Broadly speaking, however, many of the same principles apply, and (for example) the soon-to-be-introduced "absorption" technique is utilised with respect to bipartite H in Lemma 7.7 of Chapter 7.

Now we have defined $\#$*partial-H* in terms of $\#H$, it is helpful to introduce notation for the iterated application of the partial transformation. This has a natural definition; if *partial-H* is the graph obtained by applying the partial transformation to H, we define *partial*^{i}*-H* ($i \in \mathbb{N}^+$) to be the graph obtained by applying the partial transformation to *partial*$^{i-1}$*-H* , where *partial*^{0}*-H* is simply the graph H itself. Hence, *partial*^{1}*-H* is the same graph as *partial-H*.

5.7.3 Is #partial-H always at least as hard as $\#H$?

The question we are most concerned with is this: what is the complexity of approximately counting partial H-colourings, compared to the complexity of approximately counting H-colourings? (Observe that, because the "partial transformation" is simply a way of *describing* a structural relationship between two graphs *partial-H* and H, there is no reduction underpinning the transformation and hence we do not start with any a priori knowledge about the complexity relationship between $\#$*partial-H* and $\#H$.[14])

We suspect that $\#H \leq_{\mathsf{AP}} \#$*partial-H*, for all H, and this section looks at some of the evidence we have begun to collect in support of this. (We call this the "partial conjecture".)

Preliminaries

Before tackling the "partial conjecture", it is worth pointing out that, unless $\#BIS$ is AP-interreducible with $\#SAT$, $\#$*partial-H* is sometimes *strictly harder* than $\#H$. (Hence, this proves that the relationship $\#$*partial-H*$\leq_{\mathsf{AP}} \#H$ does not in general[15] hold.) This is because we have identified H for which $\#H \equiv_{\mathsf{AP}} \#BIS$ but $\#$*partial-H*$\equiv_{\mathsf{AP}} \#SAT$, as we now show.

Let H be the *2-wrench* i.e. graph 21. (See also the top left graph in Figure 5.6.) We know $\#H \equiv_{\mathsf{AP}} \#BIS$ from [8]. However, if we let H' be the graph *partial-H*, we see that $MaxPairs(H') = \{(F(H'), V(H')), (V(H'), F(H'))\}$. This is because $|F(H')||V(H')| = 2 \times 5 = 10$ whilst the nearest competing configuration in $GoodPairs(H')$ comes up $|C||C| = 3 \times 3 = 9$ times, where C consists of the two universal colours plus one of the non-universal looped colours. Hence, $\#H' \equiv_{\mathsf{AP}} \#SAT$ by Lemma 5.1. As another example, suppose H is the graph P_9^* i.e. the looped path on 9 vertices. We know $\#H \equiv_{\mathsf{AP}} \#BIS$ from Lemma 2.13 (and earlier from [8]). If we let H' be the graph *partial-H*, we note that $|F(H')||V(H')| = 1 \times 10 = 10$, whereas the nearest competing configuration in $GoodPairs(H')$ contributes at most $3 \times 3 = 9$. Hence $\#H' \equiv_{\mathsf{AP}} \#SAT$. Both these examples demonstrate the informal observation that applying the partial transformation to sufficiently "sparse" graphs often pushes the graph

[14] Contrast this with, say, the fact that $\#bi(H) \leq_{\mathsf{AP}} \#H$ because $bi(H)$ and H are related by a specific AP-reduction i.e. bipartisation.

[15] Of course, $\#$*partial-H*$\leq_{\mathsf{AP}} \#H$ will hold for some graphs and families of graphs. For example, we know *2-WR* remains in $\#BIS$ irrespective of the weight on its universal loop - see Lemma 2.10. As another example, observe that the trivial H remain trivial under the partial transformation.

into $\equiv_{\mathsf{AP}}\#SAT$. That is, if H does not have a universal loop, but the size of $|V(H)|$ is sufficiently large compared to the size of looped cliques (and other structures that might give rise to competing configurations in $GoodPairs$), $|F(\textit{partial-H})||V(\textit{partial-H})|$ often emerges dominant.

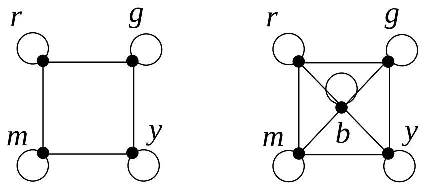
So, we now turn to the question of the partial conjecture. Most of this section is taken up by a discussion of the "absorption" reduction technique and, separately, the effect of the partial transformation on graphs that are $\equiv_{\mathsf{AP}}\#SAT$, but it is worth taking a moment to discuss the impact of the transformation on graphs that are $\equiv_{\mathsf{SP}}BIS$. In terms of the sampling domain, we see that Theorem 4.1 automatically proves the partial conjecture for $\equiv_{\mathsf{SP}}BIS$ graphs, because the partial transformation cannot make a non-trivial graph trivial and we know from the Theorem that no non-trivial, connected H are easier (in the SP sense) than BIS. As we have pointed out before, if (as seems plausible) there exists a "counting" version of Theorem 4.1, then the partial conjecture also holds in the counting sense i.e. for those H where $\#H\equiv_{\mathsf{AP}}\#BIS$.

Now we move onto a generic reduction technique which can sometimes be used to prove $\#H\leq_{\mathsf{AP}}\#$*partial-H* for a given H.

Specific reduction technique: absorption

For a given graph H, it is always worth investigating whether a *direct* reduction is possible from $\#H$ to $\#$*partial-H*. The fact that *partial-H* and H are structurally fairly similar suggests that, in certain cases, this could be a promising approach. A direct reduction also has the advantage of confirming the $\#H\leq_{\mathsf{AP}}\#$*partial-H* relationship even if the complexity of one or both graphs is unknown, as in the example we demonstrate shortly.

The direct reduction technique we have had most success with is "absorption"; for reasons we explain shortly, however, this only works with a very limited range of H. At its simplest - when H does not already have a universal loop - absorption is the tactic of coding up vertices of the graph G (where G is the input to $\#H$) in such a way that, when coloured with *partial-H*, the part of the vertex gadget which effectively determines the vertex's "colour" is exponentially likely to be bicoloured with the universal colour plus a non-universal colour. Given that the universal colour can be adjacent to everything, this makes the gadget inherit the behaviour of the non-universal colour. In other words, the extra universal colour has been absorbed by gadgetry. As an example, let H be the unclassified graph C_4^* - a sketched reduction from $\#H$ to $\#$*partial-H* is as follows.[16]



Let G be the input to $\#H$. We build G', the input to $\#$*partial-H*, as follows. For each vertex $u \in V(G)$, we introduce two disjoint sets of vertices $L[u]$ and $R[u]$, of size p and q respectively, where $q = 2p$. We connect every vertex in $L[u]$ to every vertex in $R[u]$. For each edge $\{u, v\}$ we connect every vertex in $L[u]$ to every vertex in $L[v]$. That completes the construction of G'. Now, if p and q are sufficiently large (say $p = n^3$), each $(L[.], R[.])$ pair is exponentially likely to be coloured either $(rb, rbgm)$, $(gb, gbry)$, $(yb, ybmg)$ or $(mb, mbry)$. To see why this is, note that each of the configurations listed comes up approximately $2^p 4^q = 32^p$ times, whereas competing configurations are inferior: $(b, rbmgy)$ comes up only $1^p 5^q = 25^p$ times and (brg, brg) (for example) comes up only $3^p 3^q = 27^p$ times. Now, note that because each edge is coded up by connecting respective $L[.]$ sets, the four dominant configurations act like r, g, y, m respectively[17] ; in effect we have "absorbed" the universal colour b. $\square$

There are two possible extensions to this technique that we have not yet investigated thoroughly. First, what if H already has a universal loop e.g. could we use the technique to show (for some graph H) that $\#$*partial^i-H*$\leq_{\mathsf{AP}}\#$*partial^{i+1}-H* for $i \geq 1$? Sec-

[16]We do not know the complexity of the partial version of C_4^*, either.

[17]To complete the reduction we would divide by $(\nu(p, 2)4^q)^n$ and round i.e. the standard technique.

ondly, could we use the absorption technique "in reverse" i.e. use it to show that $\#$*partial*$^{i+1}$-*H*$\leq_{\mathsf{AP}}\#$*partial*^{i}-*H* (for $i \geq 1$)? We have seen that there are complexity-related reasons for the second extension not always being possible - i.e. we know that the partial transformation sometimes makes graphs harder - but there may be classes of graphs for which both extensions are possible. In this regard, an interesting test bed for the the two extensions could be to try and prove (for H equal to C_4^* and $i \geq 1$) that $\#$*partial*^{1}-*H*$\equiv_{\mathsf{AP}}\#$*partial*^{i}-*H*. We have begun looking into this - and suspect that the reduction described above can be "tweaked" to support these extensions - but do not reproduce details here.

Comment on the absorption technique

The usefulness of the absorption technique (for proving $\#H\leq_{\mathsf{AP}}\#$*partial-H*) is, in practice, limited to graphs with highly specific characteristics. Observe that, for the technique to work, the gadget we use to encode vertices must have $|V(H)|$ maximal configurations, each one behaving like a specific colour from $V(H)$. However, as this thesis demonstrates, gadgets (in general) tend to be highly sensitive to colour characteristics such as degree, local graph structure and so on. As such, graphs H that the absorption technique can be successfully applied to tend to have highly homogeneous structures both in terms of colour degree (i.e. they tend to be regular, if universal loops are ignored) and local graph structure. This explains why the graphs corresponding to the *bi-q-col* problems (see proof of Lemma 7.5 in Chapter 7) lend themselves so well to the absorption technique; they are completely regular and all pairs of colours on the same side of the bipartition are indistinguishable.

5.7.4 $\equiv_{\mathsf{AP}}\#SAT$ and the partial conjecture

The fact that $\equiv_{\mathsf{AP}}\#SAT$ is the hardest approximation complexity class (for problems in $\#P$) makes it interesting from the point of view of the partial conjecture. Clearly, if the partial conjecture is true, then for graphs H where $\#H\equiv_{\mathsf{AP}}\#SAT$, it must be the case that $\#$*partial-H* is also $\equiv_{\mathsf{AP}}\#SAT$. In other words, if the partial conjecture is true then the family of $\equiv_{\mathsf{AP}}\#SAT$ graphs is closed under the partial transformation. Hence, if we find a $\equiv_{\mathsf{AP}}\#SAT$ graph that becomes *easier* than $\equiv_{\mathsf{AP}}\#SAT$ following the application of the partial transformation, we have disproven the whole conjecture. Conversely, if $\equiv_{\mathsf{AP}}\#SAT$-hardness of H-colouring *is* closed under the partial transformation, then this would add weight to the partial conjecture and also be a significant result in its own right.

Maintaining consistency with the overarching partial conjecture, we suspect that the $\equiv_{\mathsf{AP}}\#SAT$-hardness of H-colouring is indeed closed under the partial transformation. (We henceforth call this subset of the partial conjecture the "$\equiv_{\mathsf{AP}}\#SAT$-closure" conjecture.) In the rest of this subsection we examine some of the evidence we have unearthed in support of the $\equiv_{\mathsf{AP}}\#SAT$-closure conjecture. A sensible way of building up such evidence is to consider in turn each of the $\equiv_{\mathsf{AP}}\#SAT$-hardness lemmas from this chapter (plus the "Hell and Nešetřil" graphs), and ask the following question: if $\#H$ is shown to be $\equiv_{\mathsf{AP}}\#SAT$ by this lemma, does it follow that this lemma - or some other lemma - shows *#partial-H* to be $\equiv_{\mathsf{AP}}\#SAT$? If it turns out that *#partial-H* is $\equiv_{\mathsf{AP}}\#SAT$ by the *same* lemma, we say that the lemma domain is closed under the partial transformation. As we see shortly, a lemma can have the $\equiv_{\mathsf{AP}}\#SAT$-closure property even if the lemma's domain is not closed under the partial transformation. Finally, we define the "$\equiv_{\mathsf{SP}}SAT$-closure" conjecture simply to be the sampling version of the $\equiv_{\mathsf{AP}}\#SAT$-closure conjecture. (We prefer to work with respect to AP-reducibility where possible, however.)

"Hell and Nešetřil" graphs

In Section 2.1.2 (see page 26) we note that the "Hell and Nešetřil" graphs - non-bipartite, unlooped - are $\equiv_{\mathsf{AP}}\#SAT$. Now, whilst the set of Hell and Nešetřil graphs is not closed under the partial transformation - because of the addition of a loop - we nonetheless think that the $\equiv_{\mathsf{AP}}\#SAT$-closure conjecture may well hold for these graphs. To see why this is, note that applying the partial transformation to a Hell and Nešetřil graph produces a graph H' which fits into the domain of Lemma 5.5. This

Lemma proves that $SAT \leq_{\mathsf{SP}} H'$. Given that this is a sampling result it is not quite what we need; recall our earlier discussions that approximate sampling might be harder than approximate counting. However, by the same token we suspect that, in instances (such as this one) where we use the SP-reduction to assist us in navigating through a glut of possible hardness reductions, the resulting sampling results constitute some evidence that an AP-equivalent exists. So, while we do not yet have firm proof that the $\equiv_{\mathsf{AP}}\#SAT$-closure conjecture holds for Hell and Nešetřil graphs, we do think it likely that the conjecture holds. Note that, technically speaking, we can't actually say that the $\equiv_{\mathsf{SP}}SAT$-closure conjecture holds either because, as mentioned in Section 3.9, we don't actually know whether $SAT \leq_{\mathsf{SP}} H$ (for Hell and Nešetřil graphs H) in the first place. However, the $\equiv_{\mathsf{SP}}SAT$-closure conjecture clearly applies in spirit because, for Hell and Nešetřil graphs H, we know that sampling *partial-H* is intractable.

Lemma 5.1 (The "$(F(H), V(H)) - (V(H), F(H))$ dominance" lemma, page 160)

Arguably the most important result in this current section is as follows.

Observation 5.12 *Suppose $\#H$ is known to be $\equiv_{AP}\#SAT$ by Lemma 5.1. Then* $\#$partial-H$\equiv_{AP}\#SAT$*, also by Lemma 5.1.*

This is quite a significant result. What it means is that if

$$MaxPairs(H) = \{(F(H), V(H)), (V(H), F(H))\}$$

(thus establishing $\#H \equiv_{\mathsf{AP}} \#SAT$ by Lemma 5.1), then *irrespective of how much extra weight is loaded onto the universal colour* to produce a new graph H', the $MaxPairs$ of H' are $\{(F(H'), V(H')), (V(H'), F(H'))\}$. The proof is as follows.

Proof. Let H be any graph known to be $\equiv_{\mathsf{AP}}\#SAT$ by Lemma 5.1. We assume that $|F(H)| = k$ where $k \geq 1$. Now, by Lemma 5.1 we know that $MaxPairs(H)$ contains only the configurations $(F(H), V(H))$ and $(V(H), F(H))$. So, if we take any configuration $(L, R) \in GoodPairs(H)$ distinct from $(F(H), V(H))$ and $(V(H), F(H))$ we know that $|F(H)||V(H)| > |L||R|$. Next, we consider the structure of (L, R).

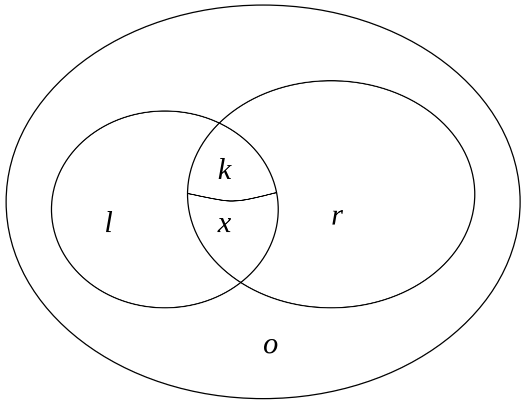


Figure 5.7: *Schematic representation of rival configuration in the context of the graph H, from Observation 5.12*

We divide vertices in L into three disjoint categories: vertices which are *not* also in R, vertices which *are* in R also but which are *not* universal colours, and universal colours. If we let the size of these three categories be l, x, k respectively it follows that $|L| = l + x + k$. Similarly, we see that $|R| = r + x + k$ where r is the number of colours in R that are not also in L. Finally, we let o equal the number of colours in H which are neither in L nor R. It follows that $|V(H)| = k + x + l + r + o$. (See Figure 5.7.) Now, given that $|F(H)||V(H)| > |L||R|$ we know that

$$k(k + x + l + r + o) > (l + x + k)(r + x + k)$$

Simplifying this inequality gives

$$ko > lr + lx + xr + xk + x^2 \tag{5.6}$$

We want to show that, if we add another universal loop y - i.e. apply the partial transformation - $(F(H) \cup \{y\}, V(H) \cup \{y\})$ and $(V(H) \cup \{y\}, F(H) \cup \{y\})$ continue to dominate over $(L \cup \{y\}, R \cup \{y\})$ and $(R \cup \{y\}, L \cup \{y\})$. Let us suppose, by way of contradiction, that after adding y, $(F(H) \cup \{y\}, V(H) \cup \{y\})$ is not dominant. In other words,

$$(k + 1)(k + 1 + x + l + r + o) \leq (l + x + k + 1)(r + x + k + 1)$$

If we tidy this up we get

$$ko + o \leq lr + lx + xr + x^2 + xk + x \tag{5.7}$$

If we study (5.6) and (5.7) we see that a necessary (but not necessarily sufficient) condition for the second inequality to apply is $o < x$. However, if $o < x$ it means that the contribution of $(F(H), V(H))$ in H was bounded above by $k(k+l+r+2x)$. So to have ever been dominant in the first place we would have needed $k(k+l+r+2x) > (l+x+k)(r+x+k)$. Simplifying this gives

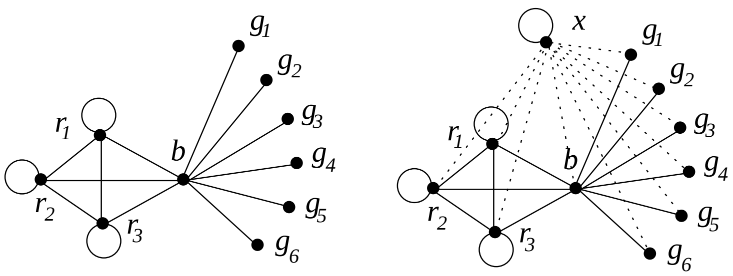
$$0 > lr + lx + xr + x^2$$

Now, if $o < x$ it follows that $x \geq 1$ (because $0 \leq o$), and hence the above inequality cannot be solved. Hence, if $o < x$ then $(F(H), V(H))$ would not have been dominant in the first place, contradiction! □

The other $\equiv_{\mathsf{AP}}\#SAT$-hardness results in this chapter respond to the partial transformation in the following ways.

Lemma 5.4 (The "$(S,T)-(T,S)$-dominance" lemma, page 164)

We have *not* been able to verify whether $\#$*partial-H*$\equiv_{\mathsf{AP}}\#SAT$ whenever $\#H$ is shown to be $\equiv_{\mathsf{AP}}\#SAT$ by Lemma 5.4. We know that the lemma's domain is *not* closed under the partial transformation, but we do not know whether the lemma has the $\equiv_{\mathsf{AP}}\#SAT$-closure property irrespective of this. To see that the domain is not closed, consider the following graphs.



The graph on the left, H, is $\equiv_{\mathsf{AP}}\#SAT$ by virtue of Lemma 5.4. To see this, note that $MaxPairs(H) = \{(S,T),(T,S)\}$ and $H[S \cup T] \neq \emptyset$, where $S = \{r_1, r_2, r_3\}$ and $T = \{r_1, r_2, r_3, b\}$. Now, consider the graph on the right, *partial-H*. (We have shown the new edges added as dotted lines.) If we let H' be the graph *partial-H*, we see that

$MaxPairs(H') = \{(S',T'),(T',S'),(P,Q),(Q,P)\}$ where $S' = S \cup \{x\}$, $T' = T \cup \{x\}$, $P = \{b,x\}$ and $Q = \{r_1, r_2, r_3, x, g_1, ..., g_6\}$. This is because $|S'||T'| = 4 \times 5 = 20$ and $|P||Q| = 2 \times 10 = 20$. In other words, adding the universal loop has allowed another configuration to "catch up" and equalise itself with the previously maximal configurations, meaning Lemma 5.4 no longer applies in this instance.

Of course, this does not mean to say that (for this *specific* H) *partial-H* is not $\equiv_{\mathsf{AP}} \#SAT$ - we could show this using Corollary 5.8 for example[18] - but just that the domain of Lemma 5.4 is not closed under partial transformation, and that it is not clear whether (in general) those graphs which fall out of the lemma's domain fall into the domain of some other $\equiv_{\mathsf{AP}} \#SAT$-hardness lemma. (It could be interesting to pursue this question as further work.)

It is worth dwelling on the above counter-example a little longer, and in particular on the size of the set $GoodPairsSimAdj(H)$, a set we define as being the subset of $GoodPairs(H)$ formed by taking all pairs $(S,T) \in GoodPairs(H)$ for which $H[S \cup T] \neq \emptyset$. In the example, $|GoodPairsSimAdj(H)| < |GoodPairsSimAdj(\textit{partial-H})|$ - that is, the introduction of the universal loop actually *expands* the domain of configurations that Lemma 5.4 can consider. Conversely, in cases where H already has a universal loop, $|GoodPairsSimAdj(H)| = |GoodPairsSimAdj(\textit{partial-H})|$, with there being a natural one-to-one mapping between elements of $GoodPairsSimAdj(H)$ and $GoodPairsSimAdj(\textit{partial-H})$. To see why this is, note first that where H has a universal loop, $GoodPairsSimAdj(H) = GoodPairs(H)$, a fact that follows because the universal loop is always a subset of $H[S \cup T]$ for all $(S,T) \in GoodPairs(H)$. Furthermore, where H has a universal loop, $|GoodPairs(H)| = |GoodPairs(\textit{partial-H})|$, because there exists a natural bijection between pairs in $GoodPairs(H)$ and $GoodPairs(\textit{partial-H})$. The bijection is as follows. Each $(S,T) \in GoodPairs(H)$ maps to $(S \cup \{x\}, T \cup \{x\}) \in GoodPairs(\textit{partial-H})$ where x is the new universal loop in *partial-H*. What is the significance of this? Well, when H already has a universal loop, no "new" configurations

[18] To see this, note that - if we plug *partial-H* into Corollary 5.8 - the graph H' (as referred to in the text of Corollary 5.8) is the IS graph with weight 4 on its looped vertex, which (by Lemma 2.3 on page 40) we already know to be $\equiv_{\mathsf{AP}} \#SAT$.

are introduced by increasing the weight on the universal loop.

It could be interesting, therefore, to determine whether there exists a graph H with $F(H) \neq \emptyset$ for which Lemma 5.4 applies but where the lemma does not apply to *partial-H*.

Lemma 5.5 (The "only loops are universal loops" lemma, page 167)

It is easy to see that the domain of Lemma 5.5 is closed under the partial transformation, so the $\equiv_{\mathsf{SP}} SAT$-closure conjecture is upheld by this lemma.

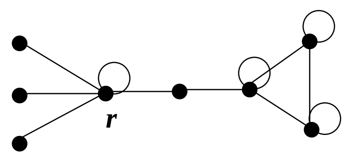
Lemma 5.7 (The "clique-grabbing" lemma, page 176)

The domain of Lemma 5.7 (and Corollary 5.8) is closed under the partial transformation. Using the definitions from that lemma, if the maximum looped clique size in H is m, the maximum looped clique size in *partial-H* is $m + 1$. There is an obvious bijection between the maximum-size looped cliques in H and those in *partial-H*; if x is the new universal loop in *partial-H*, a maximum clique C in H corresponds to the maximum clique $C \cup \{x\}$ in *partial-H*. To see that the lemma domain is closed, let C be any size-m looped clique from H for which $|H[C]| > m$. We know $H[C] \subseteq$ *partial-H*$[C \cup \{x\}]$ but because $x \notin H[C]$ and $x \in$ *partial-H*$[C \cup \{x\}]$ it follows that $H[C] \subset$ *partial-H*$[C \cup \{x\}]$. So $|$*partial-H*$[C \cup \{x\}]| > m + 1$. $\square$

The fact that Corollary 5.8 is not (strictly speaking) a $\equiv_{\mathsf{AP}} \#SAT$-hardness result makes it curious to reason about from the viewpoint of the $\equiv_{\mathsf{AP}} \#SAT$-closure conjecture. For example, suppose we have used the Corollary to prove $\#H \equiv_{\mathsf{AP}} \#SAT$ i.e. by noting that H' (the subgraph of H as defined in the text of the Corollary) is itself $\equiv_{\mathsf{AP}} \#SAT$. Using our earlier comment about natural bijections between maximum looped cliques in H and *partial-H* we note that using the Corollary with *partial-H* gives us the result #*partial-H'*$\leq_{\mathsf{AP}}$#*partial-H*. Therefore, if $\#H'$ stays in$\equiv_{\mathsf{AP}} \#SAT$ under the partial transformation, so too does $\#H$, which is interesting in the sense that it links the behaviour of H under the partial transformation to the behaviour of one of its subgraphs under the partial transformation.

Lemma 5.10 (The "no adjacent loops" lemma, page 188)

It is not too difficult to see that the domain of Lemma 5.10 is *not*, in general, closed under the partial transformation.[19] This is because the addition of a universal loop robs existing loops of their "isolated" property. Moreover, it is not clear (in general) whether *partial-H* is provably $\equiv_{\mathsf{AP}}\#SAT$ or $\equiv_{\mathsf{SP}}SAT$ by any of our other lemmas. For example, the following graph H (displayed in expanded form) is $\equiv_{\mathsf{SP}}SAT$ by Lemma 5.10, but *partial-H* does not fit into any of the $\equiv_{\mathsf{AP}}\#SAT$ or $\equiv_{\mathsf{SP}}SAT$ lemmas in this chapter.



Given that *partial-H* does not fit into any of the $\equiv_{\mathsf{AP}}\#SAT$ or $\equiv_{\mathsf{SP}}SAT$ lemmas in this chapter, the above graph also serves as an example that *partial-H* does not automatically fall into $\equiv_{\mathsf{AP}}\#SAT$ (or, for that matter, $\equiv_{\mathsf{SP}}SAT$) for graphs H already indirectly proven to be $\equiv_{\mathsf{AP}}\#SAT$ via Corollary 5.11. (That is, because $H[r]$ is a weighted independent set, and thus $\equiv_{\mathsf{AP}}\#SAT$, we know by Corollary 5.11 that $\#H\equiv_{\mathsf{AP}}\#SAT$. However, *partial-H* does not fit into any of this chapter's lemmas.)

5.7.5 Graphs that eventually fall into $\equiv_{\mathsf{AP}}\#SAT$ under the partial transformation

An interesting contribution to the "$\equiv_{\mathsf{AP}}\#SAT$-closure" and (more generally) the partial conjecture question is to identify classes of graphs that *eventually* fall provably into $\equiv_{\mathsf{AP}}\#SAT$ if the partial transformation is applied sufficiently many times. The existence of such classes of graphs supports the partial conjecture, because it reinforces

[19] In the special case - where the only loops in H are universal loops - the domain of the lemma *is*, however, closed.

the idea that increasing the weight on the universal loop pushes graphs in a harder or at-least-as-hard direction. Moreover, if (say) #*partial*^{i}-*H* is the first graph to provably[20] be $\equiv_{\mathsf{AP}}\#SAT$, and the $\equiv_{\mathsf{AP}}\#SAT$-hardness reduction used in that proof is one that has the $\equiv_{\mathsf{AP}}\#SAT$-closure property, it follows that the graph becomes "trapped" in $\equiv_{\mathsf{AP}}\#SAT$ for all higher weightings on the universal loop.

For example, consider a non-bipartite graph H (in expanded form) in which

$$GoodPairs(H) = \{(F(H), V(H)), (V(H), F(H)), (C_1, C_1), ..., (C_p, C_p)\}$$

where $C_1, ..., C_p$ are maximum-size looped cliques from H. A good example of such a graph is *3-WR*, or any weighted variant of it. Let k be the weight on the universal loop, let $c = |C_1| - k$ and let $r = |V(H)| - k$. It follows that, if $2c < r$, there must exist some value of k such that for k and higher values $\#H \equiv_{\mathsf{AP}} \#SAT$. In other words, there must come a point where applying the partial transformation iteratively tips the graph into $\equiv_{\mathsf{AP}}\#SAT$. To see this, note that the contribution of a configuration (C_i, C_i) is $(c+k)^2$, whilst the contribution of $(F(H), V(H))$ and $(V(H), F(H))$ is $k(k+r)$. So we want to know if there is a value of k for which

$$(c+k)^2 < k(k+r)$$

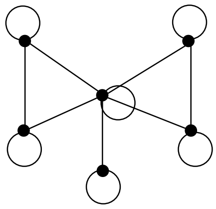
A sufficient condition for satisfying the the above inequality is

$$c^2 + 2ck < kr$$

Crucially, the c^2 does not increase with increasing k. So, if $r > 2c$, then the above inequality is inevitably satisfied for large enough k. Let k' represent this "large enough" value of k, and let H' be the graph with this weight on its universal loop. It follows that $(F(H'), V(H'))$ and $(V(H'), F(H'))$ are now the sole maximal configurations. Hence, Lemma 5.1 eventually applies, and combining this with Observation 5.12 shows that the graph remains in $\equiv_{\mathsf{AP}}\#SAT$ for all weightings on the universal loop equal to or higher than k'.

[20] The word "provably" has been used to reflect the fact that versions of the graph with less weighting on the universal loop may also be $\equiv_{\mathsf{AP}}\#SAT$, but that we have been unable to prove this fact.

The following is an example of such a graph, where $c = 2$ and $r = 5$. The fourth application of the partial transformation (which makes the weight on the universal loop 5) pushes the graph into $\equiv_{\mathsf{AP}} \#SAT$.



It is perhaps worth noting that we do not know the complexity of the above untransformed graph as it stands above (i.e. with weight 1 on its universal loop.) It is one of many graphs that have thus far evaded classification as either being $\equiv_{\mathsf{AP}} \#SAT$ or $\equiv_{\mathsf{AP}} \#BIS$-easy. (This topic - of graphs evading classification - is discussed in greater depth in Chapter 7.) It is not clear what bearing, if any, the fact that a graph can *eventually* be pushed into $\equiv_{\mathsf{AP}} \#SAT$ through repeated application of the partial transformation has on the complexity of the original, untransformed graph. Finally, it is clear that some graphs never get any harder, irrespective of how many times the partial transformation is applied: the graph *2-WR* is a principal example of this (see Lemma 2.10).

Chapter 6

Disconnected H

6.1 Introduction

We suggested in Chapter 2 that determining the complexity of a given H becomes problematic when H is disconnected, and this is why we have generally assumed that H is connected. In this chapter we explore some of the issues pertaining to disconnected H, and describe a number of results which are of assistance in determining their complexity.

We begin, in Section 6.2 (*Basic domination results*), by showing that the complexity of disconnected H is closely linked to the complexity of its dominant components, and (separately) we demonstrate a tighter upper bound on $\#H(G)$ than the conventional $\#H(G) \leq |V(H)|^n$. We also provide a simple utility lemma which provides a non-trivial upper bound for $\#H(G)$ when G has a perfect matching. Section 6.3 (*Coping with small additive quantities in counting equations*) provides us with a general result that allows us to formally disregard the single vertex and K_2 components that often litter disconnected H. We then use this result (alongside an observation by Jerrum) to categorise all the disconnected 4-vertex H graphs, where previously we had been reliant on ad-hoc reductions (as used in Chapter 2) to determine their complexity.

For this chapter only we drop the assumption that H is connected. (However, as in the rest of the paper, we stress that graphs G are presumed connected unless otherwise stated. Hence phrases like "for all G" actually mean "for all connected G".) We continue with the standard definition that n refers to $|V(G)|$. Finally, before proceeding any further, it is worth noting the following simple observation, which is really too elementary to be called a "result", but is important to be aware of nonetheless. Namely, observe that the complexity of a disconnected H is no harder than its hardest component. To see this, observe that if H has k components $H_1, ..., H_k$ and (wlog) $\#H_i \leq_{\mathsf{AP}} \#H_1$ for all H_i, we can use an $\#H_1$ oracle k times to compute each $\#H_i$ and simply add the returned values together, in each case using the same accuracy parameter that we have to compute $\#H$ to.

6.2 Basic domination results

In this Section we focus on three observations. The first, encapsulated by Lemmas 6.1 and 6.2, demonstrates that if one component exponentially dominates over another in a disconnected H graph, then the complexity of the graph is equivalent to the complexity of the dominant component. The second observation, covered by Lemma 6.3, shows that for a graph H there is an upper bound on its ability to colour a graph G which is closely related to the maximum degree of H. The third observation (Observation 6.5) provides, for G with perfect matchings, a non-trivial upper bound on $\#H(G)$ which is a function of the number of "directed" edges in H. The second and third observations are relevant to this chapter because, when attempting to classify a disconnected H, a strategy we often use (as in Lemma 6.1) is to try and prove that one component exponentially dominates over the other(s). Hence, the tighter the bounds we have on the contribution of the individual components, the more likely that this line of attack will be successful.

6.2.1 Where one component exponentially dominates over another

Lemma 6.1 *Let H be a disconnected graph with two components, H_1 and H_2. Suppose there exist constants k_1, k_2 such that $k_1 > k_2$ and, for all sufficiently large G, $\#H_1(G) \geq k_1^n$ and $\#H_2(G) \leq k_2^n$. Then $\#H \equiv_{\mathsf{AP}} \#H_1$.*

Proof. We first show that $\#H \leq_{\mathsf{AP}} \#H_1$. Let (G, ϵ) be an input to $\#H$. Our argument is that, because $\#H(G) = \#H_1(G) + \#H_2(G)$ and H_1 exponentially dominates over H_2, returning an approximation to $\#H_1(G)$ is adequate for most values of ϵ above a certain exponentially small (in n) threshold, and if ϵ is below that threshold then we have enough freedom to compute $\#H(G)$ exactly using brute force. So first we inspect ϵ. If $\epsilon < 2(k_2/k_1)^n$ then we compute $\#H(G)$ exactly by exhaustively considering all possible mappings from the colours of H to the vertices of G and discarding invalid colourings. This takes in the order of $|V(H_1)|^n + |V(H_2)|^n$ steps which is acceptable because, though exponential, is still only polynomial in ϵ^{-1}. Alternatively, if $\epsilon \geq 2(k_2/k_1)^n$ then we simply return $\#\hat{H_1}(G)$, which is the result of calling our $\#H_1$ oracle with input $(G, \epsilon/2)$. To see that this is adequate, recall that we need $\#\hat{H_1}(G)/\#H(G)$ to be in the range $[e^{-\epsilon}, e^{\epsilon}]$. The upper bound is trivially satisfied. To satisfy the lower bound, we require

$$e^{-\epsilon} \leq \frac{\#\hat{H_1}(G)}{\#H_1(G) + \#H_2(G)}$$

We know $e^{-\epsilon/2}\#H_1(G) \leq \#\hat{H_1}(G)$ so the above is satisfied if we show

$$e^{-\epsilon/2} \leq \frac{\#H_1(G)}{\#H_1(G) + \#H_2(G)}$$

Dividing the RHS through by $\#H_1(G)$ yields the requirement that $e^{-\epsilon/2} \leq (1 + (\#H_2(G)/\#H_1(G)))^{-1}$. The RHS is now minimised by maximising $\#H_2(G)/\#H_1(G)$, and we observe that this ratio is at most $(k_2/k_1)^n$ for sufficiently large n. Hence, we need to show that

$$e^{\epsilon/2} \geq 1 + (k_2/k_1)^n$$

Given that $1 + x \leq e^x$ the above inequality is satisfied by $\epsilon \geq 2(k_2/k_1)^n$. □

$\#H_1 \leq_{\mathsf{AP}} \#H$ can be shown in a similar way. Let (G, ϵ) be the input to $\#H_1$. If $\epsilon < 2(k_2/k_1)^n$ then we can use brute force to compute $\#H_1(G)$ exactly, as in the above analysis. (The running time will be approximately $|V(H_1)|^n$ which is polynomial in ϵ^{-1}.) If $\epsilon \geq 2(k_2/k_1)^n$ we simply return $\#\hat{H}(G)$ which is the result of calling the $\#H$ oracle with input $(G, \epsilon/2)$. To see that this works, observe that we need $\#\hat{H}(G)/\#H_1(G)$ to be in the range $[e^{-\epsilon}, e^{\epsilon}]$. The $e^{-\epsilon}$ bound is immediately satisfied because $\#H(G) \geq \#H_1(G)$ and we have used an accuracy parameter less than ϵ in the oracle call. To show that we have also satisfied the e^ϵ bound we must demonstrate:

$$\frac{e^{\epsilon/2}(\#H_1(G) + \#H_2(G))}{\#H_1(G)} \leq e^\epsilon$$

Cancelling this gives $1 + (\#H_2(G)/\#H_1(G)) \leq e^{\epsilon/2}$. Given that $\#H_2(G)/\#H_1(G) \leq (k_2/k_1)^n$ and $\epsilon \geq 2(k_2/k_1)^n$ we see that this inequality is adequately satisfied. □

It is helpful to describe the sampling equivalent of Lemma 6.1:

Lemma 6.2 *Let H be a disconnected graph with two components, H_1 and H_2. Suppose there exist constants k_1, k_2 such that $k_1 > k_2$ and, for all sufficiently large G, $\#H_1(G) \geq k_1^n$ and $\#H_2(G) \leq k_2^n$. Then $H_1 \leq_{SP} H$ and, in addition, if H_1 has a $PAUS$ then H has a PAUS.*

Proof. Firstly, the reader may be wondering why Lemma 6.2 does not simply say, "...then $H_1 \equiv_{SP} H$". This is because the reduction we describe for reducing sampling H to sampling H_1 does not fit well into the SP-framework. The direction H_1 to H is fine however, and is easily constructed. To see this, note that $\#H_1(G)$ is trivially at least a polynomially-small fraction of $\#H(G)$. Hence, we can simply use the powering technique (i.e. where we produce multiple copies of the input graph G) discussed in Section 3.7 (page 116[1]).

To show that a $PAUS$ for $H_1(G)$ can produce a $PAUS$ for $H(G)$ we use a sampling reduction which is not a SP reduction. Suppose we have a $PAUS$ for H_1. Now, we wish to build a $PAUS$ for H that takes as input (G, ϵ) and yields a distribution on $H(G)$ that is no further than ϵ from uniform. If

$$\epsilon \leq 2(k_2/k_1)^n$$

we can produce an exactly uniform sample by exhaustive listing. Otherwise, we simply generate an H_1 sample with accuracy $\epsilon/2$ (using the $PAUS$ for H_1) and return that.

[1]We can use the argument that, if there are p copies of the input G, $|Y_0|$ is at most k_2^{pn}, while in contrast $|Y_1|$ is crudely bound below by the exponentially superior k_1^{pn}. Thus, it is easy to guarantee that the choice of p needed to satisfy $|Y_0|/|Y_1| \leq \epsilon/4$ will not be too big.

To proceed, we therefore need to show that the failure to produce H_2 samples is not significant. First, let π be the exactly uniform distribution on H and π_1 be the exactly uniform distribution on $H_1(G)$. Therefore, $\pi_1(c) = 1/\#H_1(G)$ if $c \in H_1(G)$ and $\pi_1(c) = 0$ if $c \in H_2(G)$. We let π_1' be the approximately uniform distribution on $H_1(G)$ given by the $PAUS$ for $H_1(G)$ when called with accuracy $\epsilon/2$. Now, to meet the accuracy bound we need to prove that $\mathrm{d_{TV}}(\pi, \pi_1') \leq \epsilon$. By the triangle inequality (which holds because $\mathrm{d_{TV}}$ is a metric), we know that:

$$\mathrm{d_{TV}}(\pi, \pi_1') \leq \mathrm{d_{TV}}(\pi, \pi_1) + \mathrm{d_{TV}}(\pi_1, \pi_1')$$

Now, because we have used accuracy $\epsilon/2$ in our call to our H_1 $PAUS$, the term on the far right hand side of the above inequality is bound above by $\epsilon/2$. The only remaining task, therefore, is to show that $\mathrm{d_{TV}}(\pi, \pi_1) \leq \epsilon/2$. So, using the definition of variation distance given on page 109 we must show that

$$\sum_{c \in H_1(G)} \left| \frac{1}{\#H_1(G) + \#H_2(G)} - \frac{1}{\#H_1(G)} \right| + \sum_{c \in H_2(G)} \left| \frac{1}{\#H_1(G) + \#H_2(G)} - 0 \right| \leq \epsilon \tag{6.1}$$

(We have cancelled the two $1/2$ terms.) Tidying this up yields

$$\#H_1(G) \left(\frac{1}{\#H_1(G)} - \frac{1}{\#H_1(G) + \#H_2(G)} \right) + \frac{\#H_2(G)}{\#H_1(G) + \#H_2(G)} \leq \epsilon$$

Further gathering of terms simplifies this to

$$\frac{\#H_2(G)}{\#H_1(G) + \#H_2(G)} \leq \epsilon/2$$

So, is the above inequality satisfied? Given that $\#H_2(G) \leq k_2^n$ and $\#H_1(G) \geq k_1^n$ an upper bound on the LHS of the above inequality is $(k_2/k_1)^n$. We know from earlier that $\epsilon \geq 2(k_2/k_1)^n$ so we are done. □

The sampling result enshrined in Lemma 6.2 is of assistance in Chapter 4. There we argue that Theorem 4.1 is "best possible" because some disconnected graphs containing a non-trivial component actually have a $PAUS$. We then go on to prove this assertion by demonstrating a hand-crafted $PAUS$ for the graph in Figure 6.1. The sampling result from this section, however, allows us to develop a $PAUS$ for this graph

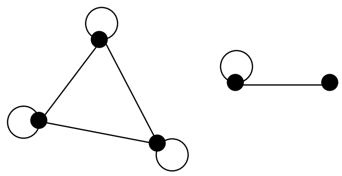


Figure 6.1: *This disconnected graph has a $PAUS$ (and an $FPRAS$).*

systematically. Label the two components of Figure 6.1 H_1 and H_2, where H_1 is the K_3^* graph and H_2 is the IS graph. (Hence $k_1 = 3$ and $k_2 = 2$.) Thus, it follows that if K_3^* has a $PAUS$ then H has a $PAUS$ also. K_3^* clearly does have a $PAUS$, so we have a graph H which does not consist solely of trivial components yet has a $PAUS$. (Of course, it also has an $FPRAS$, by Lemma 6.1.)

H with more than two components

It is also worth noting that Lemmas 6.1 and 6.2 can both be generalised to disconnected H with more than two components. In particular, note that in both cases the proofs still go through if H_1 and H_2 do not refer to single components, but instead are themselves groups of disjoint components. To see how this can be utilised, suppose H' is a graph with k components $H'_1, ..., H'_k$. Since G is connected, we know

$$\#H'(G) = \sum_{i=1}^{k} \#H'_i(G)$$

Furthermore, let h_1 and h_2 be two constants such that $h_1 > h_2$ and the following dividing process completely partitions H', with neither H_1 nor H_2 left as the empty graph:

We let H_1 consist of those components H'_i for which $\#H'_i(G) \geq h_1^n$, and we let H_2 be those components H'_i for which $\#H'_i(G) \leq h_2^n$. As a consequence, we know $\#H_1(G) \geq h_1^n$. Also, $\#H_2(G) \leq ph_2^n$ (where p is the number of components in H_2) but we can sweep p away by arguing that, for large enough G, $\#H_2(G) \leq ((h_1 + h_2)/2)^n$. Crucially, we now observe that, since H_1 and H_2 completely partition H', $\#H'(G) = \#H_1(G) + \#H_2(G)$. So it is as if we have partitioned H into two 'meta'-

components, H_1 and H_2. Hence, we can apply Lemma 6.1 or 6.2 (taking $k_1 = h_1$ and $k_2 = (1/2)(h_1 + h_2)$) which gives us $\#H_1 \equiv_{\mathsf{AP}} \#H$ or the corresponding sampling result from Lemma 6.2.

We can then apply this process recursively to H_1 for as long as we can partition it into dominating and dominated components. By transitivity we can therefore repeatedly discard those components of a disconnected graph H' which are identifiably dominated over by some other component in the graph, and H' will be interreducible (both in the counting and sampling sense) with the subgraph comprising the remaining components. The natural point to stop discarding components is when it is no longer apparent how (or, indeed, whether) remaining components dominate over each other.

6.2.2 A maximum-degree related upper bound

Lemma 6.3 *Let H be a connected graph, with maximum degree Δ. It follows that for all G, $\#H(G) \leq |V(H)|\Delta^{n-1}$.*

Proof. Since G is connected we let $T = (V(T), E(T))$ be a spanning tree of G, where $V(T) = V(G)$ and $E(T) \subseteq E(G)$. Note that, by definition, $\#H(G) \leq \#H(T)$. We try and develop an upper bound on $\#H(T)$, thus giving us an upper bound on $\#H(G)$.

We start by fixing an arbitrary vertex $r \in V(T)$ as the "root" and colouring it with some colour from $V(H)$; there are clearly at most $|V(H)|$ ways of doing this. Now, suppose we colour the rest of T by performing a depth-first search from r, and assigning a colour to every new vertex visited by the search. Our key claim is that when a new vertex is visited it can be coloured with one from at most Δ colours. To see why this is, note that every new vertex visited is connected by an edge to an already coloured vertex; given that the degree of the colour on this previous vertex is no more than Δ, it follows that there are no more than Δ colours possible on the new vertex. Given that every one of the $n-1$ remaining vertices is coloured exactly once in this manner, it follows that there are at most $|V(H)|\Delta^{n-1}$ colourings of T possible, which is also an upper bound on $\#H(G)$. $\square$

The lemma actually also holds if H is disconnected. However, in those circumstances a slightly tighter upper bound is easily described. Suppose H has k components: if H_i is the i'th component and Δ_i is the maximum degree of colours in H_i, it follows that an upper bound on $\#H(G)$ is

$$\sum_{i=1}^{k} |V(H_i)|\Delta_i^{n-1}$$

The significance of Lemma 6.3 is that it is an improvement on the standard, crude upper bound of $\#H(G) \leq |V(H)|^n$. As a consequence it allows us to prove that, in certain situations, an H graph with a small number of vertices can dominate over an H graph with a much larger number of vertices. This is demonstrated by the following example result:

Corollary 6.4 *Let H be a disconnected graph with two components, H_1 and H_2 where the maximum degree of a colour in H_2 is Δ. Suppose there exists some k such that $\Delta < k$ and for all sufficiently large G, $\#H_1(G) \geq k^n$. Then $\#H \equiv_{\mathsf{AP}} \#H_1$.*

Proof. We know that $\#H_2(G) \leq |V(H_2)|\Delta^{n-1}$ by Lemma 6.3. Given that $\Delta < k$, it is reasonable to say that, for n above a certain constant threshold, $\#H_2(G) \leq ((\Delta + k)/2)^n$. (We use $((\Delta + k)/2)^n$ simply because it is a convenient exponential, lower than k^n, which eventually bounds $|V(H_2)|\Delta^{n-1}$ above.) Now, given that $((\Delta + k)/2) < k$ we know that $\#H \equiv_{\mathsf{AP}} \#H_1$ by Lemma 6.1. $\square$

So, for example, this corollary would hold for those H where H_1 has a K_4^* subgraph and $H_2 = P_t^*$ for arbitrarily large (constant) values of t. This is because the maximum degree of a colour in P_t^* is (at most) 3, irrespective of t.

6.2.3 Matchings

A k-matching M of a graph G is a size-k subset of $E(G)$, such that no two edges in M share a vertex in common. (Recall that edges in $E(G)$ are unordered.) An $n/2$-matching in an n-vertex graph G is called a *perfect* matching, and is notable for the fact that *every* vertex in $V(G)$ comes up once as an endpoint of some edge in M. Not all graphs have perfect matchings.

Furthermore, for a given $H = (V(H), E(H))$ let $E_d(H)$ be the set of corresponding *directed* edges. (Recall that we only consider undirected H in this thesis.) For example, if H is the standard IS graph (i.e. $V(H) = \{b, r\}$ and $E(H) = \{\{b, b\}, \{b, r\}\}$) then $E_d(H) = \{(b, b), (b, r), (r, b)\}$.

Observation 6.5 *Let G be an n-vertex graph with a perfect matching. Then, for all H, $\#H(G) \leq |E_d(H)|^{n/2}$.*

To see why this holds, let M be a perfect matching of G, and let C be any colouring from $H(G)$. It follows that for all edges $\{u, v\} \in E(G)$, $(C(u), C(v)) \in E_d(H)$ and $(C(v), C(u)) \in E_d(H)$. So, if we inspect G when it has been coloured with C, and focus only on the edges from M, we gain a mapping from M to $E_d(H)$[2]. Thus, every colouring in $H(G)$ "comes up" as one of the $|E_d(H)|^{n/2}$ mappings from M to $E_d(H)$. (In general, however, not every one of the $|E_d(H)|^{n/2}$ mappings defines a colouring from $H(G)$. This is because two edges from M may well be connected to each other in $E(G)$ by edges not in M, meaning there is not complete freedom to assign pairs from $E_d(H)$ to edges in M.) □

Unlike Lemma 6.3, the above observation is dependent on G. As such, it is a tentative step in the direction of using graph-theoretic analysis to try and reason about the behaviour about H graphs on *specific families* of G. This is not a tactic we have deployed in this thesis, but it is likely to be increasingly relevant in the future as we are confronted with more and more disconnected H that we struggle to classify for general G.

[2]By looking at the colours on the endpoints of each edge in M, considering the endpoints (say) in lexicographic order.

6.3 Coping with small additive quantities in counting equations

There are various instances where two graphs H and H' can be related by counting equations such as $\#H(G') = \#H'(G) + k$ where G' is some transformation of G and k is some small, additive quantity. Lemma 6.6 and Corollary 6.7, which we present in this section, allow us to relate the complexity of $\#H'$ and $\#H$ based solely on counting equations such as these. Towards the end of the section we demonstrate a practical application of the lemma/corollary, using it to "cleanly" classify disconnected 4-vertex H (in contrast to earlier reduction techniques for disconnected 4-vertex H, which had been on a fairly messy, ad-hoc, graph by graph basis.)

It should be noted that, in both Lemma 6.6 and Corollary 6.7, the restriction to H-colouring problems is arbitrary. There is nothing to stop $\#H$ and $\#H'$ being replaced by non-H-colouring problems. (For example, we use Lemma 6.6 in Appendix A.6 in relation to a $\#DownSets$ reduction.)

Lemma 6.6 *Let H and H' be any two graphs. (That is, possibly disconnected.) Let f be a deterministic function that takes as input a graph G and produces a graph $f(G)$ in time no more than polynomial in $|V(G)|$.*[3] *If (for all G) $\#H(f(G)) = \#H'(G) + k(G)$, where $k(G)$ is a natural number no larger than poly($|V(G)|$) and computable exactly in time at most poly($|V(G)|$), then $\#H' \leq_{AP} \#H$.*

Proof: Let (G, ϵ) be the input to $\#H'$. First, we compute $k(G)$; for convenience we refer to this value as k. The fact that this only takes time polynomial in $|V(G)|$ ensures this computation does not slow down the AP-reduction unacceptably. Secondly, we construct $f(G)$ and again this does not take too long. Now, we define $N = \#H(f(G))$, so $\#H'(G) = N - k$. The reduction is as follows. Let $\hat{N}$ be the result of calling our approximation oracle for $\#H$ with input $f(G)$ and accuracy $\delta = \epsilon/20k$. (The fact that

[3] *Note that this Lemma could be extended so that f and k not only take G as their input but ϵ also, with a consequent increase in flexibility in their running times and the size of their outputs. However, we find in practice that this is unnecessary and for simplicity choose not to introduce ϵ in this way.*

k is at most poly($|V(G)|$) ensures that δ^{-1} is not too large, which we require by the definition of AP-reduction.)

To complete the reduction we round the value $\hat{N} - k$ to the nearest integer and return this value. The motivation behind this strategy is that, if N is sufficiently small in relation to ϵ, the rounding of $\hat{N} - k$ produces an exactly correct answer. Otherwise, N is "large" and the various inaccuracies do not constitute a significant deviation from the value we require, $N - k$.

First we examine the case where the reduction yields an exact answer. This happens when $\hat{N} - k$ lies in the range $((N-k)-(1/2),(N-k)+(1/2))$. To satisfy the top bound, we require

$$e^{\delta}N - k < (N-k) + \frac{1}{2} \tag{6.2}$$

Re-arranging, this becomes

$$N < \frac{1}{2(e^{\delta}-1)}$$

To ensure that bounds are tight we need to establish a lower bound on the above threshold, and this requires an upper bound on e^{δ}. Given that, for $0 < \delta \leq 1$, $e^{\delta} \leq 1+2\delta$, it follows that a lower bound on the above fraction is

$$N < \frac{5k}{\epsilon}$$

So as long as $N < 5k/\epsilon$ the top bound is satisfied. With regard to the lower bound, we require:

$$(N-k) - \frac{1}{2} < e^{-\delta}N - k \tag{6.3}$$

Rearranging, this comes out as

$$N < \frac{(1/2)}{1-e^{-\delta}}$$

As before we seek to make this threshold as low as possible. We do this by maximising the denominator of the fraction, which we do in turn by establishing a lower bound on $e^{-\delta}$. We know that $e^{-x} = 1 - x + x^2/2 - ...$ so $1 - x$ is an appropriate lower bound. Hence, to satisfy the inequality we need

$$N < \frac{10k}{\epsilon}$$

This is a less onerous bound than $5k/\epsilon$, so it is safe to say that if $N < 5k/\epsilon$ then both upper and lower bounds are met, and as a result rounding yields an exact answer for N in this range.

It remains to show that, if $N \geq 5k/\epsilon$, the answer we get out is "close enough". We seek to do this by showing that the point at which N starts being big enough is guaranteed to be below $5k/\epsilon$. When developing this threshold, the worst-case scenario is where the act of rounding adds (or subtracts) 1 in the wrong direction i.e. away from the desired answer. So, for the top bound, we require:

$$e^{\delta}N - k + 1 \leq e^{\epsilon}(N - k) \tag{6.4}$$

If we rearrange this it becomes

$$N \geq \frac{k(e^{\epsilon} - 1) + 1}{e^{\epsilon} - e^{\delta}} \tag{6.5}$$

For the purpose of analysis it is helpful to express this in another form. Note that the following inequality is identical:

$$N \geq \frac{k(e^{\delta} - 1) + 1}{e^{\epsilon} - e^{\delta}} + k \tag{6.6}$$

To ensure a tight analysis we must make the right hand side of the above inequality as large as possible. We do this by maximising the numerator and minimising the denominator. The numerator is maximised by again noting that $e^x \leq 1 + 2x$ in this range. A lower bound on the denominator is found by observing that

$$e^{\epsilon} - e^{\delta} = (1 - 1) + (\epsilon - \delta) + (1/2)(\epsilon^2 - \delta^2) + ...$$

So a lower bound on $e^{\epsilon} - e^{\delta}$ (for $\delta < \epsilon$) is $\epsilon - \delta$. Hence, the largest the right hand side of (6.6) gets is

$$\frac{(\epsilon/10) + 1}{\epsilon((20k - 1)/20k)} + k$$

Tidying this up yields

$$\frac{2k}{20k - 1} + \frac{20k}{\epsilon(20k - 1)} + k$$

The question is, then, is this value below the $5k/\epsilon$ threshold, as we require i.e. is the following true:

$$\frac{2k}{20k-1} + \frac{20k}{\epsilon(20k-1)} + k < 5k/\epsilon$$

Dividing through by k and re-arranging gives

$$\frac{2}{20k-1} + 1 < \frac{1}{\epsilon}(5 - \frac{20}{20k-1})$$

Note that the biggest the LHS gets is when $k = 1$, and this sets the LHS at $21/19$. The smallest the RHS gets is when $\epsilon = 1$ and $k = 1$, which gives $5 - (20/19)$ and this value is bigger than $21/19$. So the top bound is fine. It remains to show that the bottom bound is met. This time rounding can hinder by subtracting as much as 1 from our estimate, so we need to show that

$$e^{-\epsilon}(N-k) \leq e^{-\delta}N - k - 1 \tag{6.7}$$

Rearranging gives

$$N \geq \frac{k(1-e^{-\epsilon})+1}{e^{-\delta}-e^{-\epsilon}} \tag{6.8}$$

Again it is useful to manipulate this so that the numerator is in terms of δ, not ϵ, so this gives

$$N \geq \frac{k(1-e^{-\delta})+1}{e^{-\delta}-e^{-\epsilon}} + k \tag{6.9}$$

As before we make the numerator large and the denominator small. The numerator is maximised by minimising $e^{-\delta}$, and since $1-\delta \leq e^{-\delta}$ the largest the numerator gets is $k\delta + 1$. A lower bound on the denominator can be ascertained by minimising $e^{-\delta}$ and maximising $e^{-\epsilon}$. A lower bound on $e^{-\delta}$ is $1-\delta$, and an upper bound on e^{-x} is $1-(x/2)$ for x in the range $(0,1)$, so a lower bound on the denominator is $(\epsilon/2) - \delta$. It follows that the largest the RHS of (6.9) gets is

$$\frac{(\epsilon/20)+1}{\epsilon((10k-1)/20k)} + k$$

This rearranges to

$$\frac{k}{10k-1} + \frac{20k}{\epsilon(10k-1)} + k$$

Now, is this value less than $5k/\epsilon$? That is, does the following hold:

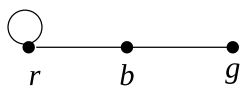
$$\frac{k}{10k-1} + \frac{20k}{\epsilon(10k-1)} + k < 5k/\epsilon$$

Dividing through by k and rearranging gives

$$\frac{1}{10k-1}+1<\frac{1}{\epsilon}(5-\frac{20}{10k-1})$$

The largest the LHS gets is when $k=1$, and this gives 10/9. The smallest the RHS gets is when $\epsilon=1$ and $k=1$, giving $5-(20/9)$, so clearly the inequality holds. $\square$

The above Lemma could be used to simplify some of the reductions from Chapter 2. For example, consider the graph H:



In the original reduction for this graph, on page 32, the graph was categorised as $\equiv_{\mathsf{AP}}\#SAT$ by showing $\#IS\leq_{\mathsf{AP}}\#H$, through the use of a K_2-cliqueset to point out $H[r]$. However, consider the following simpler reduction. Let $f(G)=G'$ where G' is a copy of G plus a new vertex x which is connected to every other vertex in G'. Now, for every graph G we have $H(G')=\#IS(G)+k(G)$ where $k(G)=3$ if G is a single vertex and $k(G)=1$ otherwise. Hence we have $\#IS\leq_{\mathsf{AP}}\#H$ by Lemma 6.6.

Corollary 6.7 *Suppose the preconditions described in Lemma 6.6 hold but we have the additional constraint that f is actually the identity function. Then $\#H\equiv_{\mathsf{AP}}\#H'$.*

Proof. We have $\#H'\leq_{\mathsf{AP}}\#H$ already from Lemma 6.6. To show that $\#H\leq_{\mathsf{AP}}\#H'$ we can adapt the proof from Lemma 6.6. Let (G,ϵ) be the input to $\#H$. First, we compute $k(G)$, referring to this value as k. We define $N=\#H'(G)$, so $\#H(G)=N+k$. Let $\hat{N}$ be the result of calling our approximation oracle for $\#H'$ with input G and accuracy $\delta=\epsilon/20k$. To complete the reduction we round $\hat{N}+k$ to the nearest integer and return this value. The difference to the previous reduction is that this time we are adding k rather than subtracting it. Rather than produce a completely new proof we can "piggy-back" on the previous proof.

Observe that the analogue to (6.2) is

$$e^{\delta}N+k<(N+k)+\frac{1}{2}$$

and the analogue to (6.3) is

$$(N+k) - \frac{1}{2} < e^{-\delta}N + k$$

Note that these two inequalities are identical to (6.2) and (6.3) respectively, i.e. simply subtract $2k$ from each side. So, as before our reduction gives an exact answer when $N < 5k/\epsilon$.

We now need to show that if $N \geq 5k/\epsilon$, our output is close enough. The analogue to (6.4) is

$$e^{\delta}N + k + 1 \leq e^{\epsilon}(N+k)$$

Rearranging to give us the analogue of (6.5) gives

$$N \geq \frac{k(1-e^{\epsilon}) + 1}{e^{\epsilon} - e^{\delta}} \tag{6.10}$$

Recall that in the previous proof we showed that if $N \geq 5k/\epsilon$ then (6.5) is satisfied. Now, notice that the RHS of (6.10) is a lower bound on the RHS of (6.5). This is because $1 - e^{\epsilon}$ is the negation of the positive term $e^{\epsilon} - 1$. Hence, if $N \geq 5k/\epsilon$ satisfies (6.5) then it automatically satisfies (6.10). Similarly, note that our analogue to (6.7) is

$$e^{-\epsilon}(N+k) \leq e^{-\delta}N + k - 1$$

and thus our analogue to (6.8) is

$$N \geq \frac{k(e^{-\epsilon} - 1) + 1}{e^{-\delta} - e^{-\epsilon}}$$

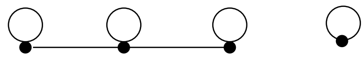
This time $e^{-\epsilon} - 1$ is the negation of the positive term $1 - e^{-\epsilon}$ so the RHS of the above inequality is a lower bound on the RHS of (6.8), and hence also satisfied by $N \geq 5k/\epsilon$.

□

6.4 A complete classification of disconnected H on 4 or fewer vertices

A useful application of Corollary 6.7 is where a disconnected graph H consists of a non-trivial component alongside various "singleton" components. We define a singleton component to be either a single vertex (with or without a loop) or a copy of K_2; such components are unique because, if H' is a singleton component, $\#H'(G)$ is bound above by a constant for *all* connected G. (For H' equal to a looped single vertex, $\#H'(G) = 1$. For H' equal to an unlooped single vertex, $\#H'(G) = 0$ unless G is a single vertex, in which case $\#H'(G) = 1$. For H' equal to K_2, $\#H'(G) = 2$ if G is bipartite and 0 otherwise.)

For example, the following graph is $\equiv_{\mathsf{AP}}\#BIS$ because, for connected G, $\#H(G) = \#H'(G) + 1$ where H' is *2-WR*.



Similarly, we can use this technique to help classify disconnected H on 4 or fewer vertices. (Disconnected H on fewer than 4 vertices can easily be subsumed into one of the cases we now describe.) For those 4-vertex disconnected H which consist solely of trivial components, $\#H$ is trivially $FPRAS$able simply because we can add together the contribution of individual components to give an exactly correct answer. For disconnected H which consist of one, non-trivial 3-vertex component and one singleton, Corollary 6.7 immediately proves that $\#H$ is of the same complexity as the 3-vertex component (which we already know because we have fully classified all 3-vertex H.)

This follows because, as in the above example, we can just take f as the identity function and $k(G) = 1$ or $k(G) = 0$ (depending on whether the singleton component is looped or unlooped and whether G is a single vertex or not.) If the largest non-trivial component in H has only two vertices, then there are three cases to consider. The only non-trivial 2-vertex component is IS (the independent set graph), so possibilities are: (a) one IS component plus singletons, (b) two IS components and (c) one IS component plus one K_2^* component. We can cope with (a) simply by using Corollary 6.7 again; we take f as the identity function and $k(G)$ equal to the sum of $\#H'(G)$ over all the singleton components H' in H. (This proves $\#H\equiv_{\mathsf{AP}}\#IS$.) Similarly, with (b) $\#H\equiv_{\mathsf{AP}}\#IS$ because $\#H(G) = 2\#IS(G)$. This leaves only (c) (i.e. Figure 6.2) to classify, and - interestingly - we now know that this graph is $FPRAS$able, thanks to an observation by Jerrum (communicated informally to the author.) More specifically, Jerrum has given an $FPAUS$ for the graph, which - because of the "counting to sampling" result in [12] - we can then easily turn into an $FPRAS$. Let H_1 be the graph's K_2^* component, and let H_2 be its IS component. Let the looped colour in H_2 be (as usual) b and the unlooped colour r.

This is the algorithm; it is a simple Monte Carlo algorithm. Firstly, let i be a counter variable that we initialise to zero. (We show how j can be chosen in due course.)

(a) With probability $1/2$, go to step (b). Otherwise, go to step (c).
(b) Choose u.a.r. an H_1 colouring of G, and terminate the algorithm by outputting that.
(c) Let Col be a mapping from $V(G)$ to $\{b, r\}$ obtained by u.a.r. mapping each vertex of G to a member of $\{b, r\}$. If Col is a valid H_2 colouring, terminate the algorithm by outputting Col. Otherwise, increment i. If i is still less than j, go to step (a), otherwise terminate the algorithm by outputting $\perp$.

Note that, for a given j, the probability of the algorithm outputting $\perp$ is at most $(1/2)^j$. (This follows because $\#H_1(G) \geq \#H_2(G)$.) Furthermore, it follows that the deviation distance of this algorithm from our desired distribution is simply the probability that $\perp$ is output. Hence, to satisfy the deviation parameter ϵ we simply need to choose j as follows:-

$$j \geq \left\lceil \frac{\ln(1/\epsilon)}{\ln(2)} \right\rceil$$

Note that this is an $FPAUS$ because j is no larger than poly$(n, \log(\epsilon^{-1}))$ $\square$.

While on this topic, it is natural to ask whether the complexity of a disconnected H can be determined knowing nothing more than the complexity of its components. This is not so:- (relative) physical structure is important. For example, consider the disconnected H with two components, IS and K_1^*. We know $\#H \equiv_{\mathsf{AP}} \#SAT$ by Lemma 6.6. How-

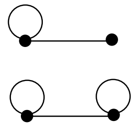


Figure 6.2: *This graph has a FPAUS (and therefore an FPRAS)*

ever, we know that if we multiply weights on the K_1^* component by 3 - turning it into K_3^* whilst preserving its $FPRAS$able status - $\#H$ becomes $FPRAS$able. So component complexity alone does not uniquely determine the complexity of the full graph H. $\square$

Finally, it is important to note that our success in classifying disconnected 4-vertex H is largely based on the fact that, because only 4 vertices are involved, there tends to be only space for one non-trivial component plus some singletons. Regrettably, for disconnected H with multiple non-trivial components, we do not know a great deal about how we might determine their complexity. In this regard, ad-hoc, graph-by-graph reductions remain an important and powerful method of determining hardness results for disconnected H, and should not be discounted. However, in terms of easiness results (i.e. determining upper bounds on complexity), we have the results from this Chapter but, beyond that, we suffer from the same general lack of knowledge about easiness results as we do with connected H.

6.5 Conclusion

This chapter has shown that we have begun to make inroads into the questions surrounding the complexity of disconnected H. However, this is clearly just the beginning and there is a need to deepen our understanding in a number of critical areas. One area that requires improvement is our understanding of how the value $\#H$ (for a given H) varies with different types of input graph G; this is likely to be an increasingly important issue when attempting to determine whether one component exponentially dominates over another. The applicability of Monte Carlo-style sampling algorithms (as used in the classification of Figure 6.2) is another area that needs to be explored further; the author feels that such algorithms could help to quite significantly extend our understanding of disconnected H, but has not had time to explore this issue, because Jerrum's $FPAUS$ was communicated to the author only in the very final stages of thesis-writing[4].

As the reader has no doubt ascertained, 4-vertex H are only really "toy" disconnected H in the sense that they are not (generally) large enough to incorporate more than one significant component. Where disconnected H incorporate two or more significant components the gaps in our knowledge are similar to the gaps in our knowledge with connected H. In other words, we are quite good at developing lower bounds on complexity (i.e. hardness results) through methods such as pointing out subgraphs, but easiness results are difficult to come by. At present we have two principal methods for completely determining the complexity of disconnected H. On one hand we may be able to point out, induce or encode $\equiv_{\mathsf{AP}}\#SAT$ problems, which is obviously a strategy we use with connected H also. Alternatively, we use Lemmas 6.1 and 6.2 to exploit those fairly limited scenarios where one component exponentially dominates over all the others, thus reducing the problem of determining the complexity of the overall graph to "simply" the problem of determining the complexity of that connected component.

[4]For example, consider the disconnected graph shown at the beginning of Section 3.7. A Monte Carlo-style algorithm similar to that shown in this chapter could probably be used to show that approximately sampling this graph is reducible to approximately sampling *2-WR*.

Chapter 7

The complexity hierarchy

7.1 Introduction

With respect to AP-reducibility, we have so far only considered three options when attempting to determine the complexity of an H-colouring problem $\#H$. We have noted that, for certain H, $\#H$ is $FPRAS$able[1], and it is natural to think of these graphs as "easy". At the other end of the spectrum we have established that (at least in the context of 4-vertex H) quite a large number of graphs are $\equiv_{\mathsf{AP}}\#SAT$ and we think of these as "hard". Additionally, we have considered graphs interreducible with $\#BIS$ to be those of "intermediate" complexity.

This chapter is concerned with the following question: what does the AP-reducibility complexity hierarchy look like from the H-colouring perspective? That is, what can we say in general about the way H-colouring problems are distributed across the complexity hierarchy? This question is pivotal because, if we recall the motivation for studying H-colouring in the first place, we see that the underlying AP-reducibility complexity hierarchy (i.e. that which concerns all the problems in $\#P$ and not just H-colouring problems) is going to be at *least* as nuanced as that suggested by H-colouring. To tackle this overarching question there are two immediate issues that we must resolve. Firstly, do any of the three classes overlap e.g. could $\equiv_{\mathsf{AP}}\#BIS$ be $FPRAS$able? Could

[1]Although it is perhaps significant that we have yet to find a *connected* non-trivial H that is $FPRAS$able

$\#BIS \equiv_{\mathsf{AP}} \#SAT$? Secondly, if the class of H-colouring problems that are respectively $FPRAS$able, $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ are mutually distinct, do all H-colouring problems fall into one of them, or do some H-colouring problems "slip through the cracks" into as-yet undiscovered complexity classes? (Note that, unless we state otherwise, we are referring to *connected* H-colouring in this chapter.[2])

Though we do not have concrete answers to these questions, we use this chapter to discuss the evidence which leads us to conjecture that the AP-reducibility complexity hierarchy might (*from the perspective of H-colouring*) look like the diagram in Figure 7.1. (We stress the H-colouring perspective because it is technically meaningless to

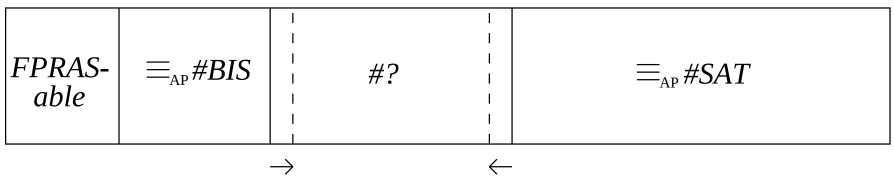


Figure 7.1: *Possible structure of the complexity hierarchy, with regard to H-colouring problems*

speak of "boundaries" between complexity classes unless we are doing so in reference to some well-defined continuum of counting problems.) To summarise Fig 7.1, we believe that the classes of $FPRAS$able, $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ H-colouring problems are all distinct complexity classes. We further believe that an $\#H$-colouring problem is either $FPRAS$able or at least as hard as $\equiv_{\mathsf{AP}}\#BIS$. As we discuss in due course, we are fairly confident that this assertion is correct but until the precise complexity relationship between approximate counting and approximate sampling is established it remains open. Most speculatively - and this is the topic that the bulk of this chapter is dedicated to - we think that a potentially large number of $\#H$-colouring problems lie in a gap between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$; too hard for $\equiv_{\mathsf{AP}}\#BIS$, not hard enough for $\equiv_{\mathsf{AP}}\#SAT$. (It may be that there are *multiple* distinct complexity classes filling this gap.) The arrows and the dotted boundaries signify our belief that, as better

[2] Recall, from Chapter 6, that the complexity landscape is more subtle when H is disconnected.

reduction techniques are uncovered, some graphs which are currently unclassified will fall into $\equiv_{\mathrm{AP}}\#BIS$ or $\equiv_{\mathrm{AP}}\#SAT$. However, this need not contradict our assertion that there is a complexity gap between the two classes: our $\equiv_{\mathrm{AP}}\#BIS$-easiness and $\equiv_{\mathrm{AP}}\#SAT$-hardness reduction techniques may improve but we do not think they will ever account for *all* non-trivial H graphs.

So, far from $\equiv_{\mathrm{AP}}\#BIS$ being some dominant feature on the H-colouring complexity landscape, we increasingly think that it is in fact quite a small complexity class, which contains very few non-bipartite, non-trivial graphs and surprisingly few bipartite, non-trivial graphs.

The chapter is laid out in several sections. In the first section, Section 7.2 (*The "easy" end of the complexity hierarchy: $\equiv_{\mathrm{AP}}\#BIS$-hardness*), we briefly discuss the results that lead us to believe that an H-colouring problem is either $FPRAS$able or at least as hard as $\equiv_{\mathrm{AP}}\#BIS$. (So, from an H-colouring perspective, we believe these classes push up tight next to one another.) In Section 7.3 (*Reduction techniques: $\equiv_{\mathrm{AP}}\#BIS$-easiness*) we move onto the main topic of this chapter and start examining the reductions we have used thus far in attempting to squeeze unclassified H graphs into $\equiv_{\mathrm{AP}}\#BIS$, with a particular emphasis on the problem $\#DownSets$ and our experiences of reducing H-colouring problems to it. This culminates in Section 7.4 (*A conjecture about $\equiv_{\mathrm{AP}}\#BIS$*), where we identify a graph H (called the *junction*) which we suspect sits neither in $\equiv_{\mathrm{AP}}\#BIS$ nor in $\equiv_{\mathrm{AP}}\#SAT$, but somewhere inbetween. In Section 7.5 (*Bipartite H and non-bipartite H*) we take a short break from the question of the complexity gap to look briefly at the question of how bipartite H as a whole relate to the non-bipartite H world. Finally, in Section 7.6 (*Other interesting relationships*) we take the first steps in firming up the notion of a complexity class between $\equiv_{\mathrm{AP}}\#BIS$ and $\equiv_{\mathrm{AP}}\#SAT$ by establishing some reductions between those H that have thus far stubbornly refused classification.

Before moving onto the substantive part of the chapter, it is worth noting that this chapter is more concerned with themes, trends and ideas than long, detailed, formal proofs. Where appropriate we have presented formal details, but in several places we demonstrate sketch proofs rather than formal proofs, reasoning that excess detail would obscure the more general point. In a similar vein, some of our later attempts at describing experimental output may seem a little "hand-waving", but in the absence of more formal characterisations such descriptions are useful for conveying informal observations and those intuitions developed throughout the research period.

7.2 The "easy" end of the complexity hierarchy: $\equiv_{\mathsf{AP}}\#BIS$-hardness

In Chapter 4 we prove that, for all non-trivial connected H (and disconnected H consisting solely of non-trivial components) $BIS{\leq_{\mathsf{SP}}}H$.[3]

We see this as a strong indication that all non-trivial, connected H-colouring approximate counting problems are at least as hard as $\equiv_{\mathsf{AP}}\#BIS$. Of course, $BIS{\leq_{\mathsf{SP}}}H$ does not give us this result exactly; it shows that a $PAUS$ for H can be turned into a $PAUS$ for BIS. As discussed in Section 3.10, this also shows that a $PAUS$ for H can be turned into an $FPRAS$ for $\#BIS$, because BIS is self-reducible. The result we are missing - that an $FPRAS$ for $\#H$ can be turned into an $FPRAS$ for $\#BIS$ - could still yet be extrapolated "for free" from Theorem 4.1, but only if we could show that all H-colouring problems are self-reducible[4].

It is not actually clear whether all H-colouring problems are self-reducible; it may be that approximately counting H-colourings is (for some H) easier than approximately sampling H-colourings. But this is a separate issue to the question of whether the existence of an SP-reduction implies the existence of an analogous AP-reduction: if approximately computing $\#H$ is (in some cases) strictly easier than approximately sam-

[3]In actual fact, Theorem 4.1 is slightly stronger than that and applies even when the input G is restricted to being bipartite.

[4]Of course, there may be other ways of proving $\#BIS{\leq_{\mathsf{AP}}}\#H$ than by piggybacking on Theorem 4.1

pling H-colourings this does not prevent both $BIS \leq_{\mathsf{SP}} H$ and $\#BIS \leq_{\mathsf{AP}} \#H$ from being simultaneously true. Indeed, as we have discussed on a number of occasions, we tend to think that SP-reductions which are underpinned by the type of proof structure used in Theorem 4.1 do constitute *some* evidence for the existence of an analogous AP-reduction.

To elaborate, recall that the SP-reduction was initially introduced to assist in the proof of Theorem 4.1, and in particular to allow the "glueing" together of multiple reductions simultaneously. The problem of having more than one reduction occurring simultaneously continues to be the main obstacle preventing the development of an AP-reduction counterpart to Theorem 4.1. Nonetheless, experience shows that, given an arbitrary, non-trivial graph H, there is a good chance that (with a view to eventually reaching P_4 i.e. $\#BIS$) we can put together ad-hoc gadgetry such that we can identify a unique non-trivial graph H', smaller than H, such that $\#H' \leq_{\mathsf{AP}} \#H$. Indeed, compared to the problem of proving a graph to be $\equiv_{\mathsf{AP}} \#BIS$-easy the task of delineating between multiple possible smaller graphs H' appears relatively achievable. Obviously this is not formal evidence for the existence of a $\#BIS \leq_{\mathsf{AP}} \#H$ result, but it hopefully explains our confidence in thinking one exists. (Proving $\#BIS \leq_{\mathsf{AP}} \#H$ is an interesting task that could be undertaken as future work leading on from this thesis, perhaps building on existing $\equiv_{\mathsf{AP}} \#BIS$-hardness results like Lemma 2.19.)

Hence, for the remainder of this chapter, we assume that the "easy" end of the complexity hierarchy (as seen by H-colouring) is relatively well understood. That is, we are fairly confident that, amongst the domain of connected H, the trivial H graphs are the only $FPRAS$able ones, and that all others are $\equiv_{\mathsf{AP}} \#BIS$-hard, even if we do not have a formal result to this effect. Note that this assumption does not deal with the *actual* complexity of $\#BIS$; as mentioned earlier, it remains a possibility that $\equiv_{\mathsf{AP}} \#BIS$ is $FPRAS$able.

7.2.1 $\equiv_{\mathsf{AP}}\#BIS$ - a distinct class?

We have a certain amount of weak, provisional evidence that $\equiv_{\mathsf{AP}}\#BIS$ is neither $FPRAS$able nor AP-reducible with $\equiv_{\mathsf{AP}}\#SAT$. As will become apparent, we are far more confident that $\#BIS\equiv_{\mathsf{AP}}\#SAT$ does not hold than we are that $\equiv_{\mathsf{AP}}\#BIS$ is not $FPRAS$able, but at the present time we think the balance of evidence is nevertheless slightly in favour of $\equiv_{\mathsf{AP}}\#BIS$ being a distinct class. First, we look at the evidence that $\equiv_{\mathsf{AP}}\#BIS$ is not $FPRAS$able. There are two reasons that can be cited in defence of this claim; one complexity-related and the other being a "weight of evidence" argument.

1. In [8], it is proven that $\#BIS$ is complete for a logically-defined subclass of $\#P$, called $\#RH\Pi_1$. (We discuss this class further in Section 7.3.3.)

2. Also in [8], it is demonstrated that $\#BIS$ is AP-interreducible with a number of H-colouring and non H-colouring problems, none of which are thus far known to admit an $FPRAS$. To this list we can add the various H-colouring problems which we show to be $\equiv_{\mathsf{AP}}\#BIS$ in this thesis, and those we identify to be $\equiv_{\mathsf{AP}}\#BIS$-easy later in this chapter. (As we go on to articulate, we suspect that "most" H-colouring problems are actually harder than $\equiv_{\mathsf{AP}}\#BIS$. However, from a complexity viewpoint the fact that the set of H-colouring problems interreducible with $\equiv_{\mathsf{AP}}\#BIS$ is non-empty, growing and shows some variety increases the weight of the argument that $\equiv_{\mathsf{AP}}\#BIS$ is a distinct class in its own right.)

These arguments must be balanced against the fact that, as this chapter testifies, we are somewhat pessimistic about how powerful (i.e. hard) $\equiv_{\mathsf{AP}}\#BIS$ actually is. Nonetheless, it is sensible to defer to the state of current knowledge and, given the two above observations, we work on the provisional assumption that $\equiv_{\mathsf{AP}}\#BIS$ is not $FPRAS$able.

Our conviction that $\equiv_{\mathsf{AP}}\#BIS$ is not AP-interreducible with $\equiv_{\mathsf{AP}}\#SAT$ is mainly based on experience. At the risk of sounding too informal, we have not even come close to finding a $\equiv_{\mathsf{AP}}\#BIS$-easy problem that is $\equiv_{\mathsf{AP}}\#SAT$: $\equiv_{\mathsf{AP}}\#BIS$-easy problems simply don't "feel" powerful enough to code up $\equiv_{\mathsf{AP}}\#SAT$ problems. (This overlaps with our discussion in Section 5.6, where we explain our suspicion that there does not exist a bipartite H such that $\#H \equiv_{\mathsf{AP}} \#SAT$.) Furthermore, the discussion in Section 7.4 of this chapter conveys our belief that there actually exists a complexity "gap" between $\equiv_{\mathsf{AP}} \#BIS$ and $\equiv_{\mathsf{AP}} \#SAT$, in which one or more complexity classes reside. This apparent "gap" bolsters the case that $\equiv_{\mathsf{AP}} \#BIS$ and $\equiv_{\mathsf{AP}} \#SAT$ are distinct.

At this point it is worth restating that it is extremely unlikely that $\equiv_{\mathsf{AP}} \#SAT$ is $FPRAS$able, because this would in turn entail that $RP = NP$. Hence, we choose to proceed on the provisional assumption that all three complexity classes are mutually distinct.

So, we now turn our attention higher up the complexity hierarchy. To recap, the work in Chapter 5 has allowed us to make some tentative assertions about inherent characteristics of $\equiv_{\mathsf{AP}} \#SAT$ problems; in this respect, Chapter 5 is chiefly concerned with dragging as many problems as possible into $\equiv_{\mathsf{AP}} \#SAT$ i.e. exploring the "frontier" between H-colouring problems that are $\equiv_{\mathsf{AP}} \#SAT$ and H-colouring problems that seem to sit below $\equiv_{\mathsf{AP}} \#SAT$. In the following section we consider the related question of dragging as many graphs as possible into $\equiv_{\mathsf{AP}} \#BIS$. Hence, the section focuses particularly on $\equiv_{\mathsf{AP}} \#BIS$-easiness reductions.

7.3 Reduction techniques: $\equiv_{\mathsf{AP}} \#BIS$-easiness

Our conjecture that there are H graphs which sit between $\equiv_{\mathsf{AP}} \#BIS$ and $\equiv_{\mathsf{AP}} \#SAT$ is heavily motivated by the difficulties we have faced in classifying many non-trivial H as either $\equiv_{\mathsf{AP}} \#BIS$-easy or $\equiv_{\mathsf{AP}} \#SAT$. These difficulties are in part due to an incomplete understanding of the power of AP-reductions. Nonetheless, our experience with $\equiv_{\mathsf{AP}} \#BIS$-easiness reductions leads us to think that our inability to show $\#H \leq_{\mathsf{AP}} \#BIS$ (for H we have also been unable to prove $\equiv_{\mathsf{AP}} \#SAT$) is in many cases not due to the limitations of our reduction techniques, but simply because "most" non-trivial H graphs lie strictly above $\equiv_{\mathsf{AP}} \#BIS$ in the complexity hierarchy.

Later in this section we conjecture, in light of results from these $\equiv_{\mathsf{AP}}\#BIS$-easiness reductions, that many bipartite graphs are harder than $\equiv_{\mathsf{AP}}\#BIS$. The idea that there may be a complexity gap between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$ emerges when we couple this conjecture with our suspicion that there is no bipartite H such that $\#H\equiv_{\mathsf{AP}}\#SAT$ (see Section 5.6 from Chapter 5). Thus, our belief in the existence of a complexity gap is particularly motivated by our experiences with bipartite H. That said, we do also think that a potentially substantial number of non-bipartite H lie somewhere between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$. (The three unclassified 4-vertex H - see Figure 2.13 on page 101 - are possible candidates for such status.) However, given the relatively high degree of success we have had adapting and modifying $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions, and the absence of apparent "structural" barriers to $\equiv_{\mathsf{AP}}\#SAT$-hardness (i.e. such as those that lead us to think no bipartite H is $\equiv_{\mathsf{AP}}\#SAT$), we would not be too surprised if many presently unclassified, non-bipartite H turn out to be $\equiv_{\mathsf{AP}}\#SAT$.

Either way, we strongly suspect that $\equiv_{\mathsf{AP}}\#BIS$ is in fact quite a small class, and to motivate this it is necessary to study both the $\equiv_{\mathsf{AP}}\#BIS$-easiness techniques that have been used so far and the graphs that they have successfully shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy.

To this end, we now look at four $\equiv_{\mathsf{AP}}\#BIS$-easiness reduction techniques which have been explored at some point, summarised as direct reduction, $\#MaxBIS$, logic and finally $\#DownSets$. All four of these techniques are rooted in reductions originally demonstrated in [8]. The general experience of using these reduction techniques is of significant overlap in terms of the H graphs that have been classified as $\equiv_{\mathsf{AP}}\#BIS$-easy using them. In fact, it seems in general that most $\equiv_{\mathsf{AP}}\#BIS$-easy graphs we have identified (through whatever method) can be shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy by reduction to $\#DownSets$. Indeed, of the four reduction techniques mentioned $\#DownSets$ has been the most significant and "prolific" identifier of $\equiv_{\mathsf{AP}}\#BIS$-easy graphs, and as a result is the one we discuss in most detail. However, it is valuable to say a little about each of the other reduction techniques in turn, both to convey some idea of how the $\equiv_{\mathsf{AP}}\#BIS$-easiness question has been tackled, and to flag up some possible future areas for research.

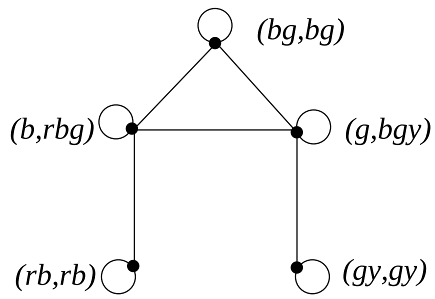
7.3.1 Reduction technique: Directly reducing $\#H$ to $\#H'$ where we already know $\#H' \leq_{\mathsf{AP}} \#BIS$.

One way of showing that a problem $\#H$ is $\equiv_{\mathsf{AP}}\#BIS$-easy is to reduce it directly to some other H-colouring problem $\#H'$ which we know already to be $\equiv_{\mathsf{AP}}\#BIS$-easy. As our collection of $\equiv_{\mathsf{AP}}\#BIS$-easy graphs grows this becomes a more useful technique, but difficulties in directly reducing one graph H to another means that, at present, it is mainly used as an "exploratory" reduction technique. By "exploratory" we mean the practice of taking a graph H' we already have relevant information about - e.g. that $\#H'$ is $\equiv_{\mathsf{AP}}\#BIS$-easy - and transforming it in some AP-reduction legitimate way, probably with gadgetry, to see what graph H comes out. This shows that $\#H \leq_{\mathsf{AP}} \#H'$. As such it is a useful technique for expanding our collection of $\equiv_{\mathsf{AP}}\#BIS$-easy graphs, but because we have no a priori knowledge of what the graph H will look like, it is of limited use in determining the complexity of any one graph in particular. Indeed, as this thesis demonstrates, being able to directly reduce one graph to another (as opposed to comparing their complexity indirectly through classes such as $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$) appears to be the exception rather than the rule.

It can sometimes be informative to search for a direct reduction from a graph H to H' when we already know, from some less direct reduction, that $\#H \leq_{\mathsf{AP}} \#H'$. For example, in due course we will establish that a number of graphs are $\equiv_{\mathsf{AP}}\#BIS$-easy by reduction to $\#DownSets$. Taking one such graph - sample[5] graph 20 from page 253, say - we provide a sketch of a possible alternative $\equiv_{\mathsf{AP}}\#BIS$-easiness proof: direct reduction to the $\equiv_{\mathsf{AP}}\#BIS$-easy graph P_4^*. Suppose the colours of P_4^*, from left-to-right, are r, b, g, y. Let $G = (V(G), E(H))$ be the input to $\#H$, where H is sample graph 20. In the input to $\#P_4^*$, we code up every vertex $u \in V(G)$ as the two disjoint sets $L[u]$ and $R[u]$, where $L[u]$ is a complete graph on s vertices and $R[u]$ is an independent set

[5] To avoid confusion with the 4-vertex H index in the appendix, we prefix references to the $\#DownSets$-generated $\equiv_{\mathsf{AP}}\#BIS$-easy graphs in this chapter with "sample"

of size t, and we connect every vertex in $L[u]$ to every vertex in $R[u]$. We code up every edge $\{u, v\} \in E(G)$ by connecting every vertex in $L[u]$ to every vertex in $L[v]$. Now, we choose s and t such that $2^s 2^t \approx 1^s 3^t$. (In Appendix A.5 we provide a brief technical justification of this technique.) This means there are 5 maximal configurations possible in each $(L[.], R[.])$ pair: (rb, rb), (bg, bg), (gy, gy), (b, rbg), (g, bgy). The permitted adjacencies between $L[.]$ sets thus points out the following colourings, as desired:



In general, however, it remains far from apparent how a direct, gadget-based reduction might work between two given graphs, even when the complexity status of the two graphs is amenable, in theory, to such a reduction.

7.3.2 Reduction technique: $\#MaxBIS$

$\#MaxBIS$ is the problem of counting all the maximum-size independent sets in a bipartite graph G. (In [8] $\#MaxBIS$ is shown to be AP-interreducible with $\#BIS$, and we use $\#MaxBIS$ in Lemma 2.8 on page 77.) We have attempted reducing various $\#H$ problems to $\#MaxBIS$ (through various deployments of gadgetry and so on), but this has not yielded any insights that have not been revealed by our $\#DownSets$ reductions, which we discuss in due course. The main reason for mentioning $\#MaxBIS$, however, is a potentially related result (discovered by DGGJ and communicated informally to this author) concerning the non-bipartite analogue of $\#MaxBIS$, $\#LargeIS$. ($\#LargeIS$ is discussed in Section 2.1.3.) We already know that, because $\#LargeIS \equiv_{\mathsf{AP}} \#SAT$, $\#H \leq_{\mathsf{AP}} \#LargeIS$ is immediate for all H. However, this is an indirect result (see [8]). DGGJ have made an elegant observation that allows the construction of a direct reduction from $\#H$ to $\#LargeIS$ for bipartite H and non-bipartite H with at least one loop. (We explain shortly why it does not seem to apply to Hell and Nešetřil graphs.)

This is the reduction.

Let G be an input to $\#H$. If H is bipartite but G is non-bipartite (which we can easily check) we simply return zero. Otherwise, let $l = |V(H)|$, and arbitrarily enumerate the vertices of $V(H)$ as $c_1, c_2, ..., c_l$. We code up G', our input to $\#LargeIS$, as follows.[6] For each vertex $u \in V(G)$, we introduce $K[u]$ which is a copy of K_l and label its l vertices as $u_1, u_2, ..., u_l$. For each edge $\{u, v\} \in E(G)$, and for all $1 \leq i, j \leq l$, we connect u_i to v_j iff $\{c_i, c_j\} \notin E(H)$. We claim that $\#H(G) = \#LargeIS(G')$. To see this, note that a maximum-size independent set in G' contains $|V(G)|$ vertices: each $K[u]$ must contain exactly one vertex IN the independent set. (This is because, owing to our constraints on G and H, $\#H(G)$ is guaranteed to be greater than zero.) Each maximum-size independent set in G' maps to precisely one $H(G)$ colouring, as follows. For each vertex $u \in V(G)$, u is coloured c_i iff u_i is the vertex in $K[u]$ that is IN the independent set. □

The above reduction does not work for Hell and Nešetřil graphs because there is a risk with such graphs that $\#H(G) = 0$. That is, if $\#H(G) = 0$, it may be that the largest independent set in G' has size less than $|V(G)|$, causing the reduction to return a spurious non-zero answer. In this context this is not much of a problem, because it is the broader reduction concept we are interested in, rather than a complete result.

As we discuss later in Section 7.4, we conjecture that there exist bipartite (and non-bipartite) H which are harder than $\equiv_{\mathsf{AP}}\#BIS$. However, it still remains a possibility that this is incorrect and that all bipartite H *are* reducible to $\#BIS$. Either way, it could help understanding of this issue to consider the feasibility of developing a bipartite analogue of the above reduction i.e. for all bipartite H, $\#H \leq_{\mathsf{AP}} \#MaxBIS$. Though we have not looked into this issue in detail, preliminary experiments with $\#MaxBIS$ suggest that a direct port of the above reduction technique is perhaps unlikely. The

[6]Technically speaking, the input to $\#LargeIS$ should also include the natural number m, where G' is guaranteed to have no independent set of size larger than m. As we justify shortly, taking $m = |V(G)|$ is suitable for this purpose.

$\#LargeIS$ reduction works because it uses the maximum-size independent set requirement in combination with complete graphs to "force" each $K[.]$ into one of $|V(H)|$ distinct states. In particular, when one vertex from a $K[.]$ is IN the independent set, it forces all other vertices to be OUT. It's not clear whether maximum-size independent sets in bipartite graphs (and gadgets) can be forced in this manner. The central issue seems to be that a vertex IN the independent set does not have much influence over those vertices on the same side of the bipartition. The switch to the bipartite domain seems to strip $\#IS$ of much of its "forcing" power, which bolsters the conjecture that in terms of completeness, $\#BIS$ is not the bipartite-world equivalent of $\#IS$.

On that note, if $\#MaxBIS$ (and thus $\#BIS$) is not hard for bipartite graphs, it begs the question as to whether there is some problem - easier than $\equiv_{\mathsf{AP}}\#SAT$ - that *can* code up all bipartite H-colouring problems directly, perhaps mimicking the way that $\#LargeIS$ can be used to directly code up many H-colouring problems. This could be an interesting topic to pursue as further work.

7.3.3 Reduction technique: Logic

In [8] it is noted that it is possible to code up certain H-colouring problems (and other counting problems) within a restricted logical framework, thereby proving that (for those graphs H for which this is possible) $\#H\leq_{\mathsf{AP}}\#BIS$.

To be more precise, Saluja, Subrahmanyam and Thakur [26] have shown how every problem in $\#P$ is equivalent to the task of counting models which satisfy some corresponding first-order logical expression. They have noted that, while all problems in $\#P$ can be coded up with some syntactically unrestricted first-order logical expression, adding syntactic restrictions can have the effect of restricting the set of counting problems that can be coded up to some subclass of $\#P$. They have also proven that, in terms of expressibility, there is a genuine containment hierarchy based on the depth to which quantifiers (i.e. universal and existence quantifiers) are used in the first-order formula. (The following notation is highly specific to the field of logical expressibility and the reader is encouraged to consult [26] and [8] for further information.) For example, the problem $\#IS$ can be coded up as follows:

$$f_{IS}(A) = |\{(I) : A \vDash \forall x, y.x \sim y \Rightarrow \neg I(x) \vee \neg I(y)\}|$$

In other words, if in the above example we let the quantifier range over a set of finite elements representing the vertices of an input graph G, and $\sim$ represents the adjacency relationship in G, we are asking how many ways there are to define a unary relation $I(.)$ over the universe of elements such that the logic expression holds for all pairs of vertices. Essentially therefore, $I(x)$ iff x is in the independent set. The fact that the outermost quantifiers are universal with no existence quantifiers occurring inside them therefore puts $\#IS$ in the class $\#\Pi_1$. (If there are no universal or existence quantifiers in the expression representing a counting problem then the problem is in $\#\Sigma_0$ or $\#\Pi_0$, which are (trivially) identical classes. Similarly, if there are only existence quantifiers used then the problem is in $\#\Sigma_1$. Problems that can be expressed with existence-universal nesting are in $\#\Sigma_2$ and those that can be expressed with universal-existence nesting are in $\#\Pi_2$.) The result of Saluja et al says:

$$\#\Sigma_0 = \#\Pi_0 \subset \#\Sigma_1 \subset \#\Pi_1 \subset \#\Sigma_2 \subset \#\Pi_2 = \#P$$

In [8] a class $\#RH\Pi_1 \subset \#\Pi_1$ is defined, which comprises the restriction of $\#\Pi_1$ as follows. The unquantified part of the expression must be in CNF, and each clause of the unquantified part has at most one unnegated occurrence of any relation symbol being "counted" and at most one negated occurrence of any relation symbol being "counted". Hence, clauses contain at most two relation symbols and never contain two negated or two unnegated relation symbols. (This is where $\#RH\Pi_1$ gets its name from: "restricted Horn" clauses.) Note that relation symbols such as $\sim$ in the above example are not affected by this restriction: informally, they are pre-defined by the input and are not free in the expression.

It is shown in [8] that the $\equiv_{\mathsf{AP}}\#BIS$ problem $\#1P1NSAT$ is complete for the class $\#RH\Pi_1$, where $\#1P1NSAT$ is the variant of $\#SAT$ where all clauses are in CNF

and have at most one unnegated literal per clause and at most one negated literal per clause. The hardness of $\#1P1NSAT$ is demonstrated with respect to parsimonious reducibility, which is obviously a special type of AP-reduction, so the result follows that all problems in $\equiv_{\mathsf{AP}}\#BIS$ are hard for $\#RH\Pi_1$. However, completeness requires membership of the class in question and [8] obliges by putting forward the relevant first-order formulae for a number of the $\equiv_{\mathsf{AP}}\#BIS$ problems in that paper. For example, in terms of $\#RH\Pi_1$ the hardness of *#2-WR* is immediate (by its AP-interreducibility with $\#1P1NSAT$) but completeness follows because it can be expressed by the following formula:

$$f_{\textit{2-WR}}(A) = |\{(C_1, C_2) : A \vDash \forall x, y.(C_1(x) \Rightarrow C_2(x)) \wedge ((C_1(x) \wedge x \sim y) \Rightarrow C_2(y))\}|$$

(We assume quantification ranges over the set of vertices in the input graph G and $\sim$ represents the adjacency relationship of G.) The restriction on clauses is obeyed because the unquantified part of the above formula can be rearranged as

$$(\neg C_1(x) \vee C_2(x)) \wedge (\neg C_1(x) \vee \neg x \sim y \vee C_2(y))$$

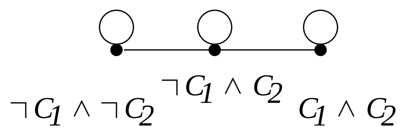
To see why this is equivalent to counting *2-WR* colourings, note that a "vertex" of G (represented by the element z, say) can be in one of three different states, which we use to represent the three different colours of *2-WR*:

$$\neg C_1(z) \wedge \neg C_2(z)$$

$$\neg C_1(z) \wedge C_2(z)$$

$$C_1(z) \wedge C_2(z)$$

Now, consider the permitted "adjacencies" between the three "colours" listed above:



Now, note that though all problems in $\equiv_{\mathsf{AP}}\#BIS$ are hard for $\#RH\Pi_1$, it is probably the case that not all problems in $\equiv_{\mathsf{AP}}\#BIS$ are in $\#RH\Pi_1$. For example, it is not clear how the $\equiv_{\mathsf{AP}}\#BIS$ problem $\#MaxBIS$ might be expressed under the restrictions of the class. Hence, it follows that if we show a counting problem $\#X$ to be in $\#RH\Pi_1$ by demonstration of an appropriate first-order formula, we automatically obtain $\#X \leq_{\mathsf{AP}} \#BIS$ (by the completeness of $\#1P1NSAT$ for the class), but the discovery that a counting problem is inexpressible within $\#RH\Pi_1$ does not necessarily mean the problem is *not* in $\equiv_{\mathsf{AP}}\#BIS$.

Hence, expressing problems in the format required by $\#RH\Pi_1$ is a technique for proving $\equiv_{\mathsf{AP}}\#BIS$-easiness. In this regard, it may not come as a surprise to the reader that we have not yet encountered a graph we can express within $\#RH\Pi_1$ that we cannot prove $\equiv_{\mathsf{AP}}\#BIS$-easy by reduction to $\#DownSets$. It is not too difficult to explain why this is. The completeness of $\#1P1NSAT$ for $\#RH\Pi_1$ is a fairly natural result; informally, we can see that a formula from $\#RH\Pi_1$ can be transformed into an instance of $\#1P1NSAT$ by explicitly instantiating pre-determined relations (such as the adjacency relationship), expanding out quantifiers and replacing relation symbols with multiple literals. Given that $\#1P1NSAT$ and $\#DownSets$ are effectively the same problem, the link between the logic and $\#DownSets$ becomes apparent.

In light of this, the reader may be wondering why we have bothered mentioning this "logic" method at all. Even though $\#DownSets$ is complete for $\#RH\Pi_1$, there is no guarantee that our explicit reductions to $\#DownSets$ are using the problem to its full power. Hence, it remains a possibility that there exist graphs which lend themselves more readily to expression within $\#RH\Pi_1$ than reduction to $\#DownSets$, and for this reason the "logic" technique could still play an important role.

There is one further point to make about $\#RH\Pi_1$. It is tempting to wonder whether we can "tweak" $\#RH\Pi_1$ to produce a logically-defined class that corresponds (in some way) to problems harder than $\equiv_{\mathsf{AP}}\#BIS$ but easier than $\equiv_{\mathsf{AP}}\#SAT$. (This is particularly tempting in light of our discussion of a "complexity gap" later in the chapter.) The obvious strategy - removing the restriction stipulating at most one negated and unnegated relation symbol per clause - unfortunately does not give us this. To see this, note that as soon as we allow two or more negative relation symbols per clause, we can code up $\#IS$:

$$I(x) \wedge x \sim y \Rightarrow \neg I(y)$$

Here, $I(x)$ means "x is in the independent set." (We have omitted the universal quantification on x and y from the above expression; it could easily be transformed into a formally correct expression if desired.) Similarly, if we allow two positive literals per clause, this also allows us to code up $\#IS$ with clauses of the form

$$\neg I'(x) \wedge x \sim y \Rightarrow I'(y)$$

where $I'(x)$ means "x is *not* in the independent set."[7]

The "crossing" property

Before leaving $\#RH\Pi_1$ and moving onto the $\#DownSets$ analysis in Section 7.3.4, we repeat an interesting $\equiv_{\mathsf{AP}}\#BIS$-easiness lemma (and corollary) that DGGJ have demonstrated informally (and communicated to this author) by proving that the H graphs in question could all be expressed within $\#RH\Pi_1$. (Technically speaking, this result makes a number of $\equiv_{\mathsf{AP}}\#BIS$-easiness results from Chapter 2 - such as Lemma 2.14 - redundant. However, these earlier results are useful in their own right, as simple demonstrations of how such reductions are used, so we have elected to keep the original proofs in.)

Lemma 7.1 [DGGJ informal] *Let $H = (V_L(H), V_R(H), E(H))$ be a bipartite graph. If the vertices of $V_L(H)$ and $V_R(H)$ can be ordered $c_0, ..., c_{|V_L(H)|-1}$ and $c'_0, ..., c'_{|V_R(H)|-1}$ such that for any pair of edges $\{c_i, c'_j\}$ and $\{c_{i+a}, c'_{j-b}\}$ which cross each other (in the sense that a and b are either both positive or negative) H contains the complete bipartite graph on $\{c_i, ..., c_{i+a}\}$ and $\{c'_{j-b}, c'_j\}$, then $\#H \leq_{\mathsf{AP}} \#BIS$.*

[7] On a related note, [8] argue that $\#RH\Pi_1$ is a proper subset of $\#\Pi_1$ because equality between these two classes would yield the unlikely result $\#IS \leq_{\mathsf{AP}} \#BIS$.

Corollary 7.2 *If $\#H$ is shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy by Lemma 7.1, it follows that any weighted variant of H can also be shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy by Lemma 7.1.*

The proofs for these two results are in Appendix A.6. Rather than reproduce the original $\#RH\Pi_1$-based proof, we have re-formulated the proof so that it operates by reduction to $\#DownSets$.

We describe the subset of bipartite H that Lemma 7.1 shows to be $\equiv_{\mathsf{AP}}\#BIS$-easy as the set of bipartite graphs with the "crossing" property. This subset of graphs crops up again shortly in the context of the following section.

7.3.4 Reduction technique: $\#DownSets$

Throughout this period of research, the bulk of what has been newly learnt about $\equiv_{\mathsf{AP}}\#BIS$-easiness has been established thanks to a set of empirical experiments conducted with the $\equiv_{\mathsf{AP}}\#BIS$ problem $\#DownSets$. The technique of coding up H-colouring problems as instances of $\#DownSets$ (and thus proving them $\equiv_{\mathsf{AP}}\#BIS$-easy) has already been demonstrated, for example, in Lemmas 2.7 and 2.11 (pages 77 and 84 respectively.) To recap, the general idea is to code up an input $G = (V(G), E(G))$ (where G is the input to $\#H$) as a single (assuming G is connected) partial order, with each vertex in G coded up uniformly as a particular partial order structure[8] and each edge in $E(G)$ coded up as further partial order constraints between the sections of the partial order representing the two relevant vertices. (For convenience, we henceforth refer to the partial order encoding of a vertex as a *cell*.) The idea then is that, within each cell, there are a constant number of downsets possible, and each one represents a colour. Adjacencies between these "colours" are therefore defined by which downsets can be next to each other.

This idea can be automated using a computer program[9] to iteratively cycle through combinations of cell and edge encodings, thus producing large numbers of H graphs

[8] Except possibly in the case of bipartite G but we return to this issue shortly

[9] Source code is available from the author on request.

for which $\#H \leq_{\mathsf{AP}} \#DownSets$. (This, of course, does not constitute an "exhaustive mapping" of the power of $\#DownSets$. Such experiments are inevitably finite and, as with most reductions in this thesis, represent only one possible style of AP-reduction.)

To date, we have conducted two principle variants of this experiment. In the first (and most general) variant, it is assumed that the input graph G is potentially non-bipartite, which means that the same cell structure has to be used to code up every vertex of G, and the introduction of constraints between two cells (to encode an edge) has to be symmetrically applied. Hence, the H graphs generated by this experiment are themselves non-bipartite, or at least potentially so.[10] In the second experiment, the input graph $G = (V_L(G), V_R(G), E(G))$ is assumed to be bipartite, offering us extra degrees of freedom both in the way the vertices of G are coded up *and* in the way the edges of G are coded up. In particular, vertices from $V_L(G)$ can be coded up with a different cell structure to vertices from $V_R(G)$, and the edge encoding does not have to be symmetric. Hence, the graphs output by this experiment are bipartite.

(Perhaps the simplest example of a graph generated by reduction to $\#DownSets$ via an asymmetric edge encoding is P_4. If we let $G = (V_L(G), V_R(G), E(G))$ be the input to $\#P_4$, we encode each $u \in V_L(G)$ as the partial order element u_0, each $v' \in V_R(G)$ as v'_0, and each edge $\{u, v'\}$ (where $u \in V_L(G)$ and $v' \in V_R(G)$) with constraint $u_0 \prec v'_0$. The edge encoding is asymmetric because it is not also the case that $v'_0 \prec u_0$.)

Owing to a reliance on exhaustive generation, the volume of output and the running time of both experiments increases exponentially with the number of partial order elements used within each cell. There is a second problem that arises because, as the size of the partial orders used to code up vertices increases, the number of colours in the induced graph H tends to increase exponentially, making it difficult to visually analyse the output. For both these reasons the samples of output we now present are drawn

[10] In practice the graphs generated are always non-bipartite, or at least have non-bipartite components, because the empty and full downsets within a cell can always be self-adjacent, introducing looped colours into the induced graph H.

from executions of the experiments where the number of partial order elements used to code up each cell is bounded above by 4 (and, in the case of the bipartite experiment, 3.) Perhaps surprisingly, cells limited to these sizes are still adequately complex to produce some interesting insights into the graphs that can be generated by reduction to $\#DownSets$.

Non-bipartite version of the experiment

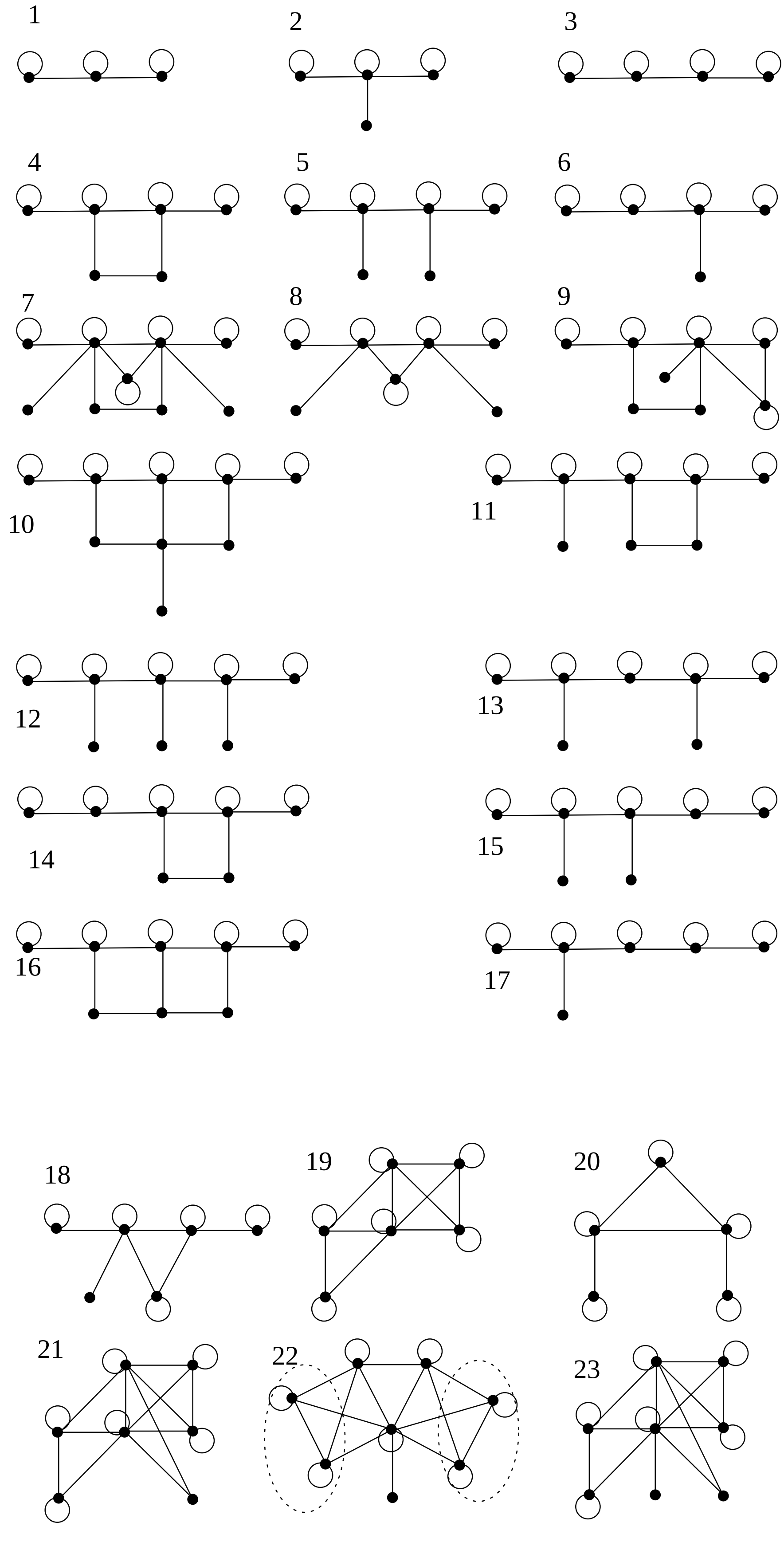
First, we present a selection of 23 graphs generated by the first experiment, with cell size less than or equal to 4. To be formally precise, we have included the cell and edge encodings that give rise to them in Appendix A.7.[11] It is important to stress that this particular sample has been specially chosen for a number of reasons.

1. First, a general observation stemming from both experiments is that many of the graphs generated are *structurally* equivalent. (We say two graphs H_1 and H_2 are structurally equivalent if one is a weighted version of the other.) Indeed, many of the graphs that are output are weighted variants of a much smaller set of graphs which are mutually structurally distinct. In choosing the graphs below, we have tried to communicate the structural variation of the graphs output, and as such we have chosen only to reproduce structurally distinct graphs.

2. Furthermore, given a choice between a number of structurally equivalent graphs, we have endeavoured to reproduce a graph with the lowest "weight metric". The weight metric is a fairly crude measurement and is simply the value obtained by expressing a graph in compact form and then adding up all the weights on the vertices. In other words, the lower the weight metric, the less the graph differs from its unweighted, compact form representation. i.e. the "less weighted" it is. Unless the graph is completely unweighted, the metric does not uniquely specify a particular weighted variant, but that isn't a problem because our main use for it is simply to give us some graph which is as close to unweighted as possible.

[11]We have not included formal proofs for the first three graphs because these are all known to be $\equiv_{\mathsf{AP}}\#BIS$-easy, also by reduction to $\#DownSets$, via results in Chapter 2

3. Also, we do not claim that these are even *structurally* exhaustive for cell size less than or equal to 4; they are reasonably representative but we have omitted a number of graphs both to keep the number of graphs down to a manageable level, and also because some graphs are simply too visually obfuscated for the reader to easily untangle.


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

There are a number of observations that can be made about the above graphs. Evidently, they are of help in positioning some of the $\equiv_{\mathsf{AP}}\#BIS$-easy graphs discovered early on (in [8]) in a wider context. For example, *2-WR*, *2-wrench* and the looped paths can be seen to belong to an infinite family of $\equiv_{\mathsf{AP}}\#BIS$-easy graphs which we have called "prickly" looped paths. This family includes sample graphs 2, 3, 5, 6, 12, 13, 15 and 17; in Appendix A.8 we define the family formally and provide a short proof that all such graphs are $\equiv_{\mathsf{AP}}\#BIS$-easy, using reduction to $\#DownSets$.[12]

Surveying the structure of the graphs more generally, it is curious to note that *2-WR* and *2-wrench* (and weighted variants thereof) are often prominent subgraphs of graphs in the sample, especially if we focus our attention on those subgraphs which lend themselves to being pointed out with the kind of gadgetry used in Chapter 2. Indeed, an informal observation is that many of the more complex graphs in the above sample seem to be amalgamations of much simpler and more "atomic" $\equiv_{\mathsf{AP}}\#BIS$-easy graphs, possibly augmented by extra "add-on" unlooped vertices and edges. (The role of unlooped vertices in non-bipartite $\equiv_{\mathsf{AP}}\#BIS$-easy H seems interesting; could it be a general rule that, if H is $\equiv_{\mathsf{AP}}\#BIS$-easy, so too is H minus all its unlooped vertices?)

Sample graphs 9, 19, 21, 22 and 23 are of interest because in each case the graph contains equivalent vertices. This raises a number of issues pertaining to the significance of weighting in graphs reducible to $\#DownSets$. In the case of sample graph 9, we know that a $\equiv_{\mathsf{AP}}\#BIS$-easy structurally equivalent version of the graph with a lower weight metric is highly unlikely to exist. This is because once the two equivalent colours (i.e. the two looped colours on the right-hand side of the graph) are collapsed into one colour, the resulting graph falls into $\equiv_{\mathsf{AP}}\#SAT$.[13](As an aside, it is interesting to note that - as in sample graph 9 - fully looped cliques in graphs often have the effect of preventing our $\equiv_{\mathsf{AP}}\#SAT$-hardness results from operating, acting almost like

[12]Despite being vaguely similar, the unclassified 4-vertex H known as the "long wrench" - the right-hand side graph in Figure 7.4 (page 271) - does not fit the definition of a "prickly" looped path, so remains unclassified.

[13]To see why, note that a maxdeg gadget could then be used to grab the single degree-5 colour, and the graph pointed out by this colour is $\equiv_{\mathsf{AP}}\#SAT$ because of Lemma 5.1.

a "counterbalance" against $\equiv_{\mathsf{AP}} \#SAT$-hardness.)

Continuing with the theme of weighting, it is not clear what happens to the complexity of sample graphs 19, 21, 22 and 23 if their respective equivalent colours are collapsed into single colours. In this regard, the following *might* be true. Suppose H is some non-trivial, non-bipartite graph containing some equivalent vertices, and H' is a version of H where some or all of the equivalent colours from H have been collapsed into single colours. It may be that, if H is generated by some cell and edge combination on d partial order elements (per cell), and H' has *not* appeared for any cell and edge combination on d or fewer elements, that H' will never appear for any cell and edge combination on $d+1$ or more elements. If this is the case, then sample graphs such as 9, 19, 21, 22 and 23 themselves constitute weak evidence that collapsing colours on these individual graphs any further renders the resulting graph inexpressible within the $\#DownSets$ framework (as it is currently used), which may itself be evidence that such graphs are harder than $\equiv_{\mathsf{AP}} \#BIS$. This could be an interesting issue to explore.

Finally, and perhaps most crucially, there is some evidence that all the graphs in the sample have an approximately "linear" structure. That is, if we disregard the unlooped vertices (which seem to be less of an influence on graph structure than looped vertices), all the above sample graphs (i.e. 1-23) have two identifiable "poles". "Pole" is an intuitive rather than formal concept and may refer to regions of a graph rather than just individual colours; for example, in sample graphs 1-17 the poles are the looped colours at either end of the looped path, and in (say) sample graph 22 the two poles have been highlighted.

This seems to tie in with the observation that there is often a kind of symmetry between the behaviour of colours corresponding to empty and full downsets.[14] Both tend to be of low degree: empty downsets (or, more generally, downsets with few elements) do not

[14]A brief examination of the proofs in Appendix A.7 certainly suggests a correlation between poles and empty/full downsets.

contain enough elements to satisfy the "demands" of other adjacent downsets, whilst full downsets (or downsets with many elements) are often too demanding in terms of the number of elements *they* require of adjacent downsets. We will return to this "linearity" theme shortly.

Bipartite version of the experiment

As mentioned earlier, in this version of the experiment we assume the input graph G is bipartite and hence we are able to code up vertices on the left bipartition of G differently to vertices on the right, and use asymmetric edge encodings.[15] Cell size was restricted to 3 or fewer elements.

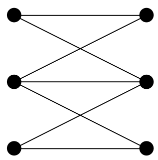
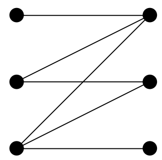
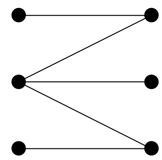
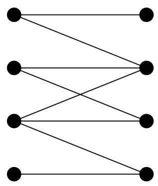
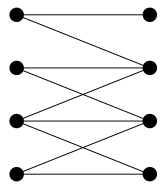
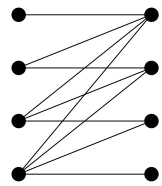
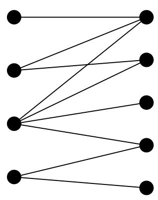
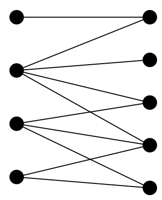
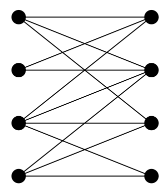
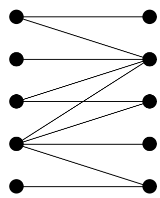
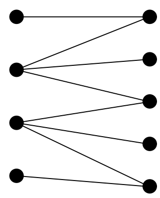
Again, many of the graphs generated by the experiment are structurally equivalent and differ only in weighting. Therefore, as with the previous experiment, we have reproduced only structurally distinct graphs, and in each case endeavoured to reproduce the least weighted (in terms of the weight metric) version encountered (for cell size less than or equal to 3.)

We have reproduced 35 graphs, and divided them into two parts. Sample graphs 1-11 all have the "crossing" property, defined earlier, and hence we already know that they are $\equiv_{\mathsf{AP}}\#BIS$-easy because of Lemma 7.1. (It follows that all weighted versions of these graphs - and, in addition, any connected subgraphs obtained by removing vertices - are $\equiv_{\mathsf{AP}}\#BIS$-easy also.[16]) We should stress that many crossing graphs have been omitted from this sample (e.g. paths and weighted paths) simply because there is little value in reproducing a large number of graphs we have already shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy. Sample graphs 12-35 do *not* have the crossing property, so in Appendix A.9 we formally reproduce the partial orders that give rise to these graphs. (We

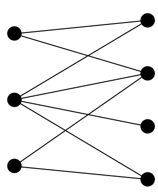
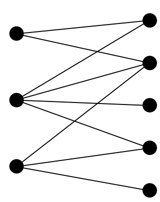
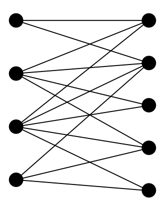
[15] Technically, therefore, this shows that $\overleftarrow{\#H}\leq_{\mathsf{AP}}\#DownSets$, but given that $\#H\leq_{\mathsf{AP}}\overleftarrow{\#H}$ (by Lemma 2.15) it follows that $\#H\leq_{\mathsf{AP}}\#DownSets$ also.

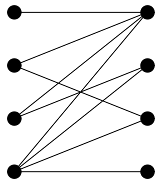
[16] As part of the proof for Corollary 7.2 it is observed that the crossing property is not lost if a vertex and all those edges connected to it are removed.

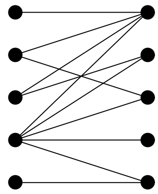
have reproduced more non-crossing graphs than crossing graphs simply because the latter category is more interesting, not because of any particular relationship between the frequencies with which they appear.)

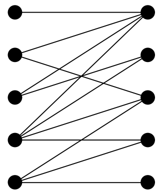

1

2

3

4

5

6

7

8

9

10

11

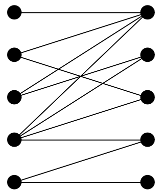
The following graphs (12-35) do *not* have the crossing property. (This has been verified using a computer program to exhaustively check all permutations of the vertices.) The relevant $\#DownSets$-proofs are listed in Appendix A.9.

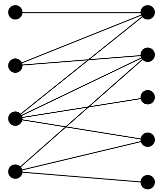

12

13

14

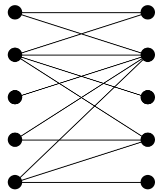

15

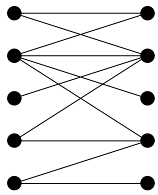

16

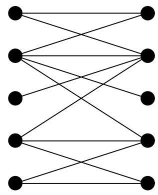

17

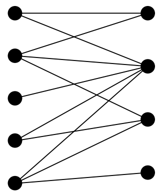

18

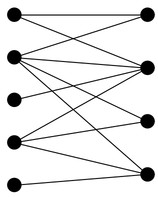

19

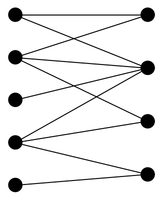

20

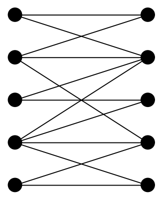

21

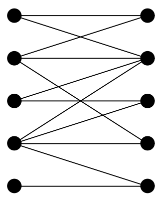

22

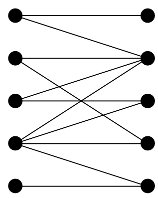

23

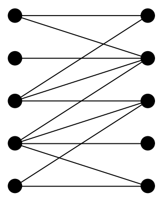

24

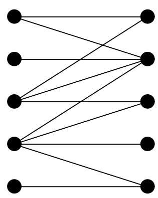

25

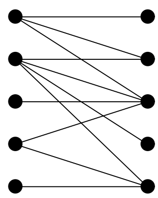

26

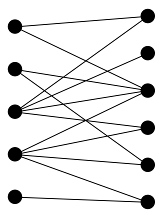

27

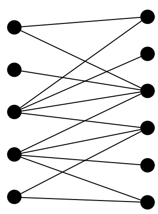

28

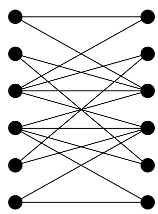

29

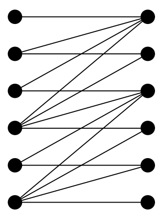

30


31


32


33


34


35

Comments

A number of observations are immediate. The fact that many of the graphs from the sample are non-symmetric - where symmetric is defined in Section 2.4.2 - confirms that exploiting the degree of freedom that bipartite input affords us does, as might be expected, allow a wider array of graphs to be reduced to $\#DownSets$. With regard to the symmetric graphs in the sample, those symmetric graphs that have symmetrical encodings (i.e. they do not exploit the assumption that G is bipartite) could instead have been coded up in the more general non-bipartite version of the experiment. (For example, sample graph 34 is the bipartite version of sample graph 4 from page 253.) Note, however, that a symmetric H may have a (necessarily) asymmetric $\#DownSets$ encoding; to take a key example, the reduction of the symmetric P_4 to $\#DownSets$ - discussed a couple of pages back - uses an asymmetric $\#DownSets$ encoding. It is highly unlikely that a symmetric encoding of P_4 exists, because then the encoding could be applied to non-bipartite H and thus yield the unlikely result of $\#IS\leq_{\mathsf{AP}}\#DownSets$. Encodings aside, it is clear that a number of the symmetric graphs from the sample are bipartite versions of known $\equiv_{\mathsf{AP}}\#BIS$-easy graphs - for example sample graph 15 is the bipartite version of *2-wrench* - and as such their $\equiv_{\mathsf{AP}}\#BIS$-easiness status was already apparent from the $\#bi(H)\leq_{\mathsf{AP}}\#H$ relationship.

More pertinently, sample graphs 12-35 obviously demonstrate that the "crossing" property does not completely characterise the set of bipartite H that can be generated by reduction to $\#DownSets$. However, upon closer inspection quite a few of the graphs lacking the crossing property *almost* have it. For example, in many cases the graphs would have the crossing property if all degree-1 vertices were removed: this is the case in sample graphs 12, 13, 15, 19, 22, 31 and so on. Indeed, it would be an interesting exercise to try and build upon the definition of "crossing" and attempt a broader general categorisation of bipartite graphs that can be constructed by reduction to $\#DownSets$. (It could be particularly interesting to examine those symmetric bipartite H which do not have the crossing property, but which we know can be reduced to $\#DownSets$ because they are the bipartisation of a non-bipartite graph H' already shown to be reducible to $\#DownSets$.)

It seems likely, however, that expanding a general definition of which bipartite H are reducible to $\#DownSets$ will be difficult, in much the same way as (notwithstanding the rigidly-defined "prickly" looped paths) it is difficult to formally pin down any recurring, characteristic traits in the graphs on page 253. Indeed, we note that even amongst the small sample of bipartite graphs above, some graphs (such as 14 and 28) fail the crossing property for more complicated reasons than simply the existence of spurious degree-1 vertices.

In terms of general comments about the graphs, it again seems that the graphs have a vaguely "linear" structure i.e. there appears to always be two distinct poles to the graph. In the non-bipartite graphs on page 253, this is characterised by the prominence of subgraphs which are looped (and potentially weighted) paths, as well as structures which are recognisably close, if not structurally equivalent, to (weighted) looped paths. Similarly, the recurring theme of these bipartite graphs seem to be that they have prominent subgraphs which are the bipartite versions of the prominent characteristics just described in the non-bipartite graphs. (For example, note the recurring subgraphs $bi(P_3^*)$ and $bi(P_4^*)$).

This is of relevance to the following section.

7.4 A conjecture about $\equiv_{\mathsf{AP}}\#BIS$

Rather than continue with vague, hand-waving attempts to define what we mean by "linear" in the context of graph structure, it is more helpful to provide a concrete example of a bipartite graph that we believe is beyond the reach of $\#DownSets$ and, moreover, beyond the reach of $\equiv_{\mathsf{AP}}\#BIS$:

This graph - the "junction" - is in many ways a test-case for our conjecture that

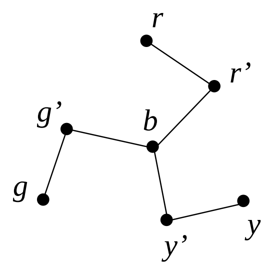


Figure 7.2: *The junction*

there are indeed graphs in a gap between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$. We suspect that it lies beyond $\equiv_{\mathsf{AP}}\#BIS$ because, unlike the vaguely "linear", bi-polar structure of the graphs output by our experiments, it has *three* poles. We do not think it is possible to reduce graphs with three or more poles to $\equiv_{\mathsf{AP}}\#BIS$. (We would not describe the $\equiv_{\mathsf{AP}}\#BIS$-easy *2-wrench* as tri-polar because its unlooped vertex does not seem significant enough, in light of what was said earlier about apparent inferiority of unlooped vertices in non-bipartite $\equiv_{\mathsf{AP}}\#BIS$-easy graphs, to constitute a pole.)

Despite many attempts we have been unable to show this graph to be $\equiv_{\mathsf{AP}}\#BIS$-easy. Note that, if just one of its degree-1 vertices was removed, the resulting graph would be $\equiv_{\mathsf{AP}}\#BIS$-easy because it would have the crossing property. However, as it stands the graph does not have the crossing property, and this is just one of many dead ends we have encountered in trying to pigeonhole the graph. (It is $\equiv_{\mathsf{AP}}\#BIS$-hard by Lemma 2.19.) It is revealing to note that, although we have observed that the crossing property only seems to identify a potentially fairly small subset of $\equiv_{\mathsf{AP}}\#BIS$-easy bipartite graphs, the junction lacks the crossing property *precisely because* it has three poles. That is, if we consider a vertex in a bipartite graph, three seems to be a threshold number in terms of the crossing property, because three or more paths (each of length ≥ 2) originating from the same initial vertex but otherwise disjoint are inevitably going to cross at some point. Edge crossing does not violate the crossing property in its own right, of course, but the absence of sufficient extra edges means the necessary complete bipartite graphs between crossing edges (in line with the definition of crossing) are simply not present.

For these reasons, it seems very important that the complexity of the junction is determined. As we discuss in Section 5.6, the fact that the graph is bipartite seems to render it unlikely that the graph is $\equiv_{\mathsf{AP}}\#SAT$. This, of course, brings us back to the overriding question: if there are one or more complexity classes between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$, how are they characterised? In Section 7.6 we have (just) begun to tackle this question, by establishing (where possible) reductions between the junction and other graphs that stubbornly refuse classification.

Another important, unclassified graph is C_6, the cycle on six vertices. (The $\equiv_{\mathsf{AP}}\#BIS$-hardness of C_6 was established in [8].) This is particularly significant because (as shown by Figure 2.6 on page 55) $\#C_6$ is the same problem[17] as *#bi-3-col*, which is the first non-trivial problem in the bipartite q-colouring hierarchy, discussed further in Section 7.6. This graph also lacks the the crossing property, and in fact it is not too difficult to show that all *#bi-q-col* problems lack the crossing property for $q \geq 3$. Clearly, C_6 has neither what could be described as a "linear" or multi-polar structure, so the complexity link between *#bi-3-col* and the junction is another area that needs investigating; as Section 7.6 shows, we are confident that $\#junction$ is AP-reducible to *#bi-4-col*, but as yet we have no reduction in either direction between $\#junction$ and *#bi-3-col*.

7.5 Bipartite H and non-bipartite H

The possibility has been explored that some bipartite H (along with some equally elusive non-bipartite H) sit somewhere between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$. Thus, as far as bipartite H is concerned, that discussion is about possible differences in complexity *within* the bipartite domain. In this section we take a step back and consider briefly two transformations that seem important in terms of generally comparing bipartite H with non-bipartite H. (In this respect, this section is related to Section 5.6 which considers whether any bipartite H can ever be $\equiv_{\mathsf{AP}}\#SAT$.)

[17] Bar an insignificant factor of 2

7.5.1 The bipartisation transformation: $bi(H)$

The relationship $\#bi(H) \leq_{\mathsf{AP}} \#H$ has already been looked at on a number of occasions throughout this thesis. In many ways it is the "bridge" between the bipartite and non-bipartite world, showing that every non-bipartite H has a (symmetric) bipartite counterpart that is no higher than it in the complexity hierarchy. There are, however, many questions about the $bi(H)$ transformation that are yet to be answered. For example, under what circumstances is $\#bi(H)$ of the same complexity as $\#H$, and under what circumstances is $\#bi(H)$ strictly easier than $\#H$? We already have numerous examples of where $\#bi(H)$ is strictly easier than $\#H$, assuming that $\equiv_{\mathsf{AP}}\#SAT$ is harder than $\equiv_{\mathsf{AP}}\#BIS$. The obvious example is $\#IS$ and $\#BIS$, but there are many more. (For example, all the $\equiv_{\mathsf{AP}}\#SAT$ 3-vertex H, apart from K_3 and the graph resembling *2-WR* minus the centre loop, have $\equiv_{\mathsf{AP}}\#BIS$-easy $bi(H)$ counterparts.) In terms of non-bipartite H for which $\#bi(H) \equiv_{\mathsf{AP}} \#H$, all the non-bipartite $\equiv_{\mathsf{AP}}\#BIS$ graphs are obvious candidates, given that $\equiv_{\mathsf{AP}}\#BIS$ is probably the lowest (provisionally) non-$FPRAS$able complexity class. The questions become particularly relevant, however, if indeed there is daylight between $\equiv_{\mathsf{AP}}\#BIS$ and $\equiv_{\mathsf{AP}}\#SAT$. For example, if there are multiple complexity classes, how does a non-bipartite H in any given class behave under the bipartisation transformation - does $\#bi(H)$ stay put, or does it slide down the complexity hierarchy, and in which case how far does it slide?

7.5.2 The $EdgeSwap$ transformation

On page 54 we repeat a proof from [8] that $\#$*2-WR*$\leq_{\mathsf{AP}}\#P_4$. The reduction used in that proof is what we call the *edge swapping* reduction. We formalise this reduction technique and then discuss why it seems to be of particular importance with regard to the complexity of bipartite H.

Let $H = (V_L(H), V_R(H), E(H))$ be any non-trivial, connected bipartite graph. We define $EdgeSwap(H) = H'$ as follows. Suppose we arbitrarily enumerate the edges of $E(H)$, and that there are k such edges: we let the ordered pair $(l_i, r_i) \in V_L(H) \times V_R(H)$

represent the ith edge. Now, we let the vertex set of H' be $\{c_1, ..., c_k\}$, and state that $\{c_i, c_j\} \in E(H')$ iff $\{l_i, r_j\} \in E(H)$ and $\{l_j, r_i\} \in E(H)$.

Observation 7.3 *If H is a non-trivial, connected bipartite graph, then $EdgeSwap(H) = H'$ is a non-trivial, non-bipartite, connected, fully-looped graph.*

Proof. First, note that H' is fully looped because, where $j = i$, it is already known that $\{l_i, r_j\}$ and $\{l_j, r_i\}$ are in $E(H)$. Given that H' contains looped colours, we immediately see that it is non-bipartite. Now, it is slightly more involved to prove that H' is connected. Let c_i and c_j be vertices from $V(H')$, and let (l_i, r_i) and (l_j, r_j) be the ordered pairs they correspond to, as defined above. Since H is connected, it follows that there is at least one path in H between l_i and l_j. If we pick one of these paths arbitrarily, it can be represented by the vertex list $l_i, x_1, ..., x_m, l_j$ where x_1,...x_m are the intermediate vertices on the path. (Since l_i and l_j are both in $V_L(H)$ it follows that m is odd.) Now, we argue that there must be a path between c_i and c_j in H'. To see this, note that the following is a valid path in H': (l_i, r_i), (l_i, x_1), (x_2, x_1), (x_2, x_3), ..., (x_{m-1}, x_m), (l_j, x_m), (l_j, r_j).

Finally, we prove that H' is non-trivial. Given that H is non-trivial (and connected) we know there must exist some $\{l_i, r_i\}$ and $\{l_j, r_j\} \in E(H)$ (where $l_i, l_j \in V_L(H)$ and $r_i, r_j \in V_R(H)$) such that $\{l_i, r_j\} \notin E(H)$. Thus, if c_i, c_j are the vertices in H' representing (l_i, r_i) and (l_j, r_j) respectively, it follows that $\{c_i, c_j\} \notin E(H')$. Now, given that H' contains a loop and is connected, it can only be trivial if it is a fully looped clique, but this is not possible because we have just shown that at least one edge is not present. □

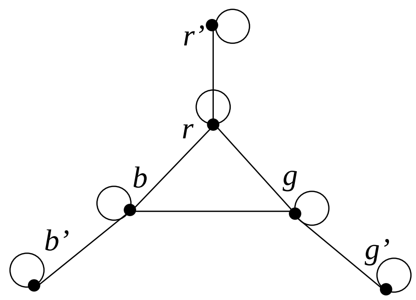
Observation 7.4 *For all bipartite H, $\#EdgeSwap(H) \leq_{AP} \#H$.*

Proof. Let H' be the graph $EdgeSwap(H)$, and let G be the input to $\#H'$. We build G', our input to $\#H$, as follows. For each $u \in V(G)$ we introduce two vertices $T[u]$ and $B[u]$, which we connect together. For each edge $\{u, v\} \in E(G)$ we connect

$T[u]$ to $B[v]$ and $T[v]$ to $B[u]$. (Note that G' is bipartite.) Now, by our definition of $EdgeSwap(H)$ it follows that $\#H'(G) = (1/2)\#H(G')$. (It does not matter whether the $T[.]$ vertices take colours from the left or right side of the bipartition; both induce the same graph, which is where the factor of 2 comes from.) $\square$

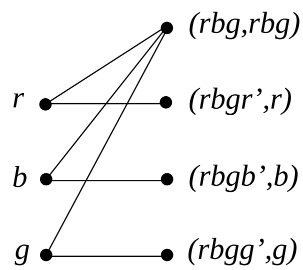
The interesting point that emerges from this Observation is that, for every non-trivial bipartite H, there exists a non-trivial, non-bipartite graph which is no harder than H. Given our suspicion that it is unlikely a $\equiv_{\mathsf{AP}}\#SAT$ bipartite H will be found, this raises the question of where exactly these $EdgeSwap(H)$ graphs actually lie in the complexity hierarchy. The fact that their structure is derived from bipartite origins gives them a curious structure which (unsurprisingly) seems to make them impervious to all of the $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions we have deployed thus far.

Are there any graphs H for which $EdgeSwap(H)$ is *strictly* easier than H, in terms of AP-reducibility? Unlike the $bi(H)$ transformation we do not yet have any positive examples of this being the case. For $H\equiv_{\mathsf{SP}}BIS$ the answer is no (because of Theorem 4.1), and for $\#H\equiv_{\mathsf{AP}}\#BIS$ the answer is, for reasons we have already discussed, also likely to be no. However, it is interesting to focus on those bipartite H (and subsequent $EdgeSwap(H)$) for which H does not seem to fit into $\equiv_{\mathsf{AP}}\#BIS$, such as the junction. It may in fact be that, for all non-bipartite H, $EdgeSwap(H)$ is of the *same* complexity as H, but we do not have enough information to conjecture either way. However, it is perhaps significant that the junction and its $EdgeSwap$ graph *do* seem to be of the same complexity. If we let H be the junction and let H' be $EdgeSwap(H)$ i.e.



the following is a sketch of a possible proof for showing that $\#H \leq_{\mathsf{AP}} \#H'$:

Let $G = (V_L(G), V_R(G), E(G))$ be the input to $\#H$. For each $u \in V_L(G)$, we introduce a vertex $X[u]$. For each $v \in V_R(G)$, we introduce a set $L[v]$ comprising s disjoint vertices and a set $R[v]$ which comprises t disjoint vertices; we connect every vertex in $L[v]$ to every vertex in $R[v]$. Next, we introduce a large complete graph on k vertices (which we call K) and connect every vertex in K to all $X[.]$ vertices and also to every vertex in each $R[.]$ set. Now, for each edge $\{u, v\} \in E(G)$ (where $u \in V_L(G)$ and $v \in V_R(G)$) we connect $X[u]$ to every vertex in $L[v]$. The idea is that, if we ensure s and t are chosen such that[18] $3^s 3^t \approx 4^s 1^t$, junction colourings are pointed out as follows. Since K is chosen to be very large (compared to the rest of the constructed graph G') it is exponentially likely to be coloured $\{r, b, g\}$. Hence, each $X[.]$ is restricted to one colour from $\{r, b, g\}$ and each of the $R[.]$ sets to some subset of $\{r, b, g\}$. Owing to our choice of s and t, it follows that each $(L[.], R[.])$ pair is exponentially likely to be coloured with one configuration from (rbg, rbg), $(rbgr', r)$, $(rbgb', b)$ and $(rbgg', g)$. This, we argue, has the effect of pointing out junction colourings, with (rbg, rbg) acting like the centre colour of the junction:



(Technically this reduction is from wlog left-orientation *junction* colourings, but as we know from Lemma 2.15 this is sufficient to yield the result we need.) This reduction is aided by the fact that the junction has a fairly "homogeneous" structure and thus lends itself to having its vertices coded up with gadgets. Determining whether $\#EdgeSwap(H) \equiv_{\mathsf{AP}} \#H$ for more obfuscated bipartite H is likely to be much more difficult.

[18] In Appendix A.10 we include some technical details explaining how this is done

7.6 Other interesting relationships

A good way of adding substance to conjectures about complexity gaps and so on is to demonstrate reductions (and preferably interreducibility) between graphs that otherwise evade classification. In this penultimate section we look at a few ad-hoc results - a mixture of formal proofs and proof sketches - that establish interesting, although admittedly only rudimentary, relationships between graphs we have already encountered such as C_6, the junction, C_4^* and the crossbow. (The crossbow is graph 50 in the graph index.)

7.6.1 The bipartite q-colouring hierarchy

The following lemma proves that, as we might expect, the complexity of counting proper q-colourings in bipartite graphs does not get any easier as q increases. This is particularly relevant given that C_6, the cycle on six vertices, is equivalent (bar a factor of 2) to the problem of counting 3-colourings in bipartite graphs. (It is also relevant to the graph C_4^*, but we come to that in the next section.) Given that this lemma is more than just an ad-hoc reduction between two graphs (i.e. it establishes a hierarchy) we have decided to formalise the result by presenting the proof, in full, below. It is worth emphasising that there remains much to be learnt about the #*bi-q-col* hierarchy besides this result, because the complexity of the individual #*bi-q-col* problems (for $q \geq 3$) is open, and we have yet to find $i, j \in \mathbb{N}$ (with $3 \leq i < j$) such that #*bi-j-col*$\leq_{\mathsf{AP}}$#*bi-i-col*. Thus, it could be a significant area of future research.

Lemma 7.5 *[Dyer, Goldberg, Kelk] For $q \geq 1$, #bi-q-col $\leq_{AP}$ #bi-(q+1)-col.*

Proof. Immediate from Lemmas 7.6 and 7.7, which follow shortly. $\square$

Thus, to prove Lemma 7.5, we first have to prove two intermediate Lemmas. To assist with this, we adopt the following labelling conventions. Let H_q be the graph defined as follows. Let $V_L(H_q) = \{c_1, ..., c_q\}$, $V_R(H_q) = \{c'_1, c'_2, ..., c'_q\}$ and say that $\{c_i, c'_j\} \in E(H_q)$ iff $i \neq j$. It follows that, for bipartite G, $\#H_q(G) = 2$#*bi-q-col*(G), so $\#H_q \equiv_{\mathsf{AP}} \#$*bi-q-col*. Now, we introduce a graph that corresponds to *semi-partial* bipartite q-colouring. More specifically, let $V_L(H_{sp(q)}) = \{d_0, d_1, ..., d_q\}$, $V_R(H_{sp(q)}) = \{d'_1, ..., d'_q\}$ and say that $\{d_i, d'_j\} \in E(H_{sp(q)})$ iff $i \neq j$.

Lemma 7.6 *[Dyer, Goldberg, Kelk] For* $q \geq 1$, $\#H_{sp(q)} \leq_{\mathsf{AP}} \#$*bi-(q+1)-col.*

Proof. For $q = 1$, $\#H_{sp(q)}$ is $FPRAS$able so we focus our attention on $q \geq 2$. We are actually going to demonstrate $\overleftarrow{\#H}_{sp(q)} \leq_{\mathsf{AP}} \#H_{q+1}$ but this is adequate because (by Lemma 2.15) we know that $\#H_{sp(q)} \leq_{\mathsf{AP}} \overleftarrow{\#H}_{sp(q)}$ and separately we know from above that $\#H_{q+1} \equiv_{\mathsf{AP}} \#$*bi-(q+1)-col*. So, let $G = (V_L(G), V_R(G), E(G))$ be the input to $\overleftarrow{\#H}_{sp(q)}$. We code up G' as follows. Let $V_L(G') = V_L(G) \cup \{x\}$ and $V_R(G') = V_R(G)$ where x is some new vertex not in $V_L(G)$ or $V_R(G)$. The edge set of G' is the same as the edge set of G except that, in addition, x is connected to every vertex in $V_R(G')$. Now, suppose we colour G' with H_{q+1}, and x is coloured (say) c_1. Then $V_R(G')$ is restricted to the q colours $V_R(H_{q+1}) \setminus \{c'_1\}$, while the vertices $V_L(G') \setminus \{x\}$ can be coloured with the $q+1$ colours from $V_L(H_{q+1})$. In other words, c_1 acts as the universal colour when it appears in $V_L(G') \setminus \{x\}$ because a vertex in $V_L(G')$ coloured c_1 can be adjacent to any vertex in $V_R(G')$. Given that x can be coloured with any colour from H_{q+1}, and all have the same effect, it follows that $\overleftarrow{\#H}_{sp(q)}(G)2(q+1) = \#H_{q+1}(G')$, and hence $\overleftarrow{\#H}_{sp(q)} \leq_{\mathsf{AP}} \#H_{q+1}$. $\square$

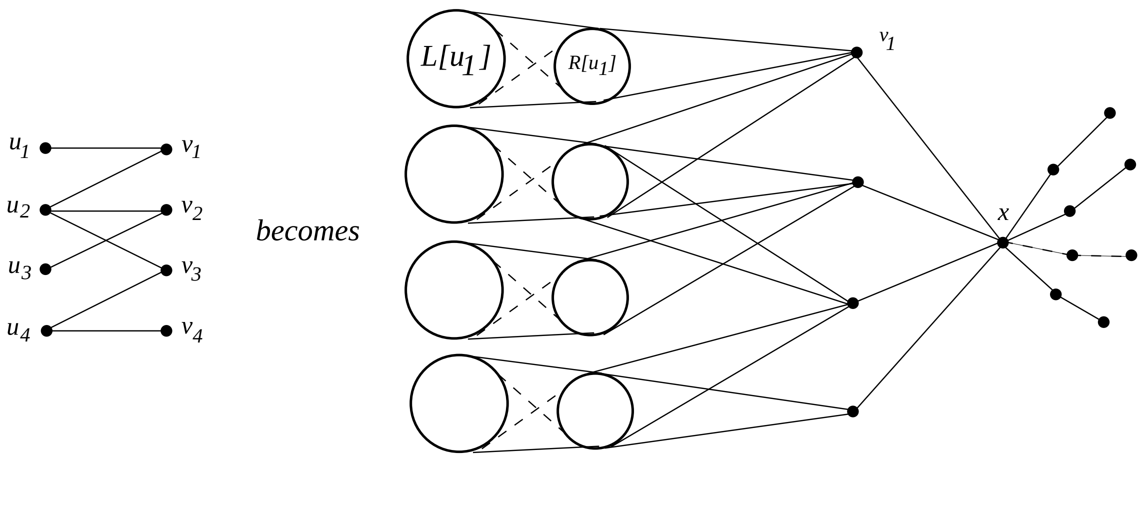


Figure 7.3: *How G' is coded up in Lemma 7.7*

Lemma 7.7 *[Dyer, Goldberg, Kelk] For $q \geq 1$, #bi-q-col $\leq_{AP}$ $\#H_{sp(q)}$*

Proof. For $q < 3$, *#bi-q-col* is $FPRAS$able so we assume $q \geq 3$. Now, assume $G = (V_L(G), V_R(G), E(G))$ is the input to *#bi-q-col*; we build G' as follows. For each $u_i \in V_L(G)$, we introduce disjoint sets $L[u_i]$ and $R[u_i]$ of size s and t respectively. We connect every vertex of $L[u_i]$ to every vertex in $R[u_i]$. For every vertex $v_i \in V_R(G)$, we simply introduce a copy of v_i. Now, for every edge $\{u_i, v_j\} \in E(G)$ (where $u_i \in V_L(G)$ and $v_j \in V_R(G)$), we connect every vertex in $R[u_i]$ to our copy of v_j. Having coded up vertices and edges, we introduce a brand new vertex x which we attach to all our copies of vertices from $V_R(G)$, and also attach k length-2 paths to x (see diagram) where k is to be determined.

Assuming our conventions from earlier, we say that a full colouring of G' (by $H_{sp(q)}$) is one where x is coloured with the universal colour from $H_{sp(q)}$ (i.e. d_0) and, for all $u_i \in V_L(G)$, $R[u_i]$ is coloured exactly with d_0 plus one other colour from $\{d_1, ..., d_q\}$. Note that every full colouring points out a *bi-q-col* colouring. To see this, note that when x is coloured d_0, all the copies of v_i take colours from $\{d'_1, ..., d'_q\}$. Now, each $R[.]$ can be coloured with one of the q configurations $\{d_0, d_1\}$, $\{d_0, d_2\}$, ..., $\{d_0, d_q\}$ and (because we have "absorbed" d_0, which can be adjacent to everything on the other side of the bipartition) this acts like a *bi-q-col* colouring. We let Y_1 be the set of full-colourings and Y_0 be the set of non-full colourings.

Every *bi-q-col* colouring comes up $Z = (q^2)^k((q-1)^s\nu(t,2))^{n_l}$ times as a full colouring of G'; we know $|Y_1| \geq Z$ because *#bi-q-col*$(G) \geq 1$. The $(q^2)^k$ term emerges in Z because, if x is coloured with d_0, each length-2 path can be coloured in q^2 ways. Therefore, to obtain our approximation of *#bi-q-col*(G), we want to divide $\#H_{sp(q)}(G')$ by Z and round in the usual fashion. Hence, we have to show $|Y_0|/Z \leq 1/4$; to do this we partition Y_0 into Y_0^- (colourings where x is not coloured d_0) and Y_0^+ (colourings where x is coloured d_0 but at least one of the $R[.]$ is not coloured exactly with any from the list $\{d_0, d_1\}$, $\{d_0, d_2\}$,..., $\{d_0, d_q\}$.) We show $|Y_0^-|/Z \leq 1/8$ and $|Y_1^+|/Z \leq 1/8$. Proceeding, if x is not coloured d_0, we see that each length-2 path can be coloured in $q + (q-1)^2$ ways if x is coloured with some d'_i ($i \in [1, q]$) and $(q-1)q$ ways if x is coloured with some d_i ($i \neq 0$). For $q \geq 2$ both these values are less than q^2. Hence, a *very* crude upper bound on $|Y_0^-|$ is

$$|V(H_{sp(q)})|(q^2-1)^k|V(H_{sp(q)})|^{n_l(s+t)+n_r}$$

(Recall that $n_l = |V_L(G)|$ and $n_r = |V_R(G)|$, meaning $n = n_l + n_r$.) Given that q^{2k} is a lower bound on Z we just need to choose k such that

$$\frac{|V(H_{sp(q)})|(q^2-1)^k|V(H_{sp(q)})|^{n_l(s+t)+n_r}}{q^{2k}} \leq 1/8$$

Choosing $k = n^2(s+t)$ adequately satisfies this for n above some fixed constant threshold.

We now have to choose s and t such that each $R[.]$ is most likely to be coloured with d_0 plus one other colour. Now, consider how many colourings are possible of a $(L[.], R[.])$ pair if $R[.]$ contains d_0 plus h other colours: $(q-h)^s\nu(t, h+1)$. (A lower bound on this quantity is $(1/2)(q-h)^s(h+1)^t$.) We wish this quantity to be maximised (for $h \in \{0, 1, ..., q-1\}$) at $h = 1$; we claim this is possible if we choose $s = (q/2)t$. To prove this, let

$$W(h) = (1+h)(q-h)^{q/2}$$

Thus, $W(h)$ represents the index of the exponential (in t) for a particular choice of h; we want $W(1)$ to maximise this value. So, if we differentiate $W(h)$ w.r.t. h we get

$$W'(h) = (q-h)^{(q/2)-1}((q-h) - (q/2)(1+h))$$

Choosing h to set $W'(h) = 0$ yields maxima and minima points; it is easy to show that the left factor represents the minima whilst the right factor represents the maxima. Thus, setting the right factor to zero yields $h = (q/2)/((q/2)+1)$. This value tends to 1 from below, so to complete the proof it just has to be shown that $h = 1$ is closer to maximal than $h = 0$; this follows from the fact that

$$q^{(q/2)} < 2(q-1)^{(q/2)}$$

for $q \geq 3$. So, we let $W_1 = W(1)$, and let $W_2 < W_1$ be the second largest value of $W(h)$ for $h \in \{0, 1, ..., q-1\}$. It follows that $q^{2k}((1/2)W_1^t)^{n_l}$ is a lower bound on Z.

Next we have to develop a crude upper bound on $|Y_0|^+$. The following is such an upper bound:

$$q^{2k}(2^{|V(H_{sp(q)})|}2^{|V(H_{sp(q)})|})^{n_l}W_1^{t(n_l-1)}W_2^t|V(H_{sp(q)})|^{n_r}$$

This is derived by assuming all configurations on the n_l $(L[.], R[.])$ pairs are possible and come up as many times as possible. So, we can prove that $|Y_0^+|/Z \leq 1/8$ by showing

$$\frac{q^{2k}(4^{|V(H_{sp(q)})|n_l})W_1^{t(n_l-1)}W_2^t|V(H_{sp(q)})|^{n_r}}{q^{2k}(1/2)^{n_l}W_1^{tn_l}} \leq 1/8$$

Noting that $n_l, n_r \leq n$, it is therefore adequate to show:

$$4^{(|V(H_{sp(q)})|+1)n}\left(\frac{W_2}{W_1}\right)^t|V(H_{sp(q)})|^n \leq 1/8$$

Given that $W_2 < W_1$ we can satisfy this (for large enough n) by setting $t = n^2$. Note that because we actually have to divide through by Z, and $s = (q/2)t$, there is a danger that Z is irrational if t is not an even number, and this could introduce another source of inaccuracy. To prevent this situation happening we can simply take $t = 2n^2$ instead. $\square$

7.6.2 Bipartite 3-colouring, bipartite 4-colouring and C_4^*

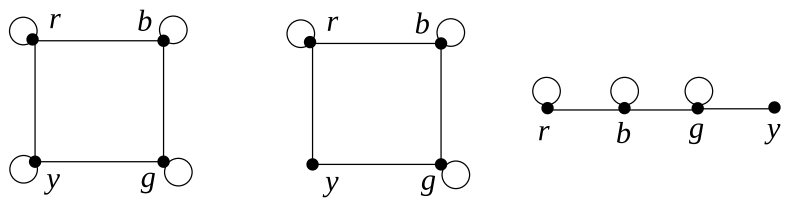


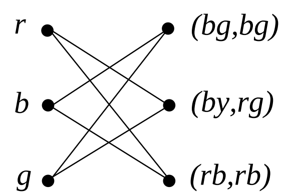
Figure 7.4: *Unclassified, connected 4-vertex H. From left to right: C_4^*, the crossbow, the long wrench*

C_4^*, the fully looped cycle on four vertices (see Figure 7.4) is presently unclassified. However, it so happens that *bi-4-col* is equivalent to $bi(C_4^*)$. (To see this, observe that every colour in C_4^* is adjacent to every colour apart from its diagonal opposite, and the degree of freedom permitted by bipartisation allows us to rearrange this so that, in the bipartite world, every colour is adjacent to every colour other than its counterpart on the other side of the bipartition.) Thus, we immediately have $\#$*bi-4-col*$\leq_{\mathsf{AP}}\#C_4^*$.

Furthermore, applying Lemma 7.5 therefore yields $\#$*bi-3-col*$\leq_{\mathsf{AP}}\#C_4^*$, thus establishing a link between C_6 and C_4^*.

7.6.3 Bipartite 3-colouring and the crossbow

It seems that $\#$*bi-3-col*$\leq_{\mathsf{AP}}\#crossbow$. (See Figure 7.4 for the definition of the *crossbow*.) Here is a sketch of a likely proof. Let $G = (V_L(G), V_R(G), E(G))$ be the input to $\#$*bi-3-col* - we build a bipartite graph G', the input to $\#crossbow$, as follows. For each $u \in V_L(G)$ we introduce a vertex $X[u]$. For each $v \in V_R(G)$ we introduce two large disjoint sets of vertices $L[v]$ and $R[v]$, both of size q, and connect every vertex in $L[v]$ to every vertex in $R[v]$. For each edge $\{u, v\} \in E(G)$ (where $u \in V_L(G)$ and $v \in V_R(G)$) we connect $X[u]$ to every vertex in $L[v]$. Having coded up the vertices and edges, we introduce a new vertex x and, for all $u \in V_L(G)$, connect x to $X[u]$. Then, for all $v \in V_R(G)$, we connect x to every vertex in $R[v]$. Finally, we attach k disjoint length-2 paths to x. (That is, the paths are mutually disjoint apart from their first vertex, which is x.) Now, the idea is that if k is large enough, x is exponentially likely to be coloured b. This is because when x is coloured b the remainder of each length-2 path can be coloured in 9 ways, but when x is coloured with any other colour the remainder of each path can be coloured in at most 8 ways. This makes all the $X[.]$ vertices exponentially likely to take colours from $\{r, b, g\}$. Now, consider one of the $(L[.], R[.])$ constructions. Since $L[.]$ and $R[.]$ are the same size it follows that an $(L[.], R[.])$ pair is exponentially likely to be coloured either $(rb, rb), (by, rg)$ or (bg, bg) - the configuration (rg, by) is not possible because each $R[.]$ is restricted by its connection to x to taking colours from $\{r, b, g\}$. Assuming $k >> q >> n$ it follows that *bi-3-col* colourings are exponentially likely to be induced, as shown:



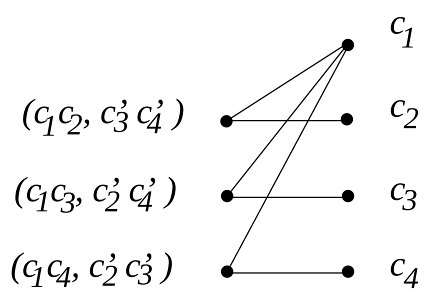
In fact, the above reduction actually demonstrates the slightly stronger result of

$$\#\textit{bi-3-col}\leq_{\mathsf{AP}}\#bi(crossbow)$$

because the graph G' is itself bipartite.

7.6.4 The junction and bipartite 4-colouring

It also appears likely that $\#junction \leq_{\mathsf{AP}} \#$*bi-4-col*$\leq_{\mathsf{AP}} \#C_4^*$. Here is a sketch of a possible proof. Suppose H is the graph representing the *junction*; let's say $V_L(H) = \{r', g', y'\}$ and $V_R(H) = \{b, r, g, y\}$ where edges in $E(H)$ are defined as in Figure 7.2. We show that $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#H_4$ which is sufficient. Let $G = (V_L(G), V_R(G), E(G))$ be the input to $\overleftarrow{\#H}$. We build an input G' (to $\#H_4$) as follows. For each vertex $u \in V_L(G)$ we introduce disjoint sets $L[u]$ and $R[u]$, both of size q, and connect every vertex in $L[u]$ to every vertex in $R[u]$. For every vertex $v \in V_R(G)$ we simply introduce a vertex $X[v]$. For every edge $\{u, v\} \in E(G)$ (where $u \in V_L(G)$ and $v \in V_R(G)$) we connect every vertex in $R[u]$ to $X[v]$. Finally, we introduce a new vertex x and, for all $u \in V_L(G)$, connect x to every vertex in $R[u]$. Now, we know that $V_L(H_4)$ contains colours $\{c_1, c_2, c_3, c_4\}$, $V_R(H_4)$ contains $\{c'_1, c'_2, c'_3, c'_4\}$ and $\{c_i, c'_i\}$ are its non-edges. If, say, x is coloured with c_1, then all the $R[.]$ sets are restricted to taking colours from $\{c'_2, c'_3, c'_4\}$. Now, given that the $L[.]$ sets are the same size as the $R[.]$ sets, it follows that the $(L[.], R[.])$ pairs are exponentially likely to be coloured either $(c_1c_4, c'_2c'_3)$, $(c_1c_3, c'_2c'_4)$ or $(c_1c_2, c'_3c'_4)$. (This is because $2 \times 2 > 1 \times 3$.) The $X[.]$ vertices are free to be coloured with $\{c_1, c_2, c_3, c_4\}$, so the following colourings are exponentially likely to be induced:



The analysis is similar irrespective of which colour is used to colour x. $\square$

Chapter 8

Future work and conclusions

8.1 Summary of main results from this thesis

In this thesis we have made an original and significant contribution to what is known about the relative complexity of approximately counting H-colourings. Our near-complete 4-vertex catalogue is useful in its own right, hinting that a large number of non-bipartite H are $\equiv_{\mathsf{AP}}\#SAT$[1], but perhaps more importantly it is accompanied by a large amount of gadgetry and reduction technology that should be of use in future research.

Perhaps the most significant contribution of the thesis has been in the area of hardness results. As we discuss in the following section we now know quite a lot about lower-bounds on complexity, and (indeed) most of our reduction techniques are geared towards hardness rather than easiness results. Theorem 4.1 proves that, until such time as we properly understand the absolute complexity of approximately counting (and sampling) independent sets in bipartite graphs, approximately sampling H-colourings can be (mostly) thought of as intractable in a complexity-theoretic sense. (The SP-reduction machinery used to prove this result, developed in conjunction with Goldberg and Paterson, should constitute a useful framework in which future research can compare the complexity of approximate sampling problems.) Additionally, our suite of $\equiv_{\mathsf{AP}}\#SAT$-hardness lemmas have helped to increase understanding about what structure a graph

[1]Recall that 47 of the 65 connected graphs on 4 or fewer vertices are known to be $\equiv_{\mathsf{AP}}\#SAT$

must have to be $\equiv_{\mathsf{AP}} \#SAT$.

On a more general level, we now have an enhanced intuition as to how H-colouring "behaves". For example, in [8] it was noted (with reference to the fluctuating complexity of the *wrench* family) that the complexity of H-colouring is non-monotonic. This is unsurprising in light of what we now know; as we have shown, the complexity of a graph H seems to be a deeply complicated non-monotonic function of the graph's structure, taking into account factors such as number of vertices, presence of universal loops, maximum clique size, sparseness, the extent to which different subgraphs compete with each other for dominance, and number of "poles". Additionally, we have noted that there is a tendency for H graphs (particularly those with loops and those that are bipartite) to prefer colouring distinct vertices of a graph G with the same colour rather than differently: in the language of statistical physics H-colouring seems to be naturally *ferromagnetic*. (Equally, our suite of $\equiv_{\mathsf{AP}} \#SAT$-hardness lemmas show that with appropriate gadgetry this behaviour can be countered.)

The other main contribution of the thesis is probably the extra information it has unearthed about the complexity hierarchy. We strongly suspect (citing the concrete sampling-based Theorem 4.1 as evidence) that a graph H with no non-trivial components is at least as hard as $\equiv_{\mathsf{AP}} \#BIS$. This, of course, does not preclude the possibility that some non-H-colouring problems exist in the continuum between being $FPRAS$able and $\equiv_{\mathsf{AP}} \#BIS$. However, basing an assertion solely on the evidence we have gleaned thus far from H-colouring, it does seem that $\equiv_{\mathsf{AP}} \#BIS$ is (at best) "not far" above the set of $FPRAS$able problems. Additionally, while we are short of formal results in this area, our experiences suggest that $\equiv_{\mathsf{AP}} \#BIS$ is a small class containing relatively few H graphs, with the complexity hierarchy exhibiting an apparent complexity gap between $\equiv_{\mathsf{AP}} \#BIS$ and $\equiv_{\mathsf{AP}} \#SAT$. The complexity hierarchy of approximate counting shows signs of being a fascinating, multi-tiered structure, and we are only just beginning to scratch the surface.

8.2 Ideas for future work

While we have significantly extended our understanding of approximately counting H-colourings, we still have a long way to go before we fully understand such problems, and in that regard this thesis asks more questions than it answers. One of the most pressing areas in which research needs to be undertaken is the AP-reduction itself. A flaw of both this thesis and [8] is that the AP-reduction has scarcely been stretched in terms of how much power it has at its disposal. Most of our AP-reductions are based on the "boosting" idea (presented by Sinclair in [27]) whereby we exponentially inflate the portion of the state space corresponding to the objects we wish to count, and then use techniques such as the "rounding" technique to extract an answer. Thus, for the most part we are using only single oracle calls with little or no randomization and hardly any post-processing of the value returned by the oracle call. It is difficult to say how much extra power we might gain by exploiting the AP-reduction more fully - it may be that we do not gain much at all - but it is an area that should be explored nonetheless.

We return to the power of reductions in a moment, but another area that it would be useful to learn more about is the complexity of approximately *sampling* H-colourings as compared to approximately *counting* them. As we have observed, it may be that approximately counting H-colourings is (in some cases) easier than approximately sampling them. It would aid our understanding of H-colourings significantly if this issue could be resolved once and for all. (Indeed, it would help us work out what connotations sampling results such as Theorem 4.1 - which proves that $BIS \leq_{\mathsf{SP}} H$ for all H without non-trivial components - have for the development of a counting-world analogue.) It would also be useful to explore in more detail the significance of working with the $FPAUS$ rather than the $PAUS$, especially with regard to sampling-preserving reductions.

One particularly important area that (at present) we barely understand at all is the apparent existence of a complexity gap. It certainly seems that H-colouring is fulfilling its function as a convenient suite of problems with which to tease out the nuances of the

AP-reducibility complexity hierarchy. Now we must take this inkling of a complexity gap forward and either strengthen it or disprove it. As such, there need to be more interreducibilities developed between graphs that may sit in this gap (bearing in mind that the complexity gap, if it exists, might itself be tiered.) There also needs to be more work done on how far our failures with $\equiv_{\mathsf{AP}}\#BIS$-easiness reductions and (to a lesser extent) $\equiv_{\mathsf{AP}}\#SAT$-hardness reductions are a reflection of complexity limitations and how far they are due to our own shortcomings. Indeed, this ties in with what must be the most glaring hole in our knowledge of H-colouring. While being quite accomplished at hardness results, we know very little whatsoever about H-colouring easiness results. To date this has been masked by the fact that a lot of graphs seem to be $\equiv_{\mathsf{AP}}\#SAT$, and hence don't require easiness results to be fully classified. However, as the graphs we study become more complex and an increasing number evade categorisation in either $\equiv_{\mathsf{AP}}\#BIS$ or $\equiv_{\mathsf{AP}}\#SAT$, coupled with the possible partitioning of the complexity hierarchy into more complexity classes in the near future, our dearth of easiness results will probably hamstring future research if it is not addressed soon. Such a focus will inevitably bring together the discussion on whether we are using AP-reductions properly with the search for new reduction techniques. It would be useful to develop more interreducibilities between H-colouring and non-H-colouring problems; recall, for example, that most of our $\equiv_{\mathsf{AP}}\#BIS$-easiness reductions have come through reduction to the remarkably flexible $\#DownSets$ problem. It would be good to increase our supply of similarly flexible problems.

There are myriad other avenues that future research could travel down. The conjecture that bipartite H are never $\equiv_{\mathsf{AP}}\#SAT$ "feels" correct but lacks firm evidence; research in this area could be undertaken as part of a wider investigation exploring the inherent structural characteristics of $\equiv_{\mathsf{AP}}\#SAT$ H and also to what extent bipartite H and non-bipartite H are different from a general point of view. The search for defining $\equiv_{\mathsf{AP}}\#SAT$-hardness characteristics could do well to investigate further the power of gadgets that step outside the rather tired "introduce two sets of vertices U and V and connect every vertex in U to every vertex in V" format. The gadget used to encode vertices in the *1-wrench* proof (graph 9) would seem to be a good starting point for new directions in gadget design. As a way of really testing how good our gadgets are it could be interesting to try them out on H with 5 or more vertices (possibly using semi-automation in the case of our $\equiv_{\mathsf{AP}} \#SAT$-hardness reductions), but this might not be the most appropriate use of research time. Another area that ties into many of the above topics is the barely-touched question of disconnected H; we really understand very little about this topic.

One final area for research that could be happily examined away from the AP-reduction framework is the *absolute* complexity of $\#BIS$. As discussed in Chapter 7 we have *some* evidence that $\#BIS$ might be intractable. However, as has also been mentioned, we would conversely not be stunned if $\#BIS$ turned out to be $FPRAS$able (although such a result would be somewhat significant.) Either way it would be useful to increase our understanding of just how difficult this seminal problem actually is.

And that concludes this thesis, save for the appendices that follow. The relative complexity of approximately counting H-colourings looks like being a challenging yet rewarding field for future researchers to study.

Steven Kelk

July 2003

Appendix A

Appendices

A.1 Informal summary of Chapter 1

(All of this section is based on work by other authors.) The background to the thesis is the class $\#P$, which essentially captures all natural counting problems. (Informally, a counting problem is in $\#P$ if potential solutions can be efficiently verified.) We know that $\#P$-complete problems are, in both a complexity and practical sense, very hard. Indeed, $\#P$-complete problems sit above the *entire* polynomial hierarchy. To put this in context, JVV showed (for example) that exact counting is probably strictly harder than exact uniform sampling, because a Probabilistic Turing Machine (PTM) equipped with an oracle from relatively low-down[1] in the polynomial hierarchy can perform the hardest[2] exact sampling problems. (In contrast, a PTM equipped with an oracle from arbitrarily high up the polynomial hierarchy can still not, in all likelihood, solve a $\#P$-complete problem.) JVV also showed that, in a general sense, approximate counting[3] and approximately uniform sampling are of the same complexity. They noted, actually, that exact uniform sampling seems to sit slightly above approximately uniform sampling (and approximate counting) in the polynomial hierarchy.

[1] Just above NP, in fact

[2] Naturally, we do not mean arbitrarily hard, we mean within the context of problems that can be expressed within $\#P$.

[3] In this thesis all references to approximate counting should technically read *randomized* approximate counting, because, as evident in JVV, the differences with *deterministic* approximate counting can be significant. However, for brevity we drop the term randomized.

However, this thesis is not really concerned with the complexity of approximate counting relative to (for example) exactly uniform sampling or exact counting. That's because, in the context of the AP-reduction introduced by [8], we already know quite a lot about where approximate counting problems sit in the wider complexity context. Indeed, we know that even the hardest approximate counting problems (those interreducible with $\equiv_{\mathsf{AP}}\#SAT$) can be solved using a PTM equipped with an oracle for an NP-complete problem. As such, these $\equiv_{\mathsf{AP}}\#SAT$ problems are - in a complexity sense - quite low-down in the polynomial hierarchy:- below exact uniform sampling and a long way below exact counting. This, of course, does not mean they are tractable; finding an $FPRAS$ (i.e. an efficient randomized approximation algorithm) for a $\equiv_{\mathsf{AP}}\#SAT$ problem is highly unlikely under standard complexity assumptions. At the other end of the complexity spectrum, many approximate counting problems do of course have an $FPRAS$, and as a consequence can be thought of as easy in an absolute sense.

So, if this thesis is not concerned with the complexity of approximate counting relative to other complexity measures, what is it concerned about? Primarily, its main preoccupation is searching for tiers of complexity *within* the AP-reducibility hierarchy. In other words, given a $\#P$-complete exact counting problem, how difficult is it to approximately solve the problem, *relative to* the difficulty of approximately solving other $\#P$-complete problems? As discussed in the introduction, previous authors have shown that some $\#P$-complete problems are just as intransigent in the approximation world (i.e. $\equiv_{\mathsf{AP}}\#SAT$) while - intriguingly - some $\#P$-complete problems actually have an $FPRAS$. Furthermore, as [8] demonstrated, some $\#P$-complete problems seem to be of "intermediate" complexity ($\equiv_{\mathsf{AP}}\#BIS$) in the approximation world. The challenge, therefore, is to learn more about what happens to $\#P$-complete problems when they are switched to the approximation domain.

Two immediate questions arise:- (1) Are these three tiers (i.e. $FPRAS$able, $\equiv_{\mathsf{AP}}\#BIS$, $\equiv_{\mathsf{AP}}\#SAT$) distinct? And, (2) Are there any more distinct tiers?

This is where H-colouring comes in. As shown in the introduction, some H-colouring problems have been shown to admit an $FPRAS$, some are $\equiv_{\mathsf{AP}} \#SAT$ and some are $\equiv_{\mathsf{AP}} \#BIS$. Many more have yet to be classified. Thus, evidence to help answer both questions (1) and (2) can be accumulated by attempting to determine tiers of complexity (with respect to AP-reducibility) within the infinite suite of H-colouring problems. That is, by identifying groups of H-colouring problems that are mutually interreducible within themselves but seemingly not interreducible with other groups.

The fundamental point is that any tiers of complexity we discover solely within the domain of H-colouring problems of course also exist when considering the wider domain of approximate counting problems. So, while it is interesting to learn more about where H-colouring problems sit in the approximate counting complexity hierarchy, the main reason for our preoccupation with them is that they are an excellent tool for learning more about the approximate counting complexity hierarchy in general. For example, if a growing cluster of H-colouring problems are found that don't seem to be AP-reducible to $\#BIS$, yet at the same time don't seem to be $\equiv_{\mathsf{AP}} \#SAT$, this could be interpreted as growing evidence that there is at least one further tier of complexity (i.e. a complexity gap) between $\equiv_{\mathsf{AP}} \#BIS$ and $\equiv_{\mathsf{AP}} \#SAT$. (As it happens, this is one of the findings of this thesis, discussed extensively in Chapter 7.)

A.2 Comment on uniqueness of compact form for a graph H in expanded form

To produce the compact form of H we first partition H into its vertex equivalence classes (i.e. two vertices of H are in the same equivalence class if they have the same adjacency set). Note that this partition is uniquely defined for a given H. Now, we will construct H', the compact form of H. First, let there be l equivalence classes $V_1 ... V_l$, which partition $V(H)$. We set $V(H') = \{v_1, ..., v_l\}$ and define $w(v_i) = |V_i|$. Finally, we say $\{v_i, v_j\} \in E(H')$ iff there exists a vertex in V_i and a vertex in V_j which are adjacent in H. (In actual fact, if there exists a vertex in V_i and a vertex in V_j

which are adjacent, every vertex in V_i is adjacent to every vertex in V_j.) Now, because the partition of H is uniquely defined for H, the graph H' is also uniquely defined for H.

Thus, the compact form is unique. Further, to see that $adj(v_i) \neq adj(v_j)$ for $i \neq j$, suppose (by way of contradiction) there do exist v_i, v_j such that $adj(v_i) = adj(v_j)$. Now, consider some vertex $x \in V_i$; the adjacency set of x (in H) is equal to the union of V_k for all k such that $v_k \in adj(v_i)$. Similarly, consider some vertex $y \in V_j$; the adjacency set of y in H is equal to the union of V_k for all k such that $v_k \in adj(v_j)$. Since $adj(v_i) = adj(v_j)$ it follows that, in H, x and y have the same adjacency set, but are not in the same equivalence class - contradiction!

A.3 Graph multiplication

On page 70 we discuss the existence of a graph "multiplication" operator. Here we formalise this notion. The explicit relevance of this operator to counting H-colourings may have been originally discovered by an earlier author; at the time of writing this has not yet been clarified. It is known, however, that the operator has been used (at least implicitly) on earlier occasions by Dyer and Greenhill (in [10]) and in joint publications by Graham Brightwell and Peter Winkler.

The "product" of two graphs $H_1 = (V(H_1), E(H_1))$ and $H_2 = (V(H_2), E(H_2))$ is defined as follows:-

$$H_1 \boxtimes H_2 = (V(H_1 \boxtimes H_2), E(H_1 \boxtimes H_2))$$

where $V(H_1 \boxtimes H_2) = V(H_1) \times V(H_2)$ (i.e. the ordered Cartesian product of $V(H_1)$ and $V(H_2)$) and

$$E(H_1 \boxtimes H_2) = \left\{ \{(c_1, d_1), (c_2, d_2)\} \middle| \{c_1, c_2\} \in E(H_1) \wedge \{d_1, d_2\} \in E(H_2) \right\}$$

Observation A.1 *Let H_1 and H_2 be any two graphs and let $H' = H_1 \boxtimes H_2$. Then $\#H'(G) = \#H_1(G)\#H_2(G)$.*

Proof. This can be proven by demonstrating an appropriate bijection between the set $H'(G)$ and the set $H_1(G) \times H_2(G)$. The bijection is natural and is described as follows.

Let C be a colouring in $H'(G)$. Assuming we have left the colours of H' in the form where they are elements from $V(H_1) \times V(H_2)$, we define the colourings $C_1 \in H_1(G)$ and $C_2 \in H_2(G)$ to be those which satisfy $(C_1(u), C_2(u)) = C(u)$ for all $u \in V(G)$. Our bijection, therefore, is to map $C \in H'(G)$ to $(C_1, C_2) \in H_1(G) \times H_2(G)$. $\square$

From a reduction point of view, this can be exploited in a number of ways. One example is given on page 70. Another is the observation that if $H = H_1 \boxtimes H_2 \boxtimes ... \boxtimes H_k$ and all the H_i are $\equiv_{\mathsf{AP}}$ $\#X$-easy (for some counting problem $\#X$), then $\#H \leq_{\mathsf{AP}} \#X$. (To see why this is, note that if we wish to approximate $\#H(G)$ to accuracy ϵ we can do this by using the $\#X$-oracle to estimate each $\#H_i(G)$ to accuracy ϵ/k, and then multiplying all the estimates together.) Similarly, if

$$H \boxtimes H_1' \boxtimes H_2' \boxtimes ... \boxtimes H_j' = H_1 \boxtimes H_2 \boxtimes ... \boxtimes H_k$$

and all the graphs other than H are $\equiv_{\mathsf{AP}}$ $\#X$-easy, then $\#H \leq_{\mathsf{AP}} \#X$, but *only* for those graphs G for which the term

$$\frac{\prod_{i=1}^{k} \#H_i(G)}{\prod_{i=1}^{j} \#H_i'(G)} \tag{A.1}$$

is defined. (To approximate $\#H(G)$ to accuracy ϵ we use accuracy $\epsilon/2k$ to estimate each $\#H_i(G)$ value, accuracy $\epsilon/2j$ to estimate each $\#H_i'(G)$ value, take the product of the $\#H_i(G)$ estimates and divide by the product of the $\#H_i'(G)$ estimates.) To see why the caveat about (A.1) emerges consider the following example. Let H be the graph representing the $\#IS$ problem and let H_1 be the graph representing the $\#BIS$ problem i.e. P_4. Now, since $H_1 = H \boxtimes K_{1,1}$ it might seem that we can reduce $\#IS$ to $\#BIS$ for all G by exploiting the relationship $\#H(G) = \#H_1(G)/\#K_{1,1}(G)$. However, because $\#K_{1,1}(G) = 0$ for non-bipartite G the expression is only defined for bipartite G, which doesn't tell us anything new. $\square$

Finally, it is worth pointing out that the natural, combinatorial bijection between $H'(G)$ (where H' is $H_1 \boxtimes H_2$) and $H_1(G) \times H_2(G)$ makes it easy to convert a uniform sampler for $H'(G)$ into a uniform sampler for (wlog) $H_1(G)$. To see this, note that, if given a sample (i.e. a colouring) $C \in H'(G)$ we can easily construct the relevant pair of colourings (C_1, C_2) in polynomial time, and just return C_1 as our sample from $H_1(G)$. Our $H_1(G)$ sampler is uniform because each colouring from $H_1(G)$ comes up $\#H_2(G)$ times in this way as a colouring of $\#H'(G)$.

A.4 Proof of Observation 3.2 from Section 3.8.1

Observation 3.2 *Let P be a set of elements, partitioned into l disjoint sets, P_1,...P_l. Let Q be a set of l elements ($q_1, ..., q_l$) and let π be a distribution on Q defined by $\pi(q_i) = |P_i|/|P|$. Now, suppose we have an approximate sampler on π, which produces a distribution π' on Q which deviates no more than ϵ' from π, and runs in time at most poly($|Q|, 1/\epsilon'$). If for all $q_i \in Q$ we can efficiently choose u.a.r. a sample from P_i when presented with q_i, we can build a $PAUS$ for P.*

Proof. Suppose we wish to build a $PAUS$ on P. So let ϵ be the desired accuracy of our $PAUS$ for P. Let ρ' be the distribution on P generated by the following experiment: we choose q_i from Q using our distribution π' with accuracy $\epsilon' = \epsilon$, choose p u.a.r. from P_i and finally output p. Now, if we let ρ be the uniform distribution on P, we need to show that the variation distance of ρ' is no more than ϵ from ρ. By the definition of variation distance on page 109, we therefore need:

$$(1/2) \sum_{p \in P} |\rho'(p) - 1/|P|| \leq \epsilon$$

The LHS of the above inequality is equivalent to

$$(1/2) \sum_{i=1}^{l} \sum_{p \in P_i} \left| \rho'(p) - 1/|P| \right| = (1/2) \sum_{i=1}^{l} \sum_{p \in P_i} \left| \pi'(q_i)/|P_i| - 1/|P| \right|$$

The value of the term inside the absolute operator is the same for all elements in the same P_i, so we can drop the inner summation operator and simply multiply through by

$|P_i|$ which gives

$$(1/2)\sum_{i=1}^{l}\left|\pi'(q_i) - |P_i|/|P|\right|$$

This value equals ϵ because π' is by definition no further than ϵ from π, and as a result we have proven this is a $PAUS$ for P. $\square$

A.5 Technical comment for sketch proof in Section 7.3.1 (page 241)

This is not intended to be a full proof. Rather, we use this section to show that the most complicated part of the sketch proof (i.e. choosing s, t such that $2^s 2^t$ is sufficiently close to $1^s 3^t$) is technically justifiable. That said, the only part we have left out is showing the inferiority of non-full colourings, where full-colourings are defined to be those where each $(L[.], R[.])$ is coloured with one of the listed configurations. This is easy in this instance because, owing to the fact that each $L[.]$ is a complete graph, the only configurations possible on $(L[.], R[.])$ other than those listed are *subsets* of those listed - e.g. (b, rb) is a subset of (rb, rb) - and thus are inherently exponentially inferior. (Let x represent the smallest that both s and t must be to ensure these inferior configurations are dwarfed; we use this value later.)

Now, note that (b, rbg) and (g, bgy) both come up 3^t times:- we can get away without using ν notation here since it is the content of the $L[.]$ sets, not the $R[.]$ sets, that define the behaviour of the configuration. The configurations (rb, rb), (bg, bg) and (gy, gy) each come up $\nu(s, 2)2^t$ times. We plan to use the normal rounding technique, taking $Z = 3^{tn}$ as the divisor. Proceeding, let:-

$$l = \frac{\nu(s, 2)2^t}{3^t}$$

and let $f = \max(l, l^{-1})$. It follows that

$$Nf^{-n} \leq \frac{\#P_4^*(G')}{Z} \leq Nf^n + \frac{|Y_0|}{Z} \tag{A.2}$$

where Y_0 is the set of non-full colourings and (as usual) G' is the graph we have constructed. Now, if $\#\hat{P_4^*}(G')$ is the value returned by our approximation oracle, we need (as argued in the proof of Lemma 2.8 on page 77) the following to hold:

$$e^{-\epsilon/21}(N-\frac{1}{4}) \le \frac{\#P_4^{\hat{*}}(G')}{Z} \le e^{\epsilon/21}(N+\frac{1}{4})$$

If we use δ (to be determined) as the accuracy to our oracle it follows from (A.2) that

$$e^{-\delta}Nf^{-n} \le \frac{\#P_4^{\hat{*}}(G')}{Z} \le e^{\delta}\left(Nf^n + \frac{|Y_0|}{Z}\right) \tag{A.3}$$

So, if we (a) take $\delta = \epsilon/42$, (b) choose s, t such that $|Y_0|/Z \le 1/4$ (which we only mention in passing here) and (c) ensure that

$$e^{-\epsilon/42n} \le \frac{3^t}{\nu(s,2)2^t} \le e^{\epsilon/42n} \tag{A.4}$$

it follows from (A.2) and (A.3) that we are done. Ideally we would choose $s, t \in \mathbb{N}^+$ such that $3^t = \nu(s,2)2^t$ but this is not possible. However, we know that $2^s(1-\exp(-s/4)) \le \nu(s,2) \le 2^s$ so if we can show $e^{-\epsilon/84n} \le (1-\exp(-s/4))$ and also show that

$$e^{-\epsilon/84n} \le (1/2)^s(3/2)^t \le e^{\epsilon/84} \tag{A.5}$$

then this satisfies (A.4). First, we deal with showing $e^{-\epsilon/84n} \le (1-\exp(-s/4))$. Note that $e^{-\epsilon/84n} \le 1-\epsilon/168n$. Hence we are satisfied as long as $s \ge s_0$ where $s_0 = \lceil 4\ln(168n/\epsilon)\rceil$. We use this information to guide our choice of a_0, below. Now, we need to prove (A.5). We do this by setting s, t to the values a, b (respectively) returned by technical Lemma 4.2 (on page 139.) The parameters we pass to this lemma are as follows: $c_1 = 3/2$, $c_2 = 2$, $a_0 = x + s_0$ and $q = 0$. The lemma guarantees that the accuracy bound is met, that our chosen values of s, t are not too big, and that both are at least as big as $x + s_0$, which is what we require.

A.6 Proof of Lemma 7.1 and Corollary 7.2

Lemma 7.1 [DGGJ informal] Let $H = (V_L(H), V_R(H), E(H))$ be a bipartite graph. If the vertices of $V_L(H)$ and $V_R(H)$ can be ordered $c_0, ..., c_{|V_L(H)|-1}$ and $c'_0, ..., c'_{|V_R(H)|-1}$ such that for any pair of edges $\{c_i, c'_j\}$ and $\{c_{i+a}, c'_{j-b}\}$ which cross each other (in the sense that a and b are either both positive or negative) H contains the complete bipartite graph on $\{c_i, ..., c_{i+a}\}$ and $\{c'_{j-b}, c'_j\}$, then $\#H \le_{\mathsf{AP}} \#BIS$.

Proof. (This proof is the work of the thesis author, but is modelled on the original sketch proof by DGGJ.) We actually demonstrate that $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#BIS$, which (by Lemma 2.15) is sufficient. So, we proceed by demonstrating how the problem $\overleftarrow{\#H}$ can be coded up as a $\#DownSets$ problem. Let $G = (V_L(G), V_R(G), E(G))$ be the input to $\overleftarrow{\#H}$, we code up our input to $\#DownSets$, P, as follows. For each vertex $u \in V_L(G)$, introduce $|V_L(H)|$ elements $u_0, ..., u_{|V_L(H)|-1}$ and relational constraints $u_i \prec u_{i+1}$ for $0 \leq i < |V_L(H)| - 1$. For each vertex $v \in V_R(G)$, introduce $|V_R(H)|$ elements $v_0, ..., v_{|V_R(H)|-1}$ and introduce constraints $v_i \prec v_{i+1}$. That completes the encoding of the vertices of G.

For each edge $\{u, v\} \in E(G)$ (where $u \in V_L(G)$ and $v \in V_R(G)$) we add constraints as follows:

For each element u_i, we add the constraint $v_j \prec u_i$ where j is the smallest index of a vertex appearing in the following set:

$$\left\{ c'_s \in V_R(H) \middle| \text{ there exists } t \text{ such that } i \leq t < |V_L(H)| \text{ and } \{c_t, c'_s\} \in E(H) \right\}$$

For each element v_i, we add constraint $u_j \prec v_i$, where j is the smallest index of vertices in the following set

$$\left\{ c_s \in V_L(H) \middle| \text{ there exists } t \text{ such that } i \leq t < |V_R(H)| \text{ and } \{c_s, c'_t\} \in E(H) \right\}$$

This completes the encoding of G as a partial order. For connected G, we claim that $\#DownSets(P) = \overleftarrow{\#H}(G)+1$, which (by Lemma 6.6) proves that $\overleftarrow{\#H} \leq_{\mathsf{AP}} \#DownSets$. (We note in Section 6.3 that Lemma 6.6 need not solely be restricted to H-colouring problems.)

To see this, consider a downset in the cell representing the vertex $u \in V_L(G)$. Ignoring the possibility that the downset is empty for a moment, assume i is the largest index of any element in the downset. We claim that this downset behaves exactly like the colour c_i. Similarly, in a downset in a right-hand cell, if i is the largest index of an element in the downset, we claim that downset behaves exactly like the colour c'_i. To prove this, we show that for all edges $\{c_i, c'_j\} \in E(H)$, the downsets corresponding to c_i and c'_j can be adjacent to one another, and that for all non-edges in $E(H)$, the two corresponding downsets *cannot* be adjacent to one another.

Consider an edge $\{c_i, c'_j\} \in E(H)$. We show that the two downsets $U = \{u_0, ..., u_i\}$ and $V = \{v_0, ..., v_j\}$ can be adjacent. Now, we know that the lowest index of a vertex adjacent to some vertex in $\{c_i, ..., c_{|V_L(H)|-1}\}$ must be less than or equal to j, because $\{c_i, c'_j\} \in E(H)$. It follows that for all vertices c_k $(k < i)$, the lowest index of a vertex adjacent to some vertex in $\{c_k, ..., c_{|V_L(H)|-1}\}$ is also bound above by j, because c'_j is adjacent to c_i and $c_i \in \{c_k, ..., c_{|V_L(H)|-1}\}$. Hence, we know that U is happy next to V, but we need to confirm that V is happy next to U. (To clarify, to say U is "happy" next to V is a necessary but not sufficient condition for the adjacency of U and V to be permissible:- adjacency is only permissible if both downsets are mutually happy next to one another. We say U is "happy" next to V if $(U \cup V)$ would be a valid downset were all constraints of the form $u \prec v$ - for $u \in U$ and $v \in V$ - ignored.) The argument is the symmetry of the above: we know that the lowest index amongst vertices adjacent to some vertex in $\{c'_j, ..., c'_{|V_R(H)|-1}\}$ is bound above by i, so V is happy to be next to U. Hence an edge in $E(H)$ is an "edge" in the $DownSets$ structure.

Finally, consider a non-edge $\{c_i, c'_j\}$. We show that the downsets $U = \{u_0, ..., u_i\}$ and $V = \{v_0, ..., v_j\}$ cannot be adjacent. Suppose by way of contradiction that they can. It follows that $v_k \prec u_i$ for some $k \leq j$, and $u_m \prec v_j$ for some $m \leq i$. Now, the fact that $v_k \prec u_i$ means there exists some edge $\{c_p, c'_k\}$ where $i \leq p \leq |V_L(H)| - 1$. Similarly, the fact that $u_m \prec v_j$ means there exists some edge $\{c_m, c'_q\}$ where $j \leq q \leq |V_R(H)| - 1$. So we know $k \leq j, m \leq i, i \leq p, j \leq q$. Now, there are several cases to consider. If $p - m > 0$ and $q - k > 0$ then these two edges cross, which gives us a contradiction, because the non-edge between c_i and c'_j prevents there being a complete bipartite graph between $\{c_m, ..., c_p\}$ and $\{c_k, ..., c_q\}$. Conversely, suppose (wlog) $q - k = 0$. This can only happen if $q = j = k$. This would mean that $p - m > 0$, because $p - m = 0$ would mean $p = i = m$ and this in turn would mean that there was an edge between

c_i and c'_j, which is not possible. In fact, the lack of edge between c_i and c'_j means that $p > i$ and $m < i$. Now, given that H is connected, *some* edge must leave c_i. The fact that $p > i$, $i > m$ and $q = j = k$ means that this edge must cross either $\{c_m, c'_q\}$ or $\{c_p, c'_k\}$; either way there is not a complete bipartite graph induced on the relevant vertices, because there is no edge between c_i and c'_j, which again gives us a contradiction. □

Note that we have ignored the possibility U and V being empty downsets. This is because an empty downset $U = \emptyset$ can be adjacent to a downset V iff $V = \emptyset$. To see this, consider an edge $\{u, v\} \in E(G)$ (where $u \in V_L(G)$ and $v \in V_R(G)$.) If the downset in the cell representing u is non-empty then it must contain at least one element - let's call it u_i - and by construction there will be some v_j in the cell representing v such that $v_j \prec u_i$. Hence it is not possible for an empty downset to be adjacent to a non-empty downset. The effect of this, in connected G, is to introduce a disconnected K_2 component, which explains where the "minus one" part comes from in the expression $\overleftarrow{\#H}(G) = \#DownSets(P) - 1$. □

Corollary 7.2 If $\#H$ is shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy by Lemma 7.1, it follows that any weighted variant of H can also be shown to be $\equiv_{\mathsf{AP}}\#BIS$-easy by Lemma 7.1.

Proof. We need to prove that both *adding* equivalent vertices to H and (where this is possible) *removing* equivalent vertices from H leaves the resulting graph H' in the domain of Lemma 7.1. The "removing vertices" direction is immediate, because removing a vertex entails also removing all its incident edges. (More specifically, if you remove a vertex and its edges from a complete bipartite subgraph of H, this does not prevent the remaining vertices in the subgraph from continuing to induce a complete bipartite graph.[4]) The more difficult direction, then, is to prove that *adding* equivalent vertices to H does not violate the crossing property. Without loss of generality we can assume just a single vertex is added to H to make H', because if this holds it follows we can

[4]In fact, this proves the slightly stronger result that, if H has the crossing property, so to does any connected graph obtained by removing vertices from H.

then prove the corollary for any (positively) weighted variant of H simply by repeatedly adding individual vertices as required.

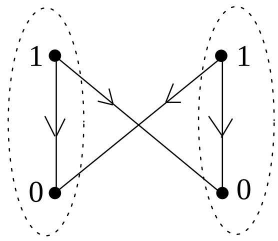
So let's assume that H has the crossing property, and the vertices of H are ordered and labelled as in the text of Lemma 7.1. Now, wlog we pick a vertex $c_i \in V_L(H)$[5] and add in a new vertex d which is equivalent to c_i (i.e. has the same adjacency set as c_i.) Note that wlog we assume that d is inserted "below" c_i so, if $i \geq 1$, d sits between c_i and c_{i-1}. (If $i = 0$ then d is "below" c_0 but above nothing else.) We can assume the new vertex is inserted "below" because, when introducing an equivalent vertex, the resulting graph is the same irrespective of where the vertex is inserted. Now, let $A(j)$ denote the index of the jth vertex that c_i is adjacent to, counting upwards i.e. $A(j) < A(j+1)$ for $j < |adj(c_i)|$. Hence, the adjacency set of both c_i and d is $\{c'_{A(1)}, c'_{A(2)}, ..., c'_{A(k)}\}$ where $k = |adj(c_i)|$. Proceeding, note that the new edge $\{d, c'_{A(j)}\}$ crosses edges $\{c_i, c'_{A(l)}\}$ for $l < j$. To see that these crossings do not violate the crossing property, note that the subgraph induced by bipartition $\{c_i, d\}$ and $adj(c_i)$ is a complete bipartite graph. Now, it remains to show that the crossing property is not violated if one of the new edges $\{d, c'_{A(j)}\}$ crosses with an edge which does not have c_i as an endpoint. Now, suppose there is an edge $\{c_p, c'_q\}$ $(p > i)$ that crosses an edge $\{d, c'_{A(j)}\}$; we know immediately that the edge also crosses $\{c_i, c'_{A(j)}\}$. We also know that $c'_q \in adj(c_i)$ because if it wasn't there the absence of edge $\{c_i, c'_q\}$ would prevent there being a complete bipartite graph between $\{c_i, ..., c_p\}$ and $\{c'_q, ..., c'_{A(j)}\}$, and this is not possible. Furthermore, since $adj(d) = adj(c_i)$ we know that $\{d, c_i, ..., c_p\}$ and $\{c'_q, ..., c'_{A(j)}\}$ also induce a complete bipartite graph. Hence the crossing of the edge $\{c_p, c'_q\}$ and $\{d, c'_{A(j)}\}$ does not violate the crossing property. (A similar argument can be deployed if $p < i$.) This completes the proof that adding equivalent vertices to H does not violate the crossing property, and hence completes the proof of Corollary 7.2 $\square$

[5]We can specify that the vertex is in $V_L(H)$ without loss of generality

A.7 $\#DownSets$ proofs (1)

Here we detail the $\#DownSets$ structures which give rise to the graphs on page 253.[6] These have all been (painstakingly!) checked by hand for correctness.

In each case, we give the two partial orders used to generate the graph in question. The first partial order is the "cell" order, which is the partial order that each vertex from G is coded up as. The second partial order is the "adjacency" order, which is used to code up edges from G. We clarify this with an example. Recall the graph *2-WR*. This can be coded up as a $\#DownSets$ problem where each cell contains two elements, 0 and 1. The cell order is $0 \prec 1$ and the adjacency order is also $0 \prec 1$. Hence, two adjacent vertices would be coded up as follows:



The downsets possible in each cell are therefore $\emptyset$, $\{0\}$ and $\{0,1\}$. Each of these three downsets can be "adjacent" to itself - i.e. two adjacent cells can both contain the same downset simultaneously - meaning the colour each downset represents is looped. Note that $\{0\}$ can be adjacent (in addition to itself) to both $\emptyset$ and $\{0,1\}$, so acts like the centre colour in *2-WR*, but the two other downsets are not mutually adjacent.

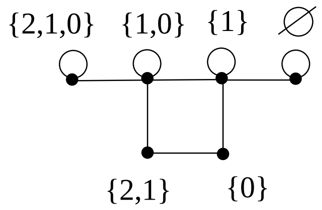
The first three graphs in our selection are coded up with cells containing three elements, and the remainder using cells with four elements. (Note that, in many cases, the graph induced by the downsets has a number of disconnected vertices. However, we know from Lemma 6.6 that these are easily discounted.)

Cell: no partial order on the three elements $\{0,1,2\}$

Adjacency: $0 \prec 2$, $1 \prec 0$

[6] We have neglected to cover the first three graphs in that section because they have already been discussed in [8].

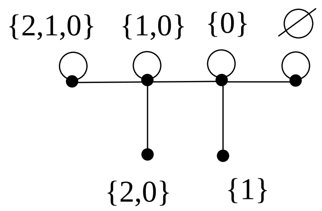
8 possible downsets in each cell, 2 corresponding to disconnected vertices: $\{2,0\},\{2\}$



Cell: no partial order on the three elements

Adjacency: $0 \prec 1$, $0 \prec 2$, $1 \prec 2$

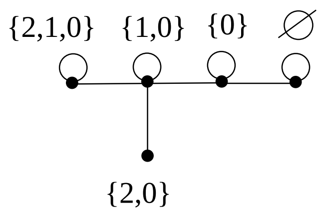
8 possible downsets in each cell, 2 corresponding to disconnected vertices: $\{2\},\{2,1\}$



Cell: $0 \prec 1$

Adjacency: $0 \prec 1$, $1 \prec 2$

6 possible downsets in each cell, 1 corresponds to a disconnected vertex: $\{2\}$

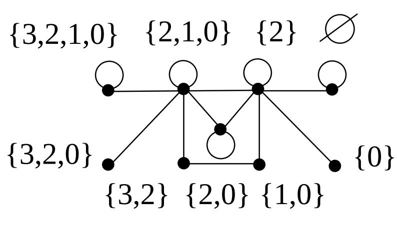


(From now on we assume four elements per cell: 0,1,2,3)

Cell: $0 \prec 1$

Adjacency: $1 \prec 3$, $2 \prec 0$

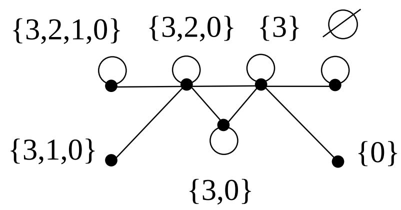
12 possible downsets in each cell, 3 correspond to disconnected vertices: $\{3\},\{3,0\},\{3,1,0\}$



Cell: $0 \prec 1$, $0 \prec 2$, $3 \prec 2$

Adjacency: $2 \prec 1$, $3 \prec 0$

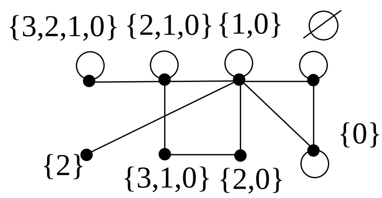
8 possible downsets in each cell, 1 corresponds to a disconnected vertex: $\{0, 1\}$



Cell: $0 \prec 1$

Adjacency: $0 \prec 3$, $1 \prec 2$, $2 \prec 3$

12 possible downsets in each cell, 4 correspond to disconnected vertices: $\{3, 2, 0\}, \{3, 2\}, \{3\}, \{3, 0\}$

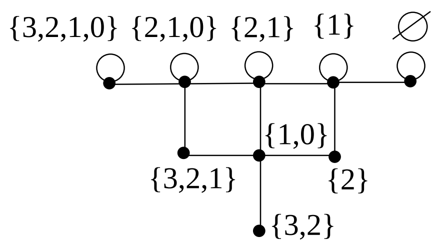


Cell: no partial order on the four elements

Adjacency: $0 \prec 3$, $1 \prec 2$, $2 \prec 0$

16 possible downsets in each cell, 7 correspond to disconnected vertices:

$\{0\}, \{2, 0\}, \{3\}, \{3, 0\}, \{3, 1\}, \{3, 0, 1\}, \{3, 2, 0\}$

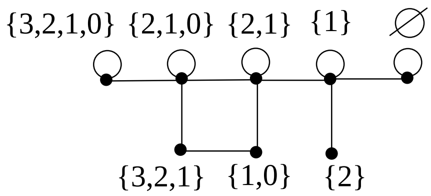


Cell: no partial order on the four elements

Adjacency: $0 \prec 3$, $1 \prec 0$, $1 \prec 2$, $2 \prec 0$

16 possible downsets in each cell, 8 correspond to disconnected vertices:

$\{0\}, \{2, 0\}, \{3\}, \{3, 0\}, \{3, 1\}, \{3, 1, 0\}, \{3, 2\}, \{3, 2, 0\}$

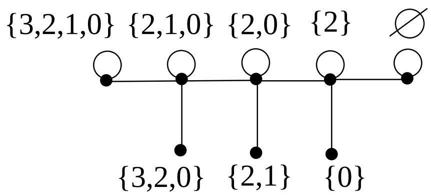


Cell: no partial order on the four elements

Adjacency: 0 ≺ 1, 0 ≺ 3, 1 ≺ 3, 2 ≺ 0, 2 ≺ 1

16 possible downsets in each cell, 8 correspond to disconnected vertices:

$\{1\}, \{0, 1\}, \{3\}, \{3, 0\}, \{3, 1\}, \{3, 1, 0\}, \{3, 2\}, \{3, 2, 1\}$

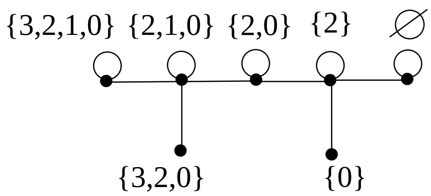


Cell: 0 ≺ 1

Adjacency: 0 ≺ 1, 1 ≺ 3, 2 ≺ 0

12 possible downsets in each cell, 5 correspond to disconnected vertices:

$\{1, 0\}, \{3\}, \{3, 0\}, \{3, 1, 0\}, \{3, 2\}$

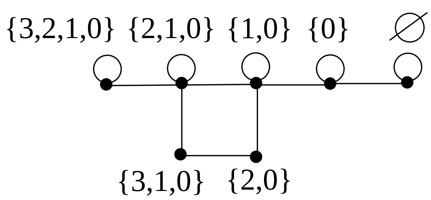


Cell: 0 ≺ 1

Adjacency: 0 ≺ 1, 1 ≺ 2, 2 ≺ 3

12 possible downsets in each cell, 5 correspond to disconnected vertices:

$\{2\}, \{3\}, \{3, 0\}, \{3, 2\}, \{3, 2, 0\}$

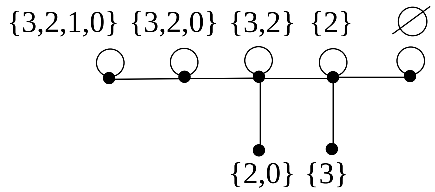


Cell: $0 \prec 1$

Adjacency: $0 \prec 1$, $2 \prec 0$, $2 \prec 3$, $3 \prec 0$

12 possible downsets in each cell, 5 correspond to disconnected vertices:

$\{0\}, \{1,0\}, \{2,1,0\}, \{3,0\}, \{3,1,0\}$

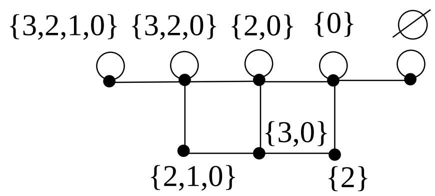


Cell: $0 \prec 1$

Adjacency: $0 \prec 2$, $2 \prec 3$, $3 \prec 1$

12 possible downsets in each cell, 4 correspond to disconnected vertices:

$\{1,0\}, \{3\}, \{3,1,0\}, \{3,2\}$

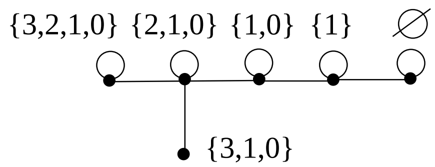


Cell: $1 \prec 0$, $0 \prec 2$, $1 \prec 2$

Adjacency: $0 \prec 2$, $1 \prec 0$, $2 \prec 3$

8 possible downsets in each cell, 2 correspond to disconnected vertices:

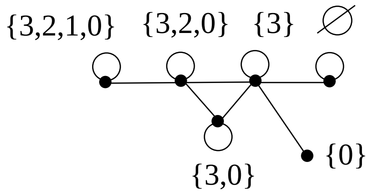
$\{3\}, \{3,1\}$



Cell: $0 \prec 1$, $0 \prec 2$, $3 \prec 2$, $2 \prec 1$

Adjacency: $2 \prec 1$, $3 \prec 0$

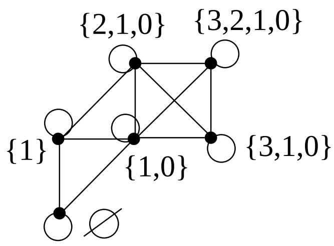
6 possible downsets in each cell



Cell: $1 \prec 0$, $0 \prec 2$, $0 \prec 3$, $1 \prec 2$, $1 \prec 3$

Adjacency: $0 \prec 3$, $1 \prec 2$

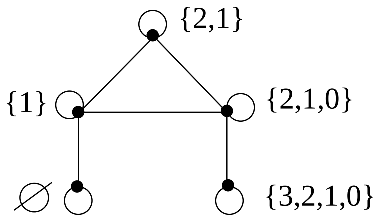
6 possible downsets in each cell



Cell: $1 \prec 0$, $2 \prec 0$, $0 \prec 3$, $1 \prec 2$, $1 \prec 3$, $2 \prec 3$

Adjacency: $0 \prec 3$, $1 \prec 2$

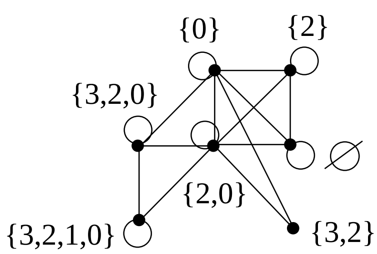
5 possible downsets in each cell



Cell: $0 \prec 1$, $2 \prec 1$, $3 \prec 1$, $2 \prec 3$

Adjacency: $0 \prec 3$, $2 \prec 1$

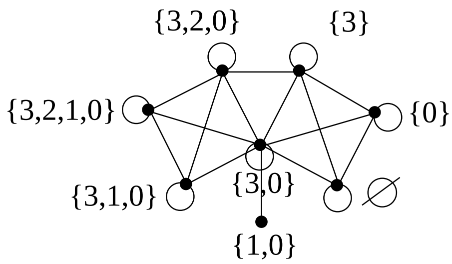
7 possible downsets in each cell



Cell: $0 \prec 1$, $0 \prec 2$, $3 \prec 2$

Adjacency: $0 \prec 1$, $3 \prec 1$, $3 \prec 2$

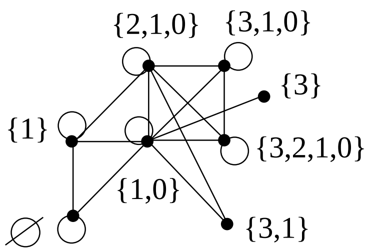
8 possible downsets in each cell



Cell: $1 \prec 0$, $0 \prec 2$, $1 \prec 2$

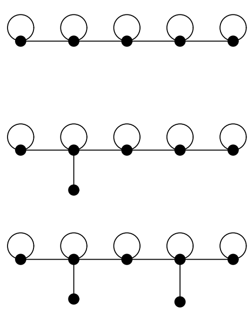
Adjacency: $0 \prec 3$, $1 \prec 2$

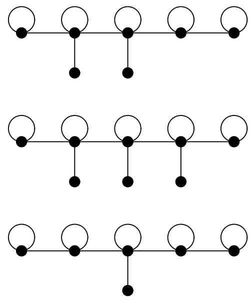
8 possible downsets in each cell



A.8 "Prickly" looped paths are $\equiv_{\mathsf{AP}} \#BIS$

We define a "prickly" looped path to be any graph comprising a copy of P_q^* ($q \geq 3$) where each non-end looped vertex has, in addition to its two immediate looped neighbours, an optional degree-1 vertex to which it is also adjacent. (Note that our definition includes P_q^* itself.) Here, for example, are all the prickly looped paths on 5 vertices.





We now show that all prickly looped paths are $\equiv_{\mathsf{AP}}\#BIS$-easy. In actual fact, it is easy to show that they are $\equiv_{\mathsf{AP}}\#BIS$-hard also. To see that they are $\equiv_{\mathsf{AP}}\#BIS$-hard, note that a prickly looped path is either P_q^* - which we already know to be $\equiv_{\mathsf{AP}}\#BIS$-hard - or it has at least one degree-1 vertex. In the latter case, we know that using a maxdeg gadget to pick out maximum degree (i.e. degree-4) vertices can be used to point out 1 or more subgraphs isomorphic to *2-wrench*, and we already know *2-wrench* is $\equiv_{\mathsf{AP}}\#BIS$-hard. The $\equiv_{\mathsf{AP}}\#BIS$-easiness proof is as follows.

Let H be a prickly looped path, and let $q \geq 3$ be the length of its looped path subgraph. We reduce $\#H$ to $\#DownSets$ as follows. Let $G = (V(G), E(G))$ be the input to $\#H$. We code up the partial order P - the input to $\#DownSets$ - as follows. For each $u \in V(G)$ we introduce a cell containing $q-1$ elements $\{u_0, u_1, ..., u_{q-2}\}$. (We impose an appropriate partial order on these elements in due course.) For every edge $\{u,v\} \in E(G)$ we add constraints $u_i \prec v_j$ and $v_i \prec u_j$ (for all $i < j$).

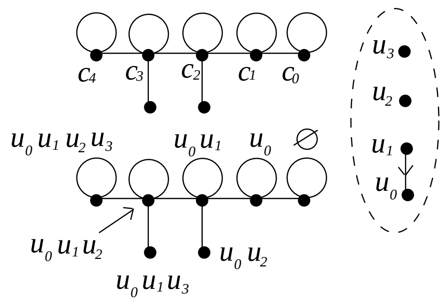
Now, consider H. Enumerate each looped colour of H in right-to-left fashion i.e. so reading from left to right the looped colours are c_{q-1}, c_{q-2},...,c_0. For each c_i ($1 \leq i \leq (q-2)$) that does *not* have a degree-1 vertex attached to it, add (for all $u \in V(G)$) the constraint $u_{i-1} \prec u_i$. This completes the construction of P. We claim that the graph induced by $DownSets(P)$ comprises H plus (potentially) a number of disconnected, unlooped (i.e. degree-0) vertices. This gives us an expression $\#DownSets(P) = \#H(G) + k$ (where k is the number of degree-0 vertices induced) and we know from Lemma 6.6 that, because k is constant, this can easily be manipulated to give us $\#H \leq_{\mathsf{AP}} \#DownSets$.

So, we need to show that H is induced by $DownSets(P)$. First, note that P_q^* is always a subgraph of the graph induced by P. To see this, consider that (for all $u \in V(G)$) the q downsets $\emptyset, \{u_0\}, \{u_0, u_1\}, ..., \{u_0, u_1, ..., u_{q-2}\}$ are always valid within a cell, and these correspond to $c_0, c_1, ..., c_{q-1}$ respectively. To see this, note that each of them corresponds to a looped colour because in each case it is permissible for the same downset to be in two adjacent cells. Furthermore, any two (distinct) downsets from the list can only occur in adjacent cells if they differ in size by one element. So this corresponds to the subgraph P_q^*. Now, it remains to show that all remaining downsets within a cell either correspond to disconnected, degree-0 colours or correspond to one of the degree-1 vertices.

Now, consider a vertex $u \in V(G)$. Suppose D_u is the downset in the cell representing u. Observe that, if the highest element in D_u is u_i (for $i \geq 2$) but D_u does not contain all elements u_0 through u_{i-2}, D_u represents a degree-0 vertex. To see why this is, suppose there exists a vertex v such that $\{u, v\} \in E(G)$; let D_v be the downset in the cell representing v. Now, we see that D_v must contain element v_{i-1} because of the constraints we put in place to encode edges of G. Following the partial order back in the direction of D_u, it follows that D_u must contain all elements u_0 through u_{i-2}, but this is not the case so D_u must represent a degree-0 colour.

So we know that, as a minimum, the elements $\{u_0, ..., u_{i-2}, u_i\}$ are in D_u. If the constraint $u_{i-1} \prec u_i$ is in the cell - and it will be if there is *not* a degree-1 vertex attached to c_i - then D_v must also contain u_{i-1}. However, if the constraint is not there (i.e. there *is* a degree-1 vertex attached to c_i) then we have the option of leaving u_{i-1} in or out. If we leave it in, then (as we have shown) D_u will behave like colour c_{i+1}. However, consider what happens if we omit element u_{i-1}. In this case D_u cannot be adjacent to an equivalent downset in an adjacent cell - so the colour D_u represents is unlooped - and in fact if D_u is adjacent to a downset D_v then D_v must be the unique downset containing elements elements v_0 through v_{i-1}, which represents c_i. (To summarise, therefore, degree-1 vertices are represented by those downsets that are missing the second-to-top element. In such cases, if the top element has index i, the degree-1 vertex is adjacent to c_i.) This completes the proof. $\square$

The following figure shows an example of the reduction in action. The structure inside the dotted ellipse shows how a vertex $u \in V(G)$ would be coded up under $\#DownSets$ if we wished to code up the H graph at the top. In the diagram, the labels on the lower copy of the graph H denote which downset corresponds to that colour (so $\{u_0\}$ corresponds to c_1, for example.) For clarity we have not shown the multiple degree-0 vertices that would also be induced, nor the edge-encodings.



Finally, note that some of the prickly looped paths shown in the Appendix A.7 catalogue do not use the precise construction described above. This is because for a given H there may be multiple distinct ways to code it up as a $\#DownSets$ problem.

A.9 $\#DownSets$ proofs (2)

Graph: 12

Left cell (on 2 elements):

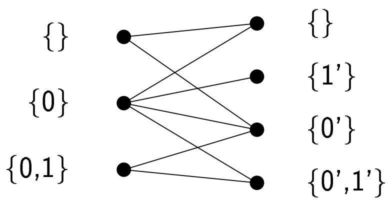
0 $\prec$ 1

Right cell (on 2 elements):

(none)

Adjacency:

0 $\prec$ 1' , 0' $\prec$ 1



Graph: 13

Left cell (on 2 elements):

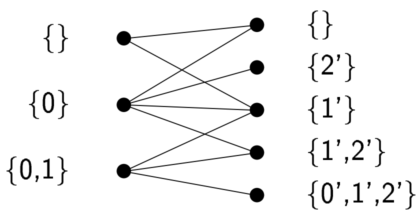
$0 \prec 1$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$0 \prec 2'$, $1 \prec 0'$, $1' \prec 1$



Graph: 14

Left cell (on 3 elements):

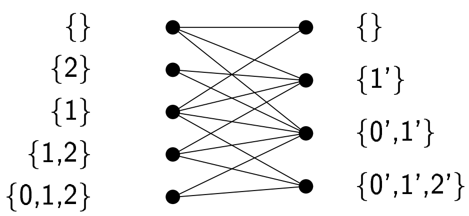
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $0' \prec 2'$, $1' \prec 2'$

Adjacency:

$0' \prec 0$, $1 \prec 2'$, $1' \prec 2$



Graph: 15

Left cell (on 2 elements):

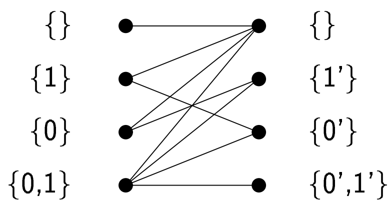
(none)

Right cell (on 2 elements):

(none)

Adjacency:

$0 \prec 1'$, $1 \prec 0'$



Graph: 16

Left cell (on 3 elements):

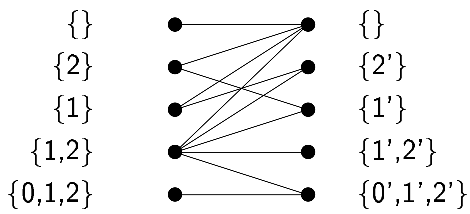
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$0' \prec 0$, $1 \prec 2'$, $2 \prec 1'$



Graph: 17

Left cell (on 3 elements):

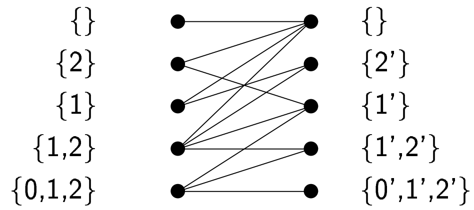
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$0 \prec 0'$, $1' \prec 0$, $1 \prec 2'$, $2 \prec 1'$



Graph: 18

Left cell (on 3 elements):

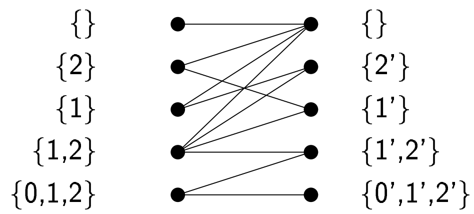
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$0 \prec 0'$, $1' \prec 0$, $2' \prec 0$, $1 \prec 2'$, $2 \prec 1'$



Graph: 19

Left cell (on 3 elements):

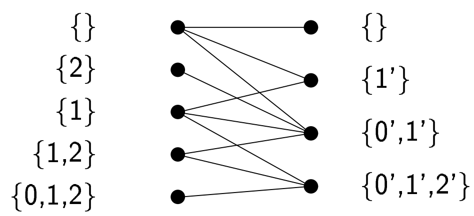
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $0' \prec 2'$, $1' \prec 2'$

Adjacency:

$2' \prec 0$, $1' \prec 1$, $1 \prec 2'$, $0' \prec 2$



Graph: 20

Left cell (on 3 elements):

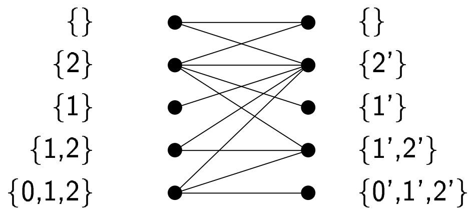
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$0 \prec 0'$, $2' \prec 1$, $2 \prec 1'$



Graph: 21

Left cell (on 3 elements):

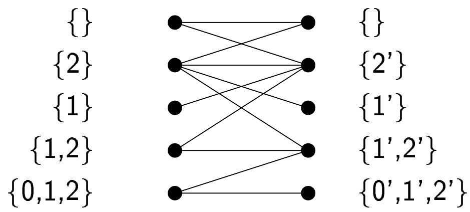
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$0 \prec 0'$, $1' \prec 0$, $2' \prec 1$, $2 \prec 1'$



Graph: 22

Left cell (on 3 elements):

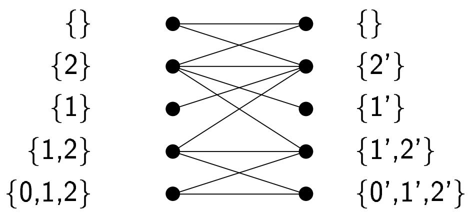
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $2' \prec 0'$

Adjacency:

$1' \prec 0$, $1 \prec 0'$, $2' \prec 1$, $2 \prec 1'$



Graph: 23

Left cell (on 3 elements):

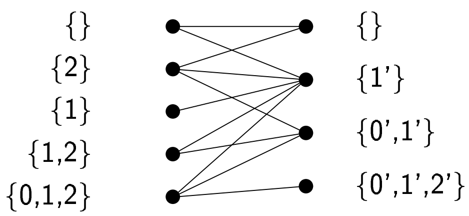
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $0' \prec 2'$, $1' \prec 2'$

Adjacency:

$0 \prec 2'$, $1' \prec 1$, $2 \prec 0'$



Graph: 24

Left cell (on 3 elements):

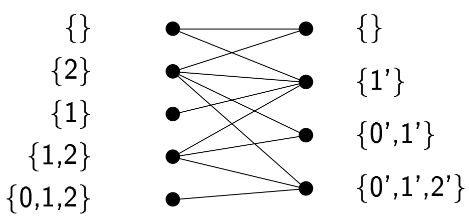
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$1' \prec 0'$, $0' \prec 2'$, $1' \prec 2'$

Adjacency:

$2' \prec 0$, $1' \prec 1$, $2 \prec 0'$



Graph: 25

Left cell (on 3 elements):

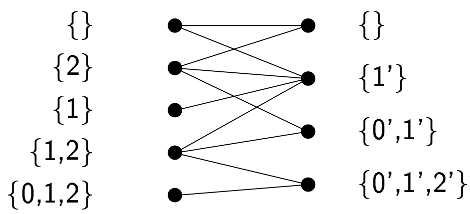
1 ≺ 0, 2 ≺ 0

Right cell (on 3 elements):

1' ≺ 0' , 0' ≺ 2' , 1' ≺ 2'

Adjacency:

2' ≺ 0 , 1' ≺ 1 , 1 ≺ 2' , 2 ≺ 0'



Graph: 26

Left cell (on 3 elements):

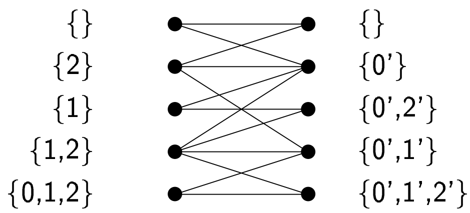
1 ≺ 0, 2 ≺ 0

Right cell (on 3 elements):

0' ≺ 1' , 0' ≺ 2'

Adjacency:

1' ≺ 0 , 0' ≺ 1 , 1 ≺ 2' , 2 ≺ 1'



Graph: 27

Left cell (on 3 elements):

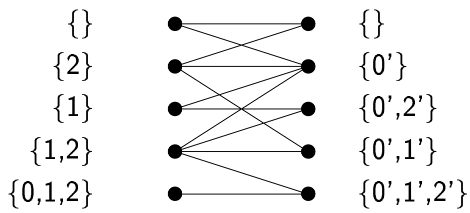
1 ≺ 0, 2 ≺ 0

Right cell (on 3 elements):

0' ≺ 1' , 0' ≺ 2'

Adjacency:

1' ≺ 0 , 2' ≺ 0 , 0' ≺ 1 , 1 ≺ 2' , 2 ≺ 1'



Graph: 28

Left cell (on 3 elements):

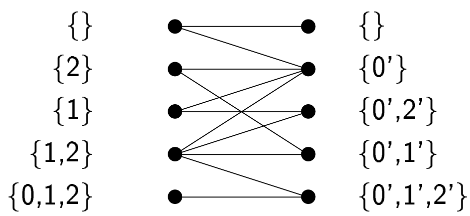
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$0' \prec 1'$, $0' \prec 2'$

Adjacency:

$1' \prec 0$, $2' \prec 0$, $0' \prec 1$, $1 \prec 2'$, $0' \prec 2$, $2 \prec 1'$



Graph: 29

Left cell (on 3 elements):

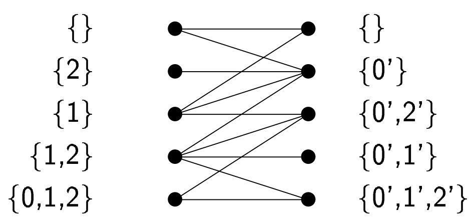
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$0' \prec 1'$, $0' \prec 2'$

Adjacency:

$2' \prec 0$, $1 \prec 1'$, $1 \prec 2'$, $0' \prec 2$, $2 \prec 1'$



Graph: 30

Left cell (on 3 elements):

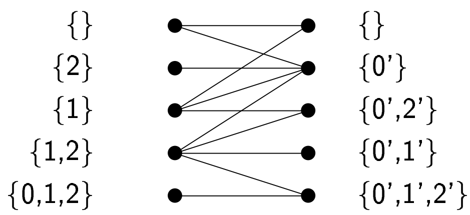
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

0' $\prec$ 1' , 0' $\prec$ 2'

Adjacency:

1' $\prec$ 0 , 2' $\prec$ 0 , 1 $\prec$ 1' , 1 $\prec$ 2' , 0' $\prec$ 2 , 2 $\prec$ 1'



Graph: 31

Left cell (on 3 elements):

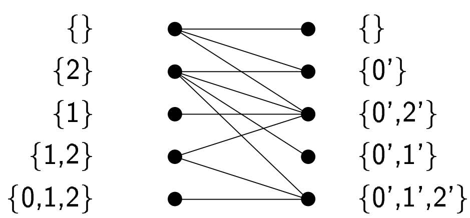
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

0' $\prec$ 1' , 0' $\prec$ 2'

Adjacency:

1' $\prec$ 0 , 2' $\prec$ 1 , 0' $\prec$ 2 , 2 $\prec$ 1'



Graph: 32

Left cell (on 3 elements):

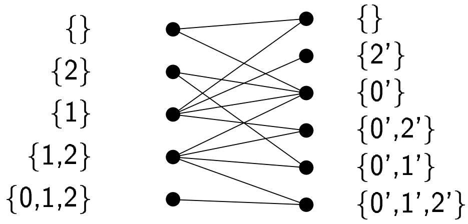
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

0' $\prec$ 1'

Adjacency:

1' $\prec$ 0 , 2' $\prec$ 0 , 1 $\prec$ 2' , 0' $\prec$ 2 , 2 $\prec$ 1'



Graph: 33

Left cell (on 3 elements):

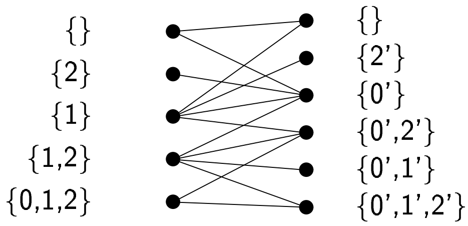
$1 \prec 0$, $2 \prec 0$

Right cell (on 3 elements):

$0' \prec 1'$

Adjacency:

$2' \prec 0$, $1 \prec 1'$, $1 \prec 2'$, $0' \prec 2$, $2 \prec 1'$



Graph: 34

Left cell (on 3 elements):

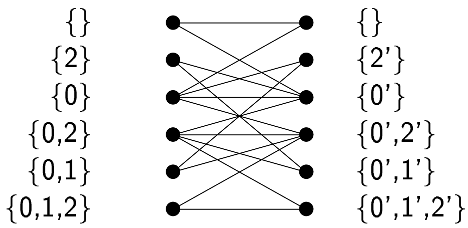
$0 \prec 1$

Right cell (on 3 elements):

$0' \prec 1'$

Adjacency:

$0 \prec 2'$, $2' \prec 1$, $0' \prec 2$, $2 \prec 1'$



Graph: 35

Left cell (on 3 elements):

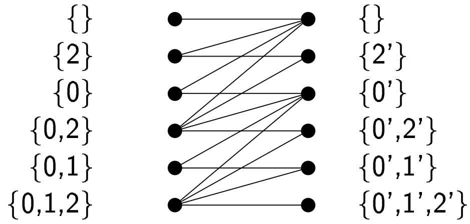
$0 \prec 1$

Right cell (on 3 elements):

$0' \prec 1'$

Adjacency:

$0 \prec 0'$, $0' \prec 1$, $1 \prec 1'$, $2 \prec 2'$



A.10 Technical comment for junction-to-EdgeSwap(junction) proof (from page 265).

Here we do not aim to provide a full proof of the reduction, but instead aim to show that s and t can be chosen such that 3^s3^t is adequately close to 4^s1^t. Indeed, it is not difficult to choose k big enough in terms of s, t and n so that K is exponentially likely to be coloured $\{r, b, g\}$. Similarly, it is not difficult to ensure both s and t are large enough so configurations on $(L[.], R[.])$ other than those listed are dwarfed. (We are helped here because the only configurations possible other than those listed are subsets of the listed configurations, and thus are inherently exponentially inferior.) To assist us later on, let x be the smallest that both s and t must be to ensure the inferior configurations are dwarfed.

As we argue, the centre colour of the junction is represented by the configuration (rbg, rbg) and the degree-1 colours of the junction by configurations $(rbgr', r)$, $(rbgg', g)$ and $(rbgb', b)$. Hence, a vertex $v \in V_R(G)$ coloured with the centre colour comes up 3^s3^t times in $(L[v], R[v])$ - note how we don't need to write $\nu(s, 3)3^t$ - and a vertex $v \in V_R(G)$ coloured with a degree-1 colour comes up $\nu(s, 4)$ times.

We are going to use the standard rounding technique, taking $Z = \nu(k, 3)(3^s3^t)^{n_r}$

as the divisor. (This is the number of times a colouring with all $V_R(G)$ vertices coloured using the centre colour comes up in G'.) Now, let:

$$l = \frac{3^s 3^t}{\nu(s,4)}$$

and let $f = \max(l, l^{-1})$. It follows that

$$Nf^{-n_r} \leq \frac{\#H'(G')}{Z} \leq Nf^{n_r} + \frac{|Y_0|}{Z} \tag{A.6}$$

where N is the number of junction colourings, and Y_0 is the set of non-full colourings. (Here, non-full colourings are those where K is either not coloured $\{r, b, g\}$ or at least one $(L[.], R[.])$ pair is not coloured with one of the listed configurations. We don't discuss Y_0 here any further, except to say that our later choice of $s, t \geq x$, and our presumption that k is huge, ensures $|Y_0|/Z \leq 1/4$.)

Now, suppose $\#\hat{H'}(G')$ is the value returned from our approximation oracle. By the same argument as in Lemma 2.8 (on page 77) we need the following to hold:

$$e^{-\epsilon/21}(N - \frac{1}{4}) \leq \frac{\#\hat{H'}(G')}{Z} \leq e^{\epsilon/21}(N + \frac{1}{4})$$

If we use δ (to be determined) as the accuracy to our oracle it follows from (A.6) that

$$e^{-\delta} N f^{-n_r} \leq \frac{\#\hat{H'}(G')}{Z} \leq e^{\delta}\left(Nf^{n_r} + \frac{|Y_0|}{Z}\right) \tag{A.7}$$

So, if we take $\delta = \epsilon/42$, choose s, t, k such that $|Y_0|/Z \leq 1/4$ (which we only mention in passing here) and most importantly ensure that

$$e^{-\epsilon/42n_r} \leq \frac{3^s 3^t}{\nu(s,4)} \leq e^{\epsilon/42n_r} \tag{A.8}$$

it follows from (A.6) and (A.7) that we are done. Ideally we would choose $s, t \in \mathbb{N}^+$ such that $3^s 3^t = \nu(s,4)$ but this is not possible. However, we know that $4^s(1 - \exp(-s/8)) \leq \nu(s,4) \leq 4^s$ so if we can show $e^{-\epsilon/84n_r} \leq (1 - \exp(-s/8))$ and also show that

$$e^{-\epsilon/84n_r} \leq (3/4)^s 3^t \leq e^{\epsilon/84n_r} \tag{A.9}$$

then this satisfies (A.8). First, we deal with showing $e^{-\epsilon/84n_r} \leq (1 - \exp(-s/8))$. Note that $e^{-\epsilon/84n_r} \leq 1 - \epsilon/168n_r$. Hence we are satisfied as long as $s \geq s_0$ where

$s_0 = \lceil 8 \ln(168 n_r/\epsilon) \rceil$. We use this information to guide our choice of a_0, below. Now, we need to prove (A.9). We do this by setting t, s to the values a, b (respectively[7]) returned by technical Lemma 4.2 (on page 139.) The parameters we pass to this lemma are as follows: $c_1 = 4/3$, $c_2 = 3$, $a_0 = x + s_0$ and $q = 0$. The lemma guarantees that the accuracy bound is met, that our chosen values of s, t are not too big, and that both are at least as big as $x + s_0$, which is what we require.

A.11 Index of connected graphs with 4 or fewer vertices

65 graphs in total. 47 of them are $\equiv_{\mathsf{AP}} \#SAT$. For those H not identified as being $\equiv_{\mathsf{AP}} \#SAT$, the complexity breakdown is as follows:

$FPRAS$able: 1, 2, 3, 5, 6, 15, 16, 46, 65

$\equiv_{\mathsf{AP}} \#BIS$: 11, 21, 24, 33, 45, 60

$\equiv_{\mathsf{AP}} \#BIS$-hard, but not proven to be $\equiv_{\mathsf{AP}} \#SAT$: 30, 50, 51

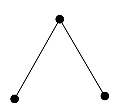
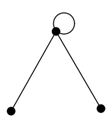
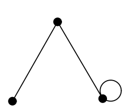
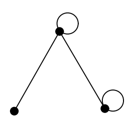
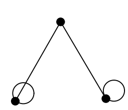
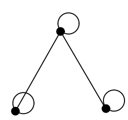
The following matrix lists all 65 graphs. Note that, for each graph, the graphic that appears here will be slightly different to that used in Chapter 2. This is because the images in the matrix were generated systematically by computer, with a view to helping the reader appreciate that Chapter 2 is exhaustive in its consideration of H with 4 or fewer vertices.

1

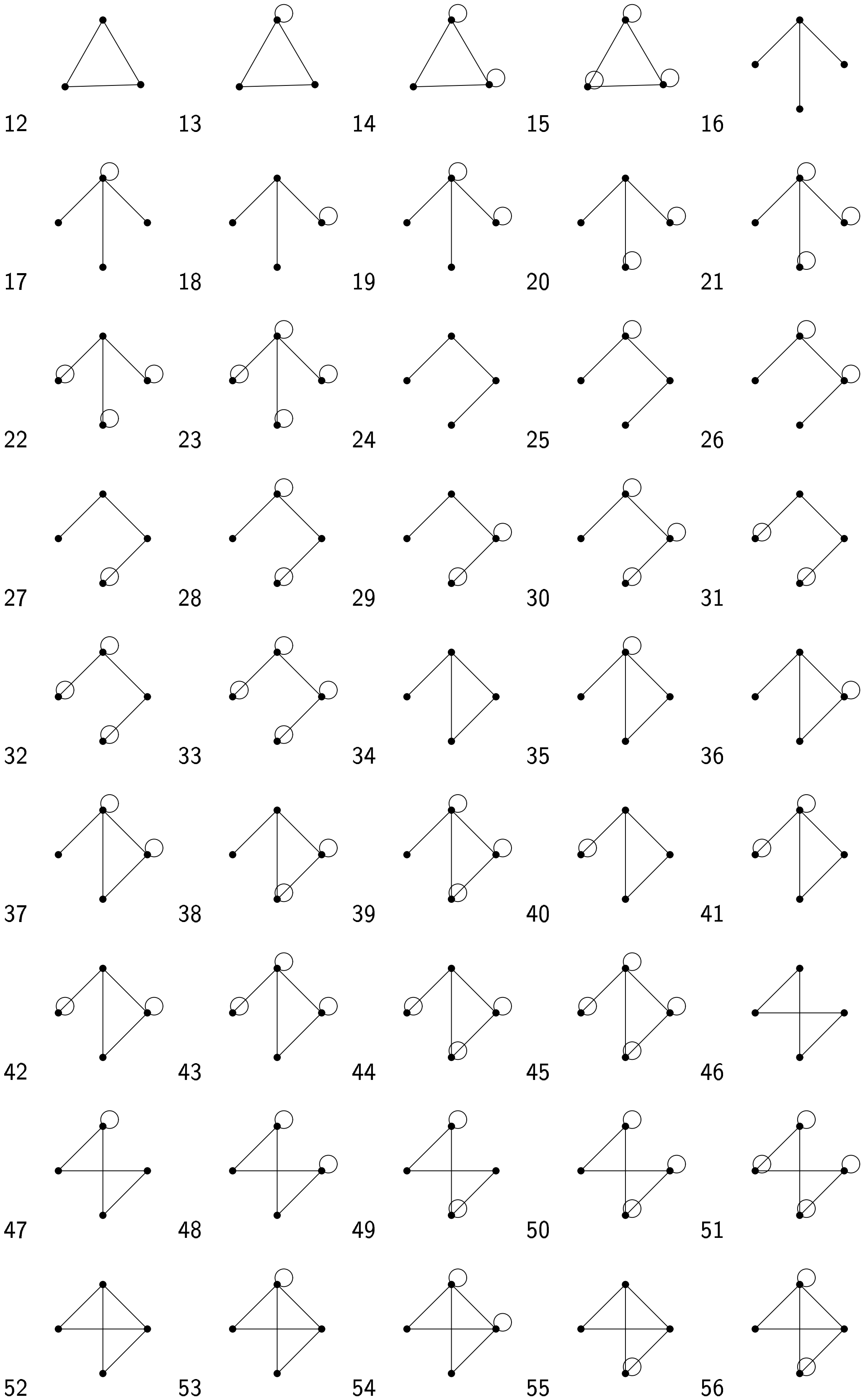
2

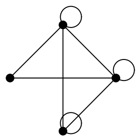
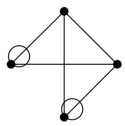
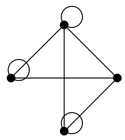
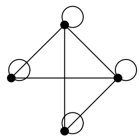
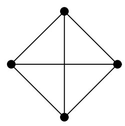
3

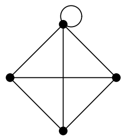
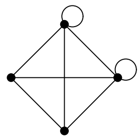
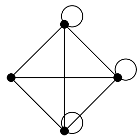
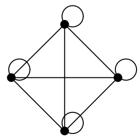
4

5

6

7

8

9

10

11

[7]Note the ordering.


12 13 14 15 16
17 18 19 20 21
22 23 24 25 26
27 28 29 30 31
32 33 34 35 36
37 38 39 40 41
42 43 44 45 46
47 48 49 50 51
52 53 54 55 56

57  58  59  60  61 

62  63  64  65 

Graph: 1

Complexity: $FPRAS$able

Comment: Complete bipartite graph (i.e. $K_{1,0}$), so trivial. See Section 2.1.1 on page 25.

Graph: 2

Complexity: $FPRAS$able

Comment: Fully looped complete graph on 1 vertex (i.e. K_1^*), so trivial. See Section 2.1.1 on page 25.

Graph: 3

Complexity: $FPRAS$able

Comment: Complete bipartite graph (i.e. $K_{1,1}$), so trivial. See Section 2.1.1 on page 25.

Graph: 4

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: This problem equivalent to counting independent sets (IS). See Section 2.1.3 on page 27.

Graph: 5

Complexity: $FPRAS$able

Comment: Fully looped complete graph on 2 vertices (K_2^*), so trivial. See Section

2.1.1 on page 25.

Graph: 6

Complexity: $FPRAS$able

Comment: Complete bipartite graph (i.e. $K_{1,2}$), so trivial. See Section 2.1.1 on page 25.

Graph: 7

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Weighted version of IS, so see Lemma 2.3 on page 40. Also classified by Lemma 5.1 (page 160).

Graph: 8

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 32. Also classified by Corollary 5.11 (page 190).

Graph: 9

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Known as the *1-wrench*. Ad-hoc reduction on page 45.

Graph: 10

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 35. Also classified by Corollary 5.11 on page 190.

Graph: 11

Complexity: $\equiv_{\mathsf{AP}} \#BIS$

Comment: Known as *2-WR* or P_3^*. Ad-hoc reduction on page 53. See also Lemma 2.10 (page 84) and Lemma 2.13 (page 90).

Graph: 12

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: One of the "Hell and Nešetřil" graphs: see Section 2.1.2 on page 26. Equivalent to 3-colouring.

Graph: 13

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Sometimes known as the "pyramid". Ad-hoc reduction on page 41. See also Lemma 5.4 on page 164.

Graph: 14

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Sometimes known as the "ear" graph. Weighted version of IS, see Lemma 2.3 on page 40.

Graph: 15

Complexity: $FPRAS$able

Comment: Complete looped graph on 3 vertices (i.e. K_3^*) so trivial, see Section 2.1.1 on page 25.

Graph: 16

Complexity: $FPRAS$able

Comment: Complete bipartite graph (i.e. $K_{1,3}$), so trivial. See Section 2.1.1 on page 25.

Graph: 17

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Weighted version of IS, see Lemma 2.3. See also Lemma 5.1 (page 160.)

Graph: 18

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 58. See also Corollary 5.11 on page 190.

Graph: 19

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 71.

Graph: 20

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 58. See also Corollary 5.11 on page 190.

Graph: 21

Complexity: $\equiv_{\mathsf{AP}} \#BIS$

Comment: Known as the *2-wrench*. Shown to be $\equiv_{\mathsf{AP}} \#BIS$ in [8]. See also Appendix A.8 on page 299.

Graph: 22

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 58. See also Corollary 5.11 on page 190.

Graph: 23

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Known as *3-WR*. Ad-hoc reduction on page 73.

Graph: 24

Complexity: $\equiv_{\mathsf{AP}} \#BIS$

Comment: Known as P_4. Essentially equivalent to $\#BIS$. See section on bipartisation on page 54.

Graph: 25

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 59. See also Corollary 5.11 on page 190.

Graph: 26

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 61.

Graph: 27

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 59. See also Corollary 5.11 on page 190.

Graph: 28

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 60. See also Corollary 5.11 on page 190 and Lemma 5.4 on page 164.

Graph: 29

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 61.

Graph: 30

Complexity: $\equiv_{\mathsf{AP}} \#BIS$-hard, but we do not know if it is $\equiv_{\mathsf{AP}} \#SAT$

Comment: See Section 2.5 on page 101.

Graph: 31

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 59. See also Corollary 5.11 on page 190.

Graph: 32

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 60.

Graph: 33

Complexity: $\equiv_{\mathsf{AP}}\#BIS$

Comment: Known as P_4^*. (Also known as "Beach" colourings in [8].) See also Lemma 2.13 on page 90.

Graph: 34

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: One of the "Hell and Nešetřil" graphs: see Section 2.1.2 on page 26.

Graph: 35

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 70. Notable for being the "square" of the graph IS - see Appendix A.3. (See also Lemma 5.5 on page 167 for a $\equiv_{\mathsf{SP}}SAT$ reduction.)

Graph: 36

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 63. See also Corollary 5.11 on page 190 and Lemma 5.4 on page 164.

Graph: 37

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 68. See also Corollary 5.8 on page 184.

Graph: 38

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 65. See also Corollary 5.8 on page 184.

Graph: 39

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 73.

Graph: 40

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 61. See also Corollary 5.11 on page 190.

Graph: 41

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 75.

Graph: 42

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 65. See also Corollary 5.11 on page 190, and Lemma 5.4 on page 164.

Graph: 43

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 68. See also Corollary 5.8 on page 184.

Graph: 44

Complexity: $\equiv_{\mathsf{AP}}\#SAT$

Comment: Ad-hoc reduction on page 65. See also Corollary 5.8 on page 184.

Graph: 45

Complexity: $\equiv_{\mathsf{AP}}\#BIS$

Comment: A weighted variant of *2-WR*. See Lemma 2.10 (page 84) and Lemma 2.13 (page 90).

Graph: 46

Complexity: $FPRAS$able

Comment: Complete bipartite graph (i.e. $K_{2,2}$), so trivial. See Section 2.1.1 on page

25.

Graph: 47

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 61. See also Corollary 5.11 on page 190.

Graph: 48

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 62. See also Corollary 5.11 on page 190.

Graph: 49

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 62.

Graph: 50

Complexity: $\equiv_{\mathsf{AP}} \#BIS$-hard, but we do not know if it is $\equiv_{\mathsf{AP}} \#SAT$

Comment: Sometimes known as the "crossbow". See Section 2.5 on page 101.

Graph: 51

Complexity: $\equiv_{\mathsf{AP}} \#BIS$-hard, but we do not know if it is $\equiv_{\mathsf{AP}} \#SAT$

Comment: Known as C_4^*. See Section 2.5 on page 101.

Graph: 52

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: One of the "Hell and Nešetřil" graphs: see Section 2.1.2 on page 26.

Graph: 53

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 69. See Lemma 5.4 on page 164.

Graph: 54

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Weighted version of IS, see Lemma 2.3. See also Lemma 5.1 (page 160.)

Graph: 55

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 62. See also Corollary 5.11 on page 190.

Graph: 56

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 68. See also Corollary 5.8 on page 184.

Graph: 57

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 72.

Graph: 58

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 62. See also Corollary 5.11 on page 190.

Graph: 59

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 69. See also Corollary 5.8 on page 184.

Graph: 60

Complexity: $\equiv_{\mathsf{AP}} \#BIS$

Comment: A weighted variant of *2-WR*. See Lemma 2.10 (page 84) and Lemma 2.13 (page 90).

Graph: 61

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: One of the "Hell and Nešetřil" graphs: see Section 2.1.2 on page 26.

Graph: 62

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 70. (See also Lemma 5.5 on page 167 for a $\equiv_{\mathsf{SP}} SAT$ reduction.)

Graph: 63

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Ad-hoc reduction on page 70. (See also Lemma 5.5 on page 167 for a $\equiv_{\mathsf{SP}} SAT$ reduction.)

Graph: 64

Complexity: $\equiv_{\mathsf{AP}} \#SAT$

Comment: Weighted version of IS, see Lemma 2.3. See also Lemma 5.1 (page 160.)

Graph: 65

Complexity: $FPRAS$able

Comment: Complete looped graph on 4 vertices (i.e. K_4^*) so trivial, see Section 2.1.1 on page 25.

A.12 Technical glossary and abbreviations

In this section we reproduce terminology that is commonly used throughout the thesis.

A.12.1 Abbreviations

AP - Approximation Preserving - Section 1.2.4
CNF - Conjunctive Normal Form
$DGGJ$ - Dyer, Goldberg, Greenhill and Jerrum - Section 1.2.4
DNF - Disjunctive Normal Form
$FPAS$ - Fully Polynomial Approximate Sampler - Section 3.10
$FPAUS$ - Fully Polynomial Almost Uniform Sampler - Section 3.10
$FPRAS$ - Fully Polynomial Randomized Approximation Scheme - Section 1.2.2
JVV - Jerrum, Valiant and Vazirani - Section 1.2.3
LHS - Left-Hand Side
$MCMC$ - Markov Chain Monte Carlo (method) - Section 1.2.2
NP - Nondeterministic Polynomial (time)
$\#P$ - The class "number-P" - Section 1.2.1
P - Polynomial (time)
$PAUS$ - Polynomial Almost Uniform Sampler - Section 3.4
RAS - Randomized Approximation Scheme - Section 1.2.2
RHS - Right-Hand Side
RP - Randomized Polynomial (time)
SP - Sampling Preserving - Section 3.4

A.12.2 Counting-problems

The problem of counting...

$\#1P1NSAT$ - Section 7.3.3
$\#BIS$ - ...independent sets in a bipartite graph - Section 1.2.4
$\#Downsets$ - ...downsets in a partial order - Section 2.4.1
$\#IS$ - ...independent sets in a graph - Section 2.1.3
$\#LargeCut$ - ...size-m cuts in a graph - page 45
$\#LargeIS$ - ...size-m independent sets in a graph - Section 2.1.3
$\#LargeIS - Cubic$ - ...size-m independent sets in a cubic graph - Section 3.9
$\#MaxBIS$ - ...maximum-size independent sets in a bipartite graph - page 77
$\#RH\Pi_1$ - Section 7.3.3
$\#SAT$ - ...satisfying assignments of a particular boolean equation - Section 1.2.1

A.12.3 H-related

$adj(c)$ - Adjacency set of the vertex c
$adj'(c)$ - "Effective" adjacency set of a vertex c - page 38
$bi(H)$ - The bipartisation of H - page 54
$bi\text{-}q\text{-}col$ - Bipartite q-colouring - Section 1.3.3, Figure 2.6, Section 7.6.1
C_k - Cycle on k vertices
C_k^* - Fully-looped cycle on k vertices
compact form - Section 2.2.1

A.12.4 Gadget-related

$X : Y$ - See proof that begins on page 45
cliqueset - See proof that begins on page 32
configuration - See proof that begins on page 41
descending polynomial - page 89
edgespan - See proof that begins on page 45
$Map(Col, I_2)$ - See proof that begins on page 41
maxcliquegrab - See proof of Lemma 5.7 (starts page 176)
maxdeg - pages 59-60
maxdegweight - page 89
maximal configuration - See end of proof that begins on page 41
maxloopgrab - Section 5.5
misguided configuration - See proof of Lemma 2.12 (starts page 84)
multiple-vertex switching - page 193
single-vertex switching - page 193

A.12.5 Miscellaneous

$\equiv_{\mathsf{AP}}\#X$ - Section 1.2.4
$\equiv_{\mathsf{SP}}X$ - Section 3.4
$\#X\leq_{\mathsf{AP}}\#Y$ - $\#X$ is AP-reducible to $\#Y$ - Section 1.2.4
$X\leq_{\mathsf{SP}}Y$ - X is SP-reducible to Y - Section 3.4
$[k]$ - $\{0, 1, ..., k\}$
d_{TV} - Variation distance - Section 3.4
$\mathbb{N}^{+}$ - The set of non-zero natural numbers
$Perm(i, j)$ - The number of ways of picking i elements from j (order matters)
$\mathbb{Q}^{+}$ - The set of non-zero rational numbers
$\nu(a, b)$ - The number of surjective functions from a elements to b elements - See proof beginning on page 41, also see Lemma 2.4 (page 50)

Bibliography

[1] G.R. Brightwell and P. Winkler, Graph homomorphisms and phase transitions, *Journal of Combinatorial Theory, Series B,* **77** (1999) 221–262.

[2] G.R. Brightwell and P. Winkler, Gibbs measures and dismantleable graphs, *Journal of Combinatorial Theory, Series B,* **78(1)** 141-166 (2000)

[3] C. Cooper, M.E. Dyer and A.M. Frieze, On Markov chains for randomly H-colouring a graph, *Journal of Algorithms,* **39(1)** (2001) 117–134.

[4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, published by MIT Press and McGraw-Hill

[5] J. Diaz, M. Serna and D.M. Thilikos, Counting list H-colourings and variants. Technical Report LSI-01-27-R, Department de Llenguatges i Sistemes InformAtics, Universitàt Politècnica de Catalunya, Barcelona, 2001. See http://www.lsi.upc.es/∼sedthilk/mypapers.html.

[6] M.E. Dyer, A.M. Frieze and M.R. Jerrum, On counting independent sets in sparse graphs, *SIAM Journal on Computing* **31** (2002) 1527–1541

[7] M.E. Dyer and C.S. Greenhill, Random walks on combinatorial objects. In J.D. Lamb and D.A. Preece, editors, *Surveys in Combinatorics*, volume 267 of *London Mathematical Society Lecture Note Series*, pages 101–136. Cambridge University Press, 1999.

[8] M.E. Dyer, C.S. Greenhill, L.A. Goldberg, and M.R. Jerrum, On the relative complexity of approximate counting problems. In *Proceedings of Approx*, volume 1913 of *Springer Lecture Notes in Computer Science*, pages 108-119, 2000.

[9] M.E. Dyer and C.S. Greenhill, On Markov chains for independent sets, *Journal of Algorithms,* **35** (2000) 17–49.

[10] M.E. Dyer and C.S. Greenhill, The complexity of counting graph homomorphisms. *Random Structures and Algorithms,* **17** (2000) 260–289.

[11] M.E. Dyer, M.R. Jerrum and E. Vigoda, Rapidly mixing Markov chains for dismantleable constraint graphs, *Lecture Notes in Computer Science* **2483**, Jos D. P. Rolim, Salil P. Vadhan (Eds.): Randomization and Approximation Techniques, 6th International Workshop, RANDOM 2002, Cambridge, MA, USA, September 13-15, 2002, pages 68-77

[12] M.E. Dyer, L.A. Goldberg, and M.R. Jerrum, Counting and Sampling H-colourings, RANDOM 2002, 51-67

[13] A. Galluccio, P. Hell and J. Nešetřil, The complexity of H-coloring of bounded degree graphs, *Discrete Mathematics,* **222** (2000) 101–109.

[14] L.A. Goldberg, S.M. Kelk and M. Paterson, The complexity of choosing an H-colouring (nearly) uniformly at random, Preliminary version: Proceedings of STOC 2002 53-62. Submitted for journal publication.

[15] P. Hell and J. Nešetřil, On the complexity of H-colouring, *Journal of Combinatorial Theory, Series B*, **48** pages 91-110, (1990).

[16] M.R. Jerrum, L.G. Valiant, and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, *Theoretical Computer Science*, **43** (1986) 169-188

[17] M.R. Jerrum and A. Sinclair, Polynomial-time approximation algorithms for the Ising model, *SIAM Journal on Computing* **22** (1993) 1087-1116

[18] M.R. Jerrum, A very simple algorithm for estimating the number of k-colorings of a low-degree graph, *Random Structures and Algorithms,* **7** (1995) 157–165.

[19] M.R. Jerrum, A. Sinclair, The Markov chain Monte Carlo method: an aproach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems* (Dorit Hochbaum, ed.), PWS, 1996, 482-520

[20] M.R. Jerrum, Counting, Sampling and Integrating: Algorithms and Complexity, Birkhäuser 2003. (See also `http://www.dcs.ed.ac.uk/home/mrj/pubs.html` .)

[21] M.R. Jerrum, A. Sinclair, E. Vigoda, A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries, Proceedings of STOC 2001, pp. 712-721

[22] R.M. Karp, M. Luby and N. Madras, Monte-Carlo approximation algorithms for enumeration problems, *Journal of Algorithms* **10** (1989), 429-448

[23] R.M. Karp, M. Luby, Monte-Carlo algorithms for enumeration and reliability problems, *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, Computer Society Press, 1983, pp. 56-64

[24] P.W. Kasteleyn, Graph theory and crystal physics. In *Graph Theory and Statistical Physics* (F. Harary, ed.), Academic Press, 1967, 43-110

[25] M. Luby and E. Vigoda. Approximately counting up to four. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 682-687. ACM Press, 1997

[26] S. Saluja, K.V. Subrahmanyam, M.N. Thakur, Descriptive complexity of $\#P$ functions, *Journal of Computer and System Sciences* **50** (1995) 493-505

[27] A. Sinclair, *Algorithms for random generation and counting: a Markov chain approach*, Progress in Theoretical Computer Science, Birkhäuser, Boston, 1993

[28] S. Toda, PP is as hard as the polynomial-time hierarchy, *SIAM Journal on Computing* **20** (1991), 865-877

[29] L.G. Valiant, The complexity of enumeration and reliability problems, *SIAM Journal on Computing* **8** (1979), 410–421.

[30] E. Vigoda, Improved bounds for sampling colorings, *Journal of Mathematical Physics*, **41** (2000) 1555–1569.