Survey of Parameter-Preserving Reductions

Philipp Kuinke

Department of Computer Science

RWTH Aachen

A thesis submitted for the degree of

Bachelor of Science

2013

Acknowledgements

I would like to acknowledge the work of my advisers Felix Reidl and Somnath Sikdar who helped me find resources and provided constant feedback and ideas for improvement. Secondly I would like to thank Walter Unger, who has shown me that theory lectures do not have to be dry and without whom I probably would not have chosen a theoretical topic for my Bachelor thesis.

Abstract

In the world of fixed parameter algorithms problem instances are broken up into a parameter k and the usual input size n. A problem is fixed parameter tractable if it can be solved in time $\mathcal{O}(f(k)g(n))$ where f is some arbitrary function and g is a polynomial.

Analogous to polynomial time reductions in the case of **NP**-hard problems, a type of reduction called parameter preserving is used to establish relations between problems in the class **FPT**. Such reductions can not only be used to re-use algorithms but also establish an internal hierarchy of running-times: informally, an increase of the parameter during the reduction hints at the target problem being harder to solve than the source problem.

The goal of this thesis is to collect, categorize and develop parameter-preserving reductions.

Contents

Contents **vii**

List of Figures **x**

1 Introduction **1**
- 1.1 On complexity . . . 2
- 1.2 Parametrized complexity theory . . . 2
  - 1.2.1 Preliminary . . . 2
  - 1.2.2 Reduction . . . 4
- 1.3 What is the point? . . . 6

2 Reductions **7**
- 2.1 Reductions, polynomial but not parameter preserving . . . 7
  - 2.1.1 Independent Set $\leq_p$ Vertex Cover . . . 8
  - 2.1.2 Clique $\leq_p$ Constraint Bipartite Vertex Cover . . . 9
- 2.2 Parameter preserving reduction in *both* directions . . . 10
  - 2.2.1 Short Non-deterministic Turing Machine Acceptance $\equiv_{pp}$ Independent Set . . . 10
  - 2.2.2 Monotone Weighted SAT $\equiv_{pp}$ Hitting Set . . . 11
  - 2.2.3 Partition into Cliques $\equiv_{pp}$ Colouring . . . 12
  - 2.2.4 Spare Allocation $\equiv_{pp}$ Constraint Bipartite Vertex Cover . . . 14
  - 2.2.5 Red-Blue Dominating Set $\equiv_{pp}$ Set Cover . . . 16
  - 2.2.6 Red-Blue Dominating Set $\equiv_{pp}$ Hitting Set . . . 18
  - 2.2.7 Vertex Cover $\equiv_{pp}$ 2-Hitting Set . . . 19
- 2.3 Parameter preserving reduction *without* parameter change . . . 20

2.3.1 Monotone Weighted SAT $\leq_{pp}$ Weighted SAT 20
2.3.2 Vertex Cover $\leq_{pp}$ Dominating Set 21
2.3.3 Vertex Cover $\leq_{pp}$ Steiner Tree 23
2.3.4 Dominating Set $\leq_{pp}$ Hitting Set 25
2.3.5 Dominating Set $\leq_{pp}$ Monotone Weighted SAT 26
2.3.6 Dominating Set $\leq_{pp}$ Center 28
2.3.7 Coloured Red-Blue Dominating set $\leq_{pp}$ Red-Blue Dominating Set 29
2.3.8 Red-Blue Dominating set $\leq_{pp}$ Steiner Tree 31
2.3.9 Coloured Small Universe Hitting Set $\leq_{pp}$ Small Universe Hitting Set 33
2.3.10 Disjoint Factors $\leq_{pp}$ Vertex Disjoint Cycles 34
2.3.11 Independent Set $\leq_{pp}$ Induced Matching 36
2.3.12 Colouring $\leq_{pp}$ Partition Into Forests 38
2.3.13 Colourful Graph Motif $\leq_{pp}$ Group Steiner Tree 41
2.3.14 Colourful Graph Motif $\leq_{pp}$ Steiner Tree 42
2.4 Parameter preserving reduction with *linear* parameter change . . 44
2.4.1 Weighted SAT $\leq_{pp}$ Dominating Set 44
2.4.2 Red-Blue Dominating Set $\leq_{pp}$ Connected Vertex Cover . . . 47
2.4.3 Red-Blue Dominating set $\leq_{pp}$ Capacitated Vertex Cover . 49
2.4.4 Group Steiner Tree $\leq_{pp}$ Directed Steiner Out-Tree 51
2.4.5 Connected Vertex Cover $\leq_{pp}$ 2-deg-Connected Feedback Vertex Set 54
2.4.6 Connected Vertex Cover $\leq_{pp}$ 2-deg-Connected Odd Cycle Transversal 56
2.4.7 Steiner Tree $\leq_{pp}$ 2-deg-Steiner Tree 57
2.4.8 Colourful Graph Motif $\leq_{pp}$ Connected Dominating Set . . 59
2.5 Parameter preserving reduction with *polynomial* parameter change 61
2.5.1 Coloured Reduced Unique Coverage $\leq_{pp}$ Unique Coverage 61
2.6 Supposedly no parameter preserving reductions 62

3 Reduction-Graph **63**
3.1 Interpretation 63

4 Conclusions **66**
4.1 Methods used 66
4.2 Further work on visualisation and reduction search 67

References **68**

Appendix: List of Problems **70**

List of Figures

1.1 The difference between the two Vertex Cover Algorithms. 3
1.2 Illustration of the reduction from Independent Set to Clique 5

2.1 Illustration of Clique to Constraint Bipartite Vertex Cover 9
2.2 Illustration of Partition into Cliques to Colouring 12
2.3 Illustration of Spare allocation to Constraint Bipartite Vertex Cover 14
2.4 Illustration of Red-Blue Dominating Set to Set Cover. . . 16
2.5 Illustration of Vertex Cover to Dominating Set 21
2.6 Illustration of Vertex Cover to Steiner Tree 23
2.7 The graph G, that will be transformed to φ 26
2.8 Illustration of coloured Red-Blue Dominating Set to Red-Blue Dominating Set. 29
2.9 Illustration of Red-Blue Dominating Set to Steiner Tree. 31
2.10 An example of the Disjoint Factors property not satisfied 34
2.11 Illustration of Disjoint Factors to Vertex Disjoint Cycles. 35
2.12 Illustration of Independent Set to Induced Matching 36
2.13 Illustration of Colouring to Partition Into Forests 38
2.14 The corresponding solution of Partition Into Forests 39
2.15 Illustration of Colourful Graph Motif to Connected Dominating Set 42
2.16 Illustration of Red-Blue Dominating Set to Connected Vertex Cover. 47
2.17 Illustration of Red-Blue Dominating Set to Capacitated Vertex Cover. 49

2.18 Illustration of Group Steiner Tree to Directed Steiner Out-Tree. 52
2.19 Left: A solution of Group Steiner Tree Right: A solution of Directed Steiner Out-Tree 53
2.20 Illustration of Connected Vertex Cover to 2-deg-Connected Feedback Vertex Set 54
2.21 Illustration of Steiner Tree to 2-deg-Steiner Tree 57
2.22 Illustration of Colourful Graph Motif to Connected Dominating Set 60

3.1 The first big connected component 64
3.2 The remaining six smaller connected components 65

Chapter 1

Introduction

> "You cannot proceed formally from an informal specification."
>
> (Jeremy Manson)

In this work we want to look at different combinatorial problems and how they are related with respect to how fast or how hard we can find a solution. We expect to find problems that are very similar and therefore behave similar, so we can arrange them to some sort of class. Maybe we find problems that on the first glance are very similar, but are in fact very different if observed more throughout or problems that seem different, but are in their core very closely related. The method of choice to conduct these comparisons will be the *reduction*.

1.1 On complexity

The main 'unit' to measure the quality of an algorithm is the efficiency. That is, how long does it take to generate a solution for a given input. In this thesis we only consider *decision problems*: For an input x we want to give the answer *yes* or *no* based on a given characteristic, like for example "is the given number $n \in \mathbb{N}$ a prime number?". And of these problems we only look at those who are decidable, which means there is an algorithm that can always find a solution in finite time. The constraint *finite* is of course quite loose, and generally we want fast solutions, that is why we need to measure the efficiency of the algorithm. For that we use the 'Big $\mathcal{O}$ notation', please see [1] for a detailed definition. It creates a ratio of run-times for different sized inputs by giving a function which maps the input length of the algorithm to a number $y \in \mathbb{R}$ (the run-time) and it only denotes *worst-case* run-times. Please note we are not interested in the runtime itself, as in "given input n how many minutes does it take to find a solution y?", and therefore we don't have any physical units attached, but rather in the growth of the runtime if we increase the input length. In practice, *polynomial*-time algorithms are considered efficient as opposed to *exponential*-time algorithms, which are not.

Before we start we have to ask ourselves "What is parametrized complexity theory?". If we look at a **NP**-hard problem we assume there is no deterministic algorithm that computes a solution in polynomial time. But of course we are interested in a solution, so one can use methods like approximation, randomisation or heuristic functions. The problem here is that these methods are often not *exact* or not *provable* (with respect to time complexity).

1.2 Parametrized complexity theory

1.2.1 Preliminary

Now let us take a look at a specific subclass of problems proposed by [7]: These are defined as an input of an object x and a positive integer k and we want to

V is the number of vertices and C the size of the Vertex Cover.
Left $O(1.19^n)$ and right $O(kn + 1.29^k)$.

C\V	**10**	**50**	**100**
5	6	5989	35867090
10	6	5989	35867090
15	6	5989	35867090

C\V	**10**	**50**	**100**
5	54	254	504
10	553	2553	5053
15	8341	38341	75841

Figure 1.1: The difference between the two Vertex Cover Algorithms.

know if x has some property that depends on k. An example for this would be Vertex Cover. The input is a graph $G = (V, E)$ and a positive integer k and the question is, does G have a vertex cover of size at most k. We know that this problem can be solved in time $\mathcal{O}(1.19^n)$ [8] which is exponential in the number of vertices n. But another algorithm that was found by [13] has a running time of $\mathcal{O}(kn + 1.29^k)$. As we see this is polynomial in n but exponential in k, but the assumption of $k \ll n$ is natural in many cases and so the second algorithm is faster for $k \leq 0.79n$. See Figure 1.1 for comparison.

As we have seen it could be of interest to look for algorithms for **NP**-hard problems that are exponential *only* in k but polynomial in the size $n = |x|$ of the input object x. These leads us to the definition of a new class of problems called **FPT** containing all *fixed-parameter tractable* problems [5]:

- A *parametrized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$ where Σ is a finite alphabet.

- A parametrized problem **L** is *fixed-parameter tractable* if the question $(x, k) \in L$? can be decided in $\mathcal{O}(f(k)g(|x|))$ where f is some arbitrary function and g is a polynomial.

If g is linear the class is called **FPL** and it is easy to see that $FPL \subseteq FPT$. Unlike the classes **P** and **NP** that only look at one single input length, **FPT** is *two-dimensional* as it has two parameters which are handled separately.

1.2.2 Reduction

Definition

The core technique of this work will be the *reduction*. And because we are dealing with parametrized problems, we will use the parameter-preserving reduction, which is defined as follows. Suppose L_1 and L_2 are parametrized problems. Then L_1 can be polynomial-time parameter-preserving reduced to L_2 ($L_1 \leq_{pp} L_2$) iff there exists a function f so that all of the following holds:

1. $(x, k) \in L_1$ iff $f((x, k)) = (y, l) \in L_2$
2. f has a runtime in $\mathcal{O}(p(|x| + k))$ where p is a polynomial
3. l is polynomial in k

This technique can be used to show whether problems, belong to the class **FPT**.

Lemma 1.2.1 ([5]) *If $L_2 \in$ **FPT** and $L_1 \leq_{pp} L_2$ then $L_1 \in$ **FPT**.*

Corollary 1.2.2 *If $L_1 \notin$ **FPT** and $L_1 \leq_{pp} L_2$ then $L_2 \notin$ **FPT***

Definition We write $L_1 \equiv_{pp} L_2$ iff $L_1 \leq_{pp} L_2$ and $L_2 \leq_{pp} L_1$

Example

To get a first grasp of this reduction technique we will give a first example. Consider these two Problems:

> INDEPENDENT SET
>
> *Input*: A Graph $G = (V, E)$ and a positive integer k.
>
> *Question*: Is there a Subset $I \subseteq V$ with $|I| \geq k$ so that for every v_1 and $v_2 \in I$, $(v_1, v_2) \notin E$?
>
> *Parameter*: k

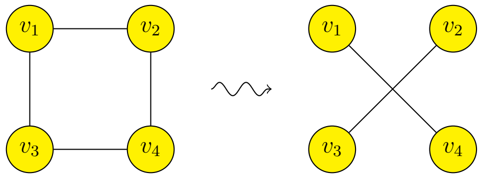

Figure 1.2: Illustration of the reduction from INDEPENDENT SET to CLIQUE

> CLIQUE
>
> *Input*: A Graph $G = (V, E)$ and a positive integer k.
>
> *Question*: Is there a Subset $C \subseteq V$ with $|C| \geq k$ so that for every v_1 and $v_2 \in C$, $(v_1, v_2) \in E$ unless $v_1 = v_2$?
>
> *Parameter*: k

One might see very easily that these two problems are basically equivalent.

Proposition 1.2.3 INDEPENDENT SET $\equiv_{pp}$ CLIQUE.

Proof INDEPENDENT SET $\leq_{pp}$ CLIQUE
Let (G, k) be the input of INDEPENDENT SET and (G', k) a transformation of that input, so that $G' = (V, E')$ where $E' = \{(u, v) \mid u, v \in V \text{ and } u \neq v \text{ and } (u, v) \notin E\}$. As illustrated in Figure 1.2
This has a running time polynomial in $|E|$ and the parameter k has not been changed.

$(x, k) \in$ INDEPENDENT SET
$\Leftrightarrow$ There exists a Subset $I \subseteq V$ with $|I| \geq k$ so that for every $v_1, v_2 \in I$, $(v_1, v_2) \notin E$
$\Leftrightarrow$ for every v_1 and $v_2 \in I$, $(v_1, v_2) \in E'$
$\Leftrightarrow$ $(G', k) \in$ CLIQUE

CLIQUE $\leq_{pp}$ INDEPENDENT SET
Analogous. $\square$

1.3 What is the point?

One might ask, what the point is of this new theory concept and rightfully so, as it looks counter-intuitive to learn a complete new theoretic model. But [3] pointed out some of the 'major highlights' of parametrized complexity theory. The main goal of parametrized complexity is to address complexity issues where we know that certain parameters will most likely be bounded. If we look at relational databases one deals, most of the time, with huge databases and queries that are asked by actual people and therefore small and of low logical complexity. If we now have two algorithms A and B and we know that A has the better run-time in the combined input size of *size of database* + *size of query* that does not have to mean that we should select A, if B has a better run-time for a small query size and thus we can choose to interpret it as a constant. In these situations parametrized complexity will help us choose the right algorithm for our ('real life') problem. Further consider two algorithms that have both the input (n, k). The first one has a run-time in $\mathcal{O}(n^{10^9})$ and the second one in $\mathcal{O}(2^k + n)$. In classical complexity theory the first one would be polynomial which is considered efficient, and the second one not. But it is easy to see that the first one is not very practical and that the second one is (for small k) way better. So parametrized complexity can help us in specific situations where we can have reasonable assumptions about the input data.

We can conclude that parametrized complexity is very suited for algorithm design for applied computer science like databases, genetics or historical linguistics, where the nature of the data is well understood.

Chapter 2

Reductions

> "Although problems and catastrophes may be inevitable, solutions are not." (Isaac Asimov)

This chapter is the core of this work. All the reductions are categorized by the change to the parameter(s), which is either no change at all, linear change or polynomial change. A separate section covers reduction which happen to be possible in both directions, making the core of the problem equivalent.

2.1 Reductions, polynomial but not parameter preserving

At first we take a look at polynomial reductions[1] of parametrized problems that do not work with the parameter preserving reduction. This is to show, that certain problems seem to be closely related at first, but if we add more constrains we see that the core of the problem has a different hardness with respect to the parameter k.

[1]These are the same as parameter preserving reduction, but allow a blow-up of the parameter dependent of the input

2.1.1 Independent Set $\leq_p$ Vertex Cover

INDEPENDENT SET

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $I \subseteq V$ with $|I| \geq k$ so that for every v_1 and $v_2 \in I$, $(v_1, v_2) \notin E$?

Parameter: k

VERTEX COVER

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$ so that for every $(v, u) \in I$, $v \in C$ or $u \in C$?

Parameter: k_1, k_2

The non parameter preserving reduction from INDEPENDENT SET to VERTEX COVER is rather easy. If C forms a vertex cover in G then $I = V - C$ has to be an independent set, because if there where an edge with both vertices in I, C would not be a vertex cover. To obtain a reduction we simply transform (G, k) to (G, k') where $k' = |V| - k$. However this is not a parameter preserving reduction because k' is not polynomial in k but only in $|V|$. So even though INDEPENDENT SET and VERTEX COVER seem to be very similar they in fact, from the parametrized viewpoint, are not.

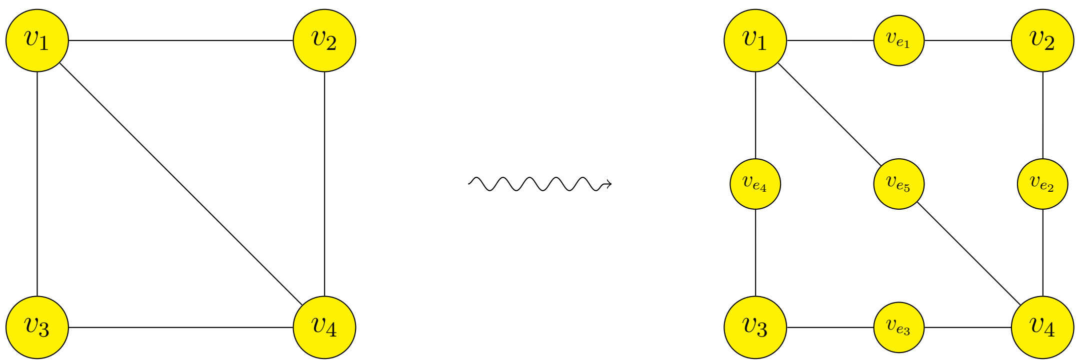

Figure 2.1: Illustration of CLIQUE to CONSTRAINT BIPARTITE VERTEX COVER

2.1.2 Clique $\leq_p$ Constraint Bipartite Vertex Cover

CLIQUE

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$ with $|C| \geq k$ so that for every v_1 and $v_2 \in C$, $(v_1, v_2) \in E$ unless $v_1 = v_2$?

Parameter: k

CONSTRAINT BIPARTITE VERTEX COVER

Input: A bipartite graph $G = (V_1 \cup V_2, E)$ and positive integers k_1, k_2.

Question: Is there a subset $C_b \subseteq V_1 \cup V_2$, with $|C \cap V_1| \leq k_1$ and $|C \cap V_2| \leq k_2$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$?

Parameter: k

The idea, as developed by [6] is to make the graph bipartite by adding an intercepting node to every edge so that the vertices can be grouped into two sets V_{old} and V_{new} (Figure 2.1). Then we proceed to set $k_1 = k$ and $k_2 = |V_{new}| - \frac{k(k-1)}{2}$. But since k_2 scales in $|V_{new}|$ which in turn is the same as $|E|$ of the original graph it is not polynomial in the parameter k.

2.2 Parameter preserving reduction in *both* directions

2.2.1 Short Non-deterministic Turing Machine Acceptance $\equiv_{pp}$ Independent Set

SHORT NON-DETERMINISTIC TURING MACHINE ACCEPTANCE

Input: A non-deterministic single tape Turing Machine M and a positive integer k.

Question: Does M accept the empty word within at most k-steps?

Parameter: k

INDEPENDENT SET

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $I \subseteq V$ with $|I| \geq k$ so that for every v_1 and $v_2 \in I$, $(v_1, v_2) \notin E$?

Parameter: k

This reduction creates a bridge to Turing machines, similar to classical complexity theory.

Proposition 2.2.1 ([3]) SHORT NTM ACCEPTANCE $\equiv_{pp}$ INDEPENDENT SET

Proof Omitted.

2.2.2 Monotone Weighted SAT $\equiv_{pp}$ Hitting Set

MONOTONE WEIGHTED SAT

Input:A Boolean expression φ in conjunctive normal form without negated literals, and a positive integer k.

Question: Is there a truth assignment of weight[1] k that satisfies φ?

Parameter: k

HITTING SET

Input: A set family $\mathfrak{F}$ over a universe U and a positive integer k.

Question: Is there a subset $H \subseteq U$ of size at most k so that for every $S \in \mathfrak{F}, H \cap S \neq \emptyset$?

Parameter: k

These problems are equivalent because both times we want to have a set of elements (the variables we set to *true*) that hits (satisfies) every set (clause). So $(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_3) \wedge (x_2 \vee x_5)$ becomes $\{\{x_1, x_2, x_3\}, \{x_4, x_3\}, \{x_2, x_5\}\}$ and vice versa, with $H = \{x_3, x_5\}$ being a satisfying (hitting) set.

Proposition 2.2.2 MONOTONE WEIGHTED SAT $\leq_{pp}$ HITTING SET

Proof Let (φ, k) be the input of MONOTONE WEIGHTED SAT where φ has m clauses and where C_i is the set of variables that are contained in clause i. Let $\mathfrak{F} = \{C_1, \ldots, C_m\}$ and $U = C_1 \cup \cdots \cup C_m$. This is polynomial in $|\varphi|$ and the parameter k has not been changed. It is easy to see that every set H of positive variables that satisfies φ is a hitting set of $\mathfrak{F}$. Therefore
$(\varphi, k) \in$ MONOTONE WEIGHTED SAT $\Leftrightarrow (\mathfrak{F}, U, k) \in$ HITTING SET $\square$

Proposition 2.2.3 HITTING SET $\leq_{pp}$ MONOTONE WEIGHTED SAT

Proof Analogous.

[1]The amount of positive variables

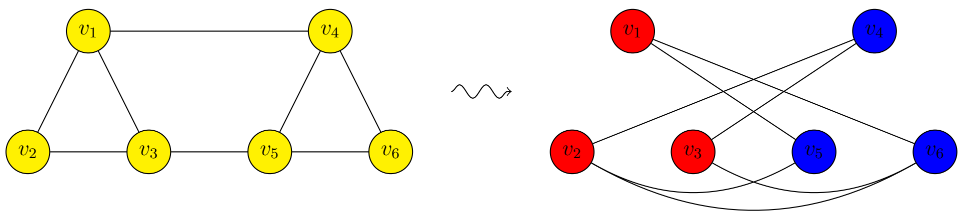

Figure 2.2: Illustration of PARTITION INTO CLIQUES to COLOURING

2.2.3 Partition into Cliques $\equiv_{pp}$ Colouring

PARTITION INTO CLIQUES

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Can V be partitioned into k many cliques, that is exists $V_1, \ldots, V_k$, with $V = \bigcup_{i=1}^{k} V_i$ where $V_i \cap V_j = \emptyset$ and V_i is clique for all $i \neq j$?

Parameter: k

COLOURING

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Can the vertices of G be coloured using k different colours, so that no two vertices that are incident to each other have the same colour?

Parameter: k

The idea is to invert the edges of the graph to make independent sets out of the cliques, so we can assign each clique its own colour (See Figure 2.2).

Proposition 2.2.4 PARTITION INTO CLIQUES $\leq_{pp}$ COLOURING

Proof Let (G, k) be the input of PARTITION INTO CLIQUES and (G', k) a transformation of that input, so that $G' = (V, E')$ where $E' = \{(u, v) \mid u, v \in$

V and $u \neq v$ and $(u, v) \notin E\}$, as illustrated in Figure 2.2.
This has a running time polynomial in $|E|$ and the parameter k has not been changed.

$(G, k) \in$ PARTITION INTO CLIQUES
$\Rightarrow$ V can be split into k many cliques
$\Rightarrow$ G' has k many independent sets
$\Rightarrow$ every node of an independent set can have the same colour
$\Rightarrow$ G' can be coloured using k many colours
$\Rightarrow$ $(G', k) \in$ COLOURING

$(G', k) \in$ COLOURING
$\Rightarrow$ G' can be coloured using k many colours
$\Rightarrow$ every set of vertices of the same colour forms an independent set
$\Rightarrow$ there are k many independent sets in G'
$\Rightarrow$ Every independent set in G' is a clique in G
$\Rightarrow$ $(G, k) \in$ PARTITION INTO CLIQUES $\square$

Proposition 2.2.5 COLOURING $\leq_{pp}$ PARTITION INTO CLIQUES

Proof Analogous.

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \rightsquigarrow$$

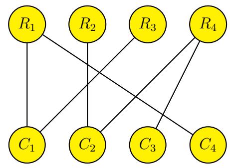

Figure 2.3: Illustration of SPARE ALLOCATION to CONSTRAINT BIPARTITE VERTEX COVER

2.2.4 Spare Allocation $\equiv_{pp}$ Constraint Bipartite Vertex Cover

SPARE ALLOCATION

Input: A binary matrix $A^{n\times m}$ representing an erroneous chip, with $a_{i,j} = 1$ iff the chip is faulty on position (i,j), and positive integers k_1 and k_2.

Question: Is there a *reconfiguration strategy*, i.e. a description of which rows and columns of A have to be replaced by spares, that repairs all faults and uses at most k_1 spare rows and at most k_2 spare columns?

Parameter: k_1, k_2

CONSTRAINT BIPARTITE VERTEX COVER

Input: A bipartite graph $G = (V_1 \cup V_2, E)$ and positive integers k_1, k_2.

Question: Is there a subset $C_b \subseteq V_1 \cup V_2$, with $|C \cap V_1| \leq k_1$ and $|C \cap V_2| \leq k_2$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$?

Parameter: k_1, k_2

These two problems are equivalent if we model it as followed. The rows of A form one set of vertices R and the columns the other set C, and then we create an edge between two vertices if the corresponding index of A is 1 (See Figure 2.3).

Proposition 2.2.6 SPARE ALLOCATION $\leq_{pp}$ CONSTRAINT BIPARTITE VERTEX COVER

Proof We transform (A, k_1, k_2) to (G, k_1, k_2), where $G = (V, E)$ is a bipartite graph with $R \cup C = V$, where $R = \{r_i \mid i \in \{1, \ldots, n\}\}$, $C = \{c_i \mid i \in \{1, \ldots, m\}\}$ and $E = \{(r_i, c_j) \mid a_{i,j} = 1\}$. The parameters k_1 and k_2 have not been changed and the transformation is constant if we interpret A as the adjacent matrix of G.

$(A, k_1, k_2) \in$ Spare Allocation
$\Leftrightarrow$ There is a description D of which rows and columns of A have to be replaced by spares, that repairs all faults and uses at most k_1 spare rows and at most k_2 spare column
$\Leftrightarrow$ D is a vertex cover in G that uses at most k_1 many vertices of R and at most k_2 many vertices of C
$\Leftrightarrow$ $G = (V, E) \in$ Constraint Bipartite Vertex Cover $\square$

Proposition 2.2.7 Constraint Bipartite Vertex Cover $\leq_{pp}$ Spare Allocation

Proof Analogous.

$R = \{v_1, \ldots, v_4\}$ and $B = \{v_5, \ldots, v_8\}$

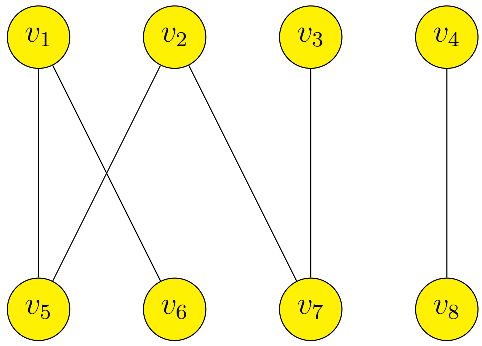

Figure 2.4: Illustration of Red-Blue Dominating Set to Set Cover.

2.2.5 Red-Blue Dominating Set $\equiv_{pp}$ Set Cover

Red-Blue Dominating Set

Input: A bipartite graph $G = (R \cup B, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq B$, with $|D| \leq k$, so that for every $v \in R$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

Set Cover

Input: A set family $\mathfrak{F}$ over a universe U and a positive integer k.

Question: Is there a sub-family $\mathfrak{F}' \subseteq \mathfrak{F}$ of size at most k, so that $\bigcup\limits_{S_i \in \mathfrak{F}'} S_i = U$?

Parameter: k

This reduction was proposed in [9] and the idea is to create an element for every node of R in the universe U and then creating $\mathfrak{F}$ so that it contains Sets for every node of B containing al the adjacent nodes of R. So the graph in Figure 2.4 becomes: $U = \{e_1, \ldots, e_4\}$, $\mathfrak{F} = \{\{e_1, e_2\}, \{e_2\}, \{e_2, e_3\}, \{e_4\}\}$.

Proposition 2.2.8 Red-Blue Dominating Set $\leq_{pp}$ Set Cover

Proof Let (G, k) be the input of Red-Blue Dominating Set with $G = (R \cup B, E)$. We transform this into $(U, \mathfrak{F}, k)$ with $U = \{e_i \mid v_i \in R\}$ and $\mathfrak{F} = \{\{e_i \mid (v_i, u_j) \in E, v_i \in R\} \mid u_j \in B\}$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(U, \mathfrak{F}, k) \in$ Set Cover
$\Leftrightarrow$ There is a sub-family $\mathfrak{F}'$ that covers U
$\Leftrightarrow$ The set of nodes corresponding to every set of $\mathfrak{F}'$ dominate R
$\Leftrightarrow (G, k) \in$ Red-Blue Dominating Set $\square$

Proposition 2.2.9 Set Cover $\leq_{pp}$ Red-Blue Dominating Set

Proof Analogous.

2.2.6 Red-Blue Dominating Set $\equiv_{pp}$ Hitting Set

RED-BLUE DOMINATING SET

Input: A bipartite graph $G = (R \cup B, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq B$, with $|D| \leq k$, so that for every $v \in R$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

HITTING SET

Input: A set family $\mathfrak{F}$ over a universe U and a positive integer k.

Question: Is there a subset $H \subseteq U$ of size at most k so that for every $S \in \mathfrak{F}, H \cap S \neq \emptyset$?

Parameter: k

The idea is that the nodes of B will become our universe and for every vertex v_i of R we create a set S_i consisting of all adjacent vertices to v_i. Thus a hitting set is a subset of B that dominates R.

Proposition 2.2.10 RED-BLUE DOMINATING SET $\leq_{pp}$ HITTING SET

Proof Let (G, k) with $G = (R \cup B, E)$ be the input of RED-BLUE DOMINATING SET. We transform this into $(\mathfrak{F}, U, k)$ with $U = B$ and $\mathfrak{F} = \{S_1, \ldots, S_{|R|}\}$ where $S_i = \{v_j \mid (v_i, v_j) \in E\}$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(G, k) \in$ RED-BLUE DOMINATING SET
$\Leftrightarrow$ There is a set $D \subseteq B$ of size at most k that dominates R
$\Leftrightarrow$ every vertex of R is adjacent to a vertex of D
$\Leftrightarrow$ $D \subseteq U$ hits every set of $\mathfrak{F}$
$\Leftrightarrow$ $(\mathfrak{F}, U, k) \in$ HITTING SET $\square$

Proposition 2.2.11 HITTING SET $\leq_{pp}$ RED-BLUE DOMINATING SET

Proof Analogous.

2.2.7 Vertex Cover $\equiv_{pp}$ 2-Hitting Set

VERTEX COVER

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$?

Parameter: k

2-HITTING SET

Input: A set family $\mathfrak{F}$ over a universe U where for every $S \in \mathfrak{F}, |S| = 2$ and a positive integer k.

Question: Is there a subset $H \subseteq U$ of size at most k so that for every $S \in \mathfrak{F}, H \cap S \neq \emptyset$?

Parameter: k

It is easy to see that these two problems are equivalent if we set $V = U$ and $E = \mathfrak{F}$.

Proposition 2.2.12 VERTEX COVER $\equiv_{pp}$ 2-HITTING SET

Proof Omitted.

2.3 Parameter preserving reduction *without* parameter change

2.3.1 Monotone Weighted SAT $\leq_{pp}$ Weighted SAT

MONOTONE WEIGHTED SAT

Input: A Boolean expression φ in conjunctive normal form without negated literals, and a positive integer k.

Question: Is there a truth assignment of weight[1] k that satisfies φ?

Parameter: k

WEIGHTED SAT

Input: A Boolean expression φ in conjunctive normal and a positive integer k.

Question: Is there a truth assignment of weight k that satisfies φ?

Parameter: k

This reduction is trivial because every *monotone* Boolean expression in conjunctive normal form is already a Boolean expression in conjunctive normal form.

Proposition 2.3.1 MONOTONE WEIGHTED SAT $\leq_{pp}$ WEIGHTED SAT

Proof Omitted.

[1] The amount of positive variables

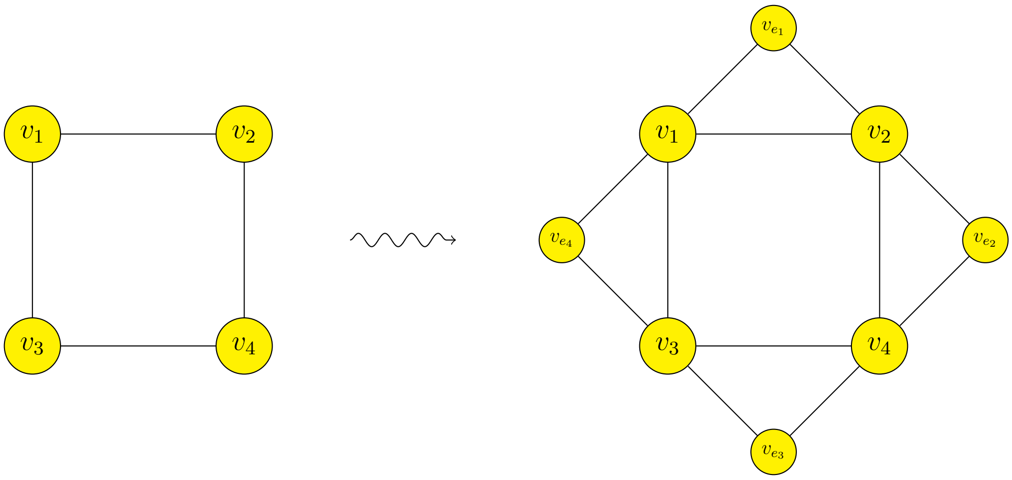

Figure 2.5: Illustration of Vertex Cover to Dominating Set

2.3.2 Vertex Cover $\leq_{pp}$ Dominating Set

Vertex Cover

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$?

Parameter: k

Dominating Set

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq V$, with $|D| \leq k$, so that for every $v \in V$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

The idea is to create *edge vertices* for every edge that are connected to the vertices of that edge (Figure 2.5). That way we enforce that a dominating set has to be a vertex cover. In turn a vertex cover is already a dominating set.

Proposition 2.3.2 Vertex Cover $\leq_{pp}$ Dominating Set

Proof Let (G, k) be the input for VERTEX COVER. We transform that input to (G', k) where $G' = (V', E')$ with $V' = V - \{v \mid v \in V$, and v is isolated$\} \cup V_{new}$ where $V_{new} = \{v_{e_i} \mid e_i \in E\}$ and $E' = E \cup E_{new}$ where $E_{new} = \{(v_{e_i}, v), (v_{e_i}, u) \mid e_i = (v, u) \in E\}$. We remove the isolated vertices, so that the dominating set in G' coincides with the vertex cover. The parameter k has not been changed and the transformation can be done in time polynomial in $|E|$.

$(G, k) \in$ VERTEX COVER
$\Rightarrow G$ has a vertex cover C of size at most k
$\Rightarrow G'$ has a dominating set D with $D = C$ of size at most k, because for every $e_i = (u, v) \in E$ either v or u are in C and v_{e_i} is adjacent to v and to u and v and u are adjacent to each other.
$\Rightarrow (G', k) \in$ DOMINATING SET

$(G', k) \in$ DOMINATING SET
$\Rightarrow G'$ has a dominating Set D with $D \subseteq V$[1]
$\Rightarrow$ for every v_{e_i} one of its two neighbours is in D and this node covers the edge e_i
$\Rightarrow G$ has a vertex cover C with $C = D$ of size at most k
$\Rightarrow (G, k) \in$ VERTEX COVER $\square$

[1] If v_{e_i} is picked we can replace it by one of its neighbours without changing the size

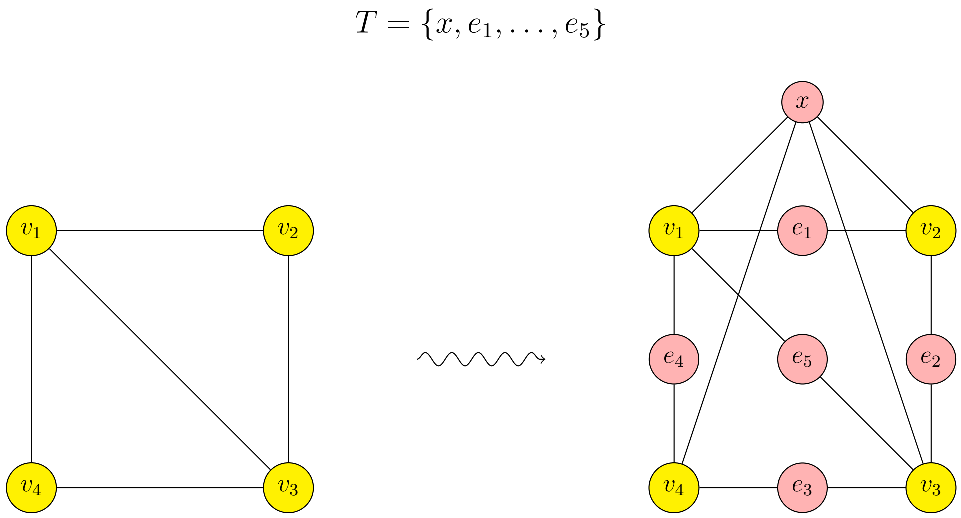

Figure 2.6: Illustration of Vertex Cover to Steiner Tree

2.3.3 Vertex Cover $\leq_{pp}$ Steiner Tree

Vertex Cover

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$?

Parameter: k

Steiner Tree

Input: A graph $G = (V, E)$ a set $T \subseteq V$ and a positive integer k.

Question: Is there a subset $S \subseteq V - T$, with $|S| \leq k$, so that the subgraph induced by $T \cup S$ is connected?

Parameter: k

The idea for this reduction is to create new vertices, one for each edge and one extra vertex that is connected to every vertex of V. This last vertex will enforce the connectivity of the Steiner tree. These new vertices will be the terminals of the new graph. The Steiner points are then the vertex cover of the original graph. This reduction can be considered folklore and happens to be parameter-preserving.

Proposition 2.3.3 Vertex Cover $\leq_{pp}$ Steiner Tree

Proof Let (G, k) with $G = (V, E)$ be the input of Vertex Cover. We transform this into $G' = (V', E')$ where $V' = V \cup V_e \cup \{x\}$ with $V_e = \{v_e \mid e \in E\}$ and $E' = \{(v, v_e), (v_e, u) \mid e = (v, u) \in E\} \cup \{(x, v) \mid v \in V\}$. Finally we set $T = V_e \cup \{x\}$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(G, k) \in$ Vertex Cover
$\Rightarrow$ There is a vertex cover $C \subseteq V$ of size at most k
$\Rightarrow$ every vertex of V_e is adjacent to a node of C
$\Rightarrow$ every vertex of $V_e \cup \{x\}$ is connected to a vertex of C
$\Rightarrow$ For $S = C$ $T \cup S$ induces a connected subgraph
$\Rightarrow$ $(G, T, k) \in$ Steiner Tree

$(G, T, k) \in$ Steiner Tree
$\Rightarrow$ There exists a subset $S \subseteq V' - T$ of size at most k so that $T \cup S$ induces a connected subgraph
$\Rightarrow$ Every vertex of V_e is connected to a node of S (since no two vertices of V_e are connected)
$\Rightarrow$ $C = S$ is a vertex cover in G
$\Rightarrow$ $(G, k) \in$ Vertex Cover $\square$

2.3.4 Dominating Set $\leq_{pp}$ Hitting Set

Dominating Set

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq V$, with $|D| \leq k$, so that for every $v \in V$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

Hitting Set

Input: A set family $\mathfrak{F}$ over a universe U and a positive integer k.

Question: Is there a subset $H \subseteq U$ of size at most k so that for every $S \in \mathfrak{F}, H \cap S \neq \emptyset$?

Parameter: k

This reduction is straightforward, because one can easily see the similarity.

Proposition 2.3.4 Dominating Set $\leq_{pp}$ Hitting Set

Proof Let (G, k) with $G = (V, E)$ be the input for Dominating Set, we transform this into $(\mathfrak{F}, U, k)$ with $U = V$ and $\mathfrak{F} = S_1, \ldots, S_{|V|}$ where $S_i = \{v_i\} \cup \{v_j \mid (v_i, v_j) \in E\}$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(G, k) \in$ Dominating Set
$\Rightarrow$ There is a dominating subset $D \subseteq V$ of size at most k
$\Rightarrow$ for every vertex v either v is in D or one of its neighbours
$\Rightarrow$ $D \subseteq U$ hits every set of $\mathfrak{F}$
$\Rightarrow$ $(\mathfrak{F}, U, k) \in$ Hitting Set

$(\mathfrak{F}, U, k) \in$ Hitting Set
$\Rightarrow$ There is a subset $H \subseteq U$ that hits every set of $\mathfrak{F}$
$\Rightarrow$ Since every set S_i contains the vertex v_i and his closed neighbourhood, either $v \in H$ or v has a neighbour $u \in H$
$\Rightarrow$ H dominates V
$\Rightarrow$ $(G, k) \in$ Dominating Set □

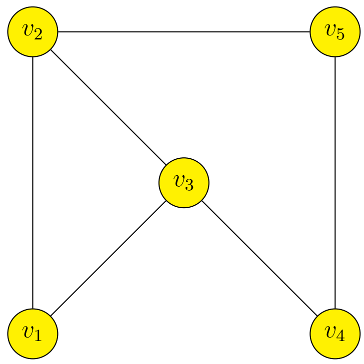

Figure 2.7: The graph G, that will be transformed to φ

2.3.5 Dominating Set $\leq_{pp}$ Monotone Weighted SAT

DOMINATING SET

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq V$, with $|D| \leq k$, so that for every $v \in V$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

MONOTONE WEIGHTED SAT

Input: A Boolean expression φ in conjunctive normal form without negated literals, and a positive integer k.

Question: Is there a truth assignment of weight[1] k that satisfies φ?

Parameter: k

To reduce DOMINATING SET to MONOTONE WEIGHTED SAT, we have to transform a graph to a formula. Each vertex becomes a variable and the set of variables set to *true*, should form a dominating set. To achieve this we form a clause for each vertex that contains itself and its closed neighbourhood. Thus the graph in Figure 2.7 becomes:
$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_1 \vee x_3 \vee x_5) \wedge (x_3 \vee x_1 \vee x_2 \vee x_4) \wedge (x_4 \vee x_3 \vee x_5) \wedge (x_5 \vee x_4 \vee x_2)$.

[1] The amount of positive variables

Proposition 2.3.5 DOMINATING SET $\leq_{pp}$ MONOTONE WEIGHTED SAT

Proof Let (G, k) with $G = (V, E)$ be the input for DOMINATING SET, then let φ be a Boolean expression in conjunctive normal form with variables $x_1, \ldots, x_i$ and clauses $C_1, \ldots, C_i$ for $i = |V|$ and

$$C_m = \{\overline{x_m} \rightarrow (x_{l_0} \vee \cdots \vee x_{l_r}) \mid 1 \leq l_0 < \cdots < l_r \leq \\ |V| \text{ and } (v_m, v_{l_0}), \ldots, (v_m, v_{l_r}) \in E\} \text{ for } m \in \{1, \ldots, |V|\}$$

$(\overline{x_m} \rightarrow (x_{l_0} \vee \cdots \vee x_{l_r}))$ expresses, if x_m is not in the dominating set (set to true), then one of its neighbours must be. This can be converted to $(x_m \vee x_{l_0} \vee \cdots \vee x_{l_r})$ using standard Boolean transformation. This transformation is polynomial in V and E and the parameter k has not been changed

$(G, k) \in$ DOMINATING SET
$\Rightarrow G$ has a dominating set D of size k
$\Rightarrow$ if we set exactly these x_i to *true* with $v_i \in D$ each clause is satisfied, because either the vertex itself or at least one of its neighbours are in D
$\Rightarrow (\varphi, k) \in$ MONOTONE WEIGHTED SAT

$(\varphi, k) \in$ MONOTONE WEIGHTED SAT
$\Rightarrow$ there exist $A = \{x_{i_1}, \ldots, x_{i_k}\}$ that are set to *true*
$\Rightarrow D = \{v_i \mid x_i \in A\}$ is a dominating set, because through $C_1, \ldots, C_{|V|}$ every vertex itself or at least on of its neighbours is in D
$\Rightarrow (G, k) \in$ DOMINATING SET □

2.3.6 Dominating Set $\leq_{pp}$ Center

DOMINATING SET

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq V$, with $|D| \leq k$, so that for every $v \in V$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

CENTER

Input: A graph $G = (V, E)$ a radius $r \in \mathbb{Q}^+$ a cost function $c\colon E \to \mathbb{Q}^+$ and a positive integer k.

Question: Is there a $Z \subseteq V$ with $|Z| \leq k$ and $\mathit{rad}(Z) \leq r$ where $\mathit{rad}(Z) = \max_{v \in V} \mathit{dist}(v, Z)$ with $\mathit{dist}(v, Z) = \min_{z \in Z} \mathit{dist}(v, z)$[1]?

Parameter: k

In the CENTER problem we are looking for a set of vertices Z so that every vertex of V has a distance of at most r from the nearest vertex of Z. The DOMINATING SET problem can be interpreted as a special case of CENTER with $r = 1$ and $c(e) = 1$ for all e.

Proposition 2.3.6 DOMINATING SET $\leq_{pp}$ CENTER

Proof Let (G, k) with $G = (V, E)$ be the input of DOMINATING SET. Let $c(e) = 1$ for all $e \in E$ and $r = 1$. The parameter k has not been changed.

$(G, k) \in$ DOMINATING SET
$\Leftrightarrow$ there is a set $D \subseteq V$ of size at most k that dominates V
$\Leftrightarrow$ $Z = D$ is a center with radius at most 1
$\Leftrightarrow$ $(G, r, c, k) \in$ CENTER $\square$

[1] $\mathit{dist}(v, u)$ is the length of the shortest path between v and u regarding the cost function c

$T = \{v_1, \ldots, v_4\}$ and $N = \{v_5, \ldots, v_8\}$

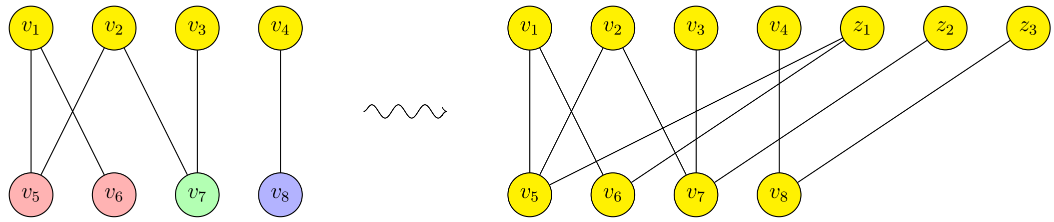

Figure 2.8: Illustration of COLOURED RED-BLUE DOMINATING SET to RED-BLUE DOMINATING SET.

2.3.7 Coloured Red-Blue Dominating set $\leq_{pp}$ Red-Blue Dominating Set

COLOURED RED-BLUE DOMINATING SET

Input: A bipartite graph $G = (T \cup N, E, col)$ with $col\colon N \to \{1, \ldots, k\}$ and a positive integer k.

Question: Is there a subset $D \subseteq N$, with $|D| \leq k$ and D contains exactly one vertex of each colour, so that for every $v \in T$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

RED-BLUE DOMINATING SET

Input: A bipartite graph $G = (T \cup N, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq N$, with $|D| \leq k$, so that for every $v \in T$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

This reduction was proposed in [9] and the idea is to replace every colour by a new node that is added to the set T. The new vertices are then connected to the vertices that were assigned the colour which is represented by this new node (see Figure 2.8).

Proposition 2.3.7 COL RB DOMINATING SET $\leq_{pp}$ RB DOMINATING SET

Proof Let (G, col, k) with $G = (T \cup N, E)$ and col: $N \to \{1, \dots, k\}$ be the input of COLOURED RED-BLUE DOMINATING SET. We transform this into $G' = (T' \cup N, E')$, where $T' = T \cup \{z_1, \dots, z_k\}$ and $E' = \{(z_a, v) \mid a \in \{1, \dots, k\}$ and $v \in N$ and $col(v) = a\}$. This is polynomial in $|G|$ and k, and the parameter k has not been changed.

$(G, col, k) \in$ COLOURED RED-BLUE DOMINATING SET
$\Rightarrow$ G has a set $D \subseteq N$ with $|D| \leq k$ that dominates T and consists of exactly one node of each colour
$\Rightarrow$ D dominates all vertices of T' since every z_i has a neighbour in D, which is that of the corresponding colour
$\Rightarrow$ $(G', k) \in$ RED-BLUE DOMINATING SET

$(G', k) \in$ RED-BLUE DOMINATING SET
$\Rightarrow$ G' has a set $D \subseteq N$ with $|D| \leq k$ that dominates T'
$\Rightarrow$ No z_a has more then one neighbour in D, because then there would be a z_x without a neighbour in D since no vertex in N has more then one edge to one of the new vertices
$\Rightarrow$ D dominates T and has exactly one node of each colour
$\Rightarrow$ $(G, col, k) \in$ RED-BLUE DOMINATING SET $\square$

$R = \{v_1, \ldots, v_4\}$ and $B = \{v_5, \ldots, v_8\}$

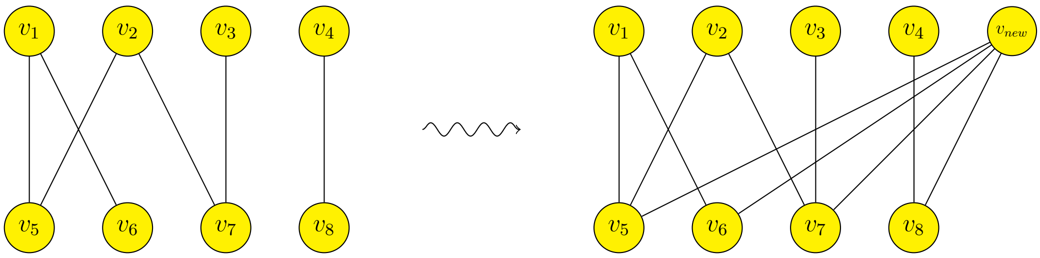

Figure 2.9: Illustration of Red-Blue Dominating Set to Steiner Tree.

2.3.8 Red-Blue Dominating set $\leq_{pp}$ Steiner Tree

Red-Blue Dominating Set

Input: A bipartite graph $G = (R \cup B, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq B$, with $|D| \leq k$, so that for every $v \in R$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

Steiner Tree

Input: A graph $G = (V, E)$ a set $T \subseteq V$ and a positive integer k.

Question: Is there a subset $S \subseteq V - T$, with $|S| \leq k$, so that the subgraph induced by $T \cup S$ is connected?

Parameter: k

This reduction was proposed in [9] and the idea is to create a new vertex that is connected to all vertices of B so that every solution of Red-Blue Dominating Set creates a connected subgraph (see Figure 2.9). Since we want to dominate R, these vertices will become the terminals of the new graph.

Proposition 2.3.8 Red-Blue Dominating Set $\leq_{pp}$ Steiner Tree

Proof Let (G, k) with $G = (R \cup B, E)$ be the input of Red-Blue Dominating Set. We transform this into $G' = (V', E')$ where $V' = R \cup B \cup \{v_{new}\}$ and $E' = E \cup \{(v_{new}, v_i) \mid v_i \in B\}$ and we set $T = R$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(G, k) \in$ Red-Blue Dominating Set
$\Rightarrow$ G has a set $D \subseteq B$ with $|D| \leq k$ that dominates R
$\Rightarrow$ every node in D is connected to v_{new}, $S = D \cup \{v_{new}\} \subseteq V' - T$ and every node of S is connected with at least one node of T
$\Rightarrow$ The subgraph induced by $S \cup T$ is connected
$\Rightarrow$ $(G', S, k) \in$ Steiner Tree

$(G', S, k) \in$ Steiner Tree
$\Rightarrow$ There is a set $S \subseteq V - T$ so that the subgraph induced by $T \cup S$ is connected
$\Rightarrow$ since the nodes of R are not connected, every node of R has to have at least one neighbour in S
$\Rightarrow$ $D = S - \{v_{new}\}$ dominates R
$\Rightarrow$ $(G, k) \in$ Red-Blue Dominating Set $\square$

2.3.9 Coloured Small Universe Hitting Set $\leq_{pp}$ Small Universe Hitting Set

COLOURED SMALL UNIVERSE HITTING SET

Input: A set family $\mathfrak{F}$ over a universe U with $|U| = d$, a colour function *col*: $U \to \{1, \ldots, k\}$ and a positive integer k.

Question: Is there a subset $H \subseteq U$ of size at most k such that for every set $S \in \mathfrak{F}, H \cap S \neq \emptyset$ and H contains at least one element of each colour?

Parameter: k, d

SMALL UNIVERSE HITTING SET

Input: A set family $\mathfrak{F}$ over a universe U with $|U| = d$ and a positive integer k.

Question: Is there a subset $H \subseteq U$ of size at most k such that for every set $S \in \mathfrak{F}, H \cap S \neq \emptyset$?

Parameter: k, d

This reduction was proposed in [9] and what we want to do is, create sets for each colour that contain the elements of this colour.

Proposition 2.3.9 COL SMALL UNIVERSE HS $\leq_{pp}$ SMALL UNIVERSE HS

Proof Let $(\mathfrak{F}, U, col, d, k)$ be the input of COLOURED SMALL UNIVERSE HITTING SET we then construct $(\mathfrak{F}', U, d, k)$. Let $U_i = \{e_j \mid col(e_j) = i, e_i \in U\}$ be the set of elements of colour i. Then $\mathfrak{F}' = \mathfrak{F} \bigcup_{i \in \{1,\ldots,k\}} U_i$. This is polynomial in k and d and the parameters k and d have not been changed.

$(\mathfrak{F}, U, col, d, k) \in$ COLOURED SMALL UNIVERSE HITTING SET
$\Leftrightarrow$ There is a subset $H \subseteq U$ that contains at least one element of each colour, that hits every set of $\mathfrak{F}$
$\Leftrightarrow$ H hits every set of $\mathfrak{F}'$
$\Leftrightarrow$ $(\mathfrak{F}', U, col, d, k) \in$ SMALL UNIVERSE HITTING SET $\square$

$$w = 1232354435\,13$$

$$F_1 = 12323544351,\ F_2 = 232,\ F_3 = 3513,\ F_4 = 44,\ F_5 = 54435$$

w does not have the Disjoint Factors property, because F_1 overlaps with all other factors and F_5 overlaps with F_3 and F_4.

Figure 2.10: An example of the Disjoint Factors property not satisfied

2.3.10 Disjoint Factors $\leq_{pp}$ Vertex Disjoint Cycles

DISJOINT FACTORS

Input: A word $w \in L_k^*$ where $L_k = \{1, \ldots, k\}$ and a positive integer k.

Question: Does w have the Disjoint Factors property?

Parameter: k

VERTEX DISJOINT CYCLES

Input: A graph $G = (V, A)$ and a positive inter k.

Question: Does G contain at least k vertex-disjoint cycles?

Parameter: k

The Disjoint Factor property is defined as followed: F_j is a factor of a word w iff w consists of a sub-string > 1 that starts and ends with the letter j. If we can find non overlapping (that is disjoint) factors $F_1, \ldots, F_k$ of w then w has the Disjoint Factor property (see Figure 2.10 for a counter example). This reduction was proposed in [10] and the idea is to create a node for every letter in w that is adjacent to every preceding and following letter and a node for every letter of the alphabet L_k that is adjacent to every letter of the word that is the same. In Figure 2.11 you can see how the graph is created and the colours of the vertices represent the circles which correspond to the disjoint factors.

Proposition 2.3.10 DISJOINT FACTORS $\leq_{pp}$ VERTEX DISJOINT CYCLES

$w = 12313414334242$

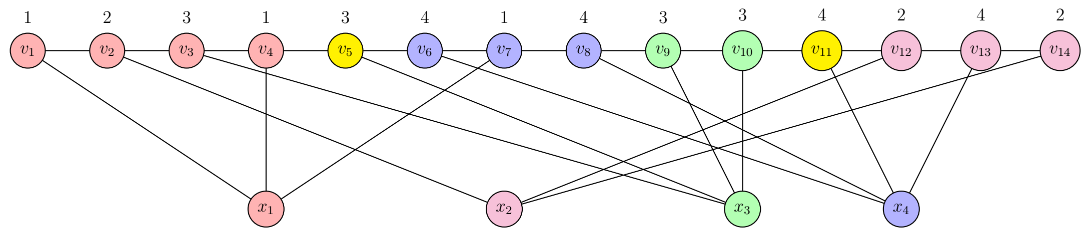

Figure 2.11: Illustration of Disjoint Factors to Vertex Disjoint Cycles.

Proof Let (w, k) be the input of Disjoint Factors with $w = w_1 \dots w_n$ over L_k^*. We transform this into (G, k) where $G = (V, E)$ with $V = \{w_i \mid w_i \in w\} \cup \{l_i \mid l_i \in L_k\}$ and $E = \{(w_i, w_{i+1}) \mid i \in \{1, \dots, n-1\}\} \cup \{(w_i, l_j) \mid w_i = j\}$. This is polynomial in $n + k$ and the parameter has not been changed.

$(w, k) \in$ Disjoint Factors
$\Rightarrow$ there exist disjoint factors $F_1, \dots, F_k$
$\Rightarrow$ For each letter $j \in L_k$ there is a cycle consisting of x_j and the vertices corresponding to the letters of F_j
$\Rightarrow$ there are k vertex disjoint cycles in G
$\Rightarrow (G, k) \in$ Vertex Disjoint Cycles

$(G, k) \in$ Vertex Disjoint Cycles
$\Rightarrow G$ has disjoint cycles $c_1, \dots, c_k$
$\Rightarrow$ Since the vertices v_i form a path, every cycle c_j must consist of a vertex x_j
$\Rightarrow$ The vertices adjacent to x_j are the same letter in w and since a cycle has a length of at least three the sub-path F_j without x_j in cycle c_j corresponds to a disjoint factor of w
$\Rightarrow w$ has the Disjunct Factors property
$\Rightarrow (w, k) \in$ Disjoint Factors $\square$

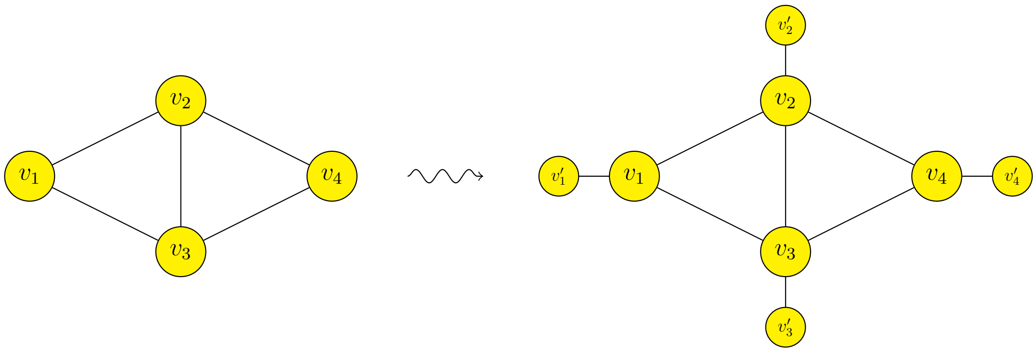

Figure 2.12: Illustration of Independent Set to Induced Matching

2.3.11 Independent Set $\leq_{pp}$ Induced Matching

Independent Set

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $I \subseteq V$ with $|I| \geq k$ so that for every v_1 and $v_2 \in I$, $(v_1, v_2) \notin E$?

Parameter: k

Induced Matching

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $M \subseteq E$, with $|M| \geq k$ so that no vertex is incident to more than one vertex regarding M and between the matched vertices there is no path of length more than one in G?

Parameter: k

The idea of this reduction is to create a new set of vertices that has a new vertex v' for every vertex $v \in V$ that is only adjacent to this vertex. The independent set of G is then an induced matching in G' (Figure 2.12).

Proposition 2.3.11 INDEPENDENT SET $\leq_{pp}$ INDUCED MATCHING

Proof Let (G, k) be the input of INDEPENDENT SET with $G = (V, E)$. We transform this into (G', k) with $G' = (V', E')$ where $V' = V \cup V_{new}$ with $V_{new} = \{v' \mid v \in V\}$ and $E' = E \cup E_{new}$ with $E_{new} = \{(v, v') \mid v \in V\}$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(G, k) \in$ INDEPENDENT SET
$\Rightarrow$ G has an independent set $I \subseteq V$ of size at least k
$\Rightarrow$ $M = \{(v, v') \mid I\}$ is an induced matching because the vertices of I are not connected in G and G'
$\Rightarrow$ $(G', k) \in$ INDUCED MATCHING

$(G', k) \in$ INDUCED MATCHING
$\Rightarrow$ G' has an induced matching M of size at least k
$\Rightarrow$ if $(v, u) \in M$ then v' and u' are not matched, otherwise it would not been induced, so we can replace (v, u) with (v, v') for all vertices in M
$\Rightarrow$ let X be the vertices matched by M, since $|M| \geq k$, $|X| \geq 2k$ and for $I = X - V_{new}$, we have $|I| \geq k$ and I is an independent set
$\Rightarrow$ $(G, k) \in$ INDEPENDENT SET $\square$

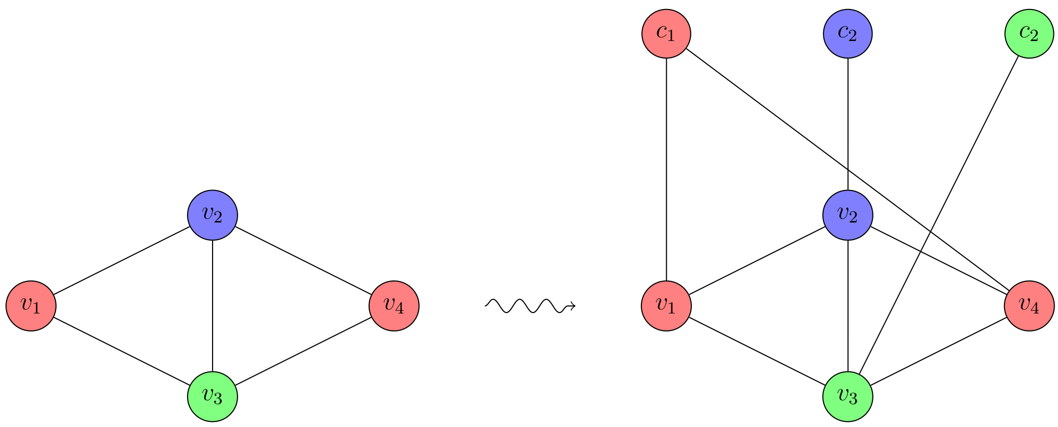

Figure 2.13: Illustration of Colouring to Partition Into Forests

2.3.12 Colouring $\leq_{pp}$ Partition Into Forests

Colouring

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Can the vertices of G be coloured using k different colours, so that no two vertices that are incident to each other have the same colour?

Parameter: k

Partition Into Forests

Input: A graph $G = (V, A)$ and a positive inter k.

Question: Can V be partitioned into k sets $V_1, \ldots, V_k$ so that the subgraph induced by each V_i is a forest[1]?

Parameter: k

The idea of this reduction is to create new vertices for every colour, and the set of vertices V_i of colour i is connected to the colour vertex c_i. This set then forms a tree and therefore a forest (see Figure 2.13). So the colouring V_1, V_2, V_3 with

[1] That is a set of trees

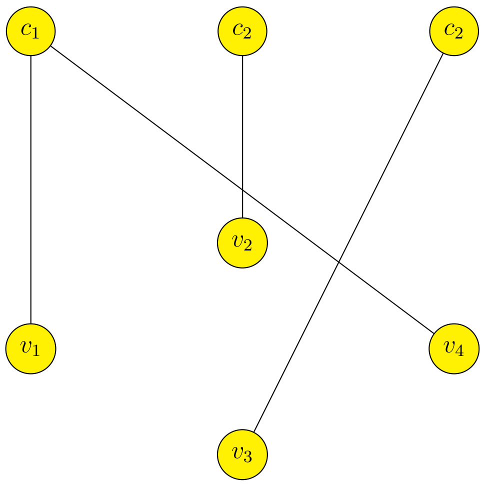

Figure 2.14: The corresponding solution of PARTITION INTO FORESTS

$V_1 = \{v_1, v_4\}$, $V_2 = \{v_2\}$ and $V_3 = \{v_3\}$ becomes the forest shown in Figure 2.14. The standard reduction which is well known, happens to be parameter preserving.

Proposition 2.3.12 COLOURING $\leq_{pp}$ PARTITION INTO FORESTS

Proof Let (G, k) with $G = (V, E)$ be the input of COLOURING. We transform this into (G', k) with $G' = (V', E')$ where $V' = V \cup C$ with $C = \{c_1, \ldots, c_k\}$ and $E' = E \cup \{(c_i, v) \mid v \in V, i \in \{1, \ldots, k\}\} \cup \{(c_i, c_j) \mid i \neq j \in \{1, \ldots, k\}\}$. This is polynomial in $|G|$ and the parameter k has not been changed.

$(G, k) \in$ COLOURING $\Rightarrow G$ can be coloured using k colours
$\Rightarrow$ There are k sets $V_1, \ldots, V_k$ where V_i is the set of vertices of colour i
$\Rightarrow V_i' = V_i \cup \{c_i\}$ induces a tree in G' since V_i is an independent set
$\Rightarrow G'$ can be partitioned into k forests
$\Rightarrow (G', k) \in$ PARTITION INTO FORESTS

$(G', k) \in$ PARTITION INTO FORESTS
$\Rightarrow G'$ can be partitioned into k sets $V_1, \ldots, V_k$ so that every V_i induces a forest
Now we have to do a case analysis to show that we have a correct colouring:

Case 1: $\exists i$ with $|V_i \cap C| > 2$

This cannot be since three nodes of C form a circle.

Case 2: $\forall i$: $|V_i \cap C| = 1$

If every V_i has exactly one vertex c_i of C then $V_i' = V_i - \{c_i\}$ for all i is a colouring for $G \Rightarrow (G, k) \in$ COLOURING

Case 3: $\exists i$ with $|V_i \cap C| = 0$

Since every vertex of C has to be in one of the sets $|V_i \cap C| = 0$ implies the existence of a set $V_j = \{c_a, c_b\}$ with $|V_j \cap C| = 2$

$\Rightarrow V_j \subseteq C$ because a vertex $v \in V$ would create a circle with two vertices of C

$\Rightarrow$ Since V_i induces a forest in G' we can find a 2-colouring $A \cup B = V_i$

$\Rightarrow$ we can swap V_i and V_j with $V_i' = A \cup \{c_a\}$ and $V_j = B \cup \{c_b\}$ without hurting the partition into forest property because A and B have to be independent sets

$\Rightarrow$ since we can do this for every pair V_i, V_j we can create Case 2

$\Rightarrow (G, k) \in$ COLOURING $\square$

2.3.13 Colourful Graph Motif $\leq_{pp}$ Group Steiner Tree

Colourful Graph Motif

Input: A graph $G = (V, E)$ a colour function $col\colon V \to \{1, \dots, k\}$ and a positive integer k.

Question: Is there a connected subset $S \subseteq V$, with $|S| \leq k$, so that $col|_S$ is bijective, that is S contains exactly one vertex of each colour?

Parameter: k

Group Steiner Tree

Input: A graph $G = (V, E)$, disjoint sets $T_1, \dots, T_k \subseteq V$ and a positive integer p.

Question: Is there a subset $S \subseteq V$, so that $G[S]$ is connected, $|S| = p$ and $S \cap T_i \neq \emptyset$ for all $i \in \{1, \dots, k\}$?

Parameter: k

This reduction was proposed in [12] and the idea is to set the T_i to all vertices of colour i. That way the questions become equivalent. But since it is not necessary that $T_1 \cup \dots \cup T_k = V$, we can not assume that Group Steiner Tree $\leq_{pp}$ Colourful Graph Motif.

Proposition 2.3.13 Colourful Graph Motif $\leq_{pp}$ Group Steiner Tree

Proof Let (G, k, col) be the input of Graph Motif with $G = (V, E)$. We transform this into $(G, p, T_1, \dots, T_k, k)$ where $p = k$ and $T_i = col^{-1}(i)$. This is polynomial in the input length and the parameter k has not been changed.

We can now see, that Steiner Tree asks whether there is a connected set S of size $p = k$ so that S hits at every T_i. But because of $|S| = k$, S can only contain exactly one vertex of each T_i. So we ask if there is a connected set S that contains exactly one vertex of each colour. □

Proposition 2.3.14 ([12]) *This reduction is d-degeneracy preserving*[1]*.*

[1] A graph G is d-degenerate iff in every subgraph of G there is a vertex with degree of at most d

$T = \{t_1, t_2, t_3\}$

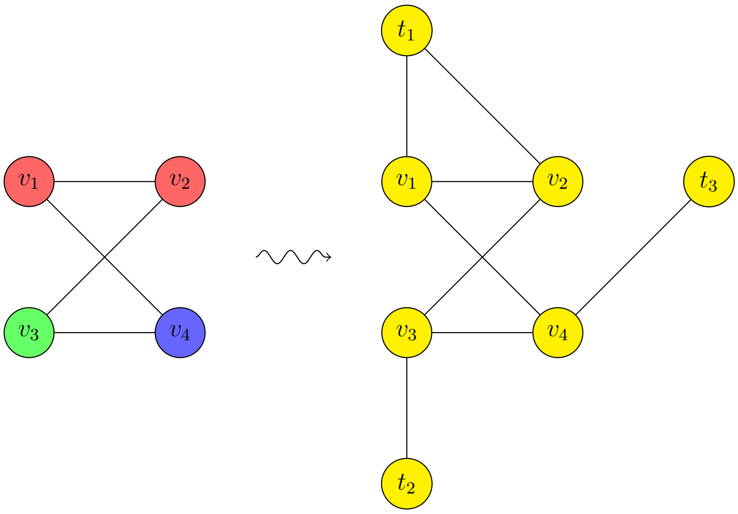

Figure 2.15: Illustration of COLOURFUL GRAPH MOTIF to CONNECTED DOMINATING SET

2.3.14 Colourful Graph Motif $\leq_{pp}$ Steiner Tree

COLOURFUL GRAPH MOTIF

Input: A graph $G = (V, E)$ a colour function *col*: $V \to \{1, \ldots, k\}$ and a positive integer k.

Question: Is there a connected subset $S \subseteq V$, with $|S| \leq k$, so that $col|_S$ is bijective, that is S contains exactly one vertex of each colour?

Parameter: k

STEINER TREE

Input: A graph $G = (V, E)$, a set $T \subseteq V$ and a positive integer k.

Question: Is there a subset $S \subseteq V - T$, with $|S| \leq k$, so that the subgraph induced by $T \cup S$ is connected?

Parameter: k

This reduction was proposed in [12] and the idea is to create terminals for each colour and connect them to the vertices of this colour. A solution for STEINER TREE is then the same as a solution for COLOURFUL GRAPH MOTIF (see Figure 2.15).

Proposition 2.3.15 COLOURFUL GRAPH MOTIF $\leq_{pp}$ STEINER TREE

Proof Let (G, col, k) with $G = (V, E)$ be the input of COLOURFUL GRAPH MOTIF. We transform this into (G', T, k) with $G' = (V', E')$ where $V' = V \cup T$ with $T = \{t_i \mid i \in \{1, \ldots, k\}\}$ and $E' = E \cup \{(v, t_i) \mid v \in col^{-1}(i)\}$. This is polynomial in the input length and the parameter k has not been changed.

$(G, col, k) \in$ COLOURFUL GRAPH MOTIF
$\Rightarrow$ There is a valid graph motif S of size at most k
$\Rightarrow$ S is connected and contains exactly one vertex of each colour
$\Rightarrow$ $G'[S \cup T]$ is connected
$\Rightarrow$ $(G', T, k) \in$ STEINER TREE

$(G', T, k) \in$ STEINER TREE
$\Rightarrow$ There is a valid solution S for Steiner tree in G' of size at most k
$\Rightarrow$ Since $G'[S \cup T]$ is connected, S has to contain exactly one vertex of each colour, otherwise the corresponding terminal could not been connected
$\Rightarrow$ Since all the terminals are leaves, S has to be connected regardless of T
$\Rightarrow$ S is a valid solution for graph motif
$\Rightarrow$ $(G, col, k) \in$ COLOURFUL GRAPH MOTIF $\square$

Proposition 2.3.16 *This reduction is d-degeneracy preserving, because if G is d-degenerate then G' is $(d+1)$-degenerate since the terminals T form an independent set and we have only added one edge to each non-terminal.*

2.4 Parameter preserving reduction with *linear* parameter change

2.4.1 Weighted SAT $\leq_{pp}$ Dominating Set

WEIGHTED SAT

Input: A Boolean expression φ in conjunctive normal form, and a positive integer k.

Question: Is there a truth assignment of weight[1] k that satisfies φ?

Parameter: k

DOMINATING SET

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq V$, with $|D| \leq k$, so that for every $v \in V$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

This reduction was proposed in [4] and what we want to do is, take the k variables that are true and convert them to a dominating set D of size $2k$. This dominating set will contain the k vertices that tell what variables we set to *true* and the k vertices that tell what intervals (considering mod n) of variables are false.
$a[3, 4] \in D$ means the third variable chosen to be set to *true* is x_4. The edges of E_4 add the constraint that every vertex of D in the set $B(3)$ has to belong to $B(3, 4)$. The index of the vertex of D in the subset $B(3, 4)$ represents the difference (mod n) between the indices of the third and fourth choices of a variable to receive the value *true*, and thus the vertex represents a range of variables to receive the value *false*. The edges of E_5 and E_9 enforce that the index t of the vertex of D in the subset $B(3, 4)$ represents the *distance* to the next variable to be set *true*, as it is represented by the unique vertex of D in the set $A(4)$.

[1] The amount of positive variables

Proposition 2.4.1 WEIGHTED SAT $\leq_{pp}$ DOMINATING SET

Proof Let φ be a Boolean expression in conjunctive normal form consisting of m clauses $C_1, \ldots, C_m$ over the set of n variables $x_0, \ldots, x_{n-1}$. We transform the input (φ, k) of WEIGHTED SAT to a graph $G = (V, E)$ and a parameter $k' = 2k$, where $V = V_1 \cup \cdots \cup V_6$ and $E = E_1 \cup \cdots \cup E_9$ with:

- $V_1 = \{a[r, s] \mid 0 \leq r \leq k-1, 0 \leq s \leq n-1\}$
- $V_2 = \{b[r, s, t] \mid 0 \leq r \leq k-1, 0 \leq s \leq n-1, 1 \leq t \leq n-k+1\}$
- $V_3 = \{c[j] \mid 1 \leq m\}$
- $V_4 = \{a'[r, u] \mid 0 \leq r \leq k-1, 1 \leq u \leq 2k+1\}$
- $V_5 = \{b'[r, u] \mid 0 \leq r \leq k-1, 1 \leq u \leq 2k+1\}$
- $V_6 = \{d[r, s] \mid 0 \leq r \leq k-1, 0 \leq s \leq n-1\}$

And with (implicitly quantified over all possible indices):

- $E_1 = \{(c[j], a[r, s]) \mid x_s \in C_j\}$
- $E_2 = \{(a[r, s], a[r, s']) \mid s \neq s'\}$
- $E_3 = \{(b[r, s, t], b[r, s, t']) \mid t \neq t'\}$
- $E_4 = \{(a[r, s], b[r, s', t]) \mid s \neq s'\}$
- $E_5 = \{(b[r, s, t], d[r, s]) \mid s' \neq s + t \mod n\}$
- $E_6 = \{(a[r, s], a'[r, u])\}$
- $E_7 = \{(b[r, s, t], b'[r, u])\}$
- $E_8 = \{(c[j], b[r, s, t]) \mid \exists i \text{ with } \overline{x_i} \in C_j, s < i < s + t\}$
- $E_9 = \{(d[r, s], a[r', s]) \mid r' = r + 1 \mod n\}$

Additionally consider these subsets of the vertices:

- $A_r = \{a[r, s] \mid 0 \leq s \leq n-1\}$

- $B_r = \{b[r,s,t] \mid 0 \leq s \leq n-1, 1 \leq t \leq n-k+1\}$
- $B_{r,s} = \{b[r,s,t] \mid 1 \leq t \leq n-k+1\}$

We show that if $(\varphi, k) \in$ Weighted SAT then $(G, k') \in$ Dominating Set. Let $\mathcal{T}$ be a truth assignment that satisfies φ and sets k variables to *true*, and these are $x_{i_0}, \ldots, x_{i_{k-1}}$, with $i_0 < i_2 < \cdots < i_{k-1}$. Let $d_r = i_{r+1 \mod k} - i_r \mod n$ for $r \in \{0, \ldots, k-1\}$. Then $D = A \cup B$ with

$$A = \{a[r, i_r] \mid r \in \{0, \ldots, k-1\}\} \text{ and } B = \{b[r, i_r, d_r] \mid r \in \{0, \ldots, k-1\}\}$$

is a dominating set in G consisting of $k' = 2k$ vertices, because A dominates the sets V_1, V_4, V_3 and B the sets V_2, V_5, V_6.
Now we show that if $(G, k') \in$ Dominating Set then $(\varphi, k) \in$ Weighted SAT. Let D be a dominating set in G with size $k' = 2k$. Since the closed neighbourhoods of $a'[0,1], \ldots, a'[k-1,1], b'[0,1], \ldots, b'[k-1,1]$ are disjointed D has to consist of vertices in each of the closed neighbourhoods. Furthermore D does not consist of vertices of $V_4 \cup V_5$, because then $2k$ vertices would not suffice for none of the vertices of $V_4 \cup V_5$ are incident to each other and this set contains more then $2k$ vertices. We conclude that D consists of exactly one vertex from each $A(r)$ and $B(r)$ for $r \in \{0, \ldots, k-1\}$.
The edges of E_4, E_5 and E_9 enforce that the $2k$ vertices in D must represent such a choice consistently. The edges E_1 and E_8 insure that the truth assignment represented by D satisfies φ. $\square$

$$R = \{v_1, \dots, v_4\} \text{ and } B = \{v_5, \dots, v_8\}$$

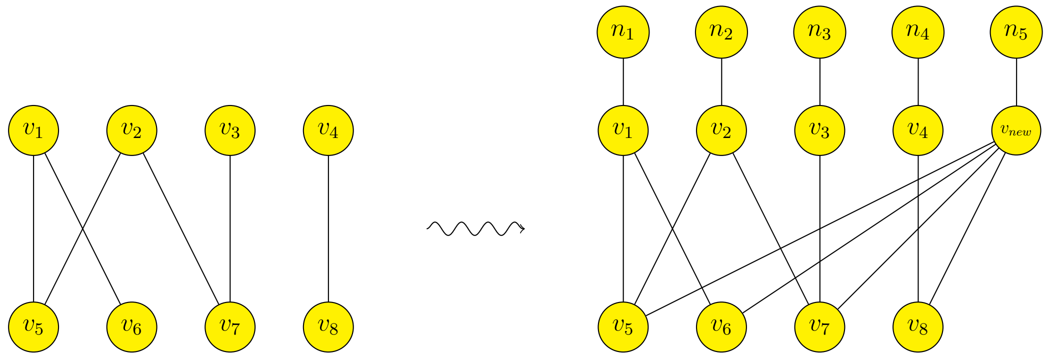

Figure 2.16: Illustration of Red-Blue Dominating Set to Connected Vertex Cover.

2.4.2 Red-Blue Dominating Set $\leq_{pp}$ Connected Vertex Cover

Red-Blue Dominating Set

Input: A bipartite graph $G = (R \cup B, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq B$, with $|D| \leq k$, so that for every $v \in R$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: $k, |R|$

Connected Vertex Cover

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$ and the subgraph induced by C is connected?

Parameter: k

Please note for this reduction to work we have to parametrize $|R|$ as well, as our transformed k will be dependent on this value. This reduction was proposed in [9] and the idea is similar to the reduction to Steiner Tree (section 2.3.8), but we also add leaf vertices to every vertex of R'. The vertex cover are now all vertices of R' plus those of B that dominate R (See Figure 2.16).

Proposition 2.4.2 Red-Blue Dominating Set $\leq_{pp}$ Connected Vertex Cover

Proof Let (G, k) with $G = (R \cup B, E)$ be the input of Red-Blue Dominating Set. We transform this into $G' = (V', E')$ where $V' = R \cup B \cup \{v_{new}\} \cup L$, with $L = \{n_1, \ldots, n_{|R|+1}\}$ and $E' = E \cup \{(v_{new}, v_i) \mid v_i \in B\} \cup \{(n_i, v_i) \mid v_i \in R\} \cup \{(n_{|R|+1}, v_{new})\}$. Then we set $k' = |R| + 1 + k$. This is polynomial in $|G|$ and the parameter has been changed linearly.

$(G, k) \in$ Red-Blue Dominating Set
$\Rightarrow$ There is a set $D \subseteq B$ of size at most k that dominates R
$\Rightarrow$ $C = D \cup R \cup \{v_{new}\}$ is a vertex cover since $R \cup \{v_{new}\}$ covers all vertices of L and v_{new} covers all vertices of B
$\Rightarrow$ C is connected because D dominates R and has size $|D| + |R| + |\{v_{new}\}| = k + |R| + 1 = k'$
$\Rightarrow$ $(G', k') \in$ Connected Vertex Cover

$(G', k') \in$ Connected Vertex Cover
$\Rightarrow$ There is a vertex cover C of size k' that is connected
$\Rightarrow$ $R \cup \{v_{new}\}$ has to be part of C because of the leaf vertices L. That leaves k vertices of R to be part of C, we call those D
$\Rightarrow$ since C was connected and no two vertices of R are connected D has to dominate R
$\Rightarrow$ $(G, k) \in$ Red-Blue Dominating Set $\square$

$$R = \{r_1, \ldots, r_4\} \text{ and } B = \{b_1, \ldots, b_3\}$$

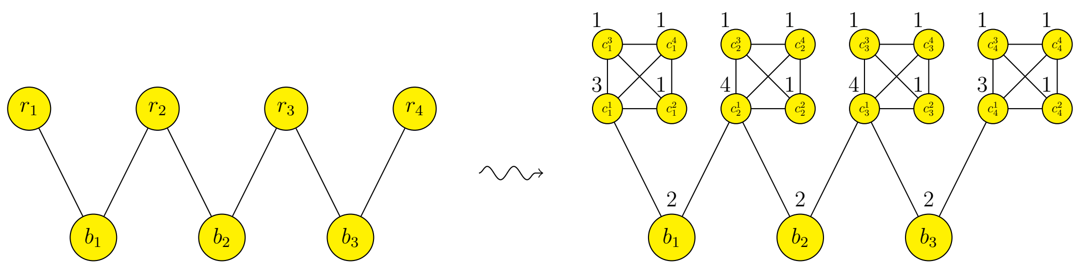

Figure 2.17: Illustration of Red-Blue Dominating Set to Capacitated Vertex Cover.

2.4.3 Red-Blue Dominating set $\leq_{pp}$ Capacitated Vertex Cover

Red-Blue Dominating Set

Input: A bipartite graph $G = (R \cup B, E)$ and a positive integer k.

Question: Is there a subset $D \subseteq B$, with $|D| \leq k$, so that for every $v \in R$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: $k, |R|$

Capacitated Vertex Cover

Input: A graph $G = (V, E)$, a capacity function $cap\colon V \to \mathbb{N}^+$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$ and a function $f\colon E \to C$ that maps every edge to one of its endpoints, so that for all $v \in C$, $|f^{-1}(v)| \leq cap(v)$?

Parameter: k

Please note for this reduction to work we have to parametrize $|R|$ as well, as our transformed k will be dependent on this value. This reduction was proposed in [9] and the idea is to convert every node of R to a clique consisting of four nodes.

The capacity is then set to one, for all clique nodes except the first one. The first clique node gets the capacity $deg - 1$ and the remaining nodes the capacity deg (see Figure 2.17). The capacity of a node is the amount of edges this node can cover.

Proposition 2.4.3 Red-Blue Dominating Set $\leq_{pp}$ Capacitated Vertex Cover

Proof Let (G, k) be the input of Red-Blue Dominating Set, with $G = (R \cup B, E)$. We transform this into (G', cap, k') with $G' = (V', E')$ where $V' = R' \cup B$ where $R' = \{c_i^1, c_i^2, c_i^3, c_i^4 \mid r_i \in R\}$ and $E' = \{(c_i^j, c_i^l) \mid j \neq l, c_i^j, c_i^l \in R'\} \cup \{(c_i^1, b_j) \mid (r_i, b_j) \in E, r_i \in R, b_j \in B\}$ with

$$cap(v_i) = \begin{cases} deg(v_i) & \text{if } v_i \in B \\ deg(v_i) - 1 & \text{if } v_i = c_i^1 \in R' \\ 1 & \text{otherwise} \end{cases}$$

and finally $k' = 4|R| + k$. This is linear in $|G|$ and the parameter k' is polynomial in $|R|$ and k.

$(G, k) \in$ Red-Blue Dominating Set
$\Rightarrow G$ has a set $D \subseteq B$ of size at most k that dominates R
$\Rightarrow C = D \cup V' - B$ where $V' - B$ are all the nodes of the cliques, is a vertex cover of size at most $|D| + |V' - B| = k + 4|R| = k'$, the cover is capacitated according to definition
$\Rightarrow (G', k') \in$ Capacitated Vertex Cover

$(G', k') \in$ Capacitated Vertex Cover
$\Rightarrow$ There exists a capacitated vertex cover $C \subseteq V'$ with size at most k'
$\Rightarrow$ Since the nodes $c_i^1, \ldots, c_i^4$ form a clique all these node have to be in C
$\Rightarrow$ Since $cap(c_i^0)$ is too small to cover all edges at least one neighbour of c_i^0 that is in B has to be in C
$\Rightarrow$ The nodes of $D = C \cap B$ dominate R
$\Rightarrow (G, k) \in$ Red-Blue Dominating Set $\square$

2.4.4 Group Steiner Tree $\leq_{pp}$ Directed Steiner Out-Tree

Group Steiner Tree

Input: An undirected graph $G = (V, E)$, vertex-disjoint subsets $S_1, \ldots, S_k$ and a positive integer p.

Question: Does G contain a tree of at most p vertices that contains at least on vertex of each S_i?

Parameter: k, p

Directed Steiner Out-Tree

Input: A directed graph $D = (V, A)$, a distinguished vertex $r \in V$, a set of terminals $D \subseteq V$ and a positive inter p.

Question: Does D contain an out-tree[1] of at most p vertices that is rooted at r and contains all the vertices of T?

Parameter: $k = |S|, p$

This reduction was proposed in [11] and the idea is to create a directed graph out of G that contains for every edge (v, u) two arcs $(v, u), (u, v)$ and further adds new nodes for every set S_i that gets an arc (v, s_i) for $v \in S_i$. Finally we add a root node r that is connected to every node of V, not directly but over a path of length $|V|$ (see Figure 2.18[2]). So for a solution for Group Steiner Tree (e.g. left in Figure 2.19) we get a solution for Directed Steiner Out-Tree (right in Figure 2.19).

Proposition 2.4.4 Group Steiner Tree $\leq_{pp}$ Directed Steiner Out-Tree

Proof Let $(G, S_1, \ldots, S_k, p)$ be the input of Group Steiner Tree. We transform this as followed: Let $S = \{r, s_1, \ldots, s_k\}$ be a set of $k + 1$ new vertices.

[1]A directed graph in which, for a vertex r called the root and any other vertex v, there is exactly one directed path from r to v.

[2]The path of r to a node v is a single arc for brevity

$$S_1 = \{v_1, v_2\}, S_2 = \{v_4, v_5\}, S_3 = \{v_7, v_8\}, S_4 = \{v_{10}, v_{11}\}$$

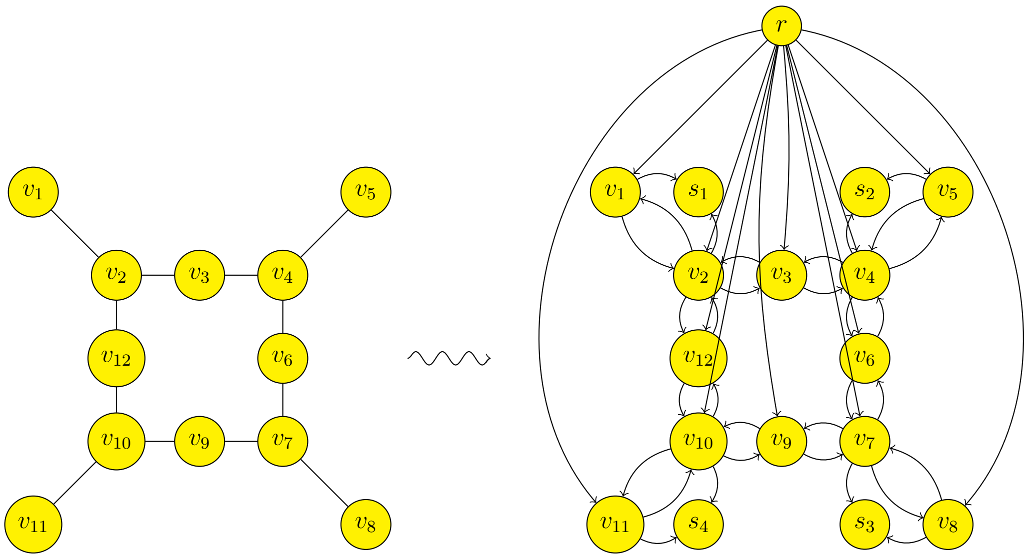

Figure 2.18: Illustration of Group Steiner Tree to Directed Steiner Out-Tree.

Further let $A = \{(u,v),(v,u) \mid (u,v) \in E\} \cup \bigcup_{i=1}^{k}\{(v,s_i) \mid v \in S_i\}$. And for every $u \in V$ create a path $P_{r,u}$ from r to u of length $n = |V|$ that is $V_u = \{u_1,\dots,u_n\}$ and $P_{r,u} = \{(r,u_1),(u_n,u),(u_i,u_{i+1}) \mid i \in \{1,\dots,n-1\}\}$. So we get $V' = V \cup S \cup \bigcup_{u \in V} V_u$ and $A' = A \cup \bigcup_{u\in V} P_{r,u}$. Finally we set $D = (V', A')$ and $p' = p + n + 1 + k$. This is polynomial in $|G| + k$ and p' is linear in p,n,k and the parameter k has only been increased by 1.

$(G, S_1, \dots, S_k, p) \in$ Group Steiner Tree
$\Rightarrow$ G contains a tree T of at most p vertices that includes at least one vertex of each S_i
$\Rightarrow$ There is a tree T' in D containing T with r as the root using one Path $P_{r,u}$ and for every $v_i \in S_i \cap T$ we can add s_i to T'
$\Rightarrow$ T' is a directed out-tree of length $p + n + 1 + k$
$\Rightarrow$ $(D, S, p', k+1) \in$ Directed Steiner Out-Tree

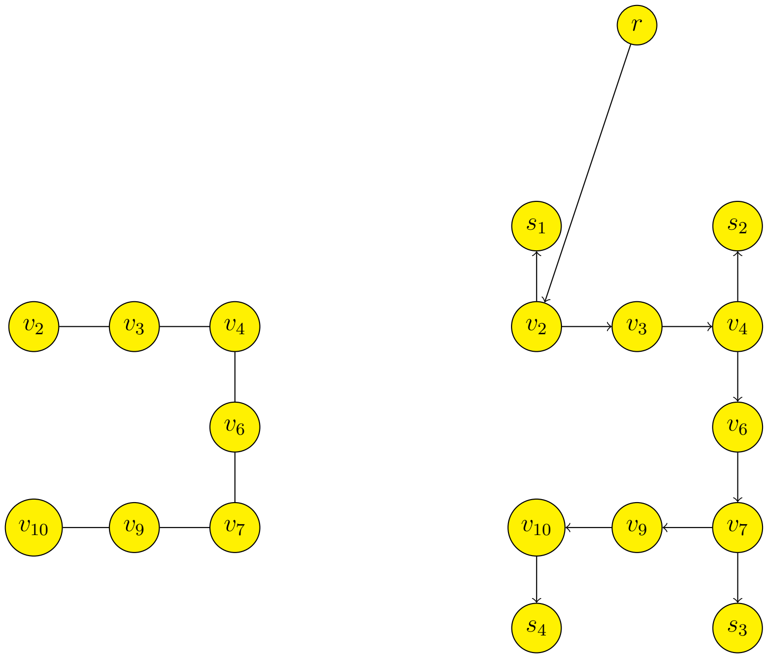

Figure 2.19: Left: A solution of Group Steiner Tree Right: A solution of Directed Steiner Out-Tree

$(D, S, p', k+1) \in$ Directed Steiner Out-Tree

$\Rightarrow$ There exists an out-tree T of at most $p + n + 1 + k$ vertices that contains r and S

$\Rightarrow$ T contains only one path $P_{r,u}$ since $2n + k > p + n + 1 + k$

$\Rightarrow$ $T' = T \cap V$ is a sub-tree of V

$\Rightarrow$ T' has at most p vertices and forms a group Steiner tree

$\Rightarrow$ $(G, S_1, \ldots, S_k, p) \in$ Group Steiner Tree $\square$

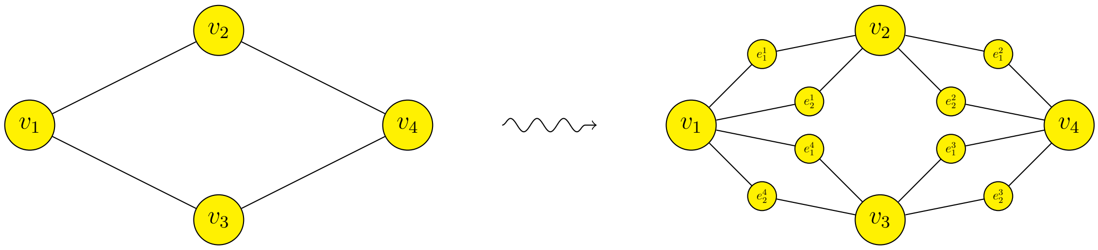

Figure 2.20: Illustration of Connected Vertex Cover to 2-deg-Connected Feedback Vertex Set

2.4.5 Connected Vertex Cover $\leq_{pp}$ 2-deg-Connected Feedback Vertex Set

Connected Vertex Cover

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$ and the subgraph induced by C is connected?

Parameter: k

2-deg-Connected Feedback Vertex Set

Input: A 2-degenerate[1] graph $G = (V, A)$ and a positive integer k.

Question: Is there a $S \subseteq V$ with $|S| \leq k$ so that the subgraph induced by S is connected and the subgraph induced by $V - S$ has no cycles?

Parameter: k

This reduction was proposed in [12] and the idea is to convert every edge of G in a cycle of length 4, where the original vertices are not adjacent to each other (see Figure 2.20).

[1] A graph G is d-degenerate iff in every subgraph of G there is a vertex with degree of at most d

Observation 2.4.5 *If in a graph G every edge has an endpoint of degree at most 2, G is 2-degenerate.*

Proposition 2.4.6 Connected Vertex Cover $\leq_{pp}$ 2-deg-Connected Feedback Vertex Set

Proof Let (G, k) with $G = (V, E)$ be the input of Connected Vertex Cover. We transform this into (G', k') with $k' = 2k + 1$ and $G' = (V', E')$ where $V' = V \cup \{e_1, e_2 \mid e \in E\}$ and $E' = \{(v, e_1), (v, e_2), (u, e_1), (u, e_2) \mid (v, u) \in E\}$. This is polynomial in $|G|$ and the parameter k has not been changed. G' is 2-degenerate (observation 2.4.5).

$(G, k) \in$ Connected Vertex Cover
$\Rightarrow$ G has a connected vertex cover $C \subseteq V$ of size at most k
$\Rightarrow$ $G'[V' - C]$ contains no cycles, because if V is an independent set in G' a cycle has to contain one e_i, but this means both vertices adjacent to e_i have to be in the cycle as well, which contradicts the assumption that C was a vertex cover since e_i would not have been covered
$\Rightarrow$ We can connect S in G' through finding a spanning tree in $G[S]$ of at most $k - 1$ edges and using one of the e_1 for every edge in the spanning tree. We call this V_{st}
$\Rightarrow$ $S = C \cup V_{st}$ is a connected feedback vertex set of size at most $2k - 1 = k'$
$\Rightarrow$ $(G', k') \in$ 2-deg-Connected Feedback Vertex Set

$(G', k') \in$ 2-deg-Connected Feedback Vertex Set
$\Rightarrow$ There is a set $S \subseteq V'$ of size at most k' so that S is connected $G'[V' - S]$ is circle free
$\Rightarrow$ $|S| \geq 2$ ($|S|$=1 is trivial) and $|S \cap V| \leq k$ since otherwise they would form at least $k + 1$ connected components, with E' connecting at most two of them, so S could not be connected
$\Rightarrow$ for $e = (u, v) \in E$ we have a cycle (u, e_1, v, e_2) in G' so S has to contain at least one of them. But as $|S| \geq 2$ and S is connected it has to be either u or v and e is covered
$\Rightarrow$ $C = S \cap V$ is a connected vertex cover
$\Rightarrow$ $(G, k) \in$ Connected Vertex Cover □

2.4.6 Connected Vertex Cover $\leq_{pp}$ 2-deg-Connected Odd Cycle Transversal

CONNECTED VERTEX COVER

Input: A graph $G = (V, E)$ and a positive integer k.

Question: Is there a subset $C \subseteq V$, with $|C| \leq k$, so that for every $(v, u) \in E$, $v \in C$ or $u \in C$ and the subgraph induced by C is connected?

Parameter: k

2-DEG-CONNECTED ODD CYCLE TRANSVERSAL

Input: A 2-degenerate[1] graph $G = (V, A)$ and a positive integer k.

Question: Is there a $S \subseteq V$ with $|S| \leq k$ so that the subgraph induced by S is connected and the subgraph induced by $V - S$ is bipartite (that is, contains no cycles of odd length)?

Parameter: k

This reduction was proposed in [12] and works the same way as seen in section 2.4.5, with the difference that we build a circle of length five for every edge instead.

Proposition 2.4.7 CONNECTED VERTEX COVER $\leq_{pp}$ 2-DEG-CONNECTED ODD CYCLE TRANSVERSAL

Proof Omitted.

[1]A graph G is d-degenerate iff in every subgraph of G there is a vertex with degree of at most d

$T = \{v_1, v_4, v_5, v_6\}$

Figure 2.21: Illustration of Steiner Tree to 2-deg-Steiner Tree

2.4.7 Steiner Tree $\leq_{pp}$ 2-deg-Steiner Tree

Steiner Tree

Input: A graph $G = (V, E)$ a set $T \subseteq V$ and a positive integer k.

Question: Is there a subset $S \subseteq V - T$, with $|S| \leq k$, so that the subgraph induced by $T \cup S$ is connected?

Parameter: k and $t = |T|$

2-deg-Steiner Tree

Input: A 2-degenerate[1] graph $G = (V, E)$ a set $T \subseteq V$ and a positive integer k.

Question: Is there a subset $S \subseteq V - T$, with $|S| \leq k$, so that the subgraph induced by $T \cup S$ is connected?

Parameter: k and $t = |T|$

This reduction was proposed in [12] and the idea is to subdivide each edge with an edge-vertex, the new solution is then the old one plus those edge-vertices, so that the graph is connected (see Figure 2.21).

Proposition 2.4.8 Steiner Tree $\leq_{pp}$ 2-deg-Steiner Tree

[1] A graph G is d-degenerate iff in every subgraph of G there is a vertex with degree of at most d

Proof Let (G, k, T) with $G = (V, E)$ be the input of Steiner Tree. We transform this into (G', k', T) with $G' = (V', E')$ where $V' = V \cup V_e$ with $V_e = \{v_e \mid e \in E\}$ and $E' = \{(v, v_e), (v_e, u) \mid e = (v, u) \in E\}$ and $k' = 2k + |T| - 1$. This is polynomial in $|G|$ and the parameter has been changed linearly. G' is 2-degenerate by observation 2.4.5.

$(G, k, T) \in$ Steiner Tree
$\Rightarrow$ There is a valid solution S so that $G[S \cup T]$ is connected
$\Rightarrow$ Let X be an arbitrary spanning tree of $G[S \cup T]$ and E_{tree} the set of its edges
$\Rightarrow |E_{tree}| \leq k + |T| - 1 = |E_{S \cup T}|$
$\Rightarrow$ Let $V_{E_{tree}}$ be the set of vertices corresponding to E_{tree} then $S' = S \cup V_{E_{tree}}$ is a valid solution of G' of size at most $2k + |T| - 1 = k'$
$\Rightarrow (G', k', T) \in$ 2-deg-Steiner Tree

$(G', k', T) \in$ 2-deg-Steiner Tree
$\Rightarrow$ There is a valid solution S in G'
$\Rightarrow S' = S \cap V$ has a cardinality of at most $k + |T|$ since $|S \cup T| \leq 2k + 2|T| - 1$
$\Rightarrow S' \cup T$ is isolated in G' and adding a single vertex from V_e connects at most two components
$\Rightarrow |S'| \leq k$ and since $S \cup T$ is connected in G', $G[S' \cup T]$ is connected
$\Rightarrow (G, k, T) \in$ Steiner Tree $\square$

2.4.8 Colourful Graph Motif $\leq_{pp}$ Connected Dominating Set

Colourful Graph Motif

Input: A graph $G = (V, E)$ a colour function col: $V \rightarrow \{1, \ldots, k\}$ and a positive integer k.

Question: Is there a connected subset $S \subseteq V$, with $|S| \leq k$, so that $col|_S$ is bijective, that is S contains exactly one vertex of each colour?

Parameter: k

Connected Dominating Set

Input: A graph $G = (V, E)$ and a positive integer p.

Question: Is there a subset $D \subseteq V$, so that $G[S]$ is connected, $|D| \leq k$ and that for every $v \in V$, we have $v \in D$ or there is a $u \in D$, so that $(v, u) \in E$?

Parameter: k

This reduction was proposed in [12] and what we want to do is, for every colour i we create two new vertices c_i and c'_i that are connected. We then connect every vertex v of colour i to c_i (see Figure 2.22).

Lemma 2.4.9 ([12]) *For $k < 2$ Colourful Graph Motif can be solved in polynomial time.*

Proposition 2.4.10 Colourful Graph Motif $\leq_{pp}$ Connected Dominating Set

Proof Because of lemma 2.4.9 let $k \geq 2$.
Let (G, col, k) with $G = (V, E)$ be the input of Colourful Graph Motif. We transform this into (G', k') with $G' = (V', E')$ with $V' = V \cup V_c$ where $V_c = \{c_i, c'_i \mid i \in \{1, \ldots, k\}\}$ and $E' = E \cup \{(c_i, c'_i) \mid i \in \{1, \ldots, k\}\} \cup \{(v, c_{col(v)}) \mid v \in V\}$ and $k' = 2k$. This is polynomial in the input and k has been changed linearly.

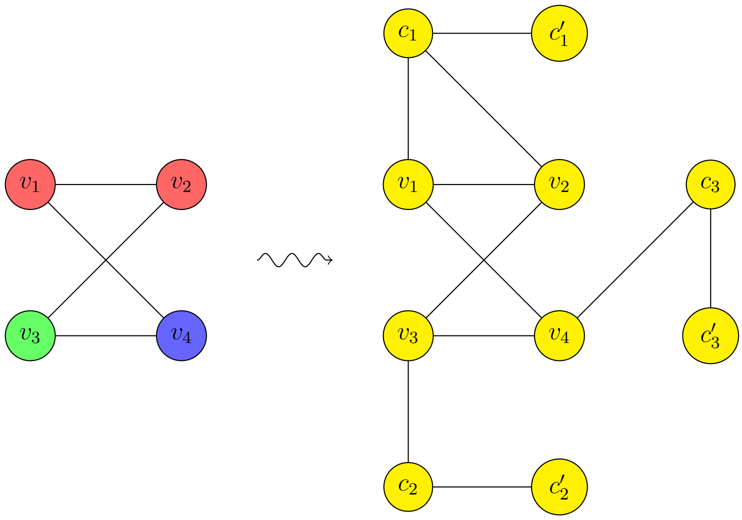

Figure 2.22: Illustration of Colourful Graph Motif to Connected Dominating Set

$(G, col, k) \in$ d-deg-Colourful Graph Motif
$\Rightarrow$ There exists a valid solution S with size at most k
$\Rightarrow$ Let $D = S \cup X$ with $X = \{c_1, \ldots, c_k\}$, since D dominates V and X dominates V_c and D and X are connected
$\Rightarrow$ D is a valid solution of size $2k = k'$ in G'
$\Rightarrow$ $(G', k') \in$ Connected Dominating Set

$(G', k') \in$ Connected Dominating Set $\Rightarrow$ There exists a valid solution D of size at most $2k$
$\Rightarrow$ $\{c_1, \ldots, c_k\} = X \subseteq D$ because we have to dominate $Y = \{c'_1, \ldots, c'_k\}$. If Y where in D we would have to take X also because of connectivity
$\Rightarrow$ For every c_i we have to take at least one neighbour $v \in V$, but since the neighbourhood of each c_i is disjoint and $|D| \leq 2k$ we have to take exactly one vertex of every neighbourhood
$\Rightarrow$ $S = D - X$ is a valid solution of size k in G
$\Rightarrow$ $(G, col, k) \in$ Colourful Graph Motif $\square$

Proposition 2.4.11 *If G is d-degenerate, then G' is $d+1$-degenerate because every vertex of V gets one new edge*

2.5 Parameter preserving reduction with *polynomial* parameter change

2.5.1 Coloured Reduced Unique Coverage $\leq_{pp}$ Unique Coverage

COLOURED REDUCED UNIQUE COVERAGE

Input: A set family $\mathfrak{F}$ over a universe U with $S \in \mathfrak{F} \Rightarrow |S| \leq k-1$ and $|U| \leq k^2$, a colour function *col*: $\mathfrak{F} \to \{1, \dots, k\}$ and a positive integer k.

Question: Is there a sub-family $\mathfrak{F}' \subseteq \mathfrak{F}$ so that at least k elements of U are contained in exactly one set in $\mathfrak{F}'$ and $\mathfrak{F}'$ has exactly on set of each colour?

Parameter: k

UNIQUE COVERAGE

Input: A set family $\mathfrak{F}$ over a universe U and a positive integer k.

Question: Is there a sub-family $\mathfrak{F}' \subseteq \mathfrak{F}$ so that at least k elements of U are contained in exactly one set in $\mathfrak{F}'$?

Parameter: k

This reduction was proposed in [9].

Proposition 2.5.1 COLOURED REDUCED UNIQUE COVERAGE $\leq_{pp}$ UNIQUE COVERAGE

Proof Let $(\mathfrak{F}, U, col, k)$ be the input of COLOURED REDUCED UNIQUE COVERAGE, we transform this into $(\mathfrak{H}, U', k')$ with $k' = k(k^2+1)+k$ and for every colour i we add a set S_i consisting of k^2+1 *new* elements to U, that is $U' = U \cup \bigcup_{i \in \{1,\dots,k\}} S_i$. Further for every set $A_i \in \mathfrak{F}$ we set $A'_i = A_i \cup S_{col(A_i)}$ and finally $\mathfrak{H} = \bigcup_{A_i \in \mathfrak{F}} A'_i$. Notice that in order to cover at least $k(k^2+1)$ elements uniquely one has to pick exactly one set of each colour. $\square$

2.6 Supposedly no parameter preserving reductions

As we have seen, there are problems that can be reduced using the standard reduction, but there has not been found a parameter-preserving reduction and it is supposed that there in fact is none. Please note that these are hypotheses that are not proven but very likely to hold (Like $P \neq NP$). Some of these are as followed:

- DOMINATING SET $\not\leq_{pp}$ INDEPENDENT SET [3]
- WEIGHTED SAT $\not\leq_{pp}$ WEIGHTED 3-SAT [3]
- INDEPENDENT SET $\not\leq_{pp}$ VERTEX COVER [2.1.1]
- CLIQUE $\not\leq_{pp}$ CONSTRAINT BIPARTITE VERTEX COVER [2.1.2]

These are just a small excerpt of reductions that are (probably) not possible with parameter-preserving reduction although conventional reductions are known. This shows us, that parametrized complexity distinguishes stronger between problems than conventional complexity theory. The notion that two problems are closely related does not have to hold if we take a closer look at the given parameters.

Chapter 3

Reduction-Graph

"An algorithm must be seen to be believed." (Donald Knuth)

We think a nice way to illustrate the results of this work is to create a directed graph, where the vertices are our problems and there is an edge from problem A to problem B iff $A \leq_{pp} B$. The text on the edges represents the parameter increase through the reduction. If there is no text then the parameter has not been changed. The reason this visual structure was chosen, is because of the transitivity of the reduction, if we have a path from A to C we know that $A \leq_{pp} C$ and it can be easily implemented as a data structure for further manipulation and analysis.

3.1 Interpretation

The graph can bee seen in Figure 3.1 and Figure 3.2 respectively. Figure 3.1 shows the biggest connected component of the graph. Here we have Vertex Cover and Colourful Graph Motif as the 'hardest' problems with no parent vertices. This is of course not absolute, as this work does not claim to be complete. Further we can see smaller circles of problems that seem to

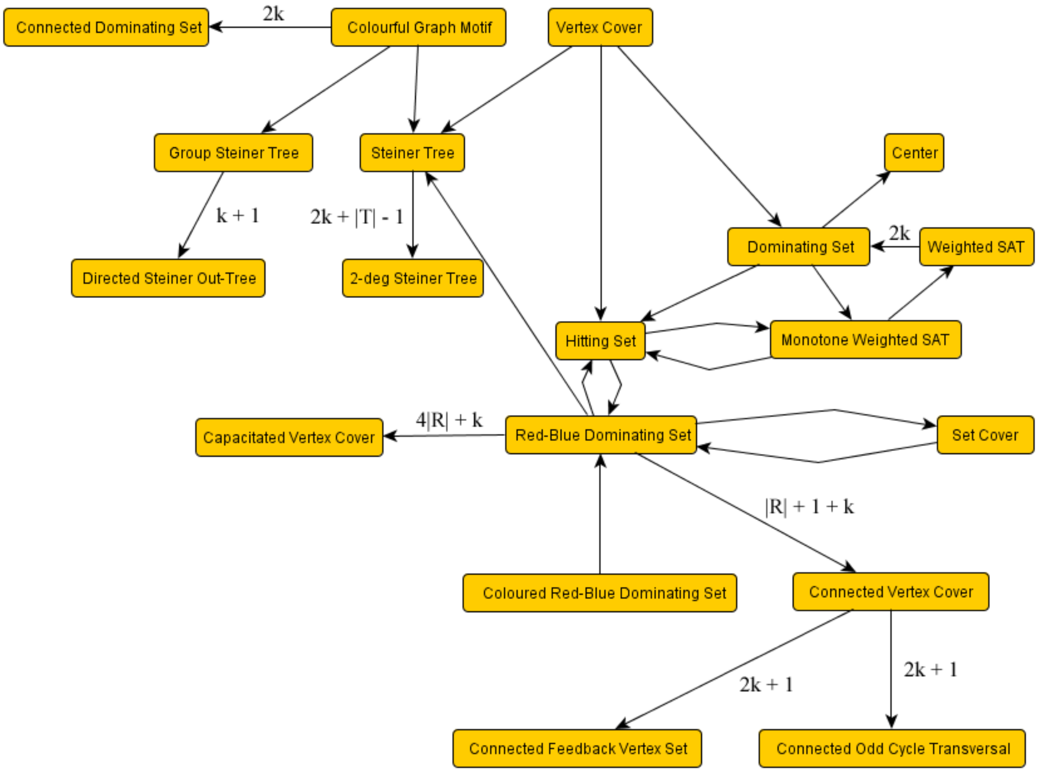

Figure 3.1: The first big connected component

be of equal hardness with respect to parametrized complexity like Dominating Set, Weighted SAT and Hitting Set. We can now use the transitivity of the reduction to infer additional statements that were not apparent at first, like Dominating Set $\equiv_{pp}$ Red-Blue Dominating Set (with a parameter increase of $2k$ in "$\leftarrow$" direction), Dominating Set $\equiv_{pp}$ Set Cover or Vertex Cover $\leq_{pp}$ Set Cover (both times no parameter increase). Figure 3.2 shows something that was expected, as this work was more of a 'breadth-first search' over different sources we have small 'islands' of problems that are not connected. The question at hand is, if it possible to find the 'missing-links' to create a connected graph. We think it is possible to create a lot of additional connections between problems.

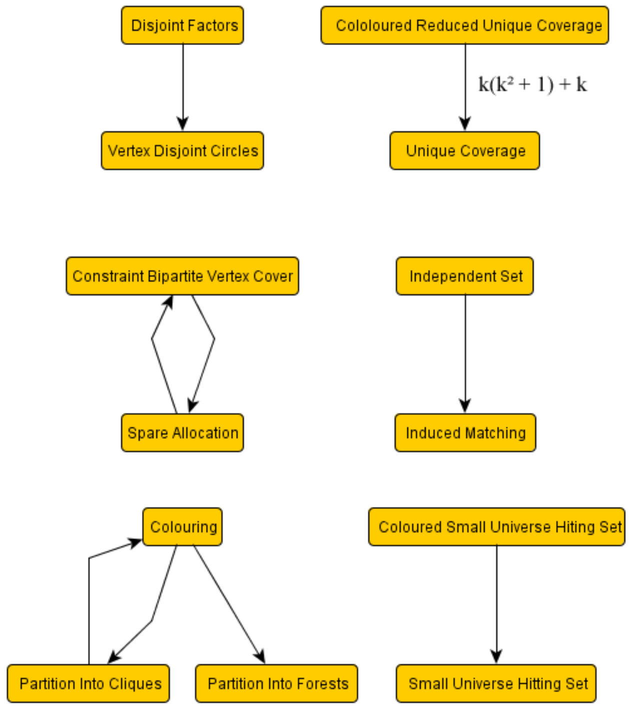

Figure 3.2: The remaining six smaller connected components

Chapter 4

Conclusions

> "The question of whether computers can think is like the question of whether submarines can swim." (Edsger W. Dijkstra)

4.1 Methods used

The goal of this work was to find, collect and categorize parameter preserving reductions. But first we wanted to give some motivation why this is a research area worth looking at, so some papers on general parametrized complexity where consulted. The results of this can be seen in Chapter 1, which helped us shape our own view on the topic and why I think this work can be at least of some use. The main source for the reductions where papers that showed the non-existence for polynomial kernels of some problems by giving reductions from problems already proven not to have polynomial kernels, general reductions that we encountered before, that happen to be parameter preserving or parameter preserving reductions I created myself.

4.2 Further work on visualisation and reduction search

As seen in chapter 3 a directed graph is a nice way to visualize reductions, because of the transitivity the search for a reduction can be as simple as a search for a path in the graph. A great tool for researchers in the field of reductions could be a web based application, that displays such a graph and has the possibilities of search operations, so that you can for example enter a problem A and get two lists, the first list is a list of all problems that can be reduced to this problem and the second one is a list of all problems that you can reduce problem A to. It would be possible to give every edge a weight which is the parameter increase of the reduction, so that we can search for 'shortest paths', that is the smallest possible parameter increase.

The success of Wikipedia shows that it could be a good idea to rely on crowd-sourced content, so people can add new problems and edges, of course it would be a necessity to reference a source for every edge added. Today there are a countless number of problems and found reductions so I think that a central resource that collects parametrized problems and reductions and first and foremost presents them in a practical way would be a valuable addition to research in the field.

References

[1] N. G. De Bruijn. *Asymptotic Methods in Analysis*. Courier Dover Publications, 1970. 2

[2] M. Cesati. Compendium of Parameterized Problems, 2006.

[3] R. Downey. Parameterized Complexity for the Skeptic. *Proceedings of The 18th IEEE Annual Conference on Computational Complexity*, 2003. 6, 10, 62

[4] R. G. Downey; M. R. Fellows. Fixed-Parameter Tractability and Completeness. 44

[5] R. G. Downey; M. R. Fellows. *Parameterized Complexity*. Springer, 1999. 3, 4

[6] S.Y. Kuo; W. K. Fuchs. Efficient Spare Allocation for Reconfigurable Arrays. *IEEE Design and Test*, 1987. 9

[7] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Habilitation thesis, University Tuebingen, 2002. 2

[8] J. M. Robson. Algorithms for Maximum Independent Sets. *Journal of Algorithms*, 7, 1986. 3

[9] M. Dom; D. Lokshtanov; S. Saurabh. Incompressibility through Colors and IDs, . 16, 29, 31, 33, 47, 49, 61

[10] N. Misra; V. Raman; S. Saurabh. Lower Bounds on Kernelization, . 34

[11] N. Misra; G. Philip; V. Raman; S. Saurabh; S. Sikdar. FPT Algorithms for Connected Feedback Vertex Set. 51

[12] M. Cygan; M. Pilipczuk; M. Pilipczuk; J. O. Wojtaszczyk. Kernelization Hardness of Connectivity Problems in d-Degenerate Graphs. 41, 43, 54, 56, 57, 59

[13] J. Chen; I. A. Kanj; G. Xia. Improved Parameterized Upper Bounds for Vertex Cover. *Mathematical Foundations of Computer Science*, 4162. 3

Appendix: List of Problems

Problem	**Reduced to**
Coloured Red-Blue Dominating Set	Red-Blue Dominating Set
Coloured Reduced Unique Coverage	Unique Coverage
Coloured Small Universe Hitting Set	Small Universe Hitting Set
Colourful Graph Motif	Group Steiner Tree, Steiner Tree, Connected Dominating Set
Colouring	Partition Into Cliques, Partition Into Forests
Connected Vertex Cover	Connected Feedback Vertex Set, Connected Odd Cycle Transversal
Constraint Bipartite Vertex Cover	Spare Allocation
Disjoint Factors	Vertex Disjoint Circles
Dominating Set	Hitting Set, Monotone Weighted SAT, Center
Group Steiner Tree	Directed Steiner Out-Tree
Hitting Set	Monotone Weighted SAT, Red-Blue Dominating Set
Independent Set	Induced Matching
Monotone Weighted SAT	Hitting Set, Weighted SAT
Partition into Cliques	Colouring
Red-Blue Dominating Set	Set Cover, Hitting Set, Steiner Tree, Connected Vertex Cover, Capacitated Vertex Cover
Set Cover	Red-Blue Dominating Set
Spare Allocation	Constraint Bipartite Vertex Cover
Steiner Tree	2-deg Steiner Tree
Vertex Cover	(2-)Hitting Set, Dominating Set, Steiner Tree
Weighted SAT	Dominating Set