# A Purely Democratic Characterization of W[1]

Michael Fellows[⋆1], Danny Hermelin[⋆⋆2], Moritz Müller[3], and Frances Rosamond[⋆⋆⋆1]

[1] Parameterized Complexity Research Unit,
The University of Newcastle, Callaghan NSW 2308 - Australia
{michael.fellows,frances.rosamond}@newcastle.edu.au
[2] Department of Computer Science, University of Haifa,
Mount Carmel, Haifa 31905 - Israel
danny@cri.haifa.ac.il
[3] Abteilung für Mathematische Logik, Albert Ludwigs Universität Freiburg,
Eckerstrasse 1, 79098 Freiburg - Germany
moritz.mueller@math.uni-freiburg.de

**Abstract.** We give a novel characterization of W[1], the most important fixed-parameter intractability class in the W-hierarchy, using Boolean circuits that consist solely of majority gates. Such gates have a Boolean value of 1 if and only if more than half of their inputs have value 1. Using majority circuits, we define an analog of the W-hierarchy which we call the $\widetilde{\mathrm{W}}$-hierarchy, and show that $\mathrm{W}[1] = \widetilde{\mathrm{W}}[1]$ and $\mathrm{W}[t] \subseteq \widetilde{\mathrm{W}}[t]$ for all $t$. This gives the first characterization of W[1] based on the weighted satisfiability problem for monotone Boolean circuits rather than antimonotone. Our results are part of a wider program aimed at exploring the robustness of the notion of weft, showing that it remains a key parameter governing the combinatorial nondeterministic computing strength of circuits, no matter what type of gates these circuits have.

⋆ Research supported by the Australian Research Council, and by an Alexander von Humboldt Foundation Research Award.
⋆⋆ Partially supported by the Caesarea Rothschild Institute.
⋆⋆⋆ Research supported by the Australian Research Council.

## 1 Introduction

Arguably the most important class in the W-hierarchy is W[1]. From a theoretical point of view, it can be viewed as the parameterized analog of NP since it contains many parameterized variants of classical NP-complete problems such as $k$-Independent Set, $k, \ell$-Longest Common Subsequence, and, most importantly, because the $k$-Step Halting Problem for Turing Machines of Unlimited Nondeterminism is complete for W[1]. This is a parameterized analog of the generic NP-complete problem used in Cook's theorem [DFKHW94,DFS99]. From a practical standpoint, W[1] is the most important complexity class for showing fixed-parameter intractability results, providing an easy accessible platform for showing such results [DF95]. Indeed, since the identification of the first complete problems for W[1], there has been a slew of fixed-parameter intractability results, reminiscent in some sense of the early days of NP-completeness.

The key combinatorial objects used to formulate W[1] and the W-hierarchy are constant depth logic circuits that model Boolean functions in the natural way. Their combinatorial nondeterministic computing strength is governed by their *weft*, defined to be the maximum number of unbounded indegree gates in any path from the input gates to the output. The generic complete problem for the W-hierarchy is the $k$-Weighted Weft-$t$ Circuit Satisfiability problem, that takes as input a constant depth weft-$t$ circuit $C$ and a parameter $k$ and asks whether $C$ has a weight $k$ satisfying assignment (*i.e.* an assignment setting exactly $k$ input gates to 1). The class W[$t$] is then defined to be the class of parameterized problems which are parameterized reducible to $k$-Weighted Weft-$t$ Circuit Satisfiability.

In this paper, we explore an alternative, purely monotone characterization of W[1] and the W-hierarchy using a different type of Boolean circuit, namely, a majority circuit. In this type of circuit, we replace the role of logical gates by majority gates which have value 1 when more than half of their inputs have value 1. Using a majority circuit analog of $k$-Weighted Weft-$t$ Circuit Satisfiability, we obtain the $\widetilde{\text{W}}$-hierarchy. Our main results are:

**Theorem 1.** $\text{W}[1] = \widetilde{\text{W}}[1]$.

**Theorem 2.** $\text{W}[t] \subseteq \widetilde{\text{W}}[t]$ *for any positive integer* $t$.

Note that in proving Theorem 2, we use Theorem 1 for the case of $t = 1$. This is not uncommon, since most proofs involving the W-hierarchy require special treatment of W[1] (see [DF99,FG06] for many examples).

The importance of these results are twofold. First, Theorem 1 gives an alternative way of showing fixed-parameter intractability results. The complete problems of W[1] are usually antimonotone by nature, where a parameterized problem is antimonotone if when an instance with a given parameter is known to be a "yes"-instance, then it is also known to be a "yes"-instance for smaller parameter values (*e.g.* maximization problems). In circuit terms, it is known that one could use an antimonotone weft-1 circuit to show W[1]-hardness, but this is not so for known for monotone circuits. Majority circuits, however, are monotone circuits and so they might come in handy in reductions for monotone problems. This is what makes the proof of Theorem 1 so combinatorially challenging.

Second, our results suggest a robustness in the notion of weft. Indeed, there has long been quite a bit of informal criticism against the naturality of this notion. This work, along with another work by almost the same set of authors [FFHMR07], aims at showing that this is not necessarily so. We do so, by showing that if one replaces the role of logical gates in circuits by nontrivial combinatorial gates, then the notion of weft still generally remains the central property governing the nondeterministic combinatorial computing power of circuits. It seems that no matter what your

favorite selection of combinatorial gates, the number of unbounded in-degree gates from the input layer to the output will still determine the parameterized complexity of finding weight $k$ satisfying assignments to your circuit.

The paper is organized as follows. In the next section we briefly review basic concepts of parameterized complexity, and formally introduce the notion of majority circuits. We then proceed in Sections 3 and 4 to prove Theorem 1, where in Section 3 we prove that $\mathrm{W}[1] \subseteq \widetilde{\mathrm{W}}[1]$, and in Section 4 we prove $\widetilde{\mathrm{W}}[1] \subseteq \mathrm{W}[1]$. Section 5 is devoted to proving Theorem 2. In Section 6 we give a brief summary of the paper, and discuss open problems.

## 2 Preliminaries

In the following we discuss notations and concepts that we use throughout the paper. In particular, we briefly review basic concepts from parameterized complexity, and formally define pure majority circuits, which play the leading role in this paper, and the corresponding $\widetilde{\mathrm{W}}$-hierarchy. We will assume that the reader is familiar with basic concepts from classical complexity and graph theory.

### 2.1 Parameterized complexity.

A *parameterized problem* (or *parameterized language*) is a subset $L \subseteq \Sigma^* \times \mathbb{N}$, where $\Sigma$ is a fixed alphabet, $\Sigma^*$ is the set of all finite length strings over $\Sigma$, and $\mathbb{N}$ is the set of natural numbers. In this way, an input $(x, k)$ to a parameterized language consists of two parts, where the second part $k$ is the *parameter*. A parameterized problem $L$ is *fixed-parameter tractable* if there exists an algorithm which on a given input $(x, k) \in \Sigma^* \times \mathbb{N}$, decides whether $(x, k) \in L$ in $f(k)poly(n)$ time, where $f$ is an arbitrary computable function solely in $k$, and $poly(n)$ is a polynomial in the total input length $n = |(x, k)|$. Such an algorithm is said to run in FPT-*time*, and FPT is the class of all parameterized problems that can be solved by an FPT-time algorithm (*i.e.* all problems which are fixed-parameter tractable).

A formal framework for proving *fixed-parameter intractability* was developed over the years, using the notion of *parameterized reductions*. A parameterized reduction from a parameterized problem $L$ to another parameterized problem $L'$ is a FPT-time computable mapping that maps an instance $(x, k) \in \Sigma^* \times \mathbb{N}$ to an instance $(x', k') \in \Sigma^* \times \mathbb{N}$ with $(x, k) \in L \iff (x', k') \in L$. Here $k'$ is required to be bounded by some function in $k$.

### 2.2 Logical circuits and the W-hierarchy.

A (*logical*) *circuit* $C$ is a connected directed acyclic graph with labels on its vertices, and a unique vertex with no outgoing edges. The vertices which are of in-degree 0 are called the *input gates* and they are labeled with Boolean variables $x_1, x_2, \ldots$. All other vertices are called the *logical gates* and are labeled with Boolean operators $\wedge$, $\vee$, and $\neg$, where vertices which are labeled $\neg$ have in-degree 1. The unique 0 out-degree vertex is the *output gate*. A *monotone logical circuit* is a logical circuit with no $\neg$-gates. An *antimonotone logical circuit* is a logical circuit where each input gate is connected to the the rest of the circuit via a $\neg$-gate, and there are no other occurrences of $\neg$-gates in the circuit.

An *assignment* $X$ for $C$ is an assignment of 0 or 1 to each of the input gates in $C$. The *weight* of $X$ is the number of input gates that it assigns a 1. The *value* of a logical gate in $C$ under $X$ is obtained straightforwardly according to the label of the gate and the value of its inputs. The *value* $C(X)$ of $C$ under $X$ is the value of the output gate of $C$. We say that $X$ *satisfies* a logical gate in

$C$ if the value of this gate under $X$ is 1, and if it satisfies the output gate of $C$ (*i.e.* $C(X) = 1$), we say that $X$ satisfies $C$.

It is convenient to consider the vertices of $C$ as organized into *layers*. The input gates constitute the *input layer*, the logical gates which are directly connected to them are the *first layer*, and more generally, a vertex is in the *i'th layer* of $C$ if the length of the maximum-length path from it to the input layer equals $i$. The *depth* of a circuit is the length of the maximum-length path from the input layer to the output gate.

There is an important distinction between two types of logical gates in $C$. Small gates are gates which have in-degree bounded by a small constant (usually we can take this constant to be 3). Large gates are vertices with unbounded in-degree. The maximum number of large gates on a path between an input gate and the output of $C$ is the *weft* of $C$.

Constant depth logical circuits are used to define a hierarchy of fixed-parameter intractability known as the W-*hierarchy*. The generic problem for this hierarchy is $k$-Weighted Weft-$t$ Circuit Satisfiability, where $t$ is a problem-dependent positive integer constant. This problem takes as input a constant depth weft-$t$ circuit $C$ and a parameter $k$ and asks to determine whether $C$ has a weight $k$ satisfying assignment. The class W[$t$] is defined to be the class of parameterized problems which are parameterized reducible to $k$-Weighted Weft-$t$ Circuit Satisfiability. The W-hierarchy is then the hierarchy of classes FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots$. It is well known that antimonotone logical circuits are sufficient to define the odd levels of the W-hierarchy, while monotone logical circuits suffice for defining the even levels of the hierarchy [DF99]. Hence, defining $k$-Weighted Weft-$t$ Antimonotone Circuit Satisfiability and $k$-Weighted Weft-$t$ Monotone Circuit Satisfiability in the natural manner, we get the former is complete for all odd $t$ and the latter for all even $t$.

### 2.3 Majority circuits and the $\widetilde{\mathrm{W}}$-hierarchy.

A majority gate is a gate which has value 1 if and only if more than half of its inputs are set to 1. They play the lead role in our novel characterization of W[1], via what we call (*purely*) *majority circuits*. These circuits have majority gates instead of logical gates, and thus have different expressive power in comparison to the ordinary logical circuits of the W-hierarchy. When speaking of majority circuits, we use the same terminology as for their logical counterparts, where all definitions and notions (in particular, the notion of weft) carry through to majority circuits. The only difference is that here we allow multiple edges, where as in logical circuits these are always redundant.

We now define a hierarchy of complexity classes where majority circuits play an analogous role to the role played by logical circuits in the W-hierarchy. This is done via the analogous generic problem $k$-Weighted Weft-$t$ Majority Circuit Satisfiability, which asks whether a given constant depth weft-$t$ majority circuit $C$ has a $k$ weight satisfying assignment, for parameter $k$. The class $\widetilde{\mathrm{W}}[t]$ is defined to be the class of parameterized problems which are parameterized reducible to $k$-Weighted Weft-$t$ Majority Circuit Satisfiability.

Finally, before proceeding, we show that we can always assume that when dealing with majority circuits, we have an additional input whose value is always set to 1. The will prove handy later on in proving both Theorem 1 and Theorem 2.

**Observation 1.** *Without loss of generality, we can assume a weft-t majority circuit $C$, $t \geq 1$, is provided with an input gate which is always assigned the value 1.*

*Proof.* To simulate a 1 in $C$, we replace the output gate of $C$ with an in-degree 2 majority gate which is connected to the old output gate and to a new input gate. Let $C'$ be $C$ after this modification.

Then $C$ has a weight $k$ satisfying assignment iff $C'$ has a weight $k+1$ satisfying assignment where the new input gate is assigned a value of 1. □

## 3 $\mathbf{W[1] \subseteq \widetilde{W}[1]}$

In this section we prove the first part of the main result of this paper, namely that $\mathrm{W}[1] \subseteq \widetilde{\mathrm{W}}[1]$. For this, we introduce an intermediate problem which we feel might also be of independent interest, the $k$-Majority Vertex Cover problem. After formally defining $k$-Majority Vertex Cover, we prove that it is in $\widetilde{\mathrm{W}}[1]$, and also W[1]-hard. From this, it will immediately imply that $\mathrm{W}[1] \subseteq \widetilde{\mathrm{W}}[1]$. We begin with a formal definition of $k$-Majority Vertex Cover.

**Definition 1.** *Given a graph $G$ and a parameter $k$, the* $k$-Majority Vertex Cover *problem asks whether there exists a subset of $k$ vertices in $G$ which covers a majority of the edges of $G$. That is, whether there exists a $V \subseteq V(G)$ with $|V| = k$ and $|\{\{u,v\} \in E(G) \mid \{u,v\} \cap V \neq \emptyset\}| > |E(G)|/2$.*

Recall that a problem is in $\widetilde{\mathrm{W}}[1]$ if it can be parameterically reduced to $k$-Weighted Weft-1 Majority Circuit Satisfiability, which is the problem of determining whether a purely majority circuit of weft 1 (and constant depth) has a weight $k$ satisfying assignment. The above definition, gives a clue to why $k$-Majority Vertex Cover is parametric reducible to $k$-Weighted Weft-1 Majority Circuit Satisfiability. This is established in the following lemma.

**Lemma 1.** $k$-Majority Vertex Cover *is in* $\widetilde{\mathrm{W}}[1]$.

*Proof.* Let $(G,k)$ be an instance of $k$-Majority Vertex Cover. We reduce $(G,k)$ to a weft-1 purely majority circuit $C$ which has a $k+1$ weighted satisfying assignment iff $G$ has a subset of $k$ vertices that cover a majority of its edges.

The construction of $C$ is as follows. Let $n = |V(G)|$ and $m = |E(G)|$. The input layer of $C$ consists of $n$ input gates – one input gate $x_v$ for each vertex $v \in V(G)$. In addition, we use the construction of Observation 1 to ensure that the input layer has an additional input gate of value 1. The first layer of $C$ consists of $m$ small in-degree 3 majority gates, one for each edge in $G$. The gate associated with the edge $\{u,v\} \in E(G)$, is connected to $x_u$, $x_v$, and the constant 1. In this way, an assignment to the input layer corresponds to a subset of vertices in $G$, and a gate in the first layer is satisfied iff the edge associated with this gate is incident to a vertex selected by the assignment. The second layer consists of the output gate which is large majority gate which is connected to all the small majority gates of the first layer.

The above construction clearly runs in FPT-time, and the circuit constructed is of weft 1 and depth 2. Furthermore, its correctness can easily be verified. To see this, first assume that $G$ has a has a subset $V$ of $k$ vertices that cover a majority of its edges, and consider the assignment $X$ which assigns $x_v = 1$ to each gate $x_v$ with $v \in V$. Then $X$ is a weight $k$ assignment which satisfies, by construction, any gate in the first layer associated to an edge covered by $V$. Therefore, it satisfies a majority of the gates in the first layer, and also the output gate. In the other direction, if $X$ is a weight $k$ assignment which satisfies $C$, then $X$ satisfies more than half of the gates in the first layer, and so the subset of $k$ vertices $V = \{v \in V(G) \mid X \text{ assigns } x_v = 1\}$ covers more than half of the edges of $G$. □

The next step is to show that $k$-Majority Vertex Cover is W[1]-hard. This might be a somewhat more surprising result than the previous lemma, since $k$-Vertex Cover and other closely related variants are known to be in FPT [GNW05]. Our proof follows along a similar line of proof used in [GNW05] for showing that $k$-Partial Vertex Cover is W[1]-hard.

**Lemma 2.** $k$-Majority Vertex Cover *is* W[1]*-hard.*

*Proof.* The proof is *via* a reduction from $k$-Independent Set which is known to be W[1]-complete [DF95,DF99]. Recall that given a graph $G$ and a parameter $k$, the $k$-Independent Set problem asks to determine whether $G$ has a subset of $k$ vertices which are pairwise non-adjacent. Let $(G, k)$ be an instance of $k$-Independent Set. We may assume w.l.o.g. that $k+1 < n/4$, since otherwise the trivial brute-force algorithm runs in FPT-time. We construct an instance $(G', k')$ for $k$-Majority Vertex Cover as follows.

Set $n = |V(G)|$, and let $d(v)$ denote the number of vertices in $G$ adjacent to the vertex $v \in V(G)$. The graph $G'$ consists of $G$ as a subgraph, along with additional new vertices and edges. For every vertex $v \in V(G)$, $G'$ consists of $v$ along with $n - 1 - d(v)$ additional new degree 1 vertices which are connected only to $v$. In this way, every vertex $v \in V(G)$ has degree $n - 1$ in $G'$. In addition, $G'$ consists of a new vertex $v^*$ which is adjacent to many new degree 1 vertices. The number of these degree 1 neighbors is cho,which is greater than half the total number of edges in $G'$.

To complete our construction, we let $k' = k + 1$. Observe that this construction is indeed a valid parametric reduction. Also, note that all edges of $G$ are preserved in $G'$, and that $|E(G')| = n(n-1)/2 + d(v^*)$. In the remaining part of the proof we show that $G$ has a subset of $k$ pairwise non-adjacent vertices iff $G'$ has a subset of $k'$ vertices which cover more than half of its edges.

Assume $V'$ is a subset of $k'$ vertices in $G'$ which cover more than $n(n-1)/4 + d(v^*)/2$ edges in $G'$. Then it must be that $v^* \in V'$ since all other vertices have degree at most $n - 1$ and since $k + 1 < n/4$. Furthermore, the remaining vertices $V' \setminus \{v^*\}$ must cover at least $k(n-1)$ edges. Since all the vertices in $V(G') \setminus \{v^*\}$ which have degree greater than 1 are vertices of $G$, it follows that $V' \setminus \{v^*\} \subseteq V(G)$. Furthermore, $V' \setminus \{v^*\}$ must be pairwise non-adjacent in $G$, since otherwise $V' \setminus \{v^*\}$ would cover less than $k(n-1)$ edges in $G'$.

Conversely, assume that $G$ has a subset $I \subseteq V(G)$ of $k$ pairwise non-adjacent vertices. Then the number of edges incident to $I$ in $G'$ is exactly $k(n-1)$. It follows that the number of edges that $I \cup \{v^*\}$ cover is $k(n-1) + d(v^*)$, which is more than half of the edges in $G'$ by construction. □

**Corollary 1.** $\mathrm{W}[1] \subseteq \widetilde{\mathrm{W}}[1]$.

## 4 $\widetilde{\mathbf{W}}[\mathbf{1}] \subseteq \mathbf{W}[\mathbf{1}]$

We next prove that $\widetilde{\mathrm{W}}[1] \subseteq \mathrm{W}[1]$, completing the proof of the main result of this paper. In the interests of a clearer presentation, we prove this in two steps. In the first step, we introduce a new problem called $k$-Majority $(p, q)$-DNF Satisfaction, and show that this problem reduces to the generic W[1]-complete $k$-Step Halting problem. Following this, we explain how this construction can be altered to show that $k$-Weighted Weft-1 Majority Circuit Satisfiability also reduces to $k$-Step Halting, proving that $\widetilde{\mathrm{W}}[1] \subseteq \mathrm{W}[1]$.

We being by introducing the $k$-Majority $(p, q)$-DNF Satisfaction problem. A *monotone $(p, q)$-DNF formula* is a boolean formula of the form $\bigvee_{i=1}^{p} \bigwedge_{j=1}^{q} x_{i,j}$, where $x_{i,j}$ are positive boolean literals which are not necessarily distinct. A *family of monotone $(p, q)$-DNF formulas* is a family $\mathcal{D} = (D_1, \ldots, D_m)$ of monotone $(p, q)$-DNF formulas, with a possible overlap in their sets of variables.

**Definition 2.** *For a given a family of monotone $(p, q)$-DNF formulas, and a parameter $k$, the $k$-*Majority $(p, q)$-DNF Satisfaction *problem asks to determine whether there exists a $k$-weight assignment to the variables of the family that satisfy more than half of its DNFs.*

We now show that $k$-Multiple Majority DNF Satisfaction parameterically reduces to the $k$-Step Halting. Recall that the $k$-Step Halting problem, which is the parameterized analog

of the classical Halting Problem, asks for a given non-deterministic single tape Turing machine (defined over an unbounded alphabet) $M$, and a parameter $k$, whether $M$ has a computation path on the empty string which halts after at most $k$ steps. The main idea of our proof for showing $\widetilde{\mathrm{W}}[1] \subseteq \mathrm{W}[1]$ is encapsulated in the following reduction.

**Lemma 3.** *There is a parameterized reduction from* $k$-Majority $(p,q)$-DNF Satisfaction *to* $k$-Step Halting.

*Proof.* Let $(\mathcal{D}, k)$ be a given instance for $k$-Majority $(p,q)$-DNF Satisfaction, with $\mathcal{D} = (D_1, \ldots, D_m)$ a family of $(p,q)$-DNFs over variables $x_1, \ldots, x_n$, and $k$ the parameter. We construct a Turing machine $M$ that halts in at most $k' = f(k)$ steps iff there exists a $k$ weighted assignment to the variables $x_1, \ldots, x_n$ which satisfies more than half of the DNFs in $\mathcal{D}$. Then main idea is to encode in the state space of $M$ all relevant information needed for determining the number of DNFs satisfied by a given weight $k$ assignment. For ease of notation, we will assume no DNF contains a conjunct with identical literals.

Consider some DNF $D_i \in \mathcal{D}$. There are $p$ possible choices of selecting $q$-conjuncts of variables to assign a 1 to, so as $D_i$ would be satisfied. We say that a subset of $q$ variables $X \subset \{x_1, \ldots, x_n\}$ *hits* $D_i$ if it is one of these possible choices. Moreover, we define the *neighborhood* $D(X)$ of $X$ to be all DNFs in $\mathcal{D}$ that it hits. The information encoded in the state space of $M$ is as follows: For every $\ell$ subsets of variables $X_1, \ldots, X_\ell \subset \{x_1, \ldots, x_n\}$, $1 \leq \ell \leq p$ and $|X_1| = \cdots = |X_\ell| = q$, we encode the size of the intersection of their neighborhoods. That is, we encode $|X_1 \cap \cdots \cap X_\ell|$, the number of DNFs in $\mathcal{D}_i$ hit by each of the $\ell$ subsets of variables. Note that this requires $\mathcal{O}(n^{pq})$ state space, which is polynomial in $n$.

We next describe the computation of $M$ on the empty string in three different *phases*:
1. First, $M$ nondeterministically guesses a subset of $k$ variables.
2. Next, $M$ identifies all $q$-subsets of variables that are implicity selected by its $k$ variable guess in the previous step.
3. Finally, $M$ calculates the total number of DNFs hit by all the $q$-subsets identified in the previous step. If this number is more than half of the DNFs in $\mathcal{D}$, $M$ halts. Otherwise, $M$ enters an infinite loop.

Due to its construction, $M$ halts in at least one of its computation paths on the empty string iff there is a weight $k$ assignment that satisfies more than half of the DNFs in $\mathcal{D}$. To complete the proof, we argue that $M$ halts after $k' = f(k)$ steps. The first phase requires $k$ nondeterministic steps since we can encode each variable by a single letter in the alphabet of $M$. In the second step, $M$ identifies $\mathcal{X} = X_1, \ldots, X_{|\mathcal{X}|}$, the set of all $q$-subsets of variables implicity selected by its $k$ variable guess. Since $|\mathcal{X}| = \mathcal{O}(k^q)$, this phase requires $\mathcal{O}(k^q)$ steps. To compute the last phase, $M$ performs an exclusion\inclusion calculation using the information stored in its state space. Since any family of $q$-subsets of size greater than $p$ necessarily has an empty intersection, we have

$$\Big|\bigcup_{X_i \in \mathcal{X}} D(X_i)\Big| = \sum_{j=1}^{|\mathcal{X}|} (-1)^{j+1} \cdot \sum_{i_1 < \cdots < i_j} |D(X_{i_1}) \cap \cdots \cap D(X_{i_j})| = \sum_{j=1}^{p} (-1)^{j+1} \cdot \sum_{i_1 < \cdots < i_j} |D(X_{i_1}) \cap \cdots \cap D(X_{i_j})|,$$

and so the information stored in $M$ suffices for this computation. The last phase therefore requires $\mathcal{O}(k^{pq})$ steps. □

We now turn to deal with weft-1 majority circuits, and our $\widetilde{\mathrm{W}}[1]$-complete $k$-Weighted Weft-1 Majority Circuit Satisfiability problem. We show that a similar construction used in the lemma above can be applied for reducing $k$-Weighted Weft-1 Majority Circuit Satisfiability to $k$-Step Halting.

Let $C$ be a weft-1 majority circuit. First, we normalize $C$ so that it first layer consists only of in-degree $q$ $\vee$-gates, and its second layer consists only of in-degree $p$ $\wedge$-gates. This can be done as follows. Consider a large gate in $C$ and the portion of $C$ which is required to evaluate one of its inputs. Since this portion involves only small gates and has constant depth, it can be viewed as boolean function of constant size (and over a constant number of variables). Also, this function is necessarily monotone, since majority gates can only compute monotone functions. We can therefore analyze its entire truth table, and convert it into a DNF using a disjunction of all the lines in the truth table in the straightforward manner. Since the function is monotone, any variable appearing in negation in some conjunct of the DNF, also appears in positive form in another conjunct with similar literals, and so it can safely be removed from the DNF. From this it follows that any portion of $C$ which is required to evaluate one of the inputs of one of its large gates can be modeled by a monotone DNF of constant size. Letting $p$ and $q$ be the largest disjunction and conjunct in all these DNFs, by an appropriate padding of 1s (recall Observation 1) and duplicate conjuncts constants, we model each such portion by a $(p, q)$-DNF.

Let us say that a weft-1 majority circuit is *simple* if it has only one big majority gate as its output gate. If $C$ is simple then the construction above suffices, and we can immediately apply the same construction of $M$ used in Lemma 3 in our reduction. Otherwise, $C$ is logically equivalent to a constant size Boolean combination of simple sub-circuits. In this case, $M$ guesses $k$ variables, and computes the value of these constantly many simple sub-circuits as described in Lemma 3. It then computes in constant time the value of the Boolean combination under this assignment given by the computed values of the simple sub-circuits.

**Corollary 2.** $\widetilde{\mathrm{W}}[1] \subseteq \mathrm{W}[1]$.

Combining Corollaries 1 and 2, we complete our proof of Theorem 1.

## 5 Higher Levels of the Hierarchies

We now turn to consider higher levels of the W-and $\widetilde{\mathrm{W}}$-hierarchies. We prove in this section that $\mathrm{W}[t] \subseteq \widetilde{\mathrm{W}}[t]$ for all positive integers $t$. For this, we will first show that this statement holds for even values of $t$. Following this, we will consider odd values greater or equal to 3, and thus by combining 1 we will obtain our desired result.

Recall that a circuit is monotone if it does not have any $\neg$-gates, and that $k$-Weighted Weft-$t$ Monotone Circuit Satisfiability is complete for all even $t \geq 2$. To show that $\mathrm{W}[t] \subseteq \widetilde{\mathrm{W}}[t]$ for even values of $t$, we prove the following lemma.

**Lemma 4.** $k$-Weighted Weft-$t$ Monotone Circuit Satisfiability *parameterically reduces to* $k$-Weighted Weft-$t$ Majority Circuit Satisfiability.

*Proof.* Let $(C, k)$ be an instance of $k$-Weighted Weft-$t$ Monotone Circuit Satisfiability, with $C$ a weft-$t$ monotone logical circuit and $k$ the parameter, and let $\ell$ be the maximum in-degree of any gate in $C$. We assume w.l.o.g. that any small gate in $C$ has in-degree at most 2.

We construct a majority circuit $C'$ of weft $t$ as follows. First, we add $\ell \cdot (k+1) - 1$ new input gates to $C'$ labeled with new pairwise distinct variables, and additional new input gate labeled with the constant 1 (which we construct according to Observation 1). $C'$ simulates the gates of $C$ as follows. The simulation of a large $\vee$-gate, say of in-degree $\ell' \leq \ell$, is straightforward: relabel so it becomes a majority gate and add $\ell' - 1$ new edges coming from the input gate labeled 1. Small $\vee$-gates can be handled similarly. Small $\wedge$-gates are simply relabeled to become small majority gates. The

interesting case is what to do when $g$ is a large $\wedge$-gate. Suppose $g$ has edges coming from $g_1, \ldots, g_{\ell'}$ (for some $\ell' \leq \ell$). We relabel $g$ to a majority gate. Then for all $1 \leq i \leq \ell'$, we replace each edge from $g_i$ to $g$ by $k+1$ parallel edges. Additionally we wire $\ell'(k+1)-1$ many new inputs to $g$.

We show that for each gate $g$ in $C$ that an arbitrary weight $k$ assignment for the variables of $C'$ satisfies $g$ in $C'$ iff its restriction to the variables in $C$ satisfies $g$ in $C$. We proceed inductively on the layer $g$ lives in (in $C$ or $C'$). The rest being easy look at the case that $g$ is a large $\wedge$-gate. Then an assignment satisfying all $g_1, \ldots, g_{\ell'}$ clearly satisfies the majority gate $g$. Conversely if a weight $k$ assignment does not satisfy all $g_1, \ldots, g_{\ell'}$ then $g$ receives at most $(\ell'-1)(k+1)$ times a 1 from these gates plus possibly some from the new variables, but at most $k$. All together, $g$ receives no more than $(\ell'-1)(k+1)+k$ values 1 and this is less than half the in-degree of $g$ which is $2\ell'(k+1)-1$. In total we have that an assignment of weight $k$ to the variables of $C'$ satisfies $C'$ iff its restriction to the variables of $C$ satisfies $C$. It follows that $C$ has a satisfying assignment to its variables of weight $k$ iff $C'$ has a satisfying assignment to its variables of weight $k$. Why? To see necessity extend an assignment for the variables of $C$ by setting all new variables to 0 and use the above equivalence. For sufficiency a satisfying weight $k$ assignment for $C'$ restricts to one of weight $\leq k$ satisfying $C$ by the above equivalence; but observe that since $C$ is monotone it has a weight $k$ satisfying assignment iff it has a satisfying assignment of weight at most $k$. □

**Corollary 3.** $\mathrm{W}[t] \subseteq \widetilde{\mathrm{W}}[t]$ *for any positive even integer* $t$.

We next show that $\mathrm{W}[t] \subseteq \widetilde{\mathrm{W}}[t]$ for odd values of $t$. For this, due to Theorem 1, it is enough to consider odd $t \geq 3$. In this case, we can consider a restricted type of antimonotone circuits which we call normalized antimonotone circuits. A *normalized antimonotone circuit* is an antimonotone circuit with its first layer containing only $\neg$-gates, its second layer containing only large $\wedge$-gates, its third only large $\vee$-gates, and so on, alternating between layers of large $\wedge$-gates and layers of large $\vee$-gates. Also, each gate is required to have incoming edges only from gates in the previous layer. It is known that for odd $t \geq 3$, $k$-Weighted Weft-$t$ Circuit Satisfiability restricted to normalized antimonotone circuits is complete for $\mathrm{W}[t]$ [FG06].

**Lemma 5.** $k$-Weighted Weft-$t$ Antimonotone Circuit Satisfiability *restricted to normalized antimonotone circuits parameterically reduces to* $k$-Weighted Weft-$t$ Majority Circuit Satisfiability.

*Proof.* Let $(C, k)$ be an instance of $k$-Weighted Weft-$t$ Antimonotone Circuit Satisfiability restricted to normalized antimonotone circuits. We transform $C$ to a majority circuit as follows: consider a large $\wedge$-gate $g$ of the second layer, *i.e.* one wired to only negations of input gates, say $\neg x_1, \ldots, \neg x_\ell$. Relabel $g$ to become a majority gate and wire it to all inputs except those labeled $x_1, \ldots, x_\ell$, say this yields $\ell'$ edges. Additionally add $\ell' - 2k + 1$ many parallel edges coming from a new input labeled with 1 – this is the smallest number $\alpha$ such that $k+\alpha$ is bigger than half of $\ell'+\alpha$. This means that this gate is satisfied by a weight $k$ assignment iff this assignment chooses $k$ variables whose negations are not wired into $g$ in $C$, *i.e.* satisfies $g$ in $C$. Observe that if we replace all first and second layer gates in this manner, we end up with a monotone circuit containing logical as well as majority gates which is equivalent to the original circuit with respect to weight $k$ assignments. For all other gates, we proceed as in Lemma 4. □

**Corollary 4.** $\mathrm{W}[t] \subseteq \widetilde{\mathrm{W}}[t]$ *for any positive odd integer* $t \geq 3$.

Combining Corollaries 1, 3, and 4, we complete our proof of Theorem 2.

## 6 Discussion

In this paper we presented an alternative characterization of W[1], using majority circuits instead of logical circuits. We also showed that this characterization holds in one direction for higher levels of the hierarchy. This gives the first monotone characterization of W[1], and is perhaps a first step in establishing a monotone characterization of the entire W-hierarchy. We believe our results may prove useful in showing fixed-parameter intractability results for monotone problems, as well as for other types of problems. Furthermore, our results exemplify the naturality of the notion of *weft*, showing that it remains the parameter governing the combinatorial nondeterministic computing strength of circuits, no matter what (nontrivial) type of gates they have.

The major open problem left by this paper is showing the other direction of Theorem 2, namely that $\widetilde{\mathrm{W}}[t] \subseteq \mathrm{W}[t]$ for all positive integers $t$. This, along with the results in this paper will prove that $\mathrm{W}[t] = \widetilde{\mathrm{W}}[t]$, giving a completely monotone characterization of the W-hierarchy. We conjecture that this is in fact the case.

## References


[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[DF95] R. Downey and M. Fellows. Fixed parameter tractability and completeness II: Completeness for $W[1]$. *Theoretical Computer Science A* 141 (1995), 109–131.

[DFKHW94] R. Downey, M. Fellows, B. Kapron, M. Hallett and H.T. Wareham. The parameterized complexity of some problems in logic and linguistics. *Proceedings of LFCS*, Springer-Verlag, *Lecture Notes in Computer Science* 813 (1994), 89–100.

[DFS99] R. Downey, M. Fellows and U. Stege. Parameterized complexity: a framework for systematically confronting computational intractability. In: *Contemporary Trends in Discrete Mathematics* (R. Graham, J. Kratochvil, J. Nesetril and F. Roberts, eds.), Proceedings of the DIMACS-DIMATIA Workshop on the Future of Discrete Mathematics, Prague, 1997, *AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 49 (1999), 49–99.

[FG06] *Parameterized Complexity Theory*, J. Flum and M. Grohe, Springer-Verlag, 2006.

[FFHMR07] M. Fellows, J. Flum, D. Hermelin, M. Müller and F. Rosamond. Parameterized complexity via combinatorial circuits. Manuscript, 2007.

[GNW05] J. Guo, R. Nidermeier and S. Wernicke. Parameterized complexity of generalized vertex cover problems. In *Proc. 9th WADS*, Springer-Verlag, *Lecture Notes in Computer Science* (2005), 36–48.