On miniaturized problems in parameterized complexity theory

Yijia Chen
Institut für Informatik,
Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany.
yijia.chen@informatik.hu-berlin.de

Jörg Flum
Abteilung für Mathematische Logik,
Universität Freiburg, Eckerstr. 1, 79104 Freiburg, Germany.
Joerg.Flum@math.uni-freiburg.de

Abstract

We introduce a general notion of miniaturization of a problem that comprises the different miniaturizations of concrete problems considered so far. We develop parts of the basic theory of miniaturizations. Using the appropriate logical formalism, we show that the miniaturization of a definable problem in $\mathrm{W}[t]$ *lies in* $\mathrm{W}[t]$*, too. In particular, the miniaturization of the dominating set problem is in* $\mathrm{W}[2]$*. Furthermore we investigate the relation between* $f(k) \cdot n^{o(k)}$ *time and subexponential time algorithms for the dominating set problem and for the clique problem.*

1. Introduction

Parameterized complexity theory provides a framework for a refined complexity analysis of algorithmic problems that are intractable in general. Central to the theory is the notion of *fixed-parameter tractability*, which relaxes the classical notion of tractability, polynomial time computability, by admitting algorithms whose runtime is exponential, but only in terms of some *parameter* that is usually expected to be small. Let FPT denote the class of all fixed-parameter tractable problems. A well-known example of a problem in FPT is the vertex cover problem, the parameter being the size of the vertex cover we ask for.

As a complexity theoretic counterpart, a theory of *parameterized intractability* has been developed. In classical complexity, the notion of NP-completeness is central to a nice and simple theory for intractable problems. Unfortunately, the world of parameterized intractability is more complex: there is a big variety of seemingly different classes of parameterized intractability. For a long while, the smallest complexity class of parameterized intractable problems considered in the literature was W[1], the first class of the so-called W-hierarchy. (In particular, FPT $\subseteq$ W[1]; moreover, FPT $\neq$ W[1] would imply PTIME $\neq$ NP.)

Recently, the situation has changed: In [6], Downey et al. consider various problems in W[1] that, apparently, are not W[1]-hard. Most of them are "miniaturizations" of well-studied problems in parameterized complexity theory; for example, mini-CIRCSAT is the problem that takes a circuit $\mathcal{C}$ of size $\leq k \cdot \log m$, where k is the parameter and m in unary is part of the input, and asks whether $\mathcal{C}$ is satisfiable. This problem is called a miniaturization of CIRCSAT, as the size ($\leq k \cdot \log m$) of $\mathcal{C}$ is small compared with m (under the basic assumption of parameterized complexity that the parameter k is small too). In [6], Downey et al. introduce the class MINI[1] as the class of parameterized problems fpt-reducible to mini-CIRCSAT. MINI[1] now provides very nice connections between classical complexity and parameterized complexity as it is known that FPT = MINI[1] if and only if n variable 3SAT can be solved in time $2^{o(n)}$. This equivalence stated in [6] is based on a result of Cai and Juedes [1].

Besides this "miniaturization route", a second route to MINI[1] has been considered by Fellows in [9]; he calls it the "renormalization route" to MINI[1]. He "renormalizes" the parameterized vertex cover problem and considers the so-called

$k \cdot \log n$ vertex cover problem: It takes as input a graph G and as parameter a natural number k; it asks if G has a vertex cover of size $k \cdot \log n$, where n is the size of G. This problem turns out to be MINI[1]-complete (cf. [9]).

Before outlining the purpose and the contents of this paper let us give two quotations, the first one from Fellows' paper [9] and the second one from Downey's paper [5]:

> Dozens of renormalized FPT problems and miniaturized arbitrary problems are now known to be MINI[1]-complete. However, what is known is quite problem specific.
>
> Can the hierarchy [starting with MINI[1]] be extended [to a hierarchy within W[1]]?

Among others, in this paper we try to develop the theory of miniaturized problems on a more abstract level and we address the problems mentioned in these quotations. Concerning the second problem, even though we introduce a hierarchy of complexity classes, we conjecture, among others encouraged by the results of this paper, that the world of parameterized intractability in W[1] is so rich that, probably, there are various more or less natural hierarchies in W[1].

We sketch the content of the different sections. In Section 2 we give the necessary preliminaries. In particular, we introduce the notion of a size function, a polynomial time function $\| \;\|$ defined on the inputs x of a given problem with the property that the length $|x|$ of x is polynomially bounded in $\|x\|$. For example, for a graph $G = (V, E)$, natural choices could be $|V|$, the number of vertices, or $|V| + |E|$, the number of vertices and edges, or $\Theta(|V| + |E| \cdot \log |V|)$, the total length of its binary description; but for graphs with many isolated vertices, $|E|$ is not a size function. Also in passing we show that the effective versions of two notions of subexponential time coincide.

In Section 3, for a given size function $\| \;\|$, we define the concept of the miniaturization $\text{mini}^{\| \;\|}\text{-}Q$ of an arbitrary problem Q. Now, a proof essentially due to Cai and Juedes [1] goes through for this concept showing that $\text{mini}^{\| \;\|}\text{-}Q$ is fixed-parameter tractable just in case $x \in Q$ is solvable in time $2^{o(\|x\|)}$. In Proposition 9 we extend the well-known fact that a linear reduction from Q to Q' yields an fpt-reduction from the miniaturization of Q to that of Q' and essentially show that the existence of a linear reduction from Q to Q' is *equivalent* to the existence of an fpt-reduction of the miniaturization of Q to that of Q' that is linear with respect to the parameters. Perhaps therefore, there are so many not fpt-equivalent miniaturizations.

There is a way of defining parameterized problems by means of first-order formulas with a free *set* variable X that has been dubbed *Fagin-definability* in [10], since it is related to Fagin's theorem characterizing NP as the class of Σ_1^1-definable problems. For example, the parameterized clique problem is Fagin-definable by the formula

$$\forall y \forall z((Xy \land Xz \land y \neq z) \rightarrow Eyz).$$

In [7], Downey et al. showed that W[t], the tth class of the W-hierarchy, contains all parameterized problems Fagin-defined by Π_t-formulas and conversely, there are W[t]-complete problems Fagin-defined by Π_t-formulas. Some miniaturized problems considered in the literature can be regarded as miniaturization of *unweighted* Fagin-definable problems, a concept we introduce in this paper. In general, the miniaturization may increase the computational complexity of a problem; e.g., the parameterized vertex cover problem is fixed-parameter tractable while its miniaturization is not (unless MINI[1] = FPT). In Section 4 we prove that in a certain sense weighted and unweighted definable problems have the same computational complexity. And using this result, we show that the miniaturization of every Fagin-definable problem in W[t] lies in W[t], too.

As mentioned above, Π_1-formulas of the form $\varphi(X) = \forall x_1 \dots \forall x_t \psi(X)$ with a set variable X and with a quantifier-free $\psi(X)$ are used to obtain the Fagin-definable problems in W[1]. We obtain a hierarchy of classes within W[1] taking the length t of the block of quantifiers into consideration. We study the basic properties of this hierarchy; in particular, we show that the (appropriate) miniaturization of tSAT is complete in the tth class of this hierarchy. Recall that Impagliazzo and Paturi [13] have shown that, assuming the *exponential time hypothesis* (stating that n variable 3SAT cannot be solved in time $2^{o(n)}$), the complexity of tSAT increases with t.

So far, when comparing the complexity of miniaturized and other parameterized problems, we used many-one reductions (more precisely, fpt many-one reductions). In some papers, Turing reductions have been considered. As we show in Section 5, most problems studied in this paper are Turing equivalent.

It is clear that testing the existence of a clique of size k in a graph $G = (V, E)$ needs at most time $|V|^{O(k)}$. So it prompts the question if there is an $f(k) \cdot |V|^{o(k)}$ time algorithm for some computable f. J. Chen et al. [3] show that if there is such an algorithm then the clique problem would be solvable in time $2^{o(|V|)}$. We study this problem in Section 6 and prove a similar result for the dominating set problem; its proof relies on the machinery of the weighted and unweighted problems we have developed in the previous sections.

In the final section, Section 7, we deal with renormalizations. Besides the renormalization of the vertex cover problem introduced in Fellows [9], we consider a slightly different renormalization and also show its fpt-equivalence to the miniaturization. We shall see that this result cannot be extended to arbitrary Fagin-definable problems, in particular not to the clique problem.

2. Preliminaries

In this section we fix our notations, recall some definitions and results, and introduce the concept of size function.

2.1. Relational structures and first-order logic. A (relational) *vocabulary* τ is a finite set of relation symbols. Each relation symbol has an *arity*. A *structure* $\mathcal{A}$ of vocabulary τ, or τ*-structure* (or, simply structure), consists of a set A called the *universe*, and an interpretation $R^{\mathcal{A}} \subseteq A^r$ of each r-ary relation symbol $R \in \tau$.

For example, let $\tau_{\text{circ}} = \{E, I, O, G_{\bigwedge}, G_{\bigvee}, G_{\neg}\}$, where E is a binary relation symbol and I, O, $G_{\bigwedge}$, $G_{\bigvee}$, $G_{\neg}$ are unary relation symbols. We view Boolean circuits as τ_{circ}-structures

$$\mathcal{C} \quad = \quad (C, E^{\mathcal{C}}, I^{\mathcal{C}}, O^{\mathcal{C}}, G^{\mathcal{C}}_{\bigwedge}, G^{\mathcal{C}}_{\bigvee}, G^{\mathcal{C}}_{\neg}),$$

where $(C, E^{\mathcal{C}})$ is the directed acyclic graph underlying the circuit, $I^{\mathcal{C}}$ is the set of all input nodes, $O^{\mathcal{C}}$ just contains the output node, $G^{\mathcal{C}}_{\bigwedge}$, $G^{\mathcal{C}}_{\bigvee}$, and $G^{\mathcal{C}}_{\neg}$ are the sets of and-gates, or-gates (and-gates and or-gates of arbitrary arity), and negation-gates, respectively. The *weight* of a truth value assignment to the input nodes of $\mathcal{C}$ is the number of input nodes set to TRUE by the assignment.

Often for graphs we shall use the more common notation $G = (V, E)$ (or, $G = (V(G), E(G))$), where V is the set of vertices of the graph G and E its set of edges.

All structures we consider in this paper have finite universe. Let B be a subset of the universe A of the τ-structure $\mathcal{A}$. We denote by $\mathcal{A} \upharpoonright B$ the substructure of $\mathcal{A}$ with universe B (i.e, the τ-structure $\mathcal{B}$ with universe B and with $R^{\mathcal{B}} = R^{\mathcal{A}} \cap B^{\text{arity}(R)}$ for $R \in \tau$).

We define the *size* $\|\mathcal{A}\|_0$ of a τ-structure $\mathcal{A}$ to be the number

$$\|\mathcal{A}\|_0 \quad := \quad |A| + \sum_{R \in \tau} \text{arity}(R) \cdot |R^{\mathcal{A}}| \cdot \log |A|.$$

In fact, the length of a reasonable binary encoding of $\mathcal{A}$ as a string is $\Theta(\|\mathcal{A}\|_0)$.

First-order formulas are built up from atomic formulas using the usual boolean connectives and existential and universal quantifications. Recall that *atomic formulas* are formulas of the form $x = y$ or $Rx_1 \dots x_r$, where $x, y, x_1, \dots, x_r$ are variables and R is an r-ary relation symbol. For $t \geq 1$, by Π_t we denote the class of all first-order formulas of the form

$$\forall x_{11} \dots \forall x_{1k_1} \exists x_{21} \dots \exists x_{2k_2} \; \dots \; Qx_{t1} \dots Qx_{tk_t} \; \psi,$$

where $Q = \forall$ if t is odd and $Q = \exists$ otherwise, and where ψ is quantifier-free.

If $\mathcal{A}$ is a structure, $a_1, \dots, a_n$ are elements of A, and $\varphi(x_1, \dots, x_n)$ is a first-order formula whose free variables are among $x_1, \dots, x_n$, then we write $\mathcal{A} \models \varphi(a_1, \dots, a_n)$ to denote that $\mathcal{A}$ satisfies φ if the variables $x_1, \dots, x_n$ are interpreted by $a_1, \dots, a_n$, respectively.

2.2. Propositional logic. Formulas of propositional logic are built up from *propositional variables* $X_1, X_2, \dots$ by taking conjunctions, disjunctions, and negations. We distinguish between *small conjunctions*, denoted by $\wedge$, which are just conjunctions of two formulas, and *big conjunctions*, denoted by $\bigwedge$, which are conjunctions of arbitrary finite sets of formulas. Analogously, we distinguish between *small* and *big disjunctions*, denoted by $\vee$ and by $\bigvee$, respectively.

For $t \geq 0$ and $d \geq 1$ we define the sets $\Gamma_{t,d}$ and $\Delta_{t,d}$ of propositional formulas by induction on t (here, by $(\lambda_1 \wedge \dots \wedge \lambda_r)$ we mean the iterated small conjunction $((\dots(\lambda_1 \wedge \lambda_2) \dots) \wedge \lambda_r)$:

$$\begin{aligned}
\Gamma_{0,d} &:= \{(\lambda_1 \wedge \dots \wedge \lambda_r) \mid \lambda_1, \dots, \lambda_r \text{ literals}, r \leq d\}, \\
\Delta_{0,d} &:= \{(\lambda_1 \vee \dots \vee \lambda_r) \mid \lambda_1, \dots, \lambda_r \text{ literals}, r \leq d\}, \\
\Gamma_{t+1,d} &:= \{\textstyle\bigwedge \Pi \mid \Pi \subseteq \Delta_{t,d}\}, \\
\Delta_{t+1,d} &:= \{\textstyle\bigvee \Pi \mid \Pi \subseteq \Gamma_{t,d}\}.
\end{aligned}$$

Often, we denote the class $\Gamma_{2,1}$, that is, the class of all propositional formulas in conjunctive normal form, by CNF. For $d \geq 1$, a formula is in *d conjunctive normal form* if it is a conjunction of disjunctions of at most d literals; the class of all such formulas is denoted by dCNF. Often, we tacitly assume that a formula $\alpha \in d$CNF is given as a set of clauses where each clause contains d literals and identify α with a τ_d-structure $\mathcal{A}(\alpha)$. Here $\tau_d := \{N, C\}$, where N is binary and C is $d$ary, and $\mathcal{A}(\alpha)$ has the set $\{X, \neg X \mid X \text{ is a variable of } \alpha\}$ as universe and

$$\begin{aligned} N^{\mathcal{A}(\alpha)} &:= \{(X, \neg X) \mid X \text{ is a variable of } \alpha\}; \\ C^{\mathcal{A}(\alpha)} &:= \{(\lambda_1, \dots, \lambda_d) \mid (\lambda_1 \vee \dots \vee \lambda_d) \text{ is a clause of } \alpha\}. \end{aligned}$$

The *weight* of a truth value assignment to the variables of a propositional formula α is the number of variables set to TRUE by the assignment. α is *k-satisfiable*, if there is an assignment satisfying α of weight k.

2.3. Size functions. Let Σ be an alphabet. We denote the length of a string $x \in \Sigma^*$ by $|x|$.

Definition 1. A function $\| \ \| : \Sigma^* \to \mathbb{N}$ is a *size function*, if it is computable in polynomial time and if, for some $c \in \mathbb{N}$, $|x| \leq \|x\|^c$ holds for all $x \in \Sigma^*$.

In particular, $| \ |$ is a size function. For a vocabulary τ, the function $\| \ \|_0$ (cf. Section 2.1) defined for τ-structures (more precisely, for the encodings of τ-structures by strings) is a size function. We introduce further size functions for τ-structures:

$$\begin{aligned} \|\mathcal{A}\|_+ &:= |A| + \sum_{R \in \tau} \text{arity}(R) \cdot |R^{\mathcal{A}}|; \\ \|\mathcal{A}\|_- &:= |A|. \end{aligned}$$

Note that

- for a graph G with n vertices and m edges: $\|G\|_- = n$ and $\|G\|_+ = \Theta(n + m)$;
- for a circuit $\mathcal{C}$ with n nodes and m lines: $\|\mathcal{C}\|_- = n$ and $\|\mathcal{C}\|_+ = \Theta(n + m)$;
- for a propositional formula $\alpha \in d$CNF with n variables and m clauses: $\|\alpha\|_- \ (:= \|\mathcal{A}(\alpha)\|_-) = \Theta(n)$ and $\|\alpha\|_+ \ (:= \|\mathcal{A}(\alpha)\|_+) = \Theta(n + m)$. [1]

2.4. Fixed-Parameter Tractability. A *parameterized problem* is a set $Q \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. If $(x, k) \in \Sigma^* \times \mathbb{N}$ is an instance of a parameterized problem, we refer to x as the *input* and to k as the *parameter*. Unless mentioned explicitly otherwise, we encode natural numbers in binary.

To illustrate our notation, let us give the example of the *parameterized weighted satisfiability problem* WSAT(Θ) for a class Θ of propositional formulas:

WSAT(Θ)	
Input:	A formula α in Θ.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if α is k-satisfiable.

Definition 2. A parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is *fixed-parameter tractable*, if there is a computable function $f : \mathbb{N} \to \mathbb{N}$, a polynomial $p \in \mathbb{N}[x]$, and an algorithm that, given a pair $(x, k) \in \Sigma^* \times \mathbb{N}$, decides if $(x, k) \in Q$ in at most $f(k) \cdot p(|x|)$ steps.

FPT denotes the complexity class consisting of all fixed-parameter tractable parameterized problems.

Often, when considering fixed-parameter tractable problems, we shall assume that the function f in Definition 2 is easily reversible. A function $f : \mathbb{N} \to \mathbb{N}$ is *easily reversible*, if f is strictly monotone and time constructible (inputs and outputs are written in unary). We shall use the following facts:

[1] Note that for arbitrary propositional formulas the number of variables does not define a size function; for formulas α in dCNF we obtain a size function, since we identify α with $\mathcal{A}(\alpha)$.

Lemma 3. *(1) For any computable function $g : \mathbb{N} \to \mathbb{N}$, there is an easily reversible f with $g(n) \leq f(n)$ for all $n \in \mathbb{N}$.*

(2) For any easily reversible $f : \mathbb{N} \to \mathbb{N}$ the inverse function $f^{-1} : \mathbb{N} \to \mathbb{N}$ given by

$$f^{-1}(n) \quad := \quad \min\{m \mid f(m) \geq n\}$$

is computable in polynomial time.

Complementing the notion of fixed-parameter tractability, there is a theory of parameterized intractability. It is based on the following notion of parameterized reduction:

Definition 4. An fpt-*reduction* (more precisely, fpt *many-one reduction*) from the parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ to the parameterized problem $Q' \subseteq (\Sigma')^* \times \mathbb{N}$ is a mapping $R : \Sigma^* \times \mathbb{N} \to (\Sigma')^* \times \mathbb{N}$ such that:

(1) For $(x, k) \in \Sigma^* \times \mathbb{N}$: $(x, k) \in Q \iff R(x, k) \in Q'$.

(2) There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$, say with $R(x, y) = (x', k')$, we have $k' \leq g(k)$.

(3) There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ and a polynomial p such that $R(x, k)$ is computable in time $f(k) \cdot p(|x|)$.

We write $Q \leq^{\text{fpt}} Q'$ if there is an fpt-reduction from Q to Q', and $Q =^{\text{fpt}} Q'$ if ($Q \leq^{\text{fpt}} Q'$ and $Q' \leq^{\text{fpt}} Q$). We set

$$[Q]^{\text{fpt}} := \{Q' \mid Q' \leq^{\text{fpt}} Q\}$$

and, for a class C of parameterized problems,

$$[\mathrm{C}]^{\text{fpt}} \quad := \quad \bigcup_{Q \in \mathrm{C}} [Q]^{\text{fpt}}.$$

For $t \geq 1$, the class W$[t]$ is defined by

$$\mathrm{W}[t] \quad := \quad [\{\text{WSAT}(\Gamma_{t,d}) \mid d \geq 1\}]^{\text{fpt}}.$$

Clearly, FPT $\subseteq$ W[1] $\subseteq$ W[2] ... and it is conjectured that FPT $\neq$ W[1] and the W-hierarchy is strict (which would imply PTIME $\neq$ NP).

2.5. Subexponential Time. There are two notions of *subexponential time*, namely $\mathbf{DTIME}(2^{o(n)})$ and $\bigcap_{\varepsilon>0} \mathbf{DTIME}(2^{\varepsilon \cdot n})$. We shall need their effective versions and first show that they are equivalent.

For computable functions $f, g : \mathbb{N} \to \mathbb{N}$, we write $f \in o^{\text{eff}}(g)$, if $f \in o(g)$ holds in an effective way, i.e., if there is a computable function h such that, given any $\ell \in \mathbb{N}$ with $\ell > 0$, we have $f(m)/g(m) \leq 1/\ell$ for all $m \geq h(\ell)$.

Proposition 5. *For a classical problem $Q \subseteq \Sigma^*$ and a size function $\| \, \| : \Sigma^* \to \mathbb{N}$ the following are equivalent:*

(1) $Q \in \mathbf{DTIME}(2^{o^{\text{eff}}(\|x\|)})$, i.e., $x \in Q$ is decidable in time $2^{f(\|x\|)}$ for some $f \in o^{\text{eff}}(\text{id})$, where id *denotes the identity function on $\mathbb{N}$.*

(2) For every rational number $\varepsilon > 0$, there is an algorithm $\mathbb{A}_\varepsilon$ deciding $x \in Q$ in time $O(2^{\varepsilon \cdot \|x\|})$. Moreover, $\mathbb{A}_\varepsilon$ can be computed from ε.

Proof: (1) ⇒ (2): Let $\mathbb{B}$ be an algorithm deciding $x \in Q$ in time $2^{f(\|x\|)}$ for some $f \in o^{\text{eff}}(\text{id})$. Choose a computable h such $f(\|x\|) \leq 1/\ell \cdot \|x\|$ for all $\|x\| \geq h(\ell)$. Given $\varepsilon > 0$, note $|x| \leq \|x\|^c$ for some $c \in \mathbb{N}$, hence there are only finitely many x satisfying $\|x\| \leq h(\lceil 1/\varepsilon \rceil)$. The required algorithm $\mathbb{A}_\varepsilon$ maintains a list of those x together with the information whether $x \in Q$. On any input instance x, the algorithm $\mathbb{A}_\varepsilon$ first checks if x is stored in the list, and if so it returns the correct answer from the list. Otherwise $\|x\| \geq h(\lceil 1/\varepsilon \rceil)$ and then, $\mathbb{A}_\varepsilon$ proceeds as $\mathbb{B}$.

(2) ⇒ (1): Let $\mathbb{A}$ be an algorithm that, given $\ell \geq 1$, computes $\mathbb{A}_{1/\ell}$ according to (2) in time $g(\ell)$. We can assume that g is easily reversible. Given $x \in \Sigma^*$, the algorithm $\mathbb{B}$ we aim at first computes $\ell := g^{-1}(\|x\|) - 1$ and then applies $\mathbb{A}_{1/\ell}$ to decide if $x \in Q$. Overall, for some $d \in \mathbb{N}$, it needs time

$$\|x\|^d + g(\ell) + 2^{\|x\|/\ell} \leq \|x\|^d + \|x\| + 2^{\|x\|/(g^{-1}(\|x\|)-1)} \in 2^{o^{\text{eff}}(\|x\|)}.$$

□

3. The miniaturization of an arbitrary problem

In this section, for a classical problem Q and a size function $\|\ \|$, we introduce its miniaturization $\text{mini}^{\|\ \|}\text{-}Q$, a parameterized problem, and study the relationship between the complexity of Q and $\text{mini}^{\|\ \|}\text{-}Q$.

Definition 6. Let $Q \subseteq \Sigma^*$ and let $\|\ \| : \Sigma^* \to \mathbb{N}$ be a size function. The *miniaturization* $\text{mini}^{\|\ \|}\text{-}Q$ of Q with respect to $\|\ \|$ is the parameterized problem:

$\text{mini}^{\|\ \|}\text{-}Q$	
Input:	$n, k \in \mathbb{N}$ in unary[2], and $x \in \Sigma^*$.
Parameter:	k.
Problem:	Decide if $\|x\| \leq k \cdot \log n$ and $x \in Q$.

Remarks 7. a) Let $Q \subseteq \Sigma^*$ and $\|\ \|$ be a size function with $|x| \leq \|x\|^c$. Consider an instance n, k, x of $\text{mini}^{\|\ \|}\text{-}Q$. Then, $|x| > (k \cdot \log n)^c$ implies $\|x\| > k \cdot \log n$. Thus, the condition $\|x\| \leq k \cdot \log n$ can be checked in time polynomial in k and n only. Therefore, often the problem $\text{mini}^{\|\ \|}\text{-}Q$ is presented in the more appealing form:

$\text{mini}^{\|\ \|}\text{-}Q$	
Input:	$n, k \in \mathbb{N}$ in unary, $x \in \Sigma^*$ with $\|x\| \leq k \cdot \log n$.
Parameter:	k.
Problem:	Decide if $x \in Q$.

b) Arguing similarly as in part a), one shows that if $\text{mini}^{\|\ \|}\text{-}Q$ is in FPT, then there is an algorithm solving $\text{mini}^{\|\ \|}\text{-}Q$ (on instance n, k, x) in $\leq f(k) \cdot p(n)$ steps for some computable function f and some polynomial p.

c) If $Q \in$ PTIME, then $\text{mini}^{\|\ \|}\text{-}Q \in$ FPT.

By part (2) of the next lemma for a problem on graphs, for example, it is irrelevant to its computational complexity, whether we consider the problem with the size function $\|G\|_+$ or whether we take the number of vertices plus the number of edges as the size of G.

Lemma 8. *Let $Q \subseteq \Sigma^*$ and $\|\ \|_1, \|\ \|_2 : \Sigma^* \to \mathbb{N}$ be size functions. Then,*

(1) $\|x\|_1 \leq \|x\|_2$ *for all* $x \in \Sigma^*$ *implies* $\text{mini}^{\|\ \|_2}\text{-}Q \leq^{\text{fpt}} \text{mini}^{\|\ \|_1}\text{-}Q$.

(2) $\|\ \|_1 \in \Theta(\|\ \|_2)$ *implies* $\text{mini}^{\|\ \|_2}\text{-}Q =^{\text{fpt}} \text{mini}^{\|\ \|_1}\text{-}Q$.

The following result relates the fixed-parameter tractability of $\text{mini}^{\|\ \|}\text{-}Q$ with the solvability of Q in subexponential time. Its proof uses an idea of [1] in the form presented in [6] (also implicit in our proof of Proposition 5).

Proposition 9. *For $Q \subseteq \Sigma^*$ and any size function $\|\ \| : \Sigma^* \to \mathbb{N}$ the following are equivalent:*

(1) $x \in Q$ is solvable in time $2^{o^{\text{eff}}(\|x\|)}$.

(2) $\text{mini}^{\|\ \|}\text{-}Q \in$ FPT.

(3) There is an algorithm that, for every instance n, k, x of $\text{mini}^{\|\ \|}\text{-}Q$ *with $\|x\| \leq k \cdot \log n$, decides if $(n, k, x) \in$* $\text{mini}^{\|\ \|}\text{-}Q$ *in time $f(k) + n$ for some computable f.*

Proof: Assume (1) and let n, k, x be an instance of $\text{mini}^{\|\ \|}\text{-}Q$ as in (3). For $\varepsilon := 1/k$ determine the algorithm $\mathbb{A}_\varepsilon$ according to (2) in Proposition 5 in time $f(k)$ and apply $\mathbb{A}_\varepsilon$ to x. Altogether, we need time

$$f(k) + 2^{1/k \cdot \|x\|} \leq f(k) + 2^{1/k \cdot k \cdot \log n} \leq f(k) + n.$$

Since the implication from (3) to (2) is clear by Remark 7 a), we turn to (2) $\Rightarrow$ (1). So assume we have an algorithm $\mathbb{A}$ deciding $\text{mini}^{\|\ \|}\text{-}Q$ in $\leq f(k) \cdot n^d$ steps for some computable and easily reversible f and $d \in \mathbb{N}$. Given $x \in \Sigma^*$ we set

$$k := f^{-1}(\|x\|) - 1 \quad \text{and} \quad n := 2^{\|x\|/k}.$$

[2] Here and later the assumption "k in unary" is redundant, since k is the parameter.

Then, $\|x\| \leq k \cdot \log n$; moreover, by our assumption on f, the number k is computable in polynomial time and n in time $\leq |x|^{O(1)} + 2^{\|x\|/k} \leq \|x\|^{O(1)} + 2^{\|x\|/k} \in 2^{o^{\text{eff}}(\|x\|)}$. Once $\|x\|$, k, and n are determined, we apply the algorithm $\mathbb{A}$ to the input n, k, x, thus getting an answer to the question $x \in Q$. For its computation, $\mathbb{A}$ needs time

$$f(k) \cdot n^d = f(f^{-1}(\|x\|) - 1) \cdot (2^{\|x\|/k})^d \leq \|x\| \cdot 2^{d \cdot \|x\|/k} = 2^{d \cdot \|x\|/k + \log \|x\|} \in 2^{o^{\text{eff}}(\|x\|)}.$$

Altogether, Q is solvable in time $2^{o^{\text{eff}}(\|x\|)}$. □

Among other things, the following result shows that a polynomial time "linear size" reduction between two problems yields an fpt-reduction of their miniaturizations:

Proposition 10. *Let $Q_1 \subseteq \Sigma_1^*$ and $Q_2 \subseteq \Sigma_2^*$ and let $\| \, \|_i : \Sigma_i^* \to \mathbb{N}$ be a size function for $i = 1, 2$. Then, the following are equivalent:*

(1) There is a function $f : \Sigma_1^* \to \Sigma_2^*$ computable in time $2^{o^{\text{eff}}(\|x\|)}$ such that $\|f(x)\|_2 \in O(\|x\|_1)$ for all $x \in \Sigma_1^*$ and such that f is a reduction from Q_1 to Q_2, i.e., $x \in Q_1 \iff f(x) \in Q_2$.

(2) There is an fpt*-reduction R from* $\text{mini}^{\| \, \|_1}$*-Q_1 to* $\text{mini}^{\| \, \|_2}$*-Q_2 such that for any instance (n_1, k_1, x_1) of* $\text{mini}^{\| \, \|_1}$*-Q_1 with $\|x_1\|_1 \leq k_1 \cdot \log n_1$ we have $R(n_1, k_1, x_1) = (n_2, k_2, x_2)$ with $k_2 \in O(k_1)$ and $\|x_2\|_2 \leq k_2 \cdot \log n_2$.*

Proof: (1) ⇒ (2): Choose f according to (1). Then, for an instance (n_1, k_1, x_1) of $\text{mini}^{\| \, \|_1}$-Q_1 with $\|x_1\|_1 \leq k_1 \cdot \log n_1$, we have $\|f(x_1)\|_2 \leq c \cdot k_1 \cdot \log n_1$ for some constant c. Thus, we can set $R(n_1, k_1, x_1) = (n_1, c \cdot k_1, f(x_1))$. And we need time

$$n_1 + c \cdot k_1 + 2^{h(k_1 \cdot \log n_1)} \text{ for some } h \in o^{\text{eff}}(\text{id})$$

to compute $R(n_1, k_1, x_1)$, hence, to show that R is an fpt-reduction, it suffices to prove that for some computable function g

$$2^{h(k_1 \cdot \log n_1)} \quad \leq \quad g(k_1) + n_1$$

for all $k_1, n_1 \in \mathbb{N}$. Again, given k_1 we can compute $m \in \mathbb{N}$ such that $h(k_1 \cdot \log n_1) \leq \log n_1$ holds for all n_1 with $k_1 \cdot \log n_1 \geq m$. Hence, $2^{h(k_1 \cdot \log n_1)} \leq 2^{\log n_1} = n_1$ for such n_1.

(2) ⇒ (1): Choose R according to (2). Arguing as in Remark 7 (b), we can assume that $R(n_1, k_1, x_1)$ is computed in time $g(k_1) \cdot n_1^c$ for some easily reversible g and some constant c. We define the reduction f of Q_1 to Q_2 fulfilling (1):

Fix an instance x_1 of Q_1. We set

$$k_1 := \min \left\{ g^{-1}(\|x_1\|_1) - 1, \frac{\|x_1\|_1}{\log \|x_1\|_1} \right\} \quad \text{and} \quad n_1 := 2^{\|x_1\|_1/k_1}.$$

Then, $\|x_1\|_1 \leq k_1 \cdot \log n_1$. Let $R(n_1, k_1, x_1) = (n_2, k_2, x_2)$. We set $f(x_1) := x_2$. By the assumption on R in (2), we have $(x_1 \in Q_1 \iff f(x_1) \in Q_2)$.

Clearly, k_1 and n_1 can be computed in time $2^{o^{\text{eff}}(\|x_1\|_1)}$. And $R(n_1, k_1, x_1)$ can be computed in time $g(k_1) \cdot n_1^c \leq \|x_1\|_1 \cdot 2^{c \cdot \|x_1\|_1/k_1} \leq 2^{c \cdot \|x_1\|_1/k_1 + \log \|x_1\|_1} \in 2^{o^{\text{eff}}(\|x_1\|_1)}$. Furthermore,

$$\begin{aligned}
\|x_2\|_2 &\leq k_2 \cdot \log n_2 \quad \text{(by the assumption on } R \text{ in (2))} \\
&\leq O(k_1) \cdot \log\,(g(k_1) \cdot n_1^c) \\
&\leq O(k_1) \cdot \log 2^{c \cdot \|x_1\|_1/k_1 + \log \|x_1\|_1} \\
&\leq O(k_1) \cdot (c \cdot \frac{\|x_1\|_1}{k_1} + \log \|x_1\|_1) \\
&\in O(\|x_1\|_1) \quad \text{(by definition of } k_1\text{)}.
\end{aligned}$$

Altogether, we see that f satisfies (1). □

Remark 11. a) Take as Q_1 a language in $2^{O(|x|)} \setminus 2^{o(|x|)}$ and as Q_2 a language in $2^{o^{\text{eff}}(|x|)}$ complete for EXPTIME under polynomial time reductions. In particular, there is a polynomial time reduction from Q_1 to Q_2. By Proposition 10, $\text{mini}^{| \, |}$-$Q_1 \notin$ FPT and $\text{mini}^{| \, |}$-$Q_2 \in$ FPT. Hence, there is no fpt-reduction from $\text{mini}^{| \, |}$-Q_1 to $\text{mini}^{| \, |}$-Q_2. This example shows that the condition "$\|f(x)\|_2 \in O(\|x\|_1)$" in (1) of the preceding proposition cannot be weakened to "$\|f(x)\|_2 \leq q(\|x\|_1)$ for some polynomial q".

b) For a natural number $d \geq 1$ replace the condition $k_2 \in O(k_1)$ in (2) of Proposition 10 by $k_2 \in O(k_1^d)$. Then, along the lines of the preceding proof, one can show that there is a reduction f from Q_1 to Q_2 according to (1) satisfying $\|f(x)\|_2 \in O(\|x\|_1^d)$.

We close this section with some examples. Let CIRCSAT, SAT, and tSAT denote the satisfiability problem for circuits, for propositional formulas in CNF, and for formulas in tCNF, respectively. In Section 2, we defined $\|\mathcal{C}\|_0$, $\|\mathcal{C}\|_+$, and $\|\mathcal{C}\|_-$. Essentially they are the (total) size of a binary encoding of $\mathcal{C}$, the number of nodes + the number of lines of $\mathcal{C}$, and the number of nodes of $\mathcal{C}$, respectively.

In the following, we abbreviate mini$^{\|\ \|_+}$-CIRCSAT and mini$^{\|\ \|_-}$-CIRCSAT by mini$^+$-CIRCSAT and mini$^-$ -CIRCSAT, respectively. The same notations are used for other problems.

Taking as Q in Proposition 9 the circuit satisfiability problem CIRCSAT, we get the following result (cf. [1, 6]); it shows, for example, that mini$^+$-CIRCSAT $\in$ FPT is quite unlike.

Proposition 12. *(1) For* $\|\ \| \in \{\|\ \|_+, \|\ \|_-\}$*:* mini$^{\|\ \|}$-CIRCSAT $\in$ FPT *if and only if there is a subexponential algorithm for* CIRCSAT*, i.e., if there is an algorithm with running time* $2^{o^{\text{eff}}(\|\mathcal{C}\|)}$ *checking if the circuit* $\mathcal{C}$ *is satisfiable.*

(2) mini$^{\|\ \|_0}$-CIRCSAT $\in$ FPT.

Proof: Part (1) is clear by Proposition 9. For part (2), it suffices to consider circuits $\mathcal{C}$, whose underlying graph is connected. Since such a graph with n nodes has at least $n-1$ edges, we see that $\|\mathcal{C}\|_0 \geq i + 2 \cdot (i-1) \cdot \log i$, where i is the number of input nodes of $\mathcal{C}$. Thus, $i \in o^{\text{eff}}(\|\mathcal{C}\|_0)$. Hence, for some polynomial p, the satisfiability of $\mathcal{C}$ can be checked in time

$$2^i \cdot p(\|\mathcal{C}\|_0) \leq 2^{o^{\text{eff}}(\|\mathcal{C}\|_0)} \cdot p(\|\mathcal{C}\|_0) \leq 2^{o^{\text{eff}}(\|\mathcal{C}\|_0)}.$$

Therefore, mini$^{\|\ \|_0}$-CIRCSAT $\in$ FPT follows by Proposition 9. □

By Proposition 10, the well-known linear reductions between CIRCSAT, SAT, and 3SAT yield:

– mini$^+$-CIRCSAT $=^{\text{fpt}}$ mini$^+$-SAT $=^{\text{fpt}}$ mini$^+$-3SAT.

Denote by VC, IS, and CLIQUE the vertex cover problem, the independent set problem, and the clique problem, respectively; e.g., the instances to CLIQUE consist of pairs (G, r), where $G = (V, E)$ is a graph and r is a natural number with $r \leq |V|$. $(G, r) \in$ CLIQUE if and only if there is a clique of size r in G. We let $\|(G, r)\|_- = \|G\|_-$ and $\|(G, r)\|_+ = \|G\|_+$ and use the analogous notations for VC and IS.

By Lemma 8, the mini$^+$ versions of these problems are fpt-reducible to their mini$^-$ versions. Using this fact and well-known linear reductions between the corresponding problems we get the following fpt-reductions between their miniaturized versions:

– mini$^+$-3SAT $\leq^{\text{fpt}}$ mini$^+$-VC$=^{\text{fpt}}$ mini$^+$-IS$\leq^{\text{fpt}}$ mini$^-$-IS$=^{\text{fpt}}$ mini$^-$-VC$=^{\text{fpt}}$ mini$^-$-CLIQUE.

For the last two equalities, we use the trivial equivalences:

$$\begin{aligned} ((V, E), r) \in \text{VC} &\iff ((V, E), |V| - r) \in \text{IS} \\ ((V, E), r) \in \text{IS} &\iff ((V, E^{\text{comp}}), r) \in \text{CLIQUE} \end{aligned}$$

(here, (V, E^{comp}) is the complement of the graph (V, E), that is, the graph that has precisely those edges that are missing from (V, E)). The last equivalence yields a linear reduction for the size function $\|\ \|_-$ only.

4. The miniaturization of Fagin-definable problems

We already mentioned in the introduction that the miniaturization may increase the computational complexity of a problem; e.g., the parameterized vertex cover problem is fixed-parameter tractable while its miniaturization is not (unless MINI[1] = FPT). As a further example consider the halting problem HP for Turing machines. It is well-known that the corresponding parameterized problem, parameterized by the number of steps, is in W[1], indeed W[1]-complete. But it is not known, whether

$\text{mini}^{|\ |}$-HP $\in$ W[1]; we conjecture that this is not the case. However, in this section we show that for every problem in W$[t]$ that is Fagin-definable by a first-order formula with a *set* variable, the miniaturization of the problem is itself in W$[t]$.

We start by recalling the definition of Fagin-definable problem. Let τ be a vocabulary and $\mathbf{C}$ a class of τ-structures decidable in polynomial time. Let $\varphi(X)$ be a first-order formula of vocabulary τ with the free set variable X; it defines a parameterized problem $W_{\varphi(X)}$ on $\mathbf{C}$ given by:

$W_{\varphi(X)}(\mathbf{C})$	
Input:	A structure $\mathcal{A}$ in $\mathbf{C}$.
Parameter:	$r \in \mathbb{N}$ with $r \leq \|A\|$.
Problem:	Decide if there is a subset S of A of cardinality r satisfying $\varphi(X)$ in $\mathcal{A}$, i.e., with $\mathcal{A} \models \varphi(S)$.

We say that $\varphi(X)$ *Fagin-defines* $W_{\varphi(X)}(\mathbf{C})$ *on* $\mathbf{C}$ and that a parameterized problem $Q \subseteq \mathbf{C}$ is *Fagin-definable*, if $Q = W_{\varphi(X)}(\mathbf{C})$ for some $\varphi(X)$.

For example, the vertex cover problem, the independent set problem, and the dominating set problem DS are Fagin-defined on the class GRAPH of all graphs by

$$\forall y \forall z\big(Eyz \rightarrow (Xy \vee Xz)\big),\ \ \forall y \forall z\big((Xy \wedge Xz) \rightarrow \neg Eyz\big),\ \text{ and }\ \forall y \exists z\big(Xz \wedge (y = z \vee Eyz)\big),$$

respectively.

If $\mathbf{C}$ is the class of all τ-structures, we denote $W_{\varphi(X)}(\mathbf{C})$ by $W_{\varphi(X)}$. For notational simplicity, we formulate most results for Fagin-definable problems $W_{\varphi(X)}$. The extensions to Fagin-definable problems $W_{\varphi(X)}(\mathbf{C})$ for other classes $\mathbf{C}$ are easy; mostly, they use the claim of the next lemma whose proof is straightforward.

We defined the miniaturization for classical problems only. Here and later when speaking of the miniaturization of a parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$, we consider it as a classical problem in some larger alphabet, say, the alphabet obtained from Σ by adding new symbols '(', ',', ')', '0' and '1'. And again for a structure $\mathcal{A}$ and $r \in \mathbb{N}$ with $r \leq |A|$, we set $\|(\mathcal{A}, r)\|_- = \|\mathcal{A}\|_-$ and $\|(\mathcal{A}, r)\|_+ = \|\mathcal{A}\|_+$. E.g., mini^--$W_{\varphi(X)}$, for a τ-formula $\varphi(X)$, is the problem

mini^--$W_{\varphi(X)}$	
Input:	$n, k \in \mathbb{N}$ in unary, a τ-structure $\mathcal{A}$ with $\|A\| \leq k \cdot \log n$, and $r \in \mathbb{N}$ with $r \leq \|A\|$.
Parameter:	k.
Problem:	Decide if there is a subset S of A of cardinality r with $\mathcal{A} \models \varphi(S)$.

Lemma 13. *For any class* $\mathbf{C}$ *of* τ*-structures in* PTIME *and any* Π_t*-formula* $\varphi(X)$ *of vocabulary* τ*,*

$$\text{mini}^{\pm}\text{-}W_{\varphi(X)}(\mathbf{C}) \quad \leq^{\text{fpt}} \quad \text{mini}^{\pm}\text{-}W_{\varphi(X)}.$$

It was shown in [7] (see [10], too) that for $t \geq 1$

$$\begin{aligned} \text{W}[t] \quad &= \quad [\{W_{\varphi(X)} \mid \varphi(X) \text{ a } \Pi_t\text{-formula}\}]^{\text{fpt}} \\ (&= \quad [\{W_{\varphi(X)}(\mathbf{C}) \mid \varphi(X) \text{ a } \Pi_t\text{-formula and } \mathbf{C} \text{ a PTIME-class of } \tau\text{-structures}\}]^{\text{fpt}}). \end{aligned}$$

In parameterized complexity theory, satisfiability problems for propositional formulas mostly are considered as weighted satisfiability problems; but, in the context of miniaturizations the unweighted form is relevant, too. We introduce the notion of unweighted Fagin-definable problems and show that the miniaturized weighted and unweighted problems have the same computational complexity. Using this result, we prove that, for every $t \geq 1$, the miniaturization of a Fagin-definable problem in W$[t]$ lies in W$[t]$, too.

Definition 14. Let τ be a vocabulary and $\mathbf{C}$ a class of τ-structures decidable in polynomial time. Let $\varphi(X)$ be a first-order formula of vocabulary τ with the free set variable X; it defines a classical problem $U_{\varphi(X)}(\mathbf{C})$, the *unweighted problem on* $\mathbf{C}$ *Fagin-defined by* φ, given by:

$U_{\varphi(X)}(\mathbf{C})$ *Input:* A structure $\mathcal{A}$ in $\mathbf{C}$. *Problem:* Decide if there is a subset S of A with $\mathcal{A} \models \varphi(S)$.

Again we write $U_{\varphi(X)}$ for $U_{\varphi(X)}(\mathbf{C})$, if $\mathbf{C}$ is the class of all τ-structures and, since the analogue of Lemma 13 holds for unweighted definable problems, we formulate most results for problems of the form $U_{\varphi(X)}$.

Example 15. A circuit is a *circuit with small gates*, if every and-gate and every or-gate has fan-in two. We denote by SMALLCIRC the class of circuits with small gates and by SMALLCIRCSAT the satisfiability problem for circuits with small gates. Then,

$$U_{\varphi_{\text{scirc}}(X)}(\text{SMALLCIRC}) \quad = \quad \text{SMALLCIRCSAT}$$

where $\varphi_{\text{scirc}}(X) := \forall x \forall y \forall z \psi_{\text{scirc}}(x, y, z, X)$ with

$$\begin{aligned} \psi_{\text{scirc}} \quad := \quad & ((G_{\neg}x \wedge Exy) \rightarrow (Xx \leftrightarrow \neg Xy)) \\ & \wedge ((G_{\wedge}x \wedge Exy \wedge Exz \wedge y \neq z) \rightarrow (Xx \leftrightarrow (Xy \wedge Xz))) \\ & \wedge ((G_{\vee}x \wedge Exy \wedge Exz \wedge y \neq z) \rightarrow (Xx \leftrightarrow (Xy \vee Xz))) \\ & \wedge (Ox \rightarrow Xx). \end{aligned}$$

Moreover, for every circuit with small gates $\mathcal{C}$ and any subset S_0 of its set $I^{\mathcal{C}}$ of input nodes, we have:

$$\mathcal{C} \models \varphi_{\text{scirc}}(S) \text{ for some } S \text{ with } S \cap I^{\mathcal{C}} = S_0 \quad \iff \quad S_0 \text{ satisfies } \mathcal{C}.$$

Here, by S_0 satisfies $\mathcal{C}$, we mean that the assignment setting the input nodes in S_0 to TRUE and all other input nodes to FALSE satisfies $\mathcal{C}$.

For the proof of the main theorem of this section, we need the following well-known result (e.g., cf. [4]):

Lemma 16. *There is a map, computable in polynomial time, that to $n \geq 1$ (in unary) and $r \leq n$ assigns a circuit with small gates $\mathcal{C}_{n,r}$ with $\|\mathcal{C}_{n,r}\|_+ \in O(n)$ satisfied exactly by the assignments of weight r.*

For $q \in \mathbb{N}$, $q \geq 1$, denote by $\Pi_{1,q}$ the set of Π_1-formulas whose block of quantifiers has length q. In the following lemma, weighted Fagin-definable problems are considered as classical problems.

Lemma 17. *Assume $q \geq 3$ and $\varphi(X) \in \Pi_{1,q}$. Then:*

(1) There is $\varphi_1(X) \in \Pi_{1,q}$ and a reduction f from $U_{\varphi(X)}$ to $W_{\varphi_1(X)}$ computable in polynomial time with $\|f(\mathcal{A})\|_{\pm} \in O(\|\mathcal{A}\|_{\pm})$ for all τ structures $\mathcal{A}$.

(2) There is $\varphi_2(X) \in \Pi_{1,q}$ and a reduction f from $W_{\varphi(X)}$ to $U_{\varphi_2(X)}$ computable in polynomial time with $\|f((\mathcal{A}, r))\|_{\pm} \in O(\|(\mathcal{A}, r)\|_{\pm})$ for all τ structures $\mathcal{A}$ and $r \in \mathbb{N}$ with $r \leq |A|$.

Remark 18. Apparently, claim (2) is not true for $q = 2$; in fact, it is not hard to show that, for $\varphi(X) \in \Pi_{1,2}$, the problem $U_{\varphi(X)}$ is reducible to 2SAT in polynomial time and hence, is solvable in polynomial time. On the other hand, the vertex cover problem has the form $W_{\varphi(X)}$ for some $\varphi \in \Pi_{1,2}$.

Before proving this lemma, we remark that by Proposition 10 it yields:

Proposition 19. *For $q \geq 3$,*

(1) $[\{\text{mini}^+\text{-}U_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,q}\}]^{\text{fpt}} = [\{\text{mini}^+\text{-}W_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,q}\}]^{\text{fpt}}$;

(2) $[\{\text{mini}^-\text{-}U_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,q}\}]^{\text{fpt}} = [\{\text{mini}^-\text{-}W_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,q}\}]^{\text{fpt}}$.

Proof of Lemma 17: Let $\varphi(X) \in \Pi_{1,q}$ be the τ-formula

$$\varphi(X) = \forall x_1 \ldots \forall x_q \psi$$

with quantifier-free ψ.

To (1): For a new unary relation symbol P, we set

$$\varphi_1(X) = \forall x_1 \ldots \forall x_q ((Px_1 \wedge \ldots \wedge Px_q) \to \psi).$$

Clearly, $\varphi_1(X) \in \Pi_{1,q}$. For a τ-structure $\mathcal{A}$, let $\hat{A} := \{\hat{a} \mid a \in A\}$ be a disjoint copy of A. Define the $\tau \cup \{P\}$-structure $\mathcal{A}_1$ by

$$A_1 := A \cup \hat{A};\ P^{\mathcal{A}_1} := A;\ R^{\mathcal{A}_1} := R^{\mathcal{A}} \text{ for } R \in \tau.$$

One easily verifies that

$$\mathcal{A} \models \exists X \varphi(X) \quad \Longleftrightarrow \quad \text{there is } S \subseteq A_1 \text{ with } (\mathcal{A}_1 \models \varphi_1(S) \text{ and } |S| = |A|).$$

Hence, $\mathcal{A} \mapsto (\mathcal{A}_1, |A|)$ yields the desired reduction from $U_{\varphi(X)}$ to $W_{\varphi_1(X)}$.

To (2): For a τ-structure $\mathcal{A}$ and $r \leq |A|$ choose the small circuit $\mathcal{C}_{|A|,r}$ according to Lemma 16. We may assume that A is the set of input nodes of $\mathcal{C}_{|A|,r}$, $I^{\mathcal{C}_{|A|,r}} = A$. We expand $\mathcal{C}_{|A|,r}$ to a $(\tau_{\text{scirc}} \dot\cup \tau)$-structure $\mathcal{B}$ by setting $R^{\mathcal{B}} := R^{\mathcal{A}}$ for $R \in \tau$. Let ψ_{scirc} be as in Example 15. We identify x, y, z (in ψ_{scirc}) with x_1, x_2, x_3, respectively, and let

$$\varphi_2(X) := \forall x_1 \ldots \forall x_q (\psi_{\text{scirc}} \wedge ((Ix_1 \wedge \ldots \wedge Ix_q) \to \psi)).$$

Then,

$$\begin{array}{ll} & (\mathcal{A}, r) \in W_{\varphi(X)} \\ \Longleftrightarrow & \text{there is } S_0 \subseteq A \text{ with } (\mathcal{A} \models \varphi(S_0) \text{ and } |S_0| = r) \\ \Longleftrightarrow & \text{there is } S_0 \subseteq A \text{ with } (\mathcal{A} \models \varphi(S_0) \text{ and } S_0 \text{ satisfies } \mathcal{C}_{|A|,r}) \qquad \text{(by Lemma 16)} \\ \Longleftrightarrow & \text{there is } S \subseteq B \text{ with } \mathcal{B} \models \varphi_2(S) \qquad \text{(by Example 15 and by definition of } \mathcal{B} \text{ and of } \varphi_2(X)) \\ \Longleftrightarrow & \mathcal{B} \in U_{\varphi_2(X)}. \end{array}$$

Therefore, the mapping $(\mathcal{A}, r) \mapsto \mathcal{B}$ is the desired reduction from $W_{\varphi(X)}$ to $U_{\varphi_2(X)}$. □

Essentially in the same way one can show:

Proposition 20. *For $t \geq 1$,*

(1) $[\{\text{mini}^+\text{-}U_{\varphi(X)} \mid \varphi(X) \in \Pi_t\}]^{\text{fpt}} = [\{\text{mini}^+\text{-}W_{\varphi(X)} \mid \varphi(X) \in \Pi_t\}]^{\text{fpt}}$;

(2) $[\{\text{mini}^-\text{-}U_{\varphi(X)} \mid \varphi(X) \in \Pi_t\}]^{\text{fpt}} = [\{\text{mini}^-\text{-}W_{\varphi(X)} \mid \varphi(X) \in \Pi_t\}]^{\text{fpt}}$.

Now, we are in a position to prove the main result of this section:

Theorem 21. *Let $t \geq 1$ and $\varphi(X) \in \Pi_t$. Then,* $\text{mini}^+\text{-}W_{\varphi(X)}$ *and* $\text{mini}^-\text{-}W_{\varphi(X)}$ *are in* W[t].
Moreover, $\text{mini}^-\text{-}U_{\varphi(X)}$ *and* $\text{mini}^+\text{-}U_{\varphi(X)}$ *are in* W[t].

In Section 4 we presented Fagin-definitions of VC and IS by Π_1-formulas and of DS by a Π_2-formula, hence:

Corollary 22. mini$^-$-VC*,* mini$^-$-IS*,* mini$^-$-CLIQUE $\in$ W[1] *and* mini$^-$-DS $\in$ W[2].

Proof of Theorem 21: By Proposition 20 it suffices to show the corresponding result for the unweighted problems. Since $\text{mini}^+\text{-}U_{\varphi(X)} \leq^{\text{fpt}} \text{mini}^-\text{-}U_{\varphi(X)}$, we have to prove that

$$\text{mini}^-\text{-}U_{\varphi(X)} \quad \in \quad \text{W}[t]$$

for $\varphi(X) \in \Pi_t$. Let t be odd (the case t even is treated similarly) and

$$\varphi(X) \quad := \quad \forall \bar{x}_1 \exists \bar{x}_2 \ldots \forall \bar{x}_t \bigwedge_{i=1}^{r} (\lambda_{i1} \vee \ldots \vee \lambda_{im_i}),$$

where the λ_{ij} are literals, i.e., atomic or negated atomic formulas. We set $d := \max\{2, m_1, \ldots, m_r\}$. We show that mini$^-$-$U_{\varphi(X)}$ is fpt-reducible to the parameterized weighted satisfiability problem WSAT$(\Gamma_{t,d})$, a problem in W$[t]$.

We consider an instance of mini$^-$-$U_{\varphi(X)}$ consisting of n, k (in unary) and a structure $\mathcal{A}$ with $|A| \leq k \cdot \log n$. We construct a formula $\alpha \in \Gamma_{t,d}$ such that

$$(n, k, \mathcal{A}) \in \text{mini}^-\text{-}U_{\varphi(X)} \quad \Longleftrightarrow \quad \alpha \text{ is } k\text{-satisfiable.}$$

We first introduce a formula α_0; for every $a \in A$, it has a propositional variable Y_a with the intended meaning "a is in (the interpretation of) X". We set

$$\alpha_0 \quad := \quad \bigwedge_{\bar{a}_1 \in A^{\text{length}(\bar{x}_1)}} \bigvee_{\bar{a}_2 \in A^{\text{length}(\bar{x}_2)}} \cdots \bigvee_{\bar{a}_{t-1} \in A^{\text{length}(\bar{x}_{t-1})}} \bigwedge_{\substack{\bar{a}_t \in A^{\text{length}(\bar{x}_t)}, \\ 1 \leq i \leq r}} \gamma(\bar{a}_1, \ldots, \bar{a}_t, i).$$

Here, $\gamma(\bar{a}_1, \ldots, \bar{a}_t, i)$ is a disjunction obtained by replacing in $(\lambda_{i1} \vee \ldots \vee \lambda_{im_i})$

– every literal λ not containing X by TRUE or FALSE according to whether the assignment $\bar{a}_1 \ldots \bar{a}_t$ (to $\bar{x}_1 \ldots \bar{x}_t$) satisfies λ in $\mathcal{A}$;

– every literal Xx_{uv} (x_{uv} denotes the vth variable in $\bar{x}_u$) and $\neg Xx_{uv}$ by $Y_{a_{uv}}$ and $\neg Y_{a_{uv}}$, respectively.

It should be clear that

$$(n, k, \mathcal{A}) \in \text{mini}^-\text{-}U_{\varphi(X)} \quad \Longleftrightarrow \quad \alpha_0 \text{ is satisfiable.}$$

Since $|A| \leq k \cdot \log n$, the formula α_0 has $\leq k \cdot \log n$ variables. To get the formula α, we apply what is called the *$k \cdot \log n$ trick* in [8]: We think of the $(\leq)\, k \cdot \log n$ variables of α_0 as being arranged in k blocks of $\log n$ variables, say, $\{Y_a \mid a \in A\} = \{Y_{uv} \mid 1 \leq u \leq k,\ 1 \leq v \leq \log n\}$. Let us obtain α by introducing k blocks of n new propositional variables, say, $\{Z_{uw} \mid 1 \leq u \leq k,\ 1 \leq w \leq n\}$ and by replacing in α_0

– any literal Y_{uv} by the big conjunction of all $\neg Z_{uw}$ such that the vth bit of the binary representation of w is 0,

– any literal $\neg Y_{uv}$ by the big conjunction of all $\neg Z_{uw}$ such that the vth bit of the binary representation of w is 1,

and by adding the conjunctions $\bigwedge_{1 \leq w < w' \leq n} (\neg Z_{uw} \vee \neg Z_{uw'})$ for $u = 1, \ldots, k$. One easily verifies that α is equivalent to a formula in $\Gamma_{t,d}$ and that

$$\begin{aligned} (n, k, \mathcal{A}) \in \text{mini}^-\text{-}U_{\varphi(X)} \quad &\Longleftrightarrow \quad \alpha_0 \text{ is satisfiable} \\ &\Longleftrightarrow \quad \alpha \text{ is } k\text{-satisfiable,} \end{aligned}$$

and hence, mini$^-$-$U_{\varphi(X)} \leq^{\text{fpt}}$ WSAT$(\Gamma_{t,d})$. □

5. The M$^-$-hierarchy

The previous analysis suggests the definition of a hierarchy $(\text{M}^-[t])_{t \geq 1}$ of classes of parameterized problems within W[1]. In this section, after introducing this hierarchy, we show that mini$^-$-tSAT is complete in M$^-[t]$.

For $t \geq 1$, we introduce the class M$^-[t]$ by

$$\text{M}^-[t] \quad := \quad [\{\text{mini}^-\text{-}U_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,t}\}]^{\text{fpt}}.$$

By Theorem 21,

$$\text{M}^-[1] \subseteq \text{M}^-[2] \subseteq \ldots \subseteq \text{M}^-[t] \subseteq \ldots \text{W}[1]. \tag{1}$$

Moreover, by Proposition 19, for $t \geq 3$,

$$\mathrm{M}^{-}[t] \quad = \quad [\{\text{mini}^{-}\text{-}W_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,t}\}]^{\text{fpt}}.$$

We need the fact that a Π_1-formula, whose block of universal quantifiers has length m, holds in a structure $\mathcal{A}$ if and only if it holds in every substructure of $\mathcal{A}$ generated by at most m elements, more precisely:

Lemma 23. *Let $\mathcal{A}$ be a τ-structure and*

$$\varphi(X) = \forall x_1 \ldots \forall x_m \psi$$

a τ-formula with quantifier-free ψ. Then, for a subset S of A, we have $\mathcal{A} \models \varphi(S)$ if and only if for all $A_0 = \{a_1, \ldots, a_m\} \subseteq A$:

$$\mathcal{A} \upharpoonright A_0 \quad \models \quad \psi(a_1, \ldots, a_m, S \cap A_0).$$

Theorem 24. *For all $t \geq 2$,* mini^{-}-tSAT *is complete in* $\mathrm{M}^{-}[t]$.

Proof: First we show that mini^{-}-tSAT $\in \mathrm{M}^{-}[t]$. Assume given a propositional formula in tCNF. Recall (cf. Section 2.2) that we identify a propositional formula α with $\mathcal{A}(\alpha)$ and that $\mathcal{A}(\alpha)$ is the $\tau_t = \{N, C\}$-structure with universe $\{X, \neg X \mid X \text{ is a variable of } \alpha\}$, and

$$\begin{aligned} N^{\mathcal{A}(\alpha)} \quad &:= \quad \{(X, \neg X) \mid X \text{ is a variable of } \alpha\}; \\ C^{\mathcal{A}(\alpha)} \quad &:= \quad \{(\lambda_1, \ldots, \lambda_t) \mid (\lambda_1 \vee \ldots \vee \lambda_t) \text{ is a clause of } \alpha\}. \end{aligned}$$

For $t \geq 2$, we have tSAT $= U_{\varphi_0(Y)}(t\text{CNF})$ for

$$\varphi_0(Y) \quad := \quad \forall x_1 \ldots \forall x_t ((N x_1 x_2 \rightarrow (Y x_1 \leftrightarrow \neg Y x_2)) \wedge (C x_1 \ldots x_t \rightarrow (Y x_1 \vee \ldots \vee Y x_t))).$$

Therefore, mini^{-}-tSAT $=$ mini^{-}-$U_{\varphi_0(Y)}(t\text{CNF})$ and hence, mini^{-}-tSAT $\in \mathrm{M}^{-}[t]$.

It remains to show that mini^{-}-tSAT is hard for $\mathrm{M}^{-}[t]$. For this purpose, let $\varphi(X) \in \Pi_{1,t}$, say

$$\varphi(X) \quad = \quad \forall x_1 \ldots \forall x_t \psi$$

with quantifier-free ψ. We show that mini^{-}-$U_{\varphi(X)} \leq^{\text{fpt}}$ mini^{-}-tSAT.

Assume given an instance $n, k, \mathcal{A}$ of mini^{-}-$U_{\varphi(X)}$ with $|A| \leq k \cdot \log n$. We define a tCNF-formula α such that ($\mathcal{A} \models \exists X \varphi(X) \iff \alpha$ is satisfiable). The formula α has propositional variables X_a for $a \in A$, again with the intended meaning "a is in X". Fix $a_1, \ldots, a_t \in A$. For $s_1, \ldots, s_t \in \{0, 1\}$ with $s_i = s_j$ if $a_i = a_j$, we let

$$S^{s_1, \ldots, s_t}_{a_1, \ldots, a_t} := \{a_i \mid 1 \leq i \leq t,\ s_i = 0\} \quad \text{and} \quad \beta^{s_1, \ldots, s_t}_{a_1, \ldots, a_t} := (X^{s_1}_{a_1} \vee \ldots \vee X^{s_t}_{a_t}),$$

where $X^1_a := X_a$ and $X^0_a := \neg X_a$. Moreover, let

$$\begin{aligned} \alpha_{a_1, \ldots, a_t} := \bigwedge \{\beta^{s_1, \ldots, s_t}_{a_1, \ldots, a_t} \mid & s_1, \ldots, s_t \in \{0, 1\} \text{ with } s_i = s_j \text{ if } a_i = a_j, \\ & \text{and with } \mathcal{A} \upharpoonright \{a_1, \ldots, a_t\} \not\models \psi(a_1, \ldots, a_t, S^{s_1, \ldots, s_t}_{a_1, \ldots, a_t})\}. \end{aligned}$$

Finally, we let $\alpha := \bigwedge \{\alpha_{a_1, \ldots, a_t} \mid a_1, \ldots, a_t \in A\}$. Then, $\|\alpha\|_{-} = |A(\alpha)| = 2 \cdot |A|$. We show:

$$\mathcal{A} \models \exists X \varphi(X) \quad \iff \quad \alpha \text{ is satisfiable.} \tag{2}$$

In fact, let b be an assignment satisfying α. We let $S := \{a \in A \mid b(X_a) = \text{TRUE}\}$ and show that $\mathcal{A} \models \varphi(S)$. By Lemma 23, it suffices to show that $\mathcal{A} \upharpoonright \{a_1, \ldots, a_t\} \models \psi(a_1, \ldots, a_t, S \cap \{a_1, \ldots, a_t\})$ for all $a_1, \ldots, a_t \in A$. Fix $a_1, \ldots, a_t \in A$ and by contradiction, assume that $\mathcal{A} \upharpoonright \{a_1, \ldots, a_t\} \not\models \psi(a_1, \ldots, a_t, S \cap \{a_1, \ldots, a_t\})$. Choose $s_1, \ldots, s_t \in \{0, 1\}$ with $s_i = s_j$ if $a_i = a_j$ such that

$$S \cap \{a_1, \ldots, a_t\} \quad = \quad S^{s_1, \ldots, s_t}_{a_1, \ldots, a_t} \tag{3}$$

Then, by definition of $\alpha_{a_1,\ldots,a_t}$, $b(\beta^{s_1,\ldots,s_t}_{a_1,\ldots,a_t}) = \text{TRUE}$ and hence, $b(X^{s_i}_{a_i}) = \text{TRUE}$ for some $i \in \{1,\ldots,t\}$. But

$$\begin{array}{rcll} a_i \in S & \iff & b(X_{a_i}) = \text{TRUE} & \text{(by definition of } S) \\ & \iff & X_{a_i} = X^{s_i}_{a_i} & \text{(since } b(X^{s_i}_{a_i}) = \text{TRUE}) \\ & \iff & s_i = 1 & \\ & \iff & a_i \notin S^{s_1,\ldots,s_t}_{a_1,\ldots,a_t} & \text{(by definition of } S^{s_1,\ldots,s_t}_{a_1,\ldots,a_t}), \end{array}$$

a contradiction to (3). The proof of the other direction of (2) is similar. □

Corollary 25. $\text{M}^-[1] = \text{M}^-[2] = \text{FPT}$.

Proof: It is well-known that 2SAT is in PTIME, thus mini$^-$-2SAT $\in$ FPT by part c) of Remark 7. Hence, $\text{M}^-[2] = \text{FPT}$ by the preceding theorem. □

Is the hierarchy in (1) (starting with $t = 2$) strict? By the preceding theorem, we know that mini$^-$-tSAT is $\text{M}^-[t]$-complete; hence, in connection with this problem one should mention the result of Impagliazzo and Paturi [13] that, assuming the exponential time hypothesis, the complexity of tSAT increases with t.

We can also look at Fagin-definability as a model-checking problem. For a size function $\| \ \|$ on structures and $t \geq 1$, we directly consider the miniaturized version:

> $\text{MC}^{\| \ \|}(\exists^1\Pi_{1,t})$
> *Input:* $n, k \in \mathbb{N}$ in unary, a vocabulary τ, a structure $\mathcal{A}$ of vocabulary τ with $\|\mathcal{A}\| \leq k \cdot \log n$, and a τ-sentence $\exists X \forall x_1 \ldots \forall x_t \psi$ with quantifier-free ψ.
> *Parameter:* k.
> *Problem:* Decide if $\mathcal{A} \models \exists X \forall x_1 \ldots \forall x_t \psi$.

Again, write $\text{MC}^+(\exists^1\Pi_{1,t})$ for $\text{MC}^{\| \ \|+}(\exists^1\Pi_{1,t})$ and $\text{MC}^-(\exists^1\Pi_{1,t})$ for $\text{MC}^{\| \ \|-}(\exists^1\Pi_{1,t})$. Then:

Proposition 26. $\text{MC}^-(\exists^1\Pi_{1,t})$ *is complete for* $\text{M}^-[t]$.

Proof: Clearly, mini$^-$-$U_{\varphi(X)} \leq^{\text{fpt}} \text{MC}^-(\exists^1\Pi_{1,t})$ for any $\varphi(X) \in \Pi_{1,t}$, hence $\text{MC}^-(\exists^1\Pi_{1,t})$ is hard for $\text{M}^-[t]$. The second part of the proof of the previous theorem shows that $\text{MC}^-(\exists^1\Pi_{1,t}) \leq^{\text{fpt}}$ mini$^-$-tSAT. □

Of course, we could also introduce a hierarchy $(\text{M}^+[t])_{t\geq 1}$ by

$$\text{M}^+[t] \quad := \quad [\{\text{mini}^+\text{-}U_{\varphi(X)} \mid \varphi(X) \in \Pi_{1,t}\}]^{\text{fpt}}.$$

By Lemma 8, $\text{M}^+[t] \subseteq \text{M}^-[t]$ for all $t \geq 1$. Arguing as in the corresponding parts of the proofs of Theorem 24 and Proposition 26, one can show that mini$^+$-tSAT and $\text{MC}^+(\exists^1\Pi_{1,t})$ are in $\text{M}^+[t]$, but we do not know whether these problems are complete for $\text{M}^+[t]$. If mini$^+$-tSAT is complete for $\text{M}^+[t]$ for $t \geq 3$, then the M^+-hierarchy collapses: $\text{M}^+[3] = \text{M}^+[4] = \ldots$, since, by Proposition 10, the usual reduction from tSAT to 3SAT yields an fpt-reduction from mini$^+$-tSAT to mini$^+$-3SAT.

Summing up, we have

$$[\text{mini}^+\text{-}t\text{SAT}]^{\text{fpt}} \subseteq \text{M}^+[t] \subseteq [\text{MC}^+(\exists^1\Pi_{1,t})]^{\text{fpt}} \subseteq \text{M}^-[t]$$

and we do not know, if (for $t \geq 3$) any inclusion can be replaced by an equality.

In [6], the class MINI[1] (sometimes denoted by M[1]) was introduced:

$$\text{MINI}[1] \quad := \quad [\text{mini}^+\text{-CIRCSAT}]^{\text{fpt}}.$$

Since mini$^+$-CIRCSAT $=^{\text{fpt}}$ mini$^+$-3SAT $\leq^{\text{fpt}}$ mini$^-$-3SAT, we know by Theorem 24 that $\text{MINI}[1] \subseteq \text{M}^-[3]$.

Sometimes (e.g., in [6, 9]), in connection with the class MINI[1], Turing reductions (more precisely, parameterized Turing reductions) have been considered; the fpt-reductions considered so far in this paper were many-one reductions.

¿From the point of view of Turing reductions nearly all problems considered here have the same complexity. In fact, it has been implicitly shown in Impagliazzo, Paturi and Zane [14] that for $t \geq 3$, there is a parameterized Turing reduction from mini$^-$-tSAT to mini$^+$-tSAT, hence, these two problems are Turing equivalent. In particular, if we denote by $\text{MINI}_T[1]$ and $\text{M}^-_T[t]$ the closure under parameterized Turing reductions of MINI[1] and $\text{M}^-[t]$, respectively, we have $\text{MINI}_T[1] = \text{M}^-_T[3]$, but also $\text{M}^-_T[t] = \text{M}^-_T[3]$ for all $t \geq 3$. We refer the reader to [11] for a detailed proof.

6. Subexponential Time and $f(k) \cdot n^{o^{\text{eff}}(k)}$ algorithms

It is well-known that CLIQUE is complete for W[1]. Hence, if CLIQUE $\in$ FPT then FPT $=$ W[1] and thus, by Corollary 22, mini$^-$-CLIQUE $\in$ FPT, which by Proposition 9 implies that $(V, E) \in$ CLIQUE is solvable in time $2^{o^{\text{eff}}(|V|)}$. Recently, J. Chen et al. strengthened this result by showing (Theorem 5.2 in [3]):

Theorem 27. *If* CLIQUE *can be decided in time $f(k) \cdot |V|^{o^{\text{eff}}(k)}$ for some computable function f, then* CLIQUE *is solvable in time $2^{o^{\text{eff}}(|V|)}$.*

Similarly as above for the clique problem, one sees that FPT $=$ W[2] implies that the dominating set problem DS is solvable in time $2^{o^{\text{eff}}(|V|)}$. In this section we present an extension of this fact in the spirit of Theorem 27.

The proof of Theorem 27 in [3] is based on an argument via the weighted satisfiability problem of formulas in 2CNF. Here we sketch another direct proof. For this purpose, we show a "decomposition" lemma for the clique problem.

Lemma 28. *Given a graph $G = (V, E)$ and $r, m \leq |V|$, in time polynomial in $|V| \cdot 2^m$ we can compute a graph $G' = (V', E')$ with $|V'| \leq |V|^2 \cdot 2^m$ such that:*

$$G \text{ has a clique of size } r \iff G' \text{ has a clique of size } r' \text{ with } r' := \lceil |V|/m \rceil. \tag{4}$$

Proof: Let G, r, m, r' be as stated. We partition the vertex set V of G arbitrarily into r' disjoint sets $V_1, \ldots, V_{r'}$, each containing not more than m vertices. For $i = 1, \ldots, k'$, we set

$$\begin{aligned} H_1 &:= \{(S, |S|) \mid S \subseteq V_1 \text{ and } S \text{ is a clique in } G\}, \\ H_i &:= \{(S, s) \mid S \subseteq V_i,\ S \text{ is a clique in } G, \text{ and } s \leq r\}, \quad \text{for } 1 < i < r', \\ H_{r'} &:= \{(S, r) \mid S \subseteq V_{r'} \text{ and } S \text{ is a clique in } G\}. \end{aligned}$$

G' has vertex set $V' := \bigcup_{i=1}^{r'} H_i$. Thus, $|V'| \leq |V|^2 \cdot 2^m$. For any $i \leq j$ and $u = (S, s) \in H_i$, $v = (T, t) \in H_j$, we let $\{u, v\} \in E'$ if

(1) $i \neq j$, and

(2) $S \cup T$ is a clique, and

(3) if $j = i + 1$ then $s + |T| = t$.

This finishes the definition of G'; note that it can be computed in the required time bound.

We show that the equivalence (4) holds: First, let C be a clique of G of size r. Set $S_i := V_i \cap C$ for $i = 1, \ldots, r'$. Moreover, set $s_1 := |S_1|$ and $s_i := s_{i-1} + |S_i|$ for $i = 2, \ldots, r'$. Then, we see that $\{(S_i, s_i) \mid 1 \leq i \leq r'\}$ is a clique in G' of size r'.

Now assume that we have a clique $C' = \{(S_i, s_i) \mid 1 \leq i \leq r'\}$ in G' of size r'. Note there is no edge between any two vertices of the same block H_i, therefore we can assume each $(S_i, s_i) \in H_i$ for $i = 1, \ldots, r'$. By the construction of G', it is easy to verify that $\bigcup_{i=1}^{r'} S_i$ is a clique in G of size r. □

Proof of Theorem 27: Assume that the algorithm $\mathbb{A}$ decides CLIQUE in time $f(k) \cdot |V|^{g(k)}$, where f is easily reversible and $g \in o^{\text{eff}}(\text{id})$ is monotone. Given an instance of CLIQUE, i.e., a graph $G = (V, E)$ and $r \leq |V|$ we set

$$m := \max\left\{ \lceil \frac{|V|}{f^{-1}(|V|) - 1} \rceil,\ \log |V| \right\} \quad \text{and} \quad r' := \lceil \frac{|V|}{m} \rceil.$$

Note that $m \geq \log |V|$ and $m \in o^{\text{eff}}(|V|)$. By Lemma 28, in time polynomial in $|V| \cdot 2^m$, that is, in time $2^{o^{\text{eff}}(|V|)}$, we can compute a graph $G' = (V', E')$ such that

$$G \text{ has a clique of size } r \iff G' \text{ has a clique of size } r'.$$

Then we apply $\mathbb{A}$ to G' and r'; this requires time

$$\begin{aligned} f(r') \cdot |V'|^{g(r')} &\leq f(f^{-1}(|V|) - 1) \cdot (|V|^2 \cdot 2^m)^{g(\lceil |V|/m \rceil)} \\ &\leq 2^{\log |V|} \cdot 2^{(2 \cdot m + m) \cdot g(\lceil |V|/m \rceil)} \qquad \text{(since } m \geq \log |V|) \\ &\in 2^{o^{\text{eff}}(|V|)} \qquad \left(\text{by} \lim_{|V| \to \infty} \frac{g(\lceil |V|/m \rceil)}{|V|/m} = 0 \right). \end{aligned}$$

□

Now we prove a similar result for the dominating set problem:

Theorem 29. *If the dominating set problem* DS *can be decided in time* $f(k) \cdot |V|^{o^{\text{eff}}(k)}$ *for some computable* f*, then* DS *can be decided in time* $2^{o^{\text{eff}}(|V|)}$.

It is not clear at all how to establish an analogue of Lemma 28 for the dominating set problem, because a dominating set of a graph G is not necessarily an "amalgamation of local dominating sets" of subgraphs of G. So we take a detour via the weighted satisfiability problem for propositional formulas in CNF. In fact, in Lemma 30 we show a weak analogue of Lemma 28 for the weighted satisfiability problem.

For a propositional formula $\alpha \in$ CNF we set

$$\text{nv}(\alpha) \quad := \quad \text{number of variables in } \alpha.$$

Recall that $|\alpha|$ denotes the length (of a reasonable encoding) of α. A class $\mathbf{C}$ of formulas in CNF is *sparse*, if for all $\alpha \in \mathbf{C}$

$$|\alpha| \in 2^{o^{\text{eff}}(\text{nv}(\alpha))}.$$

For sparse $\mathbf{C}$

$$\lim_{\alpha \in \mathbf{C},\ \text{nv}(\alpha) \to \infty} \frac{\text{nv}(\alpha)}{\log |\alpha|} = \infty,$$

holds in an effective way. Note that CNF itself is not sparse.

Let CNF^+ denote the class of all propositional formulas in CNF without negative literals.

Lemma 30. *Assume that* $(\alpha, k) \in \text{WSAT}(\text{CNF}^+)$ *can be decided in time* $f(k) \cdot |\alpha|^{o^{\text{eff}}(k)}$ *for some computable* f*. Let* $\mathbf{C} \subseteq$ CNF *be sparse. Then there is an algorithm that for* $\alpha \in \mathbf{C}$ *and* $k \in \mathbb{N}$ *decides if* α *is* k*-satisfiable in time* $2^{o^{\text{eff}}(\text{nv}(\alpha))}$.

Proof: We assume that the algorithm $\mathbb{A}$ decides $\text{WSAT}(\text{CNF}^+)$ in time $f(k) \cdot n^{g(k)}$, where f is easily reversible and $g \in o^{\text{eff}}(\text{id})$ monotone.

Let $\mathbf{C} \subseteq$ CNF be sparse. We first show:

($*$) There is an algorithm that for $\alpha \in \mathbf{C}$ decides if α is satisfiable in time $2^{o^{\text{eff}}(\text{nv}(\alpha))}$.

So, assume that $\alpha \in \mathbf{C}$, say,

$$\alpha \quad = \quad \bigwedge_{i \in I} \bigvee_{j \in J_i} \lambda_{ij},$$

where each λ_{ij} is a literal. We set

$$m := \max \left\{ \lceil \frac{\text{nv}(\alpha)}{f^{-1}(\text{nv}(\alpha)) - 1} \rceil,\ \log |\alpha| \right\} \quad \text{and} \quad r := \lceil \frac{\text{nv}(\alpha)}{m} \rceil. \tag{5}$$

Then

$$r \leq f^{-1}(\text{nv}(\alpha)) - 1 \quad \text{and} \quad 2^m = \max\{2^{o^{\text{eff}}(\text{nv}(\alpha))}, 2^{\log |\alpha|}\} \in 2^{o^{\text{eff}}(\text{nv}(\alpha))}. \tag{6}$$

By (5)

$$\lim_{\alpha\in\mathbf{C},\ \mathrm{nv}(\alpha)\to\infty} r \quad = \lim_{\alpha\in\mathbf{C},\ \mathrm{nv}(\alpha)\to\infty} \lceil \frac{\mathrm{nv}(\alpha)}{m} \rceil = \infty \quad \text{holds in an effective way.} \tag{7}$$

Now we divide the variables of α into r disjoint blocks $V_1, \ldots, V_r$, each containing at most m many variables. For each block $V_i = \{X_{ij} \mid 1 \leq j \leq m_i\}$ with $m_i \leq m$ we introduce 2^{m_i} many new variables $\{Y_{i\ell} \mid 0 \leq \ell < 2^{m_i}\}$. Then for any X_{ij} set

$$X_{ij}^* \quad := \quad \bigvee\{Y_{i\ell} \mid \text{the } j\text{th bit of the binary representation of } \ell \text{ is } 1\},$$

$$(\neg X_{ij})^* \quad := \quad \bigvee\{Y_{i\ell} \mid \text{the } j\text{th bit of the binary representation of } \ell \text{ is } 0\}.$$

Now let

$$\alpha^* \quad := \quad \bigwedge_{i\in I}\bigvee_{j\in J_i} \lambda_{ij}^* \wedge \bigwedge_{1\leq i\leq r}\ \bigvee_{0\leq \ell<2^{m_i}} Y_{i\ell}.$$

Clearly, $\alpha^* \in \mathrm{CNF}^+$ and

$$\alpha \text{ is satisfiable} \quad \iff \quad \alpha^* \text{ is } r\text{-satisfiable.}$$

Observe that for some $c \in \mathbb{N}$

$$\begin{aligned} |\alpha^*| \quad &\leq \quad c/2 \cdot (|\alpha| \cdot 2^m + r \cdot 2^m) \quad \leq \quad c \cdot |\alpha| \cdot 2^m \qquad (\text{since by (5), } r \leq |\alpha|) \\ &\leq \quad c \cdot 2^{2m} \qquad (\text{since by (5), } m \geq \log |\alpha|). \end{aligned}$$

Since $2^{2m} \in 2^{o^{\mathrm{eff}}(\mathrm{nv}(\alpha))}$ by (6), the formula α^* can be obtained in time $2^{o^{\mathrm{eff}}(\mathrm{nv}(\alpha))}$. Then we can apply $\mathbb{A}$ to α^* and r to decide the satisfiability of α in time

$$\begin{aligned} f(r) \cdot |\alpha^*|^{g(r)} \quad &\leq \quad \mathrm{nv}(\alpha) \cdot c^{g(r)} \cdot 2^{2\cdot m\cdot g(r)} \quad \leq \quad c^{g(r)} \cdot 2^{\log \mathrm{nv}(\alpha)} \cdot 2^{2\cdot(\mathrm{nv}(\alpha)/(r-1))\cdot g(r)} \qquad (\text{by (5)}) \\ &\in \quad 2^{o^{\mathrm{eff}}(\mathrm{nv}(\alpha))} \qquad\qquad (\text{by (7) and since } g \in 2^{o^{\mathrm{eff}}(\mathrm{id})}). \end{aligned}$$

This finishes the proof of $(*)$.

To prove the lemma assume that $\alpha \in \mathbf{C}$ and $k \leq m := \mathrm{nv}(\alpha)$. Let $X_1, \ldots, X_m$ be the variables of α. By Lemma 16 we can compute in polynomial time a circuit with small gates $\mathcal{C}_{m,k}$ with $\|\mathcal{C}_{m,k}\|_+ \in O(m)$, and with input nodes $X_1, \ldots, X_m$, which is satisfied exactly by the assignments of weight k. The usual reduction from CIRCSAT to 3SAT yields a formula $\beta_{m,k}$ in 3CNF such that for any assignment $b : \{X_1, \ldots, X_m\} \to \{\text{TRUE}, \text{FALSE}\}$,

$$\text{there is an assignment } b' \text{ that satisfies } \beta_{m,k} \text{ with } b' \restriction \{X_1, \ldots, X_m\} = b \quad \iff \quad \text{the weight of } b \text{ is exactly } k.$$

Furthermore $\mathrm{nv}(\beta_{m,k}) = \Theta(m)$, since $\|\mathcal{C}_{m,k}\|_+ \in O(m)$. Now set

$$\alpha_k^* \quad := \quad \alpha \wedge \beta_{m,k}.$$

It is straightforward to verify that

$$\alpha \text{ is } k\text{-satisfiable} \quad \iff \quad \alpha_k^* \text{ is satisfiable.} \tag{8}$$

Since $\mathrm{nv}(\alpha_k^*) = \Theta(m)$, the class

$$\mathbf{C}^* \quad := \quad \{\alpha_k^* \mid \alpha \in \mathbf{C} \text{ and } k \leq \mathrm{nv}(\alpha)\}.$$

is sparse too. By $(*)$ the satisfiability of formulas in $\mathbf{C}^*$ can be checked in time subexponential in the number of variables. Hence, to check the k-satisfiability of a formula α, we first produce α_k^* in polynomial time, and then check the satisfiability of α_k^* in time

$$2^{o^{\mathrm{eff}}(\mathrm{nv}(\alpha_k^*))} \quad = \quad 2^{o^{\mathrm{eff}}(\mathrm{nv}(\alpha))}.$$

□

Now, one obtains Theorem 29 by means of the following two lemmas:

Lemma 31. *If* DS *can be decided in time* $f(k) \cdot |V|^{o^{\text{eff}}(k)}$ *for some computable* f*, then* $(\alpha, k) \in$ WSAT(CNF$^+$) *can be decided in time* $f(k) \cdot |\alpha|^{o^{\text{eff}}(k)}$*.*

Lemma 32. *If for all sparse* $\mathbf{C} \subseteq$ CNF *there is an algorithm that for* $\alpha \in \mathbf{C}$ *and* $k \in \mathbb{N}$ *decides if* α *is* k*-satisfiable in time* $2^{o^{\text{eff}}(\text{nv}(\alpha))}$*, then* DS *and* CLIQUE *can be decided in time* $2^{o^{\text{eff}}(|V|)}$*.*

Proof of Lemma 31: It suffices to show that for any $\alpha \in$ CNF$^+$, we can construct in polynomial time a graph $G_\alpha = (V, E)$ such that

$$\alpha \text{ is } k\text{-satisfiable} \quad \iff \quad G_\alpha \text{ has a dominating set of size } k.$$

Set

$$\begin{aligned} V &:= \{u_X \mid X \text{ a variable of } \alpha\} \cup \{u_C \mid C \text{ a clause of } \alpha\}, \\ E &:= \big\{\{u_X, u_Y\} \mid X \neq Y \text{ variables of } \alpha\big\} \ \cup \ \big\{\{u_X, u_C\} \mid X \text{ appears in the clause } C\big\}. \end{aligned}$$

□

Proof of Lemma 32: Given an instance of DS, i.e., a graph $G = (V, E)$ and a natural number $k \leq |V|$, we define a formula $\alpha_G \in$ CNF:

$$\alpha_G \ := \ \bigwedge_{u \in V} \big(X_u \vee \bigvee_{\{u,v\} \in E} X_v\big).$$

Note that $\text{nv}(\alpha_G) = |V|$ and that

$$G \text{ has a dominating set of size } k \quad \iff \quad \alpha_G \text{ is } k\text{-satisfiable}.$$

Clearly, $\{\alpha_G \mid G \text{ a graph}\}$ is sparse. By assumption, we can test if α_G is k-satisfiable in time

$$2^{o^{\text{eff}}(\text{nv}(\alpha_G))} \ = \ 2^{o^{\text{eff}}(|V|)}.$$

Analogously, we can prove the same subexponential time bound for CLIQUE associating with a graph $G = (V, E)$ the formula in CNF

$$\bigwedge_{\{u,v\} \notin E} \neg X_u \vee \neg X_v.$$

□

Proof of Theorem 29: Assume that DS can be solved in time $f(k) \cdot |V|^{o^{\text{eff}}(k)}$. Then by Lemma 31 $(\alpha, k) \in$ WSAT(CNF$^+$) can be decided in time $f(k) \cdot |\alpha|^{o^{\text{eff}}(k)}$. Then for all sparse $\mathbf{C} \subseteq$ CNF there is an algorithm that for $\alpha \in \mathbf{C}$ and $k \in \mathbb{N}$ decides if α is k-satisfiable in time $2^{o^{\text{eff}}(\text{nv}(\alpha))}$ by Lemma 30 and hence, DS can be solved in time $2^{o^{\text{eff}}(|V|)}$ by Lemma 32. □

In view of the assertion concerning CLIQUE in Lemma 32 one obtains in the same way, thereby improving a result stated in [9].

Corollary 33. *If the dominating set problem* DS *can be decided in time* $f(k) \cdot |V|^{o^{\text{eff}}(k)}$ *for some computable* f*, then* CLIQUE *can be decided in time* $2^{o^{\text{eff}}(|V|)}$*.*

7. Renormalizations

In the context of miniaturized problems two "renormalizations" of VC have been considered, $(k \cdot \log n)^+$-VC and $(k \cdot \log n)^-$-VC. Let $\| \ \|$ be an arbitrary size function on the class of graphs. Define $(k \cdot \log n)^{\| \ \|}$-VC by

$(k \cdot \log n)^{\| \ \|}$-VC
Input: $G = (V, E)$.
Parameter: $k \in \mathbb{N}$.
Problem: Decide if G has a vertex cover of size $k \cdot \log \|G\|$.

Clearly, $(k \cdot \log\ n)^{-}$-VC denote $(k \cdot \log\ n)^{\|\ \|-}$-VC, but, as we only consider graphs here, we let $(k \cdot \log\ n)^{+}$-VC be $(k \cdot \log n)^{\|\ \|}$-VC, where

$$\|G\| = \text{number of vertices + number of edges.}$$

We show that both problems, $(k \cdot \log n)^{-}$-VC and $(k \cdot \log n)^{+}$-VC, are fpt-equivalent to mini^{-}-VC. The equivalence of the first and the third problem has been claimed in [9]. Since there no proof is given and since we do not know if the author of [9] refers to Turing reductions or to many-one reductions, we also include a proof in this paper.

Theorem 34. *$(k \cdot \log n)^{-}$-VC, $(k \cdot \log n)^{+}$-VC, and mini^{-}-VC are fpt-equivalent.*

Remark 35. Of course, in the same way one could define the $k \cdot \log n$ renormalizations of various parameterized problems, for example, of all Fagin-definable ones. We introduced the notion of renormalization for the vertex cover problem directly, since we have no substantial results for the general case. In fact, the following proposition shows that the preceding theorem does not generalize to CLIQUE (unless FPT = W[1]).

Proposition 36. *If $(k \cdot \log n)^{-}$-CLIQUE $\leq^{\text{fpt}}$ mini^{-}-CLIQUE, then FPT = W[1].*

Proof: For a parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ and $\ell \in \mathbb{N}$ the ℓth *slice* Q_ℓ of Q is the (classical) problem

$$Q_\ell := \{x \in \Sigma^* \mid Qx\ell\},$$

and the ℓth *truncation* $Q[\ell]$ of Q is the (classical) problem

$$Q[\ell] \quad := \quad \{(x, k) \in Q \mid k \leq \ell\}.$$

For parameterized problems $Q \subseteq \Sigma^* \times \mathbb{N}$ and $Q' \subseteq (\Sigma')^* \times \mathbb{N}$ with $Q \leq^{\text{fpt}} Q'$, it immediately follows from the notion of fpt-reduction that for every ℓ there is an $m \in \mathbb{N}$ such that Q_ℓ is reducible to $Q'[m]$ in polynomial time.

Now we come to our claim and assume that $(k \cdot \log n)^{-}$-CLIQUE $\leq^{\text{fpt}}$ mini^{-}-CLIQUE. The brute force algorithm shows that the mth truncation of mini^{-}-CLIQUE is in PTIME for every $m \in \mathbb{N}$. Therefore, all slices of $(k \cdot \log n)^{-}$-CLIQUE are in PTIME; in particular, the first slice, the problem

LOG-CLIQUE	
Input:	A graph $G = (V, E)$.
Problem:	Decide if there is a clique of size $\log \|V\|$.

is in PTIME. We use this to show that CLIQUE $\in$ FPT and therefore, that FPT = W[1] (since CLIQUE is W[1]-complete).

Let (G, k) be an instance of CLIQUE with $G = (V, E)$. We could solve it using the algorithm for LOG-CLIQUE, if $k = \log\ |V|$. We first consider the case $k < \log\ |V|$. Since, $\log\ (|V| + |V|) \leq k + |V|$, there is an m with $0 \leq m \leq |V|$ such that $k + m = \log\ (|V| + m)$. The graph $G' = (V', E')$ is obtained by adding m new vertices to G and connecting them with every other vertex, old or new. Then,

$$G' \text{ has a clique of size } \log\ |V'| \quad \Longleftrightarrow \quad G \text{ has a clique of size } k.$$

Now, we can apply the polynomial time algorithm for LOG-CLIQUE. If $k > \log\ |V|$, then $|V| < 2^k$ and we can apply the brute force algorithm to solve this instance of CLIQUE. □

The reader familiar with [12] will have no difficulties to (understand and to) generalize the preceding proof and to show:

For $t \geq 2$, if W[t] $\neq$ FPT then $(k \cdot \log n)^{-}$-FD$_{\varphi(X)} \not\leq^{\text{fpt}}$ mini^{-}-FD$_{\varphi(X)}$ for every generic $\Pi_{t/1}$-formula $\varphi(X)$.

In particular, for the dominating set problem, we have

If $(k \cdot \log n)^{-}$-DS $\leq^{\text{fpt}}$ mini^{-}-DS, then W[2] = FPT.

Theorem 34 is shown via Proposition 38, Proposition 40, and Proposition 41 below. For a graph $G = (V, E)$ and $p, q \in \mathbb{N}$, let $G_{p,q}$ be the graph obtained as the disjoint union of G, of a graph consisting of p isolated points, and of q copies of the graph consisting of two connected points. Several times we will use the following trivial fact:

Lemma 37. *Let $G = (V, E)$ be a graph and $p, q, s \in \mathbb{N}$ with $s \leq |V|$. Then,*

$$G \text{ has a vertex cover of size } s \iff G_{p,q} \text{ has a vertex cover of size } s + q.$$

Proposition 38. mini$^-$-VC $\leq^{\text{fpt}}$ $(k \cdot \log n)^-$-VC.

Proof: It is useful to consider the "intermediate" problem Q

Q	
Input:	$n, k \in \mathbb{N}$ in unary, a graph $G = (V, E)$ with $|V| = k \cdot n$.
Parameter:	k.
Problem:	Decide if G has a vertex cover of size $k \cdot \log n$.

and to show:

$$\text{mini}^-\text{-VC} \leq^{\text{fpt}} Q \quad \text{and} \quad Q \leq^{\text{fpt}} (k \cdot \log n)^-\text{-VC}.$$

For mini$^-$-VC $\leq^{\text{fpt}}$ Q, let an instance of mini$^-$-VC be given consisting of $n, k \in \mathbb{N}$ in unary, a graph $G = (V, E)$ with $|V| \leq k \cdot \log n$ and $r \leq |V|$. We may assume that $n \geq 3 \cdot \log n$, i.e., that $n \geq 10$. We set

$$q := k \cdot \log n - r \quad \text{and} \quad p := k \cdot n - |V| - 2 \cdot q.$$

Then, $|V(G_{p,q})| = k \cdot n$ and by Lemma 37,

$$G \text{ has a vertex cover of size } r \iff G_{p,q} \text{ has a vertex cover of size } k \cdot \log n \ (= r + q),$$

which gives the desired reduction.
For $Q \leq^{\text{fpt}} (k \cdot \log n)^-$-VC, let be given an instance of Q, i.e., natural numbers n, k in unary and a graph $G = (V, E)$ with $|V| = k \cdot n$. We may assume that $k \geq 1$ and that

$$n \quad > \quad \frac{1}{k \cdot (2^{\frac{1}{k}} - 1)}.$$

Then, for any $d \in \mathbb{N}$,

$$\log (k \cdot n + d + 1) - \log (k \cdot n + d) \quad < \quad \frac{1}{k}. \tag{9}$$

Let

$$k' \quad := \quad \lfloor \frac{k \cdot \log n}{\log (k \cdot n)} \rfloor.$$

Then, $1 \leq k' \leq k$, and

$$k' \cdot \log (k \cdot n) \leq k \cdot \log n \leq (k' + 1) \cdot \log (k \cdot n).$$

Using (9), we see that for some $0 \leq m \leq k^2 \cdot n^2$,

$$k' \cdot \log (k \cdot n + m) \quad = \quad k \cdot \log n.$$

By Lemma 37,

$$\begin{aligned} & G \text{ has a vertex cover of size } k \cdot \log n \\ \iff & G_{m,0} \text{ has a vertex cover of size } k' \cdot \log (k \cdot n + m) \ (= k \cdot \log n), \\ \iff & G_{m,0} \text{ has a vertex cover of size } k' \cdot \log |V(G_{m,0})|. \end{aligned}$$

This shows that $Q \leq^{\text{fpt}} (k \cdot \log n)^-$-VC. □

In the proof of the next proposition, we need the following result of Nemhauser and Trotter [15].

Theorem 39. *There is a polynomial time algorithm that, given a graph $G = (V, E)$, computes disjoint subsets D_0 and D_1 of V with the following properties:*

(1) For any vertex cover S of the subgraph $G[D_1]$ induced by D_1, the set $S \cup D_0$ is a vertex cover of G.

(2) There is a minimum vertex cover S of G with $D_0 \subseteq S$.

(3) Any vertex cover of $G[D_1]$ has size at least $|D_1|/2$.

Here, by a minimum vertex cover of G we mean a vertex cover of G of minimal cardinality. Note that the first two properties imply that the union of any minimum vertex cover of $G[D_1]$ and D_0 is a minimum vertex cover of G.

Proposition 40. $(k \cdot \log n)^{-}$-VC $\leq^{\text{fpt}}$ mini^{-}-VC.

Proof: Assume given an instance of $(k\cdot\log n)^{-}$-VC consisting of the graph $G = (V, E)$ (as input) and of $k \in \mathbb{N}$ as parameter. We look for a vertex cover of size $k \cdot \log |V|$. We assume $k \cdot \log |V| \leq |V|$. We apply the algorithm of Theorem 39 to G and obtain the corresponding sets D_0 and D_1. Then,

$$\begin{aligned} & G \text{ has a vertex cover of size } k \cdot \log |V| \\ \iff\ & G \text{ has a minimum vertex cover of size} \leq k \cdot \log |V|, \\ \iff\ & G[D_1] \text{ has a minimum vertex cover of size} \leq k \cdot \log |V| - |D_0|, \\ \iff\ & G[D_1] \text{ has a vertex cover of size } \min\{|D_1|, k \cdot \log |V| - |D_0|\}. \end{aligned}$$

We get the following instance n, k', G', r of mini^{-}-VC:

$$n := |V|, \quad k' := 2 \cdot k, \quad G' := G[D_1], \quad \text{and} \quad r := \min\{|D_1|, k \cdot \log |V| - |D_0|\}.$$

Then, the map $G, k \mapsto n, k', G', r$ is an fpt-reduction from $(k \cdot \log n)^{-}$-VC to mini^{-}-VC; note the following: if $|D_1| > k' \cdot \log n$, then $|D_1| > 2 \cdot (k \cdot \log |V| - |D_0|)$, and hence by the third property in Theorem 39, $G[V_0]$ has no vertex cover of size r, and therefore, G has no vertex cover of size $k \cdot \log |V|$. □

Finally we show that for the $k \cdot \log n$ versions of vertex cover it is irrelevant what size functions of our two standard ones we choose:

Proposition 41. $(k \cdot \log n)^{-}$-VC *and* $(k \cdot \log n)^{+}$-VC *are* fpt-*equivalent.*

Proof: $(k \cdot \log n)^{+}$-VC $\leq^{\text{fpt}}$ $(k \cdot \log n)^{-}$-VC: Assume given an instance of $(k \cdot \log n)^{+}$-VC consisting of $G = (V, E)$ and k. Observe that by Lemma 37

$$\begin{aligned} & G \text{ has a vertex cover of size } k \cdot \log (|V| + |E|) \\ \iff\ & G_{|E|,0} \text{ has a vertex cover of size } k \cdot \log (|V| + |E|), \\ \iff\ & G_{|E|,0} \text{ has a vertex cover of size } k \cdot \log |V(G_{|E|,0})|. \end{aligned}$$

$(k \cdot \log n)^{-}$-VC $\leq^{\text{fpt}}$ $(k \cdot \log n)^{+}$-VC: Assume given an instance of $(k \cdot \log n)^{-}$-VC consisting of the graph $G = (V, E)$ and the natural number k. We want to know whether there is a vertex cover of size $k \cdot \log |V|$. Since by Lemma 37, for every $p, q \in \mathbb{N}$,

$$G \text{ has a vertex cover of size } k \cdot \log |V| \quad \iff \quad G_{p,q} \text{ has a vertex cover of size } k \cdot \log |V| + q,$$

in order to obtain a reduction from $(k \cdot \log n)^{-}$-VC to $(k \cdot \log n)^{+}$-VC we want to fix p and q in such a way that

$$k \cdot \log (|V(G_{p,q})| + |E(G_{p,q})|) \quad = \quad k \cdot \log |V| + q,$$

i.e.,

$$k \cdot \log (|V| + |E| + 3 \cdot q + p) - q = k \cdot \log |V|. \tag{10}$$

We can assume that $k \geq 1$ and that

$$|V| + |E| \quad > \quad \frac{3}{2^{\frac{1}{k}} - 1}.$$

Then, for any $d \in \mathbb{N}$,

$$\log\left(|V| + |E| + d + 3\right) - \log\left(|V| + |E| + d\right) < \frac{1}{k}. \tag{11}$$

An easy calculation shows that for some $0 \leq q \leq 4 \cdot k^2 \cdot (|V| + |E|)$,

$$0 \leq k \cdot \log\left(|V| + |E| + 3 \cdot q\right) - q \leq k \cdot \log|V|.$$

Fix such an q. Now, the inequality (11) shows that the equality (10) can be fulfilled for some p with $0 \leq p \leq (|V|+|E|+3 \cdot q)^2$. □

Proof of Theorem 34: We get

$$\text{mini}^{-}\text{-VC} \leq^{\text{fpt}} (k \cdot \log n)^{-}\text{-VC} \leq^{\text{fpt}} (k \cdot \log n)^{+}\text{-VC} \leq^{\text{fpt}} (k \cdot \log n)^{-}\text{-VC} \leq^{\text{fpt}} \text{mini}^{-}\text{-VC}$$

by applying Proposition 38, Proposition 41, Proposition 41, and Proposition 40, respectively. □

8. Conclusions

We have introduced a general notion of miniaturization of a problem that comprises the different miniaturizations of concrete problems considered so far. Using the appropriate logical formalism, we were able to show that the miniaturizations of definable problems in W$[t]$ are in W$[t]$, too. Based on this logical formalism we introduced a hierarchy of complexity classes in W$[1]$.

Some problems were raised but not settled in this paper, and we think they deserve further study. Let us mention a further question: Can we drop the sparseness condition in Proposition 31 (a proof might require some kind of *Sparsification Lemma* for formulas in CNF, cf. [14]). If so, a result in [2] would prove that the existence of an $f(k) \cdot n^{o^{\text{eff}}(k)}$ time algorithm for the dominating set problem DS implies W$[1]$ = FPT. This would improve a result in [3].

References

[1] L. Cai and D. Juedes. Subexponential parameterized algorithms collapse the W-hierarchy. In *Proceedings of the 28st International Colloquium on Automata, Languages and Programming (ICALP'01)*, Lecture Notes in Computer Science 2076, 273-284, 2001.

[2] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and Ge Xia. Tight Lower Bounds for Certain Parameterized NP-Hard Problems. In *Proceedings of The 19th IEEE Annual Conference on Computational Complexity (CCC'04)*, 150-160, 2004.

[3] J. Chen, X. Huang, I. Kanj, and G. Xia. Linear FPT Reductions and Computational Lower Bounds. In *Proceedings of the 36th annual ACM symposium on Theory of computing*, 212-221, 2004.

[4] P. Clote and E. Kranakis. *Boolean Functions and Models of Computation*. Springer-Verlag, 2003.

[5] R. Downey. Parameterized complexity for the sceptic. In *Proceedings of The 18th IEEE Annual Conference on Computational Complexity (CCC'03)*, 147-169, 2003.

[6] R. Downey, V. Estivill, M. Fellows, E. Prieto, and F. Rosamond. Cutting up is hard to do: the parameterized complexity of k-cut and related problems. In *Proceedings of Computing: The Australasian Theory Symposium (CATS'03)*, Electronic Notes in Theoretical Computer Science, 78(0), 1-14, 2003.

[7] R. Downey, M. Fellows, and K. Regan. Descriptive complexity and the W-hierarchy. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetic*, volume 39 of *AMS-DIMACS Volume Series*, 119–134, AMS, 1998.

[8] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[9] M. Fellows. New directions and new challenges in algorithm design and complexity, parameterized. In *Proceedings of the 8th International Workshop on Algorithms and Data Structures (WADS 2003)*, Lecture Notes in Computer Science 2748, 505-519, 2003

[10] J. Flum and M. Grohe. Fixed-parameter tractability, definability, and model checking. *SIAM Journal of Computing*, 31(1):113–145, 2001.

[11] J. Flum and M. Grohe. Parameterized complexity and subexponential time. In *the Complexity Column of the Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 84, October, 2004.

[12] J. Flum, M. Grohe, and M. Weyer. Limited nondeterminism and parameterized complexity theory. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, Lecture Notes in Computer Science 3142, 555-567, 2004.

[13] R. Impagliazzo and R. Paturi. Complexity of k-SAT. *Journal of Computer System and Science*, 62(2): 367-375, 2001.

[14] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer System and Science*, 63(4):512-530, 2001.

[15] G. Nemhauser and L. Trotter. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.