

Parameterized Complexity of Weighted Satisfiability Problems

Nadia Creignou^{1*} and Heribert Vollmer^{2†}

¹ Laboratoire d'Informatique Fondamentale de Marseille, CNRS UMR 7279,
Aix-Marseille Université, 163 avenue de Luminy, F-13288 Marseille Cedex 9, France.
creignou@lif.univ-mrs.fr

² Institut für Theoretische Informatik, Leibniz Universität Hannover, Appelstr. 4,
30167 Hannover, Germany.
vollmer@thi.uni-hannover.de

Abstract. We consider the weighted satisfiability problem for Boolean circuits and propositional formulæ, where the weight of an assignment is the number of variables set to true. We study the parameterized complexity of these problems and initiate a systematic study of the complexity of its fragments. Only the monotone fragment has been considered so far and proven to be of same complexity as the unrestricted problems. Here, we consider all fragments obtained by semantically restricting circuits or formulæ to contain only gates (connectives) from a fixed set B of Boolean functions. We obtain a dichotomy result by showing that for each such B , the weighted satisfiability problems are either $W[P]$ -complete (for circuits) or $W[SAT]$ -complete (for formulæ) or efficiently solvable. We also consider the related counting problems.

1 Introduction

Satisfiability of circuits and formulæ are fundamental problems, which are the core of many complexity classes. This is true not only in the “classical” complexity setting but also in parameterized complexity theory. Here, with each problem instance we associate a parameter. Instances with the same parameter are thought to share a common structure. A parameterized problem is fixed-parameter tractable (in FPT) if it can be solved in polynomial time for each fixed value of the parameter, where the degree of the polynomial does not depend on the parameter. Much like in the classical setting, to give evidence that certain algorithmic problems are not in FPT one shows that they are complete for superclasses of FPT, like the classes in the so-called W -hierarchy.

Weighted satisfiability (where the weight of a solution is given by the number of variables assigned true) gives rise to a parameterized version of the problems of satisfiability of circuits or formulæ. The goal is then to decide the existence of satisfying assignments of weight exactly k , where k is the parameter. From a complexity theoretic viewpoint, these parameterized problems are very hard since they are $W[P]$ -complete for circuits and $W[SAT]$ -complete for formulæ (see, e.g., [8]).

* Supported by Agence Nationale de la Recherche under grant ANR-09-BLAN-0011-01.

† Work done while on leave at the University of Oxford. Supported by DFG VO 630/6-2 and EPSRC EP/G055114/1.

This intractability result raises the question for restrictions leading to fragments of lower complexity. Concerning formulae such restrictions have been considered in previous work. Indeed Marx [10] studied the parameterized complexity of satisfiability problems in the famous Schaefer’s framework where formulae are restricted to generalized conjunctive normal form with clauses from a fixed set of relations (the constraint language). He obtained a dichotomy classification by showing that for every possible constraint language the weighted satisfiability problem for generalized CNF formulae is either in FPT or W[1]-complete (thus, in any case, much lower than the W[SAT]-completeness for general weighted SAT). A similar yet different approach is not to restrict the *syntactic shape* of the formulae by stipulating a certain normal form but rather to require formulae to be constructed from a restricted set of Boolean functions B (in contrast to the Schaefer framework, one might say that these are *semantic* restrictions). Such formulae are called *B-formulae*. This approach has first been taken by Lewis, who showed that deciding satisfiability of B -formulae is NP-complete if and only if the set of Boolean functions B has the ability to express the negation of implication \nrightarrow [9]. Since then this approach has been applied to a wide range of algorithmic problems from the area of circuits [14,3] or propositional formulae in, e.g., temporal logics [2] or non-monotonic logics [5].

The goal of this paper is to follow this approach and to show that Post’s lattice allows to completely classify the complexity of weighted satisfiability for all possible sets of allowed Boolean functions. We consider both circuits and formulae, and the complexity of deciding whether they admit a satisfying assignment of weight exactly k . We show that depending on the set B of allowed connectives the parameterized weighted satisfiability problem is either W[P]-complete (for circuits) and W[SAT]-complete for formulae, or in P. More precisely, we prove that the complexity of these problems is W[P]-complete or W[SAT]-complete (depending on whether they concern circuits or formulae) as soon as B can express either the function $x \wedge (y \vee z)$, or any 2-threshold function as for example the ternary majority function. The problem becomes solvable in polynomial time in all remaining cases. Thus, in a sense, we exactly pinpoint the reason for intractability of weighted satisfiability by exhibiting which Boolean functions make the problem hard.

Besides the decision problem, we study the complexity of the corresponding counting problems. We prove here also a dichotomy theorem in showing that the problems are either $\#W[P]$ -complete (or $\#W[SAT]$ -complete), or in FP. The frontier of this dichotomy is not the same as in the decision case, since some tractable decision problems, as, e.g., the weighted satisfiability problem in which only the connective \rightarrow is allowed, become hard counting problems.

Our results are summarized in Fig. 1. White sets B of Boolean functions lead to easy problems, black sets lead to hard problems. The gray colored nodes correspond to those sets B for which the decision problems are easy, but the counting problems are hard.

The rest of the paper is structured as follows. We first give the necessary preliminaries. Afterwards, we define the weighted satisfiability considered herein. We then classify the parameterized complexity of these problems. Next, we consider the counting problems and finally conclude with a discussion of the results.

2 Preliminaries

Parameterized complexity. We assume familiarity with the basic classes and reducibility notions from parameterized complexity theory, see, e.g., [8,12], such as FPT, W[P], W[SAT], fpt-reductions and Turing fpt-reductions.

Boolean circuits and propositional formulae. We assume familiarity with propositional logic. A *Boolean function* is an n -ary function $f: \{0, 1\}^n \rightarrow \{0, 1\}$. We define *Boolean circuits* (see also [17]) in the standard way as directed acyclic graphs with each node of in-degree $k > 0$ labeled by a Boolean function of arity k . For non-commutative functions, there is in addition an ordering on the incoming edges. Nodes of in-degree 0 are either labeled as Boolean constants 0 or 1, or as input nodes. In addition, one node of out-degree 0 is labeled as the output node. We think of the input nodes as being numbered $1, \dots, n$. This definition of a Boolean circuit corresponds to the intuitive idea that a circuit consists of a set of gates which are either input gates, or compute some Boolean function with arguments taken from the predecessor gates. The value computed by the circuit is the result computed in the distinguished output-gate. So, a circuit \mathcal{C} with n input nodes naturally computes an n -ary Boolean function, we denote it by $f_{\mathcal{C}}$.

We denote the value computed by \mathcal{C} on input $a \in \{0, 1\}^n$ by $\mathcal{C}(a)$. If $\mathcal{C}(a) = 1$, we say that a *satisfies* \mathcal{C} . We call \mathcal{C} *satisfiable* if there is some tuple $a \in \{0, 1\}^n$ that satisfies \mathcal{C} . We define the *weight of a tuple* $a = (a_1, \dots, a_n) \in \{0, 1\}^n$ to be $\sum_{i=1}^n a_i$, the number of 1-entries of a . We say \mathcal{C} is *k -satisfiable* if it is satisfied by a tuple of weight k . A circuit \mathcal{C} is *monotone* if for all $a = (a_1, \dots, a_n) \in \{0, 1\}^n$ such that $\mathcal{C}(a) = 1$, $a_1 \leq b_1, \dots, a_n \leq b_n$ implies $\mathcal{C}(b_1, \dots, b_n) = 1$.

A formula φ is a circuit where the underlying graph forms a tree. Hence, such circuits can always be written as a formula in the usual string representation without growing significantly in size. For a general circuit, the length of its “formula representation” can be exponential in the size of the original circuit. Further we denote by $\varphi[\alpha/\beta]$ the formula obtained from φ by replacing all occurrences of α with β . The set $\text{Var}(\varphi)$ denotes the set of variables occurring in the formula

Deciding the k -satisfiability of a Boolean circuit, p-WCIRCUIT-SAT, where k is taken to be the parameter, is of fundamental importance for parameterized complexity theory. Indeed, p-WCIRCUIT-SAT is W[P]-complete under fpt-reductions (see [8, Theorem 3.9]). Deciding the k -satisfiability of a Boolean formula, p-WSAT, is W[SAT]-complete by definition.

Given B a finite set of Boolean functions, a B -circuit (resp. a B -formula) is a Boolean circuit (a formula) using only functions (connectives) from B .

Clones of Boolean functions. A *clone* is a set of Boolean functions that is closed under superposition, i.e., it contains all projections (that is, the functions $f(a_1, \dots, a_n) = a_k$ for $1 \leq k \leq n$ and $n \in \mathbb{N}$) and is closed under arbitrary composition. Let B be a finite set of Boolean functions. We denote by $[B]$ the smallest clone containing B and call B a *base* for $[B]$. The set $[B]$ corresponds to the set of all Boolean functions that can be computed by B -circuits. All closed classes of Boolean functions were identified by Post ([13]). Post also found a finite base for each of them and detected their inclusion structure, hence the name of *Post’s lattice* (see Figure 1).

In order to define the clones, we require the following notions, where f is an n -ary Boolean function:

- f is *c-reproducing* if $f(c, \dots, c) = c$, $c \in \{0, 1\}$.
- f is *monotonic* (or, *monotone*) if $a_1 \leq b_1, a_2 \leq b_2, \dots, a_n \leq b_n$ implies $f(a_1, \dots, a_n) \leq f(b_1, \dots, b_n)$.
- f is *c-separating of degree k* if for all $A \subseteq f^{-1}(c)$ of size $|A| = k$ there exists an $i \in \{1, \dots, n\}$ such that $(a_1, \dots, a_n) \in A$ implies $a_i = c$, $c \in \{0, 1\}$.
- f is *c-separating* if f is *c-separating of degree $|f^{-1}(c)|$* .
- f is *self-dual* if $f(x_1, \dots, x_n) \equiv \neg f(\neg x_1, \dots, \neg x_n)$.
- f is *affine* if it is of the form $f(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n \oplus c$ with $c \in \{0, 1\}$.

In the following we will often use well-known Boolean functions, as \wedge , \vee , \neg , \oplus , \rightarrow the implication function and the ternary majority operation maj (defined by $\text{maj}(x_1, x_2, x_3) = 1$ if and only if $x_1 + x_2 + x_3 \geq 2$). We will also refer to *q-threshold functions* as functions f verifying $f(x_1, \dots, x_n) = 1$ if and only if $\sum_{i=1}^n x_i \geq q$. Observe that maj is thus a ternary 2-threshold function. More generally T_k^n will denote the k -threshold function of arity n .

A list of all clones with definitions and finite bases is given in Table 1 on page 5, see also, e.g., [4]. Clones of particular importance in this paper, either because they are of technical importance or because they mark points in Post's lattice where the complexity of our problems changes, are the following:

- The clone of all Boolean functions $\text{BF} = [\wedge, \neg] = [\wedge, \vee, \neg, 0, 1]$.
- The monotonic clones M_* , e.g., $M_2 = [\wedge, \vee]$ and $M = [\wedge, \vee, 0, 1]$.
- The dual clones D_* , e.g., $D_2 = [\text{maj}]$.
- The disjunctive clones V_* , e.g., $V = [\vee, 0, 1]$.
- The conjunctive clones E_* , e.g., $E = [\wedge, 0, 1]$.
- The affine clones L_* , e.g., $L = [\oplus, 0, 1]$.
- The implication clone S_0^* , e.g., $S_0 = [\rightarrow]$.

We will often add some function $f \notin C$ to a clone C and consider the clone $C' = [C \cup \{f\}]$ generated out of C and f . With Post's lattice one can determine this C' quite easily: it is the lowest clone above C that contains f . The following list contains identities we will frequently use.

- $[S_{02} \cup \{0, 1\}] = [S_{12} \cup \{0, 1\}] = \text{BF}$
- $[S_{00} \cup \{0, 1\}] = [D_2 \cup \{0, 1\}] = [M_2 \cup \{0, 1\}] = [S_{10} \cup \{0, 1\}] = M$
- $[S_{10} \cup \{1\}] = [M_2 \cup \{1\}] = M_1$
- $[S_{00} \cup \{0\}] = [M_2 \cup \{0\}] = M_0$
- $[D_2 \cup \{1\}] = S_{01}^2, [D_2 \cup \{0\}] = S_{11}^2$

Let f be an n -ary Boolean function. A B -formula φ such that $\text{Var}(\varphi) \supseteq \{x_1, \dots, x_n\}$ is a *B-representation* of $f(x_1, \dots, x_n)$ if it holds that $\sigma: \text{Var}(\varphi) \rightarrow \{0, 1\}$ satisfies φ if and only if $f(\sigma(x_1), \dots, \sigma(x_n)) = 1$. Such a B -representation exists for every $f \in [B]$. Yet, it may happen that the B -representation of some function uses some input variable more than once.

Clone	Definition	Base
BF	All Boolean functions	$\{x \wedge y, \neg x\}$
R ₀	$\{f \in \text{BF} \mid f \text{ is 0-reproducing}\}$	$\{x \wedge y, x \oplus y\}$
R ₁	$\{f \in \text{BF} \mid f \text{ is 1-reproducing}\}$	$\{x \vee y, x \leftrightarrow y\}$
R ₂	$R_0 \cap R_1$	$\{x \vee y, x \wedge (y \leftrightarrow z)\}$
M	$\{f \in \text{BF} \mid f \text{ is monotone}\}$	$\{x \wedge y, x \vee y, 0, 1\}$
M ₀	$M \cap R_0$	$\{x \wedge y, x \vee y, 0\}$
M ₁	$M \cap R_1$	$\{x \wedge y, x \vee y, 1\}$
M ₂	$M \cap R_2$	$\{x \wedge y, x \vee y\}$
S ₀	$\{f \in \text{BF} \mid f \text{ is 0-separating}\}$	$\{x \rightarrow y\}$
S ₀ ⁿ	$\{f \in \text{BF} \mid f \text{ is 0-separating of degree } n\}$	$\{x \rightarrow y, T_2^{n+1}\}$
S ₁	$\{f \in \text{BF} \mid f \text{ is 1-separating}\}$	$\{x \nrightarrow y\}$
S ₁ ⁿ	$\{f \in \text{BF} \mid f \text{ is 1-separating of degree } n\}$	$\{x \nrightarrow y, T_n^{n+1}\}$
S ₀₂ ⁿ	$S_0^n \cap R_2$	$\{x \vee (y \wedge \neg z), T_2^{n+1}\}$
S ₀₂	$S_0 \cap R_2$	$\{x \vee (y \wedge \neg z)\}$
S ₀₁ ⁿ	$S_0^n \cap M$	$\{T_2^{n+1}, 1\}$
S ₀₁	$S_0 \cap M$	$\{x \vee (y \wedge z), 1\}$
S ₀₀ ⁿ	$S_0^n \cap R_2 \cap M$	$\{x \vee (y \wedge z), T_2^3\}$ if $n = 2$, $\{T_2^{n+1}\}$ if $n \geq 3$
S ₀₀	$S_0 \cap R_2 \cap M$	$\{x \vee (y \wedge z)\}$
S ₁₂ ⁿ	$S_1^n \cap R_2$	$\{x \wedge (y \vee \neg z), T_n^{n+1}\}$
S ₁₂	$S_1 \cap R_2$	$\{x \wedge (y \vee \neg z)\}$
S ₁₁ ⁿ	$S_1^n \cap M$	$\{T_n^{n+1}, 0\}$
S ₁₁	$S_1 \cap M$	$\{x \wedge (y \vee z), 0\}$
S ₁₀ ⁿ	$S_1^n \cap R_2 \cap M$	$\{x \wedge (y \vee z), T_2^3\}$ if $n = 2$, $\{T_n^{n+1}\}$ if $n \geq 3$
S ₁₀	$S_1 \cap R_2 \cap M$	$\{x \wedge (y \vee z)\}$
D	$\{f \in \text{BF} \mid f \text{ is self-dual}\}$	$\{\text{maj}(x, \neg y, \neg z)\}$
D ₁	$D \cap R_2$	$\{\text{maj}(x, y, \neg z)\}$
D ₂	$D \cap M$	$\{\text{maj}(x, y, z)\}$
L	$\{f \in \text{BF} \mid f \text{ is affine}\}$	$\{x \oplus y, 1\}$
L ₀	$L \cap R_0$	$\{x \oplus y\}$
L ₁	$L \cap R_1$	$\{x \leftrightarrow y\}$
L ₂	$L \cap R_2$	$\{x \oplus y \oplus z\}$
L ₃	$L \cap D$	$\{x \oplus y \oplus z \oplus 1\}$
E	$\{f \in \text{BF} \mid f \text{ is constant or a conjunction}\}$	$\{x \wedge y, 0, 1\}$
E ₀	$E \cap R_0$	$\{x \wedge y, 0\}$
E ₁	$E \cap R_1$	$\{x \wedge y, 1\}$
E ₂	$E \cap R_2$	$\{x \wedge y\}$
V	$\{f \in \text{BF} \mid f \text{ is constant or a disjunction}\}$	$\{x \vee y, 0, 1\}$
V ₀	$V \cap R_0$	$\{x \vee y, 0\}$
V ₁	$V \cap R_1$	$\{x \vee y, 1\}$
V ₂	$V \cap R_2$	$\{x \vee y\}$
N	$\{f \in \text{BF} \mid f \text{ is essentially unary}\}$	$\{\neg x, 0, 1\}$
N ₂	$N \cap D$	$\{\neg x\}$
I	$\{f \in \text{BF} \mid f \text{ is constant or a projection}\}$	$\{\text{id}, 0, 1\}$
I ₀	$I \cap R_0$	$\{\text{id}, 0\}$
I ₁	$I \cap R_1$	$\{\text{id}, 1\}$
I ₂	$I \cap R_2$	$\{\text{id}\}$

Table 1. The list of all Boolean clones with definitions and bases, where T_k^n denotes the k -threshold function of arity n ,

Question: Does φ have a satisfying assignment of weight exactly k ?

The corresponding problem for circuits is denoted by $\text{p-WCIRCSAT}(B)$.

Our goal is to obtain a complexity classification of these problems according to B . Observe that if B_1 and B_2 are two finite sets of Boolean functions such that $B_1 \subseteq [B_2]$, then every function of B_1 can be expressed by a B_2 -formula, its so-called B_2 -representation. This provides a canonical fpt-reduction from $\text{p-WCIRCSAT}(B_1)$ to $\text{p-WCIRCSAT}(B_2)$: Given (\mathcal{C}_1, k) an input of the first problem, construct in logarithmic space the pair (\mathcal{C}_2, k) in which \mathcal{C}_2 is obtained from \mathcal{C}_1 in replacing all B_1 -functions by their B_2 -representation (observe that since B_1 and B_2 are not part of the input the cost of computing these representations is not taken into account in the complexity of the reduction). However, since the B_2 -representation of some function may use some input variable more than once (see Example 1) this reduction is not necessarily polynomial when we turn to formulæ. In order to avoid an exponential blow-up when dealing with formulæ we will seek *short representations*, i.e., representations in which every variable appears exactly once. We say that a set B *efficiently implements* an n -ary function f if there is a B -formula φ that is equivalent to $f(x_1, \dots, x_n)$ and in which each x_i appears exactly once. In that case the above reduction applies to formulæ as well, and we get a canonical fpt-reduction from $\text{p-WSAT}(\{f\})$ to $\text{p-WCIRCSAT}(B)$.

4 The Complexity of Weighted Satisfiability Problems

In the following B denotes a finite set of Boolean functions,

Lemma 2. $\text{p-WSAT}(B)$ is in W[SAT] and $\text{p-WCIRCSAT}(B)$ is in W[P] .

Proof. For circuits it follows from the fact that $\{\wedge, \vee, \neg\}$ can represent all Boolean functions. Therefore, $\text{p-WCIRCSAT}(B) \leq^{\text{fpt}} \text{p-WCIRCSAT}(\{\wedge, \vee, \neg\})$, the latter being in W[P] by definition, thus proving membership in W[P] . For formulæ the fact that $\text{p-WSAT}(B) \leq^{\text{fpt}} \text{p-WSAT}(\{\wedge, \vee, \neg\})$ follows from [16, Theorem 4.7]. \square

Lemma 3. If B contains the constants 0 and 1, and if $M \subseteq [B]$, then $\text{p-WSAT}(B)$ is W[SAT] -complete, and $\text{p-WCIRCSAT}(B)$ is W[P] -complete.

Proof. It is proved in [1] that $\text{p-WSAT}(\{\wedge, \vee\})$ is W[SAT] -complete and that $\text{p-WCIRCSAT}(\{\wedge, \vee\})$ is W[P] -complete. Since $M \subseteq [B]$, either $[B] = M$ or $[B] = \text{BF}$. In both cases, since by assumption B contains the two constants and according to [15, Lemma 4], B efficiently implements the functions \wedge and \vee . This shows that $\text{p-WSAT}(\{\wedge, \vee\}) \leq^{\text{fpt}} \text{p-WSAT}(B)$, thus concluding the proof for formulæ. The same reduction actually shows the hardness result for the circuit problem. \square

The following lemma shows that we can freely use the constant 1 as soon as the function \wedge can be computed by a B -circuit.

Lemma 4. If $[B]$ contains the conjunction function then $\text{p-WSAT}(B \cup \{1\})$ fpt-reduces to $\text{p-WSAT}(B)$ and $\text{p-WCIRCSAT}(B \cup \{1\})$ fpt-reduces to $\text{p-WCIRCSAT}(B)$.

Proof. Let φ be a $B \cup \{1\}$ -formula. Let $\varphi' := \varphi[1/t] \wedge t$, where t is a fresh variable. Since $\wedge \in [B]$ the formula φ' can be represented by a B -formula. Moreover, it is clear that φ has a satisfying assignment of weight k if and only if φ' has a satisfying assignment of weight $k + 1$, thus showing that $\text{p-WSAT}(B \cup \{1\}) \leq^{\text{fpt}} \text{p-WSAT}(B)$. The same proof shows a similar result for circuits. \square

Dealing with the constant 0 requires additional tricks. First let us introduce variants of our problems as technical tools: $\text{p-WSAT}^+(B)$ and $\text{p-WCIRCSAT}^+(B)$ denote our original problems restricted to monotone instances. Let us observe that Lemmas 3 and 4 still hold for these variants. Obviously proving hardness for these variants is enough for proving hardness for the original problems.

Definition 5. Let l be a positive integer. A formula ψ is l -costly if every satisfying assignment of ψ has weight at least l .

Lemma 6. If for every non-negative integer k there exists a $(k + 1)$ -costly B -formula, then we obtain the reductions $\text{p-WSAT}^+(B \cup \{0\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B)$ and $\text{p-WCIRCSAT}^+(B \cup \{0\}) \leq^{\text{fpt}} \text{p-WCIRCSAT}^+(B)$.

Proof. Let φ be a monotone $B \cup \{0\}$ -formula. Let ψ be a $(k + 1)$ -costly B -formula over m variables. Consider the B -formula φ' obtained from φ in replacing every occurrence of 0 by $\psi(y_1, \dots, y_m)$, where the y_i 's are fresh variables. If there is a satisfying assignment for φ of weight k , then it can be extended to a satisfying assignment of φ' of same weight by setting all the y_i 's to 0. Conversely, any truth assignment to the variables of φ' of weight k makes the formula ψ false (since it is $(k + 1)$ -costly). Therefore, the restriction of such an assignment to the variables of φ provides a satisfying assignment of φ of weight at most k . Since by assumption φ is monotone this implies that φ is k -satisfiable. To sum up, φ has a satisfying assignment of weight k if and only if φ' has a satisfying assignment of weight k , thus concluding the proof. The same proof holds for circuits. Observe that in the reduction the size of φ' is in the worse case the size of φ times the size of ψ , which is an arbitrary function depending on k . Therefore, the reduction here is an fpt-reduction, but not a many-one-log-space reduction as in Lemma 4. \square

Lemma 7. If $[B]$ contains some q -threshold function (of arbitrary arity $n \geq q$) where $q \geq 2$, then there exists an l -costly B -formula for any $l \geq 1$.

Proof. Build a balanced tree of depth d whose gates are q -threshold functions, with $q \geq 2$. Every satisfying assignment of this tree must have weight at least q^d . Thus, in choosing d large enough, say such that $q^d > l$, we can build this way an l -costly B -formula. \square

Lemma 8. If $D_2 \subseteq [B]$, then $\text{p-WSAT}(B)$ is W[SAT] -complete and $\text{p-WCIRCSAT}(B)$ is W[P] -complete.

Proof. Suppose that $D_2 \subseteq [B]$. Note that in this case $M \subseteq B \cup \{0, 1\}$. Therefore, $\text{p-WSAT}^+(B \cup \{0, 1\})$ is W[SAT] -complete and $\text{p-WCIRCSAT}^+(B \cup \{0, 1\})$ is W[P] -complete according to Lemma 3. The hardness result for the formula problem is proved by the following sequence of reductions: $\text{p-WSAT}^+(B \cup \{0, 1\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B \cup$

$\{0\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B) \leq^{\text{fpt}} \text{p-WSAT}(B)$. The first reduction holds according to Lemma 4 since $[B \cup \{0\}]$ is a superset of S_{11}^2 , and thus $\wedge \in [B]$. Observe that B can express the ternary majority function (see Table 1), which is a 2-threshold function. Thus, the second reduction follows from Lemma 7 and Lemma 6. As mentioned above the last reduction is trivial. The same sequence of reductions provides the desired result for circuits. \square

Lemma 9. *If $S_{10} \subseteq [B]$, then $\text{p-WSAT}(B)$ is W[SAT] -complete, and $\text{p-WCIRCSAT}(B)$ is W[P] -complete.*

Proof. Suppose that $S_{10} \subseteq [B]$. We still have $M \subseteq [B \cup \{0, 1\}]$, therefore $\text{p-WSAT}^+(B \cup \{0, 1\})$ is W[SAT] -complete according to Lemma 3. Hardness is proved by the following sequence of reductions: $\text{p-WSAT}^+(B \cup \{0, 1\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B \cup \{1\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B)$. The first reduction holds according to Lemma 7 and Lemma 6. Indeed $[B \cup \{1\}]$ is a superset of M_1 , and thus a superset of D_2 . For this reason $[B \cup \{1\}]$ contains the ternary majority function, which is a 2-threshold function. The second reduction follows from Lemma 4 since $\wedge \in S_{10}$ and thus $\wedge \in [B]$. The same sequence of reductions provides the desired result for circuits. \square

Lemma 10. *If $S_{00}^n \subseteq [B]$ for some $n \geq 2$, then $\text{p-WSAT}(B)$ is W[SAT] -complete and $\text{p-WCIRCSAT}_{\leq}(B)$ is W[P] -complete.*

Proof. Observe that $M = [S_{00}^n \cup \{0, 1\}]$, and hence that $M \subseteq [B \cup \{0, 1\}]$. Thus, according to Lemma 3, $\text{p-WSAT}^+(B \cup \{0, 1\})$ is W[SAT] -complete. The lemma follows then from the following sequence of reductions: $\text{p-WSAT}^+(B \cup \{0, 1\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B \cup \{0\}) \leq^{\text{fpt}} \text{p-WSAT}^+(B)$. The first reduction holds according to Lemma 4 since $M_0 = [S_{00}^n \cup \{0\}]$, and hence $\wedge \in [B \cup \{0\}]$. Observe that S_{00}^n contains the 2-threshold function T_{n+1}^2 . Hence, the second reduction follows from Lemma 6 and Lemma 7. The same sequence of reductions provides the desired result for circuits. \square

In the following lemmas we prove tractability results for circuit problems (tractability for formulae follows trivially as a special case).

Lemma 11. *If $[B] \subseteq \vee$, or $[B] \subseteq \text{E}$, or $[B] \subseteq \text{L}$, then $\text{p-WSAT}(B)$ and $\text{p-WCIRCSAT}(B)$ are in P .*

Proof. The basic idea is to compute a normal form of the functions computed by such B -circuits, from which it is easy to decide whether the circuits are k -satisfiable.

First, let $\vee_2 \subseteq [B] \subseteq \vee$. Let $\mathcal{C}(x_1, \dots, x_n)$ be a B -circuit. The Boolean function described by \mathcal{C} can be expressed as $f_{\mathcal{C}}(x_1, \dots, x_n) = a_0 \vee (a_1 \wedge x_1) \vee \dots \vee (a_n \wedge x_n)$, where the a_i 's are in $\{0, 1\}$. The values a_i , where $0 \leq i \leq n$, can be determined easily by using the following simple facts: $a_0 = 0$ if and only if $f_{\mathcal{C}}(0, \dots, 0) = 0$ and $a_i = 0$ for $1 \leq i \leq n$ if and only if $a_0 = 0$ and $f_{\mathcal{C}}(0^{i-1}, 1, 0^{n-i}) = 0$. This can be checked in polynomial time since the value problem for B -circuits is known to be in P (see [14]). We conclude that the normal form, from which deciding k -satisfiability is easy, can be computed efficiently.

Tractability for $\text{E}_2 \subseteq [B] \subseteq \text{E}$ follows as above in computing the dual normal form, i.e., $f_{\mathcal{C}}(x_1, \dots, x_n) = a_0 \wedge (a_1 \vee x_1) \wedge \dots \wedge (a_n \vee x_n)$.

Finally let $L_2 \subseteq [B] \subseteq L$. The proof follows by computing the normal form as $f_C(x_1, \dots, x_n) = a_0 \oplus (a_1 \wedge x_1) \oplus \dots \oplus (a_n \wedge x_n)$. Similar to the above the values a_i 's can be easily determined by n well-chosen oracles to the circuit value problem. Again deciding k -satisfiability is easy from this normal form. \square

Lemma 12. *If $S_{00} \subseteq [B] \subseteq S_0$ then p -WSAT(B) and p -WCIRCSAT(B) are in P.*

Proof. Observe that \rightarrow is the basis of S_0 . Since $B \subseteq S_0$ any B -circuit can be transformed in logarithmic space into an $\{\rightarrow\}$ -circuit in locally replacing each gate $f \in B$ by its $\{\rightarrow\}$ -representation. Note that such a circuit has satisfying assignments of all possible weights (except may be the all-0 one). To see this, start from the output gate and go backwards. At every gate take backwards the edge corresponding to the right argument of the implication. Thus we get a path from the output gate to a 'target-gate' which is either a variable or the constant 1 (the constant 0 does not appear by assumption). In case of a variable, setting this variable to 1 is sufficient to satisfy the circuit. Therefore from this we can build satisfying assignments of any weight ≥ 1 . If by the described procedure we reach the constant 1, then the circuit represents a tautology. The special case of the all-0 assignment has to be dealt with separately. \square

We are now in a position to state our main result for weighted satisfiability decision problems. Indeed a careful examination of Post's lattice shows that the above lemmas cover all the lattice and thus provide a complete classification.

Theorem 13. *Let B be a finite set of Boolean functions.*

1. *If $D_2 \subseteq [B]$ or $S_{10} \subseteq [B]$ or $S_{00}^n \subseteq [B]$ for some $n \geq 2$, then p -WSAT(B) is W[SAT]-complete, and p -WCIRCSAT(B) is W[P]-complete under fpt-reductions.*
2. *In all other cases p -WSAT(B) and p -WCIRCSAT(B) are in P.*

5 Complexity of the Counting Problems

There are natural counting problems associated with the decision problems studied above.

Problem: p -#WSAT(B)
Input: a B -formula φ and $k \in \mathbb{N}$
Parameter: k
Output: Number of satisfying assignments for φ of weight exactly k

The corresponding problem for circuits is denoted by p -#WCIRCSAT(B).

In the following, as proposed in [8], we use \leq^{fpt} to designate a parsimonious fpt-reduction, while $\leq^{\text{fpt-T}}$ will refer to a Turing fpt-reduction. Let us now introduce two complexity classes, which are the counting analogues of W[SAT] and W[P]. To the best of our knowledge, the class #W[SAT] has not been considered in the literature so far.

Definition 14. *The class #W[SAT] is the closure of p -#WSAT($\{\wedge, \vee, \neg\}$) under fpt-parsimonious reductions, that is*

$$\#W[\text{SAT}] := [p\text{-}\#\text{WSAT}(\{\wedge, \vee, \neg\})]^{\text{fpt}}.$$

Whereas it was originally defined in terms of counting accepting runs of a κ -restricted nondeterministic Turing machines, the class $\#W[P]$ can be defined in a similar way (see, e.g., [8, page 366]):

Definition 15. *The class $\#W[P]$ is the closure of $p\text{-}\#\text{WCIRCSAT}(\{\wedge, \vee, \neg\})$ under fpt -parsimonious reductions, that is*

$$\#W[P] := [p\text{-}\#\text{WCIRCSAT}(\{\wedge, \vee, \neg\})]^{\text{fpt}}.$$

Proposition 16. *$p\text{-}\#\text{WSAT}(B)$ is in $\#W[\text{SAT}]$ and $p\text{-}\#\text{WCIRCSAT}(B)$ is in $\#W[P]$.*

Proof. This follows from the proof of Lemma 2 in observing that all reductions are parsimonious. \square

We first state two lemmas which allow to take care of the constants. The first one is simply the observation that the reduction in Lemma 4 is parsimonious.

Lemma 17. *If $[B]$ contains the conjunction function then $p\text{-}\#\text{WSAT}(B \cup \{1\})$ parsimoniously fpt -reduces to $p\text{-}\#\text{WSAT}(B)$ and $p\text{-}\#\text{WCIRCSAT}(B \cup \{1\})$ parsimoniously fpt -reduces to $p\text{-}\#\text{WCIRCSAT}(B)$.*

Given a formula φ , let $\#\text{Sat}_k(\varphi)$ denote its number of satisfying assignments of weight exactly k .

Lemma 18. *If $[B]$ contains the disjunction function then $p\text{-}\#\text{WSAT}(B \cup \{0\})$ Turing fpt -reduces to $p\text{-}\#\text{WSAT}(B)$ and $p\text{-}\#\text{WCIRCSAT}(B \cup \{0\})$ Turing fpt -reduces to $p\text{-}\#\text{WCIRCSAT}(B)$.*

Proof. Let φ be a $B \cup \{0\}$ -formula. Let $\varphi' := \varphi[0/f] \vee f$, where f is a fresh variable. Since $\vee \in [B]$ the formula φ' can be represented by a B -formula. Moreover, $\#\text{Sat}_k(\varphi') = \#\text{Sat}_k(\varphi) + \binom{n}{k-1}$, thus showing that $p\text{-}\#\text{WSAT}(B \cup \{0\}) \leq^{\text{fpt}} p\text{-}\#\text{WSAT}(B)$. The reduction consists of a precomputation phase, one oracle call, and then some postcomputation, namely the summation of the result from the oracle and the binomial coefficient; hence it is actually a 1-Turing fpt -reduction. The same reduction holds for circuits. \square

Lemma 19. *If $M_2 \subseteq [B]$, then $p\text{-}\#\text{WSAT}(B)$ is complete for the class $\#W[\text{SAT}]$ and $p\text{-}\#\text{WCIRCSAT}(B)$ is complete for $\#W[P]$, both under Turing fpt -reductions.*

Proof. First we prove that $p\text{-}\#\text{WSAT}(\{\wedge, \vee, \neg\}) \leq^{\text{fpt-T}} p\text{-}\#\text{WSAT}(\{\wedge, \vee\})$. Let φ be a $\{\wedge, \vee, \neg\}$ -formula. Without loss of generality one can suppose that φ is in negation normal form, NNF. Indeed, if it is not the case one can transform it in NNF in polynomial time in pushing the negation symbols in front of variables in applying de Morgan's laws and the double negation elimination. Now we use a well-known reduction to express a general formula as conjunction of a monotone and a negated monotone formula. The formula $\varphi(\bar{x}) = \varphi(x_1, \dots, x_n)$ is mapped to $\psi(x_1, \dots, x_n, y_1, \dots, y_n)$ where $\neg x_i$ is replaced by a fresh variable y_i . This gives

$$\begin{aligned} \#\text{Sat}_k(\varphi(\bar{x})) &= \#\text{Sat}_k(\psi(\bar{x}, \bar{y}) \wedge \bigwedge_{i=1}^n (x_i \vee y_i) \wedge \neg \bigvee_{i=1}^n (x_i \wedge y_i)) \\ &= \#\text{Sat}_k(\alpha(\bar{x}, \bar{y}) \wedge \neg \beta(\bar{x}, \bar{y})) \end{aligned}$$

where α and β are $\{\wedge, \vee\}$ -formulae defined by

$$\alpha(\bar{x}, \bar{y}) = \psi(\bar{x}, \bar{y}) \wedge \bigwedge_{i=1}^n (x_i \vee y_i) \text{ and } \beta(\bar{x}, \bar{y}) = \bigvee_{i=1}^n (x_i \wedge y_i).$$

Thus we have

$$\#\text{Sat}_k(\varphi) = \#\text{Sat}_k(\alpha) - \#\text{Sat}_k(\alpha \wedge \beta).$$

Indeed, if a k -assignment satisfies α but not $\alpha \wedge \beta$, then it satisfies $\alpha \wedge \neg\beta$. Conversely a k -assignment that satisfies $\alpha \wedge \neg\beta$, satisfies α and does not satisfy $\alpha \wedge \beta$. This proves that $\text{p-}\#\text{WSAT}(\{\wedge, \vee, \neg\})$ 2-Turing fpt-reduces to $\text{p-}\#\text{WSAT}(\{\wedge, \vee\})$.

Now, $M \subseteq [B \cup \{0, 1\}]$, hence there are short $B \cup \{0, 1\}$ -representations of \wedge and \vee [15], therefore $\text{p-}\#\text{WSAT}(\{\wedge, \vee\}) \leq^{\text{fpt}} \text{p-}\#\text{WSAT}(B \cup \{0, 1\})$. Since both \vee and \wedge are in M_2 , and thus in B we can get rid of the constants by applying successively Lemmas 18 and 17.

For circuits $\text{p-}\#\text{WCIRCSAT}(\{\wedge, \vee, \neg\})$ is $\#\text{W[P]}$ -complete by definition (see Def. 15). The completeness of $\text{p-}\#\text{WCIRCSAT}(\{\wedge, \vee\})$ results from the fact that the reduction from the weighted satisfiability of Boolean circuits to the weighted satisfiability of monotone circuits given in [8, Thm 3.14] is parsimonious. Then the same sequence of reductions as in the case of formulae allows to conclude. \square

Lemma 20. *If $S_{10} \subseteq [B]$, then $\text{p-}\#\text{WSAT}(B)$ is complete for the class $\#\text{W[SAT]}$ and $\text{p-}\#\text{WCIRCSAT}(B)$ is complete for $\#\text{W[P]}$, both under Turing fpt-reductions.*

Proof. Observe that $M_1 \subseteq [B \cup \{1\}]$, therefore, by Lemma 19, $\text{p-}\#\text{WSAT}(B \cup \{1\})$ is $\#\text{W[SAT]}$ -complete. The result is then obtained by the reduction $\text{p-}\#\text{WSAT}(B \cup \{1\}) \leq^{\text{fpt}} \text{p-}\#\text{WSAT}(B)$, which follows from Lemma 17 since $\wedge \in S_{10} \subseteq [B]$. A similar proof provides the result for circuits. \square

Lemma 21. *If $S_{00} \subseteq [B]$, then $\text{p-}\#\text{WSAT}(B)$ is complete for the class $\#\text{W[SAT]}$ and $\text{p-}\#\text{WCIRCSAT}(B)$ is complete for $\#\text{W[P]}$ -complete, both under Turing fpt-reductions.*

Proof. Similar to the proof above in using the fact that $\text{p-}\#\text{WSAT}(B \cup \{0\})$ is $\#\text{W[SAT]}$ -complete and the reduction $\text{p-}\#\text{WSAT}(B \cup \{0\}) \leq^{\text{fpt-T}} \text{p-}\#\text{WSAT}(B)$, which is obtained through Lemma 18. \square

Lemma 22. *If $D_2 \subseteq [B]$, then $\text{p-}\#\text{WSAT}(B)$ is $\#\text{W[SAT]}$ -complete and $\text{p-}\#\text{WCIRCSAT}(B)$ is $\#\text{W[P]}$ -complete, both under Turing fpt-reductions.*

Proof. Observe that $[D_2 \cup \{1\}] = S_{01}^2$, therefore according to Lemma 21 we get hardness for $\text{p-}\#\text{WSAT}(B \cup \{1\})$. It remains to show that $\text{p-}\#\text{WSAT}(B \cup \{1\}) \leq^{\text{fpt}} \text{p-}\#\text{WSAT}(B)$. For this we will use some specific functions g_l which belong to D_2 (for they are self-dual) and which are defined as follows:

$$g_l(x_1, \dots, x_l, 0) = x_1 \wedge \dots \wedge x_l, \text{ and } g_l(x_1, \dots, x_l, 1) = x_1 \vee \dots \vee x_l.$$

Let maj denote the ternary majority function, which is also a function from D_2 . Let φ be a $B \cup \{1\}$ -formula. Consider φ' defined by:

$$\varphi' := \text{maj}(\varphi[1/t], t, g_{k+2}(y_1, \dots, y_{k+2}, t)),$$

where t and the y_i 's are fresh variables. Then we map (φ, k) to $(\varphi', k + 1)$. Observe that every assignment which sets t to 0 and that satisfies φ' has at least weight $k + 2$ (and thus is too costly). Now consider assignments that set t to 1 and that satisfies φ' . Either they satisfy $g_{k+2}(y_1, \dots, y_{k+2}, 1) = y_1 \vee \dots \vee y_{k+2}$ or they don't. In the latter case they have to satisfy φ . To sum up we have the following equality:

$$\#\text{Sat}_k(\varphi') = \sum_{j=1}^{k-1} \binom{k+2}{j} \binom{n}{k-j-1} + \#\text{Sat}_{(k-1)}(\varphi).$$

As in Lemma 18 above, we thus obtain a 1-Turing fpt-reduction. Note that we do not obtain a Turing polynomial-time-reduction, since the time required to compute the B -representation of g_{k+2} maybe too high; we only have parameterized polynomial-time. \square

Lemma 23. *If $B \subseteq \vee$, or $B \subseteq \text{E}$, or $B \subseteq \text{L}$, then $\text{p-}\#\text{WSAT}(B)$ is in FP.*

Proof. Easy after having computed the normal form as in Lemma 11.

We are now in a position to state the full classification for the counting problems.

Theorem 24. *Let B be a finite set of Boolean functions.*

1. *If $\text{D}_2 \subseteq [B]$ or $\text{S}_{10} \subseteq [B]$ or $\text{S}_{00} \subseteq [B]$, then $\text{p-}\#\text{WSAT}(B)$ is $\#\text{W}[\text{SAT}]$ -complete and $\text{p-}\#\text{WCIRCSAT}(B)$ is $\#\text{W}[\text{P}]$ -complete, both under Turing fpt-reductions.*
2. *In all other cases $\text{p-}\#\text{WSAT}(B)$ and $\text{p-}\#\text{WCIRCSAT}(B)$ are in FP.*

6 Conclusion

In this paper we obtained a complete classification of the parameterized complexity of the weighted satisfiability problem, depending on the Boolean functions allowed to appear, both for formulas and for Boolean circuits, and both in the decision and in the counting context. It may seem a little disappointing not to see any involved FPT algorithm in our classification, contrary to the classification of Marx [10] in Schaefer's framework that revealed some nontrivial FPT algorithms. However let us advocate that Post's framework does not seem well adapted to such nontrivial algorithms. Indeed in the classifications that appeared in the literature in the past, the tractable cases usually turned out to be trivially algorithmically solvable (with the possible exception of auto-epistemic logic [5] in which a nontrivial algorithm was developed for the affine fragment).

Parameterized counting complexity was introduced in [7,11], but surprisingly is not much developed so far. We see our paper also as a contribution to this study. While the class $\#\text{W}[\text{P}]$ was introduced in [7] in analogy to $\text{W}[\text{P}]$, we here introduced $\#\text{W}[\text{SAT}]$ in analogy to $\text{W}[\text{SAT}]$, and we present natural complete satisfiability problems for both classes. So we believe our study makes a step towards a better understanding of counting problems within parameterized complexity.

One might also consider the variants of the weighted satisfiability problem in which the task is to find a satisfying assignment of weight at most k or at least k . Preliminary results exist for the monotone fragment, see, e.g., [8,6]. We leave further results for these variants as future work.

Acknowledgement. We are grateful to Arne Meier (Hannover) and Steffen Reith (Wiesbaden) for helpful discussions.

References

1. K. R. Abrahamson, R.G. Downey, and M.R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for $W[P]$ and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995.
2. M. Bauland, T. Schneider, H. Schnoor, I. Schnoor, and H. Vollmer. The complexity of generalized satisfiability for linear temporal logic. *Logical Methods in Computer Science*, 5(1), 2008.
3. E. Böehler, N. Creignou, M. Galota, S. Reith, H. Schnoor, and H. Vollmer. Boolean circuits as a representation for Boolean functions: Efficient algorithms and hard problems. *Logical Methods in Computer Science*, 2012. To appear.
4. E. Böehler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks I: Post’s lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.
5. N. Creignou, A. Meier, M. Thomas, and H. Vollmer. The complexity of reasoning for fragments of autoepistemic logic. *ACM Transactions on Computational Logic*, 2012. To appear.
6. S. Dantchev, B. Martin, and S. Szeider. Parameterized proof complexity. *Computational Complexity*, 20(1):51–85, 2011.
7. J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM J. Comput.*, 33(4):892–922, 2004.
8. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
9. H. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.
10. D. Marx. Parameterized complexity of constraint satisfaction problems. *Computational Complexity*, 14(2):153–183, 2005.
11. C. McCartin. Parameterized counting problems. *Annals of Pure and Applied Logic*, 138(1-3):147–182, 2006.
12. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
13. E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.
14. S. Reith and K. Wagner. The complexity of problems defined by Boolean circuits. In *Proceedings Mathematical Foundation of Informatics (MFI99)*, pages 141–156. World Science Publishing, 2005.
15. H. Schnoor. The complexity of model checking for boolean formulas. *Int. J. Found. Comput. Sci.*, 21(3):289–309, 2010.
16. M. Thomas. On the applicability of Post’s lattice. *CoRR*, abs/1007.2924, 2010.
17. H. Vollmer. *Introduction to Circuit Complexity*. Springer Verlag, Berlin Heidelberg, 1999.