Algorithmic Aspects of Upper Domination: A Parameterised Perspective

C. Bazgan[1,*], L. Brankovic[2,3], K. Casel[3], H. Fernau[3], K. Jansen[4], K. Klein[4], M. Lampis[1], M. Liedloff[5], J. Monnot[1], and V. Th. Paschos[1]

[1] PSL, University of Paris-Dauphine, CNRS, LAMSADE UMR 7243, F-75775 Paris Cedex 16, France
{bazgan, michail.lampis, jerome.monnot, paschos}@lamsade.dauphine.fr
[2] School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia
ljiljana.brankovic@newcastle.edu.au
[3] Universität Trier, Fachbereich 4, Informatikwissenschaften, D-54286 Trier, Germany
{casel, fernau}@uni-trier.de
[4] Universität Kiel, Institut für Informatik D-24098 Kiel, Germany
{kj, kmk}@informatik.uni-kiel.de
[5] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067 Orléans, France
mathieu.liedloff@univ-orleans.fr

Abstract. This paper studies Upper Domination, *i.e.*, the problem of computing the maximum cardinality of a minimal dominating set in a graph, with a focus on parameterised complexity. Our main results include W[1]-hardness for Upper Domination, contrasting FPT membership for the parameterised dual Co-Upper Domination. The study of structural properties also yields some insight into Upper Total Domination. We further consider graphs of bounded degree and derive upper and lower bounds for kernelisation.

1 Introduction

Domination, independence and irredundance are basic concepts in graph theory and most of the overall six respective minimisation and maximisation problems, which are related via the so-called domination chain (see [15]), are very well-studied. Especially for parameterised complexity, Minimum Domination and Maximum Independent Set and their respective parameterised duals are sort of fundamental. With the exception of Upper Domination, all problems of the domination chain are known to be complete for either W[1] or W[2] while their corresponding parameterised dual is in FPT. This paper therefore studies the so far neglected parameter $\Gamma(G)$, which denotes the maximum cardinality of a minimal dominating set in G. More precisely, we discuss the following problems:

* Institut Universitaire de France

Upper Domination	Co-Upper Domination
Input: A graph $G = (V, E)$, a nonnegative integer k.	**Input:** A graph $G = (V, E)$, a nonnegative integer ℓ.
Question: Is $\Gamma(G) \geq k$?	**Question:** Is $\Gamma(G) \geq \|V\| - \ell$?

Notice that Co-Upper Domination could be also addressed as Minimum Maximal Nonblocker or as Minimum Maximal Star Forest; see [1] for further discussion. From the perspective of classical complexity theory, both problems are trivially equivalent and were shown to be NP-complete quite some time ago [7]. Aside from this, very little is known, especially with respect to parameterised complexity. From this perspective, k and ℓ turn out to be the natural parameters, which turn them into dual problems in the parameterised complexity sense of this word. As we will only consider this natural parameterisation, we refrain from explicitly mentioning the parameter throughout this paper. Slightly abusing notation, we will therefore use the names Upper Domination and Co-Upper Domination to also refer to the parameterised problems.

In the next section, we link minimal dominating sets to a decomposition of the vertex set that turns out to be a crucial tool for deriving our combinatorial and computational results. Section three then discusses properties of upper dominating sets from a parameterised point of view and reveals W[1]-hardness for Upper and Upper Total Domination. Conversely, Co-Upper Domination is shown to be in FPT, which we prove by providing both a kernelisation and a branching algorithm. In section four, we consider graphs of bounded degree and derive kernelisations for Upper and Co-Upper Domination for this restricted graph class. This section also includes an exact $O^*(1.3481^n)$-algorithm for subcubic graphs which builds on the decomposition derived in section two. We further discuss general questions of exact algorithms for Upper Domination, as well as some related questions for total domination variants (see [16]) in the last section. For reasons of space, proofs and other details were moved into an appendix to this extended abstract.

Basic notions. Throughout this paper, we only deal with undirected simple graphs $G = (V, E)$. The number of vertices $|V|$ is also known as the order of G. $N(v)$ denotes the open neighbourhood of v in a graph G, and $N[v]$ is the closed neighbourhood of v in G, *i.e.*, $N[v] = N(v) \cup \{v\}$. These notions can be easily extended to vertex sets X, *e.g.*, $N(X) = \bigcup_{x \in X} N(x)$. The cardinality of $N(v)$ is also known as the degree of v, denoted as $deg(v)$. The maximum degree in a graph is written as Δ. A graph of maximum degree three is called subcubic.

Given a graph $G = (V, E)$, a subset S of V is a *dominating set* if every vertex $v \in V \setminus S$ has at least one neighbour in S, *i.e.*, if $N[S] = V$. A dominating set is called *minimal* if no proper subset of it is a dominating set. Likewise, a vertex set I is *independent* if $N(I) \cap I = \emptyset$. An independent set is maximal if no proper superset is independent. In the following we use classical notations: $\alpha(G)$ denotes the cardinality of a maximum independent set in $G = (V, E)$ and $\tau(G) := |V| - \alpha(G)$ is the cardinality of a minimum vertex cover.

For any subset $S \subseteq V$ and $v \in S$ we define the private neighbourhood of v with respect to S as $pn(v, S) := N[v] - N[S - \{v\}]$. Any $w \in pn(v, S)$ is called a *private neighbour of* v *with respect to* S. If the set S is clear from the context, we will omit the "with respect to S" part. A dominating set $S \subseteq V$ is minimal if and only if $|pn(v, S)| > 0$ for every $v \in S$. Observe that v can be a private neighbour of itself.

Parameterised Complexity. We mainly refer to a recent textbook [9] in the area. Important notions that we will make use of include the parameterised complexity classes FPT, W[1] and W[2], parameterised reductions and kernelisation. In this area, it has also become customary not only to suppress constants (as in the O notation), but also even polynomial-factors, leading to the so-called O^*-notation.

2 Graph Decompositions for Minimal Dominating Sets

The following exposition is crucial for the development of the algorithms we derive in this paper and also for the general investigation of properties of minimal dominating sets. Any minimal dominating set D for a graph $G = (V, E)$ can be associated with a partition of the set of V into four sets F, I, P, O given by: $I := \{v \in D\colon v \in pn(v, D)\}$, $F := D \setminus I$, $P \in \{B \subseteq N(F) \cap (V \setminus D)\colon |pn(v, D) \cap B| = 1$ for all $v \in F\}$ and $O := V \setminus D \cup P)$. This representation is not necessarily unique since there might be different choices for the sets P and O, but for every partition of this kind, the following properties hold:

1. Every vertex $v \in F$ has at least one neighbour in F, called a **f**riend.
2. The set I is an **i**ndependent set in G.
3. The subgraph induced by the vertices $F \cup P$ has an edge cut set separating F and P that is, at the same time, a perfect matching; hence, P can serve as the set of **p**rivate neighbours for F.
4. The neighbourhood of a vertex in I is always a subset of O, which are otherwise the **o**utsiders.

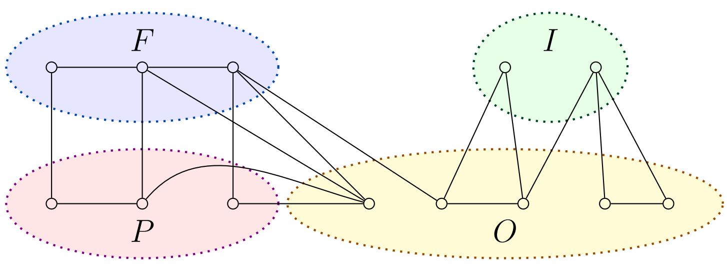
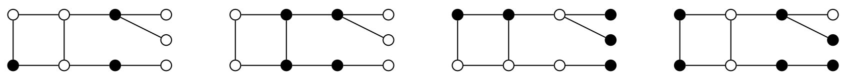


Fig. 1. Illustration of the FIPO structure imposed by minimal dominating sets

This partition is also related to a different characterisation of $\Gamma(G)$ in terms of so-called upper perfect neighbourhoods [15]. Observe two important special cases of the partition (F, I, P, O): If $F = \emptyset$, then I is an independent dominating set. If $I = \emptyset$, then F is a minimal total dominating set, *i.e.*, a set $S \subset V$ such that $V = N(S)$ and $N(S') \neq V$ for all $S' \subset S$. Both notions have been thoroughly studied in the literature. Observe that finding a maximum cardinality minimal dominating set for which $I = \emptyset$ holds in an (F, I, P, O) partitioning (called (F, P, O)-Domination set in the following) is not equivalent to the problem Upper Total Domination, which asks for a maximum cardinality minimal total dominating set. The following example illustrates the differences between optimal solutions (illustrated by the black vertices) for Minimum, (F, P, O)-, Upper and Upper Total Domination:



min DS $\quad$ (F, P, O) DS $\quad$ upper DS $\quad$ upper total DS

From the domination chain we know $\alpha(G) \leq \Gamma(G)$ for all graphs G, which is simply due to the fact that any maximal independent set is also a minimal dominating set. Considering the partition (F, I, P, O) for a minimal dominating set S for a graph G of order $n > 0$, we immediately know that $|I| \leq \alpha(G)$. Further, we know $|F| = |P|$ and hence $|F| = 1/2(n - |I| - |O|) \leq 1/2(n - \alpha(G))$. With $|S| = |F| + |I|$, we see that $|S| \leq 1/2(n + \alpha(G))$ and since this inequality holds for all mimimal dominating sets S, we can conclude:

$$\alpha(G) \;\leq\; \Gamma(G) \;\leq\; \frac{n}{2} + \frac{\alpha(G)}{2} \tag{1}$$

3 Fixed Parameter Tractability

In this section we will investigate the fixed parameter tractability of Upper Domination, its dual and related problems. The problems Minimum Domination, Minimum Independent Domination and Maximum Independent Set were among the first problems conjectured not to be in FPT [8]. In fact, aside from Upper Domination, all other problems from the domination chain are now known to be complete for either W[1] or W[2] (see [2] and [10] for upper and lower irredundance respectively). It is perhaps not very surprising that Upper Domination is also unlikely to belong to FPT, and it looks rather unexpected that this question has been open for such a long time. We show that Upper Domination is W[1]-hard by a reduction from Multicoloured Clique, a problem introduced in [12] to facilitate W[1]-hardness proofs. While the construction used in our reduction itself is not very complicated, proving its correctness turns out to be quite complex and technical.

Theorem 1. Upper Domination *is W[1]-hard.*

Proof. (Sketch) Let $G = (V, E)$ be a graph with k different colour-classes given by $V = V_1 \cup V_2 \cup \dots \cup V_k$. Multicoloured Clique asks if there exists a clique $C \subseteq V$ in G such that $|V_i \cap C| = 1$ for all $i = 1, \dots, k$. For this problem, one can assume that each set V_i is an independent set in G, since edges between vertices of the same colour-class have no impact on the existence of a solution. Multicoloured Clique is known to be W[1]-complete, parameterised by k. We construct a graph $G' = (V', E')$ by: $V' := V \cup \{v_e : e \in E\}$ and

$$E' := \bigcup_{i=1}^{k} V_i \times V_i \quad \cup \quad \bigcup_{i=1}^{k} \bigcup_{j=1}^{k} \{(v_e, v_{e'}) : e, e' \in (V_i \times V_j) \cap E\}$$
$$\cup \bigcup_{i=1}^{k} \bigcup_{j=1}^{k} \left\{(v_{(u,w)}, x) : (u, w) \in (V_i \times V_j) \cap E, x \in ((V_i \cup V_j) - \{u, w\})\right\} .$$

It can be shown that there exists a minimal dominating set S of cardinality $k + \frac{1}{2}(k^2 - k)$ for G' if and only if $|S \cap V_i| = 1$ for all $i = 1, \dots, k$ and $|S \cap V_{i,j}| = 1$ for all $i \neq j$, where $V_{i,j} := \{v_e : e \in E \cap (V_i \times V_j)\}$. With this property, it is easy to see that S is minimal if and only if $S \cap V$ is a clique in the original graph; observe that if S contains two vertices v_i and v_j from V_i and V_j, respectively, which are not adjacent in G, then these already dominate all vertices of $V_{i,j}$ in G'. Overall, it can be shown that G' has an upper dominating set of cardinality $k + \frac{1}{2}(k^2 - k)$ if and only if G is a "yes"-instance for Multicoloured Clique, which proves W[1]-hardness for Upper Domination, parameterised by $\Gamma(G')$. □

We want to point out that the above reduction also works for the restriction of Upper Domination to solutions for which I is empty:

Corollary 1. (F, P, O)-Domination*, that is the restriction of* Upper Domination *to solutions S such that $V = N(S)$, is W[1]-hard.*

This result means that if we consider somehow splitting the problem Upper Domination into the subproblems of computing the independent vertices I and (F, P, O)-Domination, we end up with two W[1]-hard problems. Considering Upper Total Domination, the construction in the proof of Theorem 1 is not very helpful, since unfortunately any set S with $|S \cap V_i| = 1$ for all $i = 1, \dots, k$ and $|S \cap V_{i,j}| = 1$ for all $i \neq j$, regardless of the structure of the original graph G, is a minimal total dominating set for G'. We can however use a much simpler construction to show W[1]-hardness for Upper Total Domination, a result which cannot be inferred from the known NP-hardness of the problem, see [11].

Theorem 2. Upper Total Domination *is W[1]-hard.*

Proof. (Sketch) We reduce from Multicoloured Independent Set. Let $G = (V, E)$ be a graph with k different colour-classes given by $V = V_1 \cup V_2 \cup \dots \cup V_k$. We construct a graph $G' = (V', E')$ as follows: Starting from G, we add k vertices $C = \{c_1, \dots, c_k\}$ and turn each vertex set $V_j \cup \{c_j\}$ into a clique. We claim that G admits a multicoloured independent set (of size k) if and only if G' has a minimal total dominating set with $2k$ vertices. □

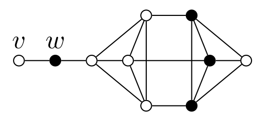
We do not know if Upper Domination belongs to W[1], but we can at least place it in W[2], the next level of the W hierarchy. We obtain this result by describing a suitable multi-tape Turing machine that solves this problem, see [4].

Proposition 1. Upper Domination *belongs to W[2].*

Proof. Recall how Minimum Domination can be seen to belong to W[2] by providing an appropriate multi-tape Turing machine [4]. First, the k vertices that should belong to the dominating set are guessed, and then this guess is verified in k further (deterministic) steps using n further tapes in parallel, where n is the order of the input graph. We only need to make sure that the guessed set of vertices is minimal. To this end, we copy the guessed vertices k times, leaving one out each time, and we also guess one vertex for each of the $k-1$-element sets that is not dominated by this set. Such a guess can be tested in the same way as sketched before using parallel access to the $n+1$ tapes. The whole computation takes $O(k^2)$ parallel steps of the Turing machine, which shows the claim. □

Let us notice that very similar proofs also show membership in W[2] and hardness for W[1] for the question whether, given some hypergraph G and parameter k, there exists a minimal hitting set of G with at least k vertices. This also means that Upper Total Domination belongs to W[2].

In the context of parameterised complexity, would like to point out another difference between Upper Domination and Minimum Domination. Despite its W[2]-hardness, there is at least a reduction-rule for Minimum Domination, which deals with vertices of degree one, as they can be assumed not to be contained in a minimum dominating set. One might suspect that any upper dominating set would conversely always choose to contain degree-one vertices. As the example on the right illustrates, there can not be such a rule for Upper Domination, since the degree-one vertex v is never part of a maximum solution; in fact, the black vertices form the unique optimal solution for this graph.



Another interesting question is to consider the dual parameter ℓ, that is to decide the existence of an upper dominating set of size at least $n-\ell$. This is in fact the natural parameterisation for Co-Upper Domination.

Theorem 3. Co-Upper Domination *is in FPT. More precisely, it admits a kernel of at most $\ell^2+\ell$ many vertices and at most ℓ^2 many edges.*

Proof. Let $G=(V,E)$ be an input graph of order n. Consider a vertex $v \in V$ with $deg(v) > \ell$ and any minimal dominating set D with partition (F,I,P,O):

- If $v \in I$, all neighbours of v have to be in O which means $|O| \geq |N(v)| > \ell$.
- If $v \in F$, exactly one neighbour p of v is in P and $N[v]-\{p\} \subseteq F \cup O$, which gives $|O|+|P| = |O|+|F| \geq |N[v]-\{p\}| > \ell$.
- If $v \in P$, exactly one neighbour p of v is in F and $N[v]-\{p\} \subseteq P \cup O$, so $|O|+|P| > \ell$.

We always have either $v \in O$ or $|O| + |P| > \ell$, which means a "no"-instance for Co-Upper Domination. Consider the graph G' built from G by deleting the vertex v and all its edges. For any minimal dominating set D for G with partition (F, I, P, O) such that $v \in O$, D is also minimal for G', since $pn(w, D) \supseteq \{w\}$ for all $w \in I$ and $|pn(u, D) \cap P| = 1$ for all $u \in F$. Also, any set $D' \subset V - \{v\}$ which does not dominate v has a cardinality of at most $|V - N[v]| < n - \ell$, so if G' has a dominating set D' of cardinality at least $n - \ell$, $N(v) \cap D' \neq \emptyset$; hence, D' is also dominating for G. These observations allow us to successively reduce (G, ℓ) to (G', ℓ') with $\ell' = \ell - 1$, as long as there are vertices v with $deg(v) > \ell$. Any isolated vertex in the resulting graph G' originally only has neighbours in O which means it belongs to I in any dominating set D with partition (F, I, P, O) and can hence be deleted from G' without affecting the existence of an upper dominating set with $|P| + |O| \leq \ell'$.

Let (G', ℓ') be the instance obtained after the reduction above with $G' = (V', E')$ and let $n' = |V'|$. If there is an upper dominating set D for G' with $|D| \geq n' - \ell'$, any associated partition (F, I, P, O) for D satisfies $|P| + |O| \leq \ell'$. Since G' does not contain isolated vertices, every vertex in I has at least one neighbour in O. Also, any vertex in V', and hence especially any vertex in O, has degree at most ℓ', which means that $|I| \leq |N(O)| \leq \ell'|O|$. Overall:

$$|V'| \leq |I| + |F| + |P| + |O| \leq (\ell' + 1)|O| + 2|P| \leq \max_{j=0}^{\ell'}\{j(\ell' + 1), 2(\ell' - j)\},$$

and hence $|V'| \leq \ell'(\ell' + 1)$, or (G', ℓ') and consequently (G, ℓ) is a "no"-instance. Concerning the number of edges, we can derive a similar estimate. There are at most ℓ edges incident with each vertex in O. In addition, there is exactly one edge incident with each vertex in P that has not yet been accounted for, and, in addition, there could be $\ell - 1$ edges incident to each vertex in F that have not yet been counted. This shows the claim. □

We just derived a kernel result for Co-Upper Domination, in fact a kernel of quadratic size in terms of the number of vertices and edges. This poses the natural question if we can do better also with respect to the question weather the brute-force search we could perform on the quadratic kernel is the best we can do to solve Co-Upper Domination in FPT time.

Proposition 2. Co-Upper Domination *can be solved in time* $O^*(4.3077^\ell)$.

Proof. (Sketch) This result can be shown by designing a branching algorithm that takes a graph $G = (V, E)$ and a parameter ℓ as input. A complete description of the algorithm, as well as its correctness and running time analysis are given in the appendix. Due to space restriction, we only describe here the rough ideas without any proof. As in Section 2, to each graph $G = (V, E)$ and (partial) dominating set, we associate a partition (F, I, P, O). We consider $\kappa = \ell - (\frac{|F|}{2} + \frac{|P|}{2} + |O|)$ as a measure of the partition and for the running time of the algorithm. Note that $\kappa \leq \ell$. At each branching step, our algorithm picks some vertices from R (the set of yet undecided **r**emaining vertices). They are either added to the current dominating set $D := F \cup I$ or to $\overline{D} := P \cup O$. Each time a vertex is added to P (resp. to O) the value of κ decreases by $\frac{1}{2}$ (resp. by 1). Also, whenever a vertex x is added to F, the value of κ decreases by $\frac{1}{2}$.

Let us describe the two halting rules. First, whenever κ reaches zero, we are facing a "no"-instance. Then, if the set R of undecided vertices is empty, we check whether the current domination set D is minimal and of size at least $n - \ell$, and if so, the instance is a "yes"-instance. Then, we have a simple reduction rule: whenever the neighbourhood of a undecided vertex $v \in R$ is included in $\overline{D}$, we can safely add v to I. Finally, vertices are placed to F, I or $\overline{D}$ according to three branching rules. The first one considers undecided vertices with a neighbour already in F (in such a case, v cannot belongs to I). The second one considers undecided vertices with only one undecided neighbour (in such a case, several cases may be discarded as, *e.g.*, they cannot be both in I or both in $\overline{D}$). The third branching rule considers all the possibilities for an undecided vertex and due to the previous branching rules, it can be assumed that each undecided vertex has at least two undecided neighbours (which is nice since such vertices have to belong to $\overline{D}$ whenever an undecided neighbour is added to I). □

Of course, the question remains to what extent the previously presented parameterised algorithm can be improved on. In this context, we briefly discuss the issue of (parameterised) approximation for this parameter.

Theorem 4. Co-Upper Domination *is 4-approximable in polynomial time, 3-approximable with a running time in $O^*(1.0883^{\tau(G)})$ and 2-approximable in time $O^*(1.2738^{\tau(G)})$ or $O^*(1.2132^n)$.*

Proof. First of all, observe by subtracting n from Eq. (1) that $\tau(G)$ relates to the co-upper domination number in the following way:

$$\frac{\tau(G)}{2} + 1 \leq n - \Gamma(G) \leq \tau(G) \tag{2}$$

Using any 2-approximation algorithm one can compute a vertex cover V' for G, and define $S' = V \setminus V'$. Let S be a maximal independent set containing S'. $V \setminus S$ is a vertex cover of size $|V \setminus S| \leq |V'| \leq 2\tau(G) \leq 4(n - \Gamma(G))$. Moreover, S is maximal independent and hence minimal dominating set which makes $V \setminus S$ a feasible solution for Co-Upper Domination with $|V \setminus S| \leq 4(n - \Gamma(G))$. The claimed running time for the factor-2 approximation stems from the best parameterised and exact algorithms Minimum Vertex Cover by [6] and [18], the factor-3 approxmation from the parameterised approximation in [3]. □

4 Graphs of bounded degree

In contrast to the case of general graphs, Upper Domination turns out to be easy (in the sense of paramterised complexity) for graphs of bounded degree.

Proposition 3. *Fix $\Delta > 2$.* Upper Domination *has a problem kernel with at most Δk many vertices.*

Proof. First, we can assume that the input graph G is connected, as otherwise we can apply the following argument separately on each connected component. Assume G is a cycle or a clique. Then, the problem Upper Domination can be optimally solved in polynomial time, *i.e.*, we can produce a kernel as small as we want. Otherwise, Brooks' Theorem yields a polynomial-time algorithm that produces a proper colouring of G with (at most) Δ many colours. Extend the biggest colour class to a maximal independent set I of G. As I is maximal, it is also a minimal dominating set. So, there is a minimal dominating set I of size at least n/Δ, where n is the order of G. So, $\Gamma(G) \geq n/\Delta$. If $k < n/\Delta$, we can therefore immediately answer YES. In the other case, $n \leq \Delta k$ as claimed. □

With some more combinatorial effort, we obtain:

Proposition 4. *Fix $\Delta > 2$.* Co-Upper Domination *has a problem kernel with at most $(\Delta + 0.5)\ell$ many vertices.*

Proof. Consider any graph $G = (V, E)$. For any partition (F, I, P, O) corresponding to an upper dominating set $D = I \cup F$ for G, isolated vertices in G always belong to I and can hence be deleted in any instance of Co-Upper Domination without changing ℓ. For any graph G without isolated vertices, the set $P \cup O$ is a dominating set for G, since $\emptyset \neq N(v) \subset O$ for all $v \in I$ and $N(v) \cap P \neq \emptyset$ for all $v \in F$. Maximum degree Δ hence immediately implies $n = |N[P \cup O]| \leq (\Delta + 1)\ell$.

Since any connected component can be solved separately, we can assume that G is connected. For any $v \in P$, the structure of the partition (F, I, P, O) yields $|N[v] \cap D| = 1$, so either $|N[v]| = 1 < \Delta$ or there is at least one $w \in P \cup O$ such that $N[v] \cap N[w] \neq \emptyset$. For any $v \in O$, if $N[v] \cap F \neq \emptyset$, the F-vertex in this intersection has a neighbour $w \in P$, which means $N[w] \cap N[v] \neq \emptyset$. If $N[v] \subset I$ and $N[v] \neq V$, at least one of the I-vertices in $N[v]$ has to have another neighbour to connect to the rest of the graph. Since $N[I] \subset O$, this also implies the existence of a vertex $w \in O$, $w \neq v$ with $N[w] \cap N[v] \neq \emptyset$. Finally, if $N[v] \not\subset I \cup F$, there is obviously a $w \in P \cup O$, $w \neq v$ with $N[w] \cap N[v] \neq \emptyset$.

Assume that there is an upper dominating set with partition (F, I, P, O) such that $|P \cup O| = l \leq \ell$ and let $v_1, \ldots, v_l$ be the $l > 1$ vertices in $P \cup O$. By the above argued domination-property of $P \cup O$, we have:

$$n = |\bigcup_{i=1}^{l} N[v_i]| = \tfrac{1}{2} \sum_{i=1}^{l} |N[v_i] \setminus \bigcup_{j=1}^{i-1} N[v_j]| + \tfrac{1}{2} \sum_{i=1}^{l} |N[v_i] \setminus \bigcup_{j=i+1}^{l} N[v_j]|$$

Further, by the above argument about neighbourhoods of vertices in $P \cup O$, maximum degree Δ yields for every $i \in \{1, \ldots, l\}$ either $|N[v_i] \setminus \bigcup_{j=1}^{i-1} N[v_j]| \leq \Delta$ or $|N[v_i] \setminus \bigcup_{j=i+1}^{l} N[v_j]| \leq \Delta$ which gives:

$$n = \tfrac{1}{2} \sum_{i=1}^{l} |N[v_i] \setminus \bigcup_{j=1}^{i-1} N[v_j]| + |N[v_i] \setminus \bigcup_{j=i+1}^{l} N[v_j]| \leq \tfrac{1}{2} l(2\Delta + 1) \leq (\Delta + 0.5)\ell.$$

Any graph with more than $(\Delta + 0.5)\ell$ vertices is consequently a "no"-instance which yields the stated kernelisation, as the excluded case $|P \cup O| = 1$ (or in other words $N[v] = V$ for some $v \in O$) can be solved trivially. □

This implies that we have a $3k$-size vertex kernel for UPPER DOMINATION, restricted to subcubic graphs, and a 3.5ℓ-size vertex kernel for CO-UPPER DOMINATION, again restricted to subcubic graphs. With [5, Theorem 3.1], we can conclude the following consequence:

Corollary 2. *Unless P equals NP, for any $\varepsilon > 0$,* UPPER DOMINATION*, restricted to subcubic graphs, does not admit a kernel with less than $(1.4 - \varepsilon)k$ vertices; neither does* CO-UPPER DOMINATION*, restricted to subcubic graphs, admit a kernel with less than $(1.5 - \varepsilon)\ell$ vertices.*

Exact Algorithms

Let us recall one important result on the pathwidth of subcubic graphs from [14].

Theorem 5. *Let $\epsilon > 0$ be given. For any subcubic graph G of order $n > n_\epsilon$, a path decomposition proving $pw(G) \leq n/6 + \epsilon$ is computable in polynomial time.*

This result immediately gives an $O^*(1.2010^n)$-algorithm for solving MINIMUM DOMINATION on subcubic graphs. We will take a similar route to prove moderately exponential-time algorithms for UPPER DOMINATION.

Proposition 5. UPPER DOMINATION *on graphs of pathwidth p can be solved in time $O^*(7^p)$, given a corresponding path decomposition.*

We are considering all partitions of each bag of the path decomposition into 6 sets: F, F^*, I, P, O, O^*, where

- F is the set of vertices that belong to the upper dominating set and have already been matched to a private neighbour;
- F^* is the set of vertices that belong to the upper dominating set and still need to be matched to a private neighbour;
- I is the set of vertices that belong to the upper dominating set and is independent in the graph induced by the upper dominating set;
- P is the set of private neighbours that are already matched to vertices in the upper dominating set;
- O is the set of vertices that are not belonging neither to the upper dominating set nor to the set of private neighbours but are already dominated;
- O^* is the set of vertices not belonging to the upper dominating set that have not been dominated yet.

The exact recursions can be found in the appendix. Observe that the upper bound on the running time can be improved for graphs of a certain maximum degree to $O^*(6^p)$, so that we can conclude:

Corollary 3. UPPER DOMINATION *on subcubic graphs of order n can be solved in time $O^*(1.3481^n)$, using the same amount of space.*

We like to point out that the idea from the pathwidth algorithm above can be adapted to work for treewidth.

Proposition 6. UPPER DOMINATION *on graphs of treewidth p can be solved in time $O^*(11^p)$, given a corresponding nice tree decomposition.*

5 Discussions and Open Problems

The motivation to study Upper Domination (at least for some in the group of authors) was based on the following observation:

Proposition 7. Upper Domination *can be solved in time* $O^*(1.7159^n)$ *on general graphs of order* n.

Proof. The suggested algorithm simply lists all minimal dominating sets and then picks the biggest one. It has been shown in [13] that this enumeration problem can be performed in the claimed running time. □

It is of course a bit nagging that there seems to be no better algorithm (analysis) than this enumeration algorithm for Upper Domination. Recall that the minimisation counterpart can be solved in better than $O^*(1.5^n)$ time [17,19]. As this appears to be quite a tough problem, it makes a lot of sense to study it on restricted graph classes. This is what we did above for subcubic graphs, see Corollary 3. We summarise some open problems.

- Is Upper Domination in W[1]? Or, hard for W[2]?
- Can we improve on the 4-approximation of Co-Upper Domination?
- Can we find smaller kernels for Upper or Co-Upper Domination on degree-bounded graphs?
- Can we find exact (*e.g.,* branching) algorithms that beat the enumeration or pathwidth-based ones for Upper Domination, at least on cubic graphs?

Also for Upper Total Domination, the best exact algorithm seems to be based on enumeration. Recall that Fomin *et al.* establish that a graph with n vertices has at most 1.7159^n minimal dominating sets [13]. To achieve this result, they first design a branching algorithm to enumerate all minimal set covers of an instance $(\mathcal{U}, \mathcal{S})$, where $\mathcal{S}$ is a collection of subsets over a universe $\mathcal{U}$ and then use a simple reduction from a dominating set instance to a set cover instance. It is implicit from their analysis (see Section 4 of [13]) that a Set Cover instance has at most $1.156154^{|\mathcal{U}|+2.720886|\mathcal{S}|}$ minimal set covers which can be enumerated in time $O^*(1.156154^{|\mathcal{U}|+2.720886|\mathcal{S}|})$. As a easy consequence, minimal total dominating sets of a graph $G = (V, E)$ can be enumerated in time $O^*(1.7159^n)$, by picking as the universe $\mathcal{U} = V$ and $\mathcal{S} = \{N(v) : v \in V\}$. This allows us to conclude that Upper Total Domination can be solved in the same time. Similarities to Upper Domination continue to some extent; however, the general picture is not very clear and still needs some research.

Acknowledgements. We would like to thank our colleagues Serge Gaspers, David Manlove and Daniel Meister for some discussions on (total) upper domination. Part of this research was supported by Deutsche Forschungsgemeinschaft, grant FE 560/6-1.

References

1. F. N. Abu-Khzam, C. Bazgan, M. Chopin, and H. Fernau. Data reductions and combinatorial bounds for improved approximation algorithms. *Journal of Computer and System Sciences*, 82(3):503–520, 2016.
2. D. Binkele-Raible, L. Brankovic, M. Cygan, H. Fernau, J. Kneis, D. Kratsch, A. Langer, M. Liedloff, M. Pilipczuk, P. Rossmanith, and J. O. Wojtaszczyk. Breaking the 2^n-barrier for IRREDUNDANCE: Two lines of attack. *Journal of Discrete Algorithms*, 9:214–230, 2011.
3. L. Brankovic and H. Fernau. A novel parameterised approximation algorithm for MINIMUM VERTEX COVER. *Theoretical Computer Science*, 511:85–108, 2013.
4. M. Cesati. The Turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67:654–685, 2003.
5. J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: lower bounds and upper bounds on kernel size. *SIAM Journal on Computing*, 37:1077–1108, 2007.
6. J. Chen, I. A. Kanj, and G. Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40–42):3736–3756, 2010.
7. G. A. Cheston, G. Fricke, S. T. Hedetniemi, and D. P. Jacobs. On the computational complexity of upper fractional domination. *Discrete Applied Mathematics*, 27(3):195–207, 1990.
8. R. G. Downey and M. R. Fellows. Fixed parameter tractability and completeness. *Congressus Numerantium*, 87:161–187, 1992.
9. R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
10. R. G. Downey, M. R. Fellows, and V. Raman. The complexity of irredundant set parameterized by size. *Discrete Applied Mathematics*, 100:155–167, 2000.
11. Q. Fang. On the computational complexity of upper total domination. *Discrete Applied Mathematics*, 136(1):13–22, 2004.
12. M. R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.
13. F. V. Fomin, F. Grandoni, A. V. Pyatkin, and A. A. Stepanov. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Trans. Algorithms*, 5(1):1–17, 2008.
14. F. V. Fomin and K. Høie. Pathwidth of cubic graphs and exact algorithms. *Information Processing Letters*, 97:191–196, 2006.
15. T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs*, volume 208 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, 1998.
16. M. Hennings and A. Yeo. *Total Domination in Graphs*. Springer, 2013.
17. Y. Iwata. A faster algorithm for dominating set analyzed by the potential method. In D. Marx and P. Rossmanith, editors, *Parameterized and Exact Computation - 6th International Symposium, IPEC 2011*, volume 7112 of *LNCS*, pages 41–54. Springer, 2012.
18. J. Kneis, A. Langer, and P. Rossmanith. A fine-grained analysis of a simple independent set algorithm. In R. Kannan and K. Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, volume 4 of *LIPIcs*, pages 287–298. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2009.
19. J. M. M. van Rooij and H. L. Bodlaender. Exact algorithms for dominating set. *Discrete Applied Mathematics*, 159(17):2147–2164, 2011.

6 Appendix: Omitted Proofs

In this section, we collect proofs that have been omitted or considerably shortened in the main text body.

6.1 Proof of Theorem 1

Let $G = (V, E)$ be a graph with k different colour-classes given by $V = V_1 \cup V_2 \cup \cdots \cup V_k$. Multicoloured Clique asks if there exists a clique $C \subseteq V$ in G such that $|V_i \cap C| = 1$ for all $i = 1, \ldots, k$. For this problem, one can assume that each set V_i is an independent set in G, since edges between vertices of the same colour-class have no impact on the existence of a solution. Multicoloured Clique is known to be W[1]-complete, parameterised by k. We construct a graph G' such that G' has an upper dominating set of cardinality (at least) $k + \frac{1}{2}(k^2 - k)$ if and only if G is a "yes"-instance for Multicoloured Clique which proves W[1]-hardness for Upper Domination, parameterised by $\Gamma(G')$.

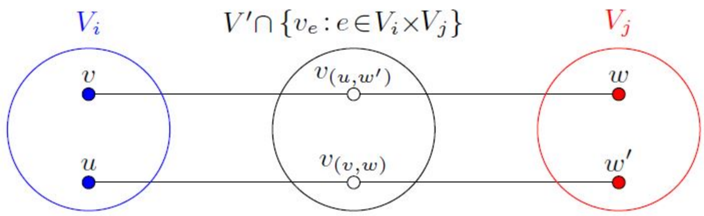
Consider $G' = (V', E')$ given by: $V' := V \cup \{v_e : e \in E\}$ and

$$
\begin{aligned}
E' := & \bigcup_{i=1}^{k} V_i \times V_i \quad \cup \quad \bigcup_{i=1}^{k} \bigcup_{j=1}^{k} \{(v_e, v_{e'}) : e, e' \in (V_i \times V_j) \cap E\} \\
& \cup \bigcup_{i=1}^{k} \bigcup_{j=1}^{k} \left\{(v_{(u,w)}, x) : (u, w) \in (V_i \times V_j) \cap E, x \in ((V_i \cup V_j) - \{u, w\})\right\} .
\end{aligned}
$$

If $C \subset V$ is a (multi-coloured) clique of cardinality k in G, the set $S' := C \cup \{v_{(u,v)} : u, v \in C\}$ is an upper dominating set for G' of cardinality $k + \frac{1}{2}(k^2 - k)$: First of all, $\{v_{(u,v)} : u, v \in C\} \subset V'$ since $(u, v) \in E$ for all $u, v \in C$. Further, by definition of the edges E', $u, v \notin N_{G'}(v_{(u,v)})$ and $u \notin N_{G'}(v)$ for u and v from different colour classes so S' is an independent set in G' and hence a minimal dominating set. It can be easily verified that S' is also dominating for G' – observe that it contains exactly one vertex for each clique in the graph.

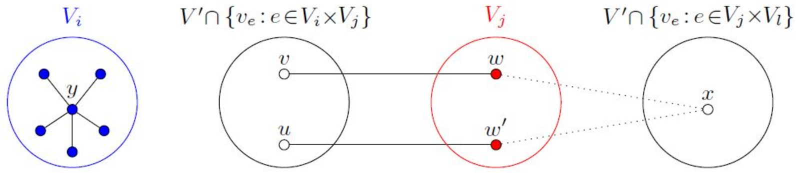
Suppose S is a minimal dominating set for G'. Consider the partition $S = \left(\bigcup_{i=1}^{k} S_i\right) \cup \left(\bigcup_{1 \leq i < j \leq k} S_{\{i,j\}}\right)$ defined by: $S_i := S \cap V_i$ for $i = 1, \ldots, k$ and $S_{\{i,j\}} := S \cap \{v_e : e \in V_i \times V_j\}$ for all $1 \leq i < j \leq k$. The minimality of S gives the following properties for these subsets of S:

1. If $|S_i| > 1$ for some index $i \in \{1, \ldots, k\}$, minimality implies $|S_i| = 2$ and for all $j \neq i$ either $S_{\{i,j\}} = \emptyset$ or $S_j = \emptyset$:



Since for every $u \in V_i$ and every j, $j \neq i$, by construction $V_i \subset N[u]$, and if there is more than one vertex in S_i, then their private neighbours have to be in $\{v_e \colon e \in E\}$. A vertex v_e with $e \in V_i \times V_j$ is not adjacent to a vertex $u \in V_i$ if and only if $e = (u, w)$ for some $w \in V_j$. For two different vertices $u, v \in V_i$ consequently all v_e with $e \in V_i \times V_j$ are adjacent to either u or v, a third vertex $w \in V_i$ consequently can not have any private neighbour. This also means that any vertex $v_e \in S_{\{i,j\}}$ has to have a private neighbour in V_j, so if $S_{\{i,j\}} \neq \emptyset$ the set S_j has to be empty because one vertex from S_j dominates all vertices in V_j. These observations hold for all $j \neq i$.

2. If $|S_{\{i,j\}}| > 1$ for some indices $i, j \in \{1, \ldots, k\}$ we find that $|S_{\{i,j\}}| = 2$, $|S_i|, |S_j| \leq 1$ and that $S_i \neq \emptyset$ implies $S_j = S_{\{j,l\}} = \emptyset$ for all $l \in \{1, \ldots, k\} - \{i, j\}$ (and equivalently $S_j \neq \emptyset$ implies $S_i = S_{\{i,l\}} = \emptyset$ for all $l \in \{1, \ldots, k\} - \{i, j\}$):



Since for any two vertices u, v from $S_{\{i,j\}}$ we have $\{v_e \colon e \in (V_i \times V_j) \cap E\} \cup V_i \cup V_j \subset N(u) \cup N(v)$, the cardinality of $S_{\{i,j\}}$ can be at most two. If there is a vertex y in S_i, it already dominates all of V_i so private neighbours for $u, v \in S_{\{i,j\}}$ have to be in S_j. For any two vertices $w, w' \in V_j$ any $v_e \in V' \cap \{v_e \colon e \in V_j \times V_l,\ l = 1, \ldots, k\}$ is either adjacent to at least w or w', so especially for the private vertices of u and v every $x \in S_{j,l}$ would be adjacent to one of them and can consequently not be in a minimal dominating set, so $S_j = S_{\{j,l\}} = \emptyset$. Dominating the vertices in $S_{\{j,l\}}$ for $l \neq i$ then requires $|S_l| = 2$ for all $l \neq i$, which leaves no possible private vertices outside V_i for vertices in V_i, so $|S_i| = 1$.

3. If $|S_i| = 2$ there exists an index $j \neq i$ such that $S_{\{i,j\}} = \emptyset$ and $|S_j| \leq 1$:
Let $u, v \in S_i$. By the structure of G', u and v share all neighbours in V_i and v_e such that $e = (x, y) \in V_i \times V_l$ with $x \notin \{u, v\}$ for all $l \neq i$, so especially the private neighbourhood of u is restricted to $pn(u, S) \subseteq \{v_e \colon e = (v, y) \in E\}$. Let j be an index such that there is a vertex $z \in V_j$ with $v_{(u,z)} \in pn(v, S)$ (there is at least one such index). No neighbour of $v_{(u,z)}$ beside v can be in S, which means that $S_{\{i,j\}} = \emptyset$ and $S_j \subseteq \{z\}$.

4. $|S_{\{i,l\}}| = 2$ implies $|S_{\{j,l\}}| \leq 1$ for all $j \neq i$.
Suppose $|S_{\{i,l\}}|, |S_{\{j,l\}}| \geq 2$ for some indices $i, j, l \in \{1, \ldots, k\}$. By property 2 both sets $S_{\{i,l\}}, S_{\{j,l\}}$ have cardinality two so let $u_i, w_i \in S_{\{i,l\}}$ and $u_j, w_j \in S_{\{j,l\}}$. Since each set $\{v_e \colon e \in E \cap (V_s \times V_t)\}$ is a clique, the private neighbours for these vertices have to be in V_i, V_j, V_l. Suppose $v \in pn(u_i, S) \cap V_l$ which means that w_i, u_j, w_j are not adjacent to v. This is only possible if w_i represents some edge $(v, x) \in E \cap V_l \times V_i$ and u_j, w_j represent some edges $(v, y), (v, y') \in E \cap V_l \times V_j$. By definition of E', w_i, u_j, w_j

then share their neighbourhood in V_l (namely $V_l - \{v\}$) which means that $pn(w_i, S) \subset V_i$ and $pn(u_j) \cup pn(w_j) \subset V_j$ which implies $S_i = S_j = \emptyset$. So in any case, even if there is no $v \in pn(u_i, S) \cap V_l$, at least one of the sets V_i or V_j contains two vertices which are private neighbours for $S_{\{i,j\}}$ and $S_i = S_j = \emptyset$.
Suppose V_j contains two private vertices $y \neq y'$ for u_j and w_j respectively. For any two arbitrary vertices $n_1, n_2 \in V_j$, any vertex $x \in \{v_e \colon e \in E \cap (V_i \times V_j)\}$ is adjacent to at least one of them, which means that any $x \in S_{\{i,j\}}$ would steal at least $y \in pn(u_j)$ or $y' \in pn(w_j)$ as private neighbour. Minimality of S hence demands $S_i = S_j = S_{\{i,j\}} = \emptyset$. A set with this property however does not dominate any of the vertices v_e with $e \in E \cap (V_i \times V_j)$. (The set $E \cap (V_i \times V_j)$ is not empty unless the graph G is a trivial "no"-instance for Multicoloured Clique.)

According to these properties, the indices of these subsets of S can be divided into the following six sets: $C_i := \{j \colon |S_j| = i\}$ and $D_i := \{(j,l) \colon |S_{\{j,l\}}| = i\}$ for $i = 0, 1, 2$ which then give $|S| = 2(|C_2| + |D_2|) + |C_1| + |D_1|$. If $|C_2| + |D_2| \neq 0$ and $k > 3$, we can construct an injective mapping $f \colon C_2 \cup D_2 \cup \{a\} \to C_0 \cup D_0$ with some $a \notin V'$ in the following way:

- For every $i \in C_2$ choose some $j \neq i$ with $(i,j) \in D_0$ and $j \notin C_2$ which exists according to property 3 and set $f(i) = (i,j)$. Since $j \notin C_2$ this setting is injective.
  If $D_2 = \emptyset$ and $C_2 = \{i\}$, choose some $l \neq i$ and map a via f either to l or to (i,l), since, by property 1, one of them is in C_0 or D_0 respectively. If $D_2 = \emptyset$ and $|C_2| > 1$, choose some $i, l \in C_2$ and set $f(a) = (i,l)$ since $S_{\{i,l\}} = \emptyset$ by property 1 and neither i nor l is mapped to (i,l).
- For $(i,j) \in D_2$, property 2 implies at least i or j lies in C_0. By Property 4 we can choose one of them arbitrarily without violating injectivity. If both are in C_0 we can use one of them to map a. If for all $(i,j) \in D_2$ only one of the indices i, j is in C_0, we still have to map a, unless $f(a)$ has been already defined. Assume for $(i,j) \in D_2$ that $i \notin C_0$. By property 2 $\{(j,l) \colon l \notin \{i,j\}\} \subset D_0$. If we cannot choose one of these index-pairs as injective image for a, they have all been used to map C_2 which means $\{1, \ldots, k\} - \{i,j\} \subseteq C_2$ and hence, by property 1, all index-pairs (l,h) with $l, h \in \{1, \ldots, k\} - \{i,j\}$ are in D_0 and so far not in the image of f, so we are free to chose one of them as image of a, unless $f(a)$ has been already defined.

This injection proves that $|C_2| + |D_2| > 0$ implies that $|C_2| + |D_2| < |C_0| + |D_0|$. This means that, regardless of the structure of the original graph G, the subsets S_i and $S_{i,j}$ of S either all contain exactly one vertex or $k + \frac{1}{2}(k^2 - k) = |C_1| + |D_1| + |C_0| + |D_0| + |C_2| + |D_2| > |C_1| + |D_1| + 2(|C_2| + |D_2|) = |S|$.

So if $|S| = k + \frac{1}{2}(k^2 - k)$, the above partition into the sets $S_i, S_{i,j}$ satisfies $|S_i| = |S_{\{i,j\}}| = 1$ for all i, j. A set with this property is always dominating for G' but only minimal if each vertex has a private neighbour. For some $v_e \in S_{\{i,j\}}$ this implies that there is some private neighbour $e' = (u, v) \in V' \cap (V_i \times V_j)$ that is not dominated by the (existing) vertex u' in S_i or the vertex v' in S_j; (all vertices V_i and V_j are already dominated by $\{u', v'\} \subset S$ and cannot be private neighbours for v_e). By construction of E', this is only possible if $(u, v) = (u', v') \in E$. Since this is true for all index-pairs (i, j), the vertices $\{v \colon v \in S_i, i = 1, \ldots, k\}$ form a clique in the original graph G.

6.2 Proof of Theorem 2

In analogy to the private neighbourhood, the *private open neighbourhood* of v with respect to S is $pon(v, S) := N(v) - N(S - \{v\})$. Any $w \in pon(v, S)$ is called a *private open neighbour of* v *with respect to* S.

We again reduce from Multicoloured Clique. Let $G = (V, E)$ be a graph with k different colour-classes given by $V = V_1 \cup V_2 \cup \cdots \cup V_k$. We construct a graph $G' = (V', E')$ as follows: Starting from G, we add k vertices $C = \{c_1, \ldots, c_k\}$ and turn each vertex set $V_j \cup \{c_j\}$ into a clique. We claim that G admits a multicoloured independent set (of size k) if and only if G' has a minimal total dominating set with $2k$ vertices.

If $K = \{v_1, \ldots, v_k\} \subseteq V$ is a multi-coloured independent set, then $D := K \cup C$ is a total dominating set. It is minimal, because removing a vertex $v \in \{v_j, c_j\}$ from D would yield $u \notin N(D)$ for $u \in \{v_j, c_j\} \setminus \{v\}$, since both v_j and c_j are not adjacent to any c_i with $i \not= j$ or any vertex in $K \setminus \{v_j\}$.

Conversely, any dominating set must contain at least one vertex from $V_j \cup \{c_j\}$ for each j in order to dominate c_j. Let D be some minimal total dominating set for G', with $|D| \geq 2k$. If for some j, $|D \cap (V_j \cup \{c_j\})| > 2$, then, as $V_j \cup \{c_j\}$ forms a clique, all $\ell > 2$ private open neighbours $p_1, \ldots, p_\ell$ of the vertices from $\{u_1, \ldots, u_\ell\} = D \cap (V_j \cup \{c_j\})$ are from $V' \setminus (V_j \cup \{c_j\})$, so in fact from $V \setminus V_j$. Each p_i belongs to some color class $f(i) \in \{1, \ldots, k\}$, and $f : \{1, \ldots, \ell\} \to \{1, \ldots, k\}$ is an injective mapping; namely, suppose there were $i \neq i'$ with $f(i) = f(i') = s$. The vertex c_s needs to be dominated, which is then impossible without stealing the private neighbour from either u_i or $u_{i'}$.

With the same argument, it is also clear that u_i is the only vertex from $D \cap (V_s \cup \{c_s\})$ for $s = f(i)$. Hence, $|D \cap \{x \in (V_r \cup \{c_r\}) : r = j \vee r \in f(\{1, \ldots, \ell\})\}| = 2\ell$, but this affects $\ell + 1$ colour classes. Hence, D contains less than $2k$ vertices, a contradiction. Therefore, for all j, $1 \leq |D \cap (V_j \cup \{c_j\})| \leq 2$. In order to satisfy $|D| \geq 2k$, this means that, for all j, $1 \leq |D \cap (V_j \cup \{c_j\})| = 2$. We can argue as before that all vertices from $D \cap (V_j \cup \{c_j\})$ must find their private open neighbours within $D \cap (V_j \cup \{c_j\})$. This also means that $K := D \cap V$ forms an independent set in G with $|I| \geq k$.

6.3 Details on Proposition 2

Proposition 8. *Given a graph* $G = (V, E)$ *and a parameter* ℓ*, a call of Algorithm* ***ComputeCoUD*** *with parameters* $(G, \ell, \emptyset, \emptyset, \emptyset, \ell)$ *solves* Co-Upper Domination *in time* $O^*(4.3077^\ell)$.

Algorithm 1: `ComputeCoUD(`G, ℓ, F, I, $\overline{D}$, κ`)`

input : *a graph* $G = (V, E)$*, parameter* $\ell \in \mathbb{N}$*, three disjoint sets* $F, I, \overline{D} \subseteq V$ *and* $\kappa \leq \ell$*.*

output : *"yes" if* $\Gamma(G) \geq |V| - \ell$*; "no" otherwise.*

Let $R \leftarrow V \setminus (F \cup I \cup \overline{D})$

if $\kappa < 0$ **then return** "no" ; (H1)

if R *is empty* **then** (H2)

- **if** $F \cup I$ *is a minimal dominating set of* G *and* $|F \cup I| \geq n - \ell$ **then**
  - **return** "yes"
- **else return** "no"

if *there is a vertex* $v \in R$ *s.t.* $N(v) \subseteq \overline{D}$ **then** (R1)

- **return** `ComputeCoUD(`G, ℓ, F, $I \cup \{v\}$, $\overline{D}$, κ`)`

if *there is a vertex* $v \in R$ *s.t.* $|N(v) \cap F| \geq 1$ **then** (B1)

- **return** `ComputeCoUD(`G, ℓ, $F \cup \{v\}$, I, $\overline{D}$, $\kappa - \frac{1}{2}$`)` $\vee$
  `ComputeCoUD(`G, ℓ, F, I, $\overline{D} \cup \{v\}$, $\kappa - \frac{1}{2}$`)`

if *there is a vertex* $v \in R$ *s.t.* $|N(v) \cap R| = 1$ **then** (B2)

- Let u be the unique neighbour of v in R
- **return** `ComputeCoUD(`G, ℓ, $F \cup \{u, v\}$, I, $\overline{D}$, $\kappa - 1$`)` $\vee$
  `ComputeCoUD(`G, ℓ, $F \cup \{u\}$, I, $\overline{D} \cup \{v\}$, $\kappa - 1$`)` $\vee$
  `ComputeCoUD(`G, ℓ, F, $I \cup \{v\}$, $\overline{D} \cup \{u\}$, $\kappa - 1$`)`

else (B3)

- Let v be a vertex of R
- **return** `ComputeCoUD(`G, ℓ, F, $I \cup \{v\}$, $\overline{D} \cup N(v)$, $\kappa - 2$`)` $\vee$
  `ComputeCoUD(`G, ℓ, $F \cup \{v\}$, I, $\overline{D}$, $\kappa - \frac{1}{2}$`)` $\vee$
  `ComputeCoUD(`G, ℓ, F, I, $\overline{D} \cup \{v\}$, $\kappa - \frac{1}{2}$`)`

Proof. Algorithm `ComputeCoUD` is a branching algorithm, with halting rules (H1) and (H2), reduction rule (R1), and three branching rules (B1)-(B3). We denote by $G = (V, E)$ the input graph and by ℓ the parameter. At each call, the set of vertices V is partitioned into four sets: F, I, $\overline{D}$ and R. The set of **r**emaining vertices R is equal to $V \setminus (F \cup I \cup \overline{D})$, and thus can be obtained from G and the three former sets.

At each recursive call, the algorithm picks some vertices from R. They are either added to the current dominating set $D := F \cup I$, or to the set $\overline{D}$ to indicate that they do not belong to any extension of the current dominating set. The sets F and I are as previously described (*i.e.*, if we denote by D the dominating set we are looking for, $I := \{v \in D : v \in pn(v, D)\}$ and $F := D - I$).

Note that parameter κ corresponds to our "budget", which is initially set to $\kappa := \ell$. Recall that any minimal dominating set of a graph $G = (V, E)$ can be associated with a partition (F, I, P, O) (see Section 2 for the definitions of the sets and for some properties). If we denote by D a minimal dominating set of G and by $\overline{D}$ the set $V \setminus D$, then by definition, F, I is a partition of D and P, O is a partition of $\overline{D}$. Also, by definition of F and P, it holds that $|F| = |P|$ and there is a perfect matching between vertices of F and P. Since each vertex of F will (finally) be matched with its private neighbour from P, we define our budget as $\kappa = \ell - \left(\frac{|F|}{2} + \frac{|P|}{2} + |O|\right)$. One can observe that if D is a minimal dominating set of size at least $n - \ell$ then $\kappa \geq 0$. Conversely, if $\kappa < 0$ then any dominating set D such that $F \cup I \subseteq D$ is of size smaller than $n - \ell$. This shows the correctness of **(H1)**. We now consider the remaining rules of the algorithm. Note that by the choice of κ, each time a vertex x is added to $\overline{D}$, the value of κ decrease by $\frac{1}{2}$ (or by 1 if we can argue that x is not matched with a vertex of F and thus belongs to O). Also, whenever a vertex x is added to F, the value of κ decreases by $\frac{1}{2}$.

(H2) If R is empty, then all vertices have been decided: they are either in $D := F \cup I$ or in $\overline{D}$. It remains to check whether D is a minimal dominating set of size at least $n - \ell$.

(R1) All neighbours (if any) of v are in $\overline{D}$ and thus v has to be in $I \cup F$. As v will also belong to $pn(v, D)$, we can safely add v to I. Observe also that this reduction rule does not increase our budget.

(B1) Observe that if v has a neighbour in F, then v cannot belong to I. When a vertex v in added to F the budget is reduced by at least $\frac{1}{2}$; when v is added to $\overline{D}$, the budget is reduced by $\frac{1}{2}$, as well. So (B1) gives a branching vector of $(\frac{1}{2}, \frac{1}{2})$.

(B2) If (R1) and (B1) do not apply and $N(v) \cap R = \{u\}$, then the vertex v has to either dominate itself or be dominated by u. Every vertex in F has a neighbour in F, which in this case means that $v \in F$ implies $u \in F$ (first branch). Moreover, the budget is the reduced by at least $2 \cdot \frac{1}{2}$.
If v is put in I, u has to go to $\bar{D}$ (third branch). Thus u cannot be a private neighbour of some F-vertex, and the budget decreases by at least 1 ($u \in O$). If v does not dominate itself, u has to be in $F \cup I$. In this last case it suffices to consider the less restrictive case $u \in F$, as v can be chosen as the private neighbour for u (second branch). If u is indeed in I for a minimal dominating set which extends the current $I \cup F$, there is a branch which puts all the remaining neighbours of u in $\bar{D}$. Observe that we only dismiss branches with halting rule (H2) where we check if $F \cup I$ is a minimal dominating set, we do not require the chosen partition to be correct. As for the counting in halting rule (H1): whether we count $u \in F$ and $v \in P$ (recall that $P \subseteq \overline{D}$) each with $\frac{1}{2}$ or count $v \in O$ (recall that $O \subseteq \overline{D}$) with 1 does not make a difference for κ. So the budget decreases by at least 1.
Altogether (B2) gives a branching vector of $(1, 1, 1)$.

(B3) The correctness of (B3) is easy as all possibilities are explored for vertex v. Observe that by (R1) and (B2), vertex v has at least two neighbours in R. When v is added to I, these two vertices are removed (and cannot be the private neighbours of some F-vertices). Thus we reduce the budget by at least 2. When v is added to F, the budget decreases by at least $\frac{1}{2}$. When v is added to $\overline{D}$, we reduce the budget by at least $\frac{1}{2}$. Thus (B3) gives a branching vector of $(2, \frac{1}{2}, \frac{1}{2})$. However, we can observe that the second branching rule (*i.e.,* when v is added to F) implies a subsequent application of (B1) (or rule (H1) would stops the recursion). Thus the branching vector can be refined to $(2, 1, 1, \frac{1}{2})$.

Taking the worst-case over all branching vectors, establishes the claimed running time. □

6.4 Proof of Proposition 5

We are considering all partitions of each bag of the path decomposition into 6 sets: F, F^*, I, P, O, O^*, where

- F is the set of vertices that belong to the upper dominating set and have already been matched to a private neighbour;
- F^* is the set of vertices that belong to the upper dominating set and still need to be matched to a private neighbour;
- I is the set of vertices that belong to the upper dominating set and is independent in the graph induced by the upper dominating set;
- P is the set of private neighbours that are already matched to vertices in the upper dominating set;
- O is the set of vertices that are not belonging neither to the upper dominating set nor to the set of private neighbours but are already dominated;
- O^* is the set of vertices not belonging to the upper dominating set that have not been dominated yet.

(Sets within the partition can be also empty.) For each such partition, we determine the largest minimal dominating set in the situation described by the partition, assuming optimal settings in the part of the graph already forgotten.

We can assume that we are given a nice path decomposition. So, we only have to describe the table initialisation (the situation in a bag containing only one vertex) and the table updates necessary when we introduce a new vertex into a bag and when we finally forget a vertex.

initialisation We have six cases to consider:
- $T[\{v\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset] \leftarrow -1$,
- $T[\emptyset, \{v\}, \emptyset, \emptyset, \emptyset, \emptyset] \leftarrow 1$,
- $T[\emptyset, \emptyset, \{v\}, \emptyset, \emptyset, \emptyset] \leftarrow 1$,
- $T[\emptyset, \emptyset, \emptyset, \{v\}, \emptyset, \emptyset] \leftarrow -1$,
- $T[\emptyset, \emptyset, \emptyset, \emptyset, \{v\}, \emptyset] \leftarrow -1$.
- $T[\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \{v\}] \leftarrow 1$.

Here, -1 signals the error cases when we try to introduce already dominated vertices.

forget Assume that we want to update table T to table T' for the partition F, F^*, I, P, O, O^*, eliminating vertex v:
- $T'[F \setminus \{v\}, F^*, I, P, O, O^*] \leftarrow T[F, F^*, I, P, O, O^*]$,

- $T'[F, F^* \setminus \{v\}, I, P, O, O^*] \leftarrow -1$,
- $T'[F, F^*, I \setminus \{v\}, P, O, O^*] \leftarrow T[F, F^*, I, P, O, O^*]$,
- $T'[F, F^*, I, P \setminus \{v\}, O, O^*] \leftarrow T[F, F^*, I, P, O, O^*]$,
- $T'[F, F^*, I, P, O \setminus \{v\}, O^*] \leftarrow T[F, F^*, I, P, O, O^*]$,
- $T'[F, F^*, I, P, O, O^* \setminus \{v\}] \leftarrow -1$.

Clearly, it is not feasible to eliminate vertices whose promises have not yet been fulfilled.

introduce We are now introducing a new vertex v into the bag. The neighbourhood N refers to the situation in the new bag, *i.e.*, to the corresponding induced graph. T' is the new table and T the old one.

- $T'[F \cup \{v\}, F^*, I, P, O, O^*] \leftarrow -1$ if $N(v) \cap (I \cup O^*) \neq \emptyset$ or $|N(v) \cap P| \neq 1$;
  $T'[F \cup \{v\}, F^*, I, P, O, O^*] \leftarrow \max\{T[F, F^*, I, P \setminus \{x\}, O \setminus X, O^* \cup X \cup \{x\}] : x \in N(v), X \subseteq (N(v) \setminus \{x\}) \cap O\} + 1$ otherwise;
  this means that exactly one neighbour x of v that was previously labelled to be dominated in the future is selected as a private neighbour of v; all other neighbours of v are labelled dominated;
- $T'[F, F^* \cup \{v\}, I, P, O, O^*] \leftarrow -1$ if $N(v) \cap (I \cup P \cup O^*) \neq \emptyset$;
  $T'[F, F^* \cup \{v\}, I, P, O, O^*] \leftarrow \max\{T[F, F^*, I, P, O \setminus X, O^* \cup X] : X \subseteq N(v) \cap O\} + 1$ otherwise;
  in contrast to the previous situation, no private neighbour has been selected;
- $T'[F, F^*, I \cup \{v\}, P, O, O^*] \leftarrow -1$ if $N(v) \cap (I \cup F \cup F^* \cup P \cup O^*) \neq \emptyset$;
  $T'[F, F^*, I \cup \{v\}, P, O, O^*] \leftarrow \max\{T[F, F^*, I, P, O \setminus X, O^* \cup X] : X \subseteq N(v) \cap O\} + 1$ otherwise;
- $T'[F, F^*, I, P \cup \{v\}, O, O^*] \leftarrow -1$ if $N(v) \cap I \neq \emptyset$ or $|N(v) \cap F| \neq 1$;
  $T'[F, F^*, I, P \cup \{v\}, O, O^*] \leftarrow T[F \setminus N(v), F^* \cup (N(v) \cap F), I, P, O, O^*]$ otherwise;
  this means that exactly one neighbour x of v that was previously labelled as dominating but looking for a private neighbour in the future is selected as pairing up with v; all other neighbours of v are not in the dominating set;
- $T'[F, F^*, I, P, O \cup \{v\}, O^*] \leftarrow T[F, F^*, I, P, O, O^*]$ if $N(v) \cap (F \cup F^* \cup I) \neq \emptyset$ and $T'[F, F^*, I, P, O \cup \{v\}, O^*] \leftarrow -1$ otherwise;
- $T'[F, F^*, I, P, O, O^* \cup \{v\}] \leftarrow T[F, F^*, I, P, O, O^*]$ unless $N(v) \cap (F \cup F^* \cup I) \neq \emptyset$; in that case, $T'[F, F^*, I, P, O, O^* \cup \{v\}] \leftarrow -1$.

The formal induction proof showing the correctness of the algorithm is an easy standard exercise. As to the running time, observe that we cycle only in one case potentially through all subsets of O, so that the running time follows by applying the binomial formula:

$$\sum_{i=0}^{p} \binom{p}{i} 5^i 2^{p-i} = 7^p .$$

6.5 Proof of Proposition 6

For a given nice tree decomposition use the same partition into the six sets F, F^*, I, P, O, O^* for each bag as in the proof of Proposition 5. The procedures for initialisation, forget and introduce can be used just like before. The only case missing for a treewidth algorithm is a procedure which deals with a join-bag (a bag with two children which both contain the same vertices as their parent). In terms of a table update, this join-procedure has to create a new table T' from the two given tables T_1, T_2 of the children. The only important things to consider are that private neighbourhoods with forgotten vertices only exist in exactly one of the child-bags and that dominated outsiders only need domination from at most one of the child-bags. This can be handled with the following procedure:

join To create the new table entry $T'[F, F^*, I, P, O, O^*]$ from existing tables T_1 and T_2, consider all partitions $F_1 \cup F_2$ of $F - N(P)$ and $P_1 \cup P_2$ of $P - N(F)$ and $O_1 \cup O_{12} \cup O_2$ of $O - N(I \cup F \cup F^*)$ and choose the partitions for which $v_1 := T_1[F - F_2, F^* \cup F_2, I, P - P_2, O - O_2, O^* \cup P_2 \cup O_2] \neq -1$ and $v_2 := T_2[F - F_1, F^* \cup F_1, I, P - P_1, O - O_1, O^* \cup P_1 \cup O_1] \neq -1$ such that $T'[F, F^*, I, P, O, O^*] := v_1 + v_2 - |F \cup F^* \cup I|$ is maximised.

Considering all these partitions for a join-bag results in a worst-case running time in $O^*(11^p)$.