Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)
Main Track

Symbolic Solutions for Symbolic Constraint Satisfaction Problems

Alexsander Andrade de Melo[1] , **Mateus de Oliveira Oliveira**[2]
[1]Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
[2]University of Bergen, Bergen, Norway
aamelo@cos.ufrj.br, mateus.oliveira@uib.no

Abstract

A fundamental drawback that arises when one is faced with the task of deterministically certifying solutions to computational problems in PSPACE is the fact that witnesses may have superpolynomial size, assuming that NP is not equal to PSPACE. Therefore, the complexity of such a deterministic verifier may already be super-polynomially lower-bounded by the size of a witness. In this work, we introduce a new symbolic framework to address this drawback. More precisely, we introduce a PSPACE-hard notion of symbolic constraint satisfaction problem where both instances and solutions for these instances are implicitly represented by ordered decision diagrams (i.e. read-once, oblivious, branching programs). Our main result states that given an ordered decision diagram D of length k and width w specifying a CSP instance, one can determine in time $f(w, w') \cdot k$ whether there is an ODD of width at most w' encoding a solution for this instance. Intuitively, while the parameter w quantifies the complexity of the instance, the parameter w' quantifies the complexity of a prospective solution. We show that CSPs of constant width can be used to formalize natural PSPACE hard problems, such as reachability of configurations for Turing machines working in nondeterministic linear space. For such problems, our main result immediately yields an algorithm that determines the existence of solutions of width w in time $g(w) \cdot n$, where $g : \mathbb{N} \to \mathbb{N}$ is a suitable computable function, and n is the size of the input.

1 Introduction

One of the main drawbacks when dealing with computational problems which lie in complexity classes beyond NP is the fact that witnesses for YES instances of these problems may not have polynomial size. Indeed, in the case of PSPACE-hard problems, YES instances cannot be certified in polynomial time, unless NP=PSPACE. One approach towards solving PSPACE complete problems is to use generalizations of propositional proof systems to the context of quantified formulas (Balabanov, Widl, and Jiang 2014; Narizzano et al. 2009). In this case, proofs of satisfiability/unsatisfiability play the role of certificates for YES/NO instances respectively. The drawback however, is that proofs may still have exponential size, and that the search for such proofs is non-deterministic.

In this work, we introduce a new symbolic framework for certifying YES instances of PSPACE-complete problems. More precisely, we introduce a new symbolic notion of constraint satisfaction problem (CSP) where both instances and assignments are encoded using the notion of ordered decision diagrams (ODDs), a straightforward generalization of the notion of ordered binary decision diagrams (OBDDs) to non-binary alphabets. Our main result (Theorem 1) states that for each $w, w' \in \mathbb{N}$, the process of determining whether a given CSP of width w has a solution of width w' can be solved in time $f(w, w') \cdot k$, where f is a suitable function depending only on w and on w' and k is the length of the input ODD. In the terminology of parameterized complexity theory (Downey and Fellows 1999; Cygan et al. 2015), our algorithm is fixed parameter tractable with respect to the parameters w and w'.

Our second main result (Theorem 11) states that the task of determining whether a CSP of constant width has a solution is already PSPACE-hard. The family of CSPs that we use to prove this hardness result encode natural grid-like CSPs with uniform horizontal and vertical nearest-neighbor constraints. Satisfiability for these CSPs generalize natural PSPACE-hard reachability problems, such as configuration reachability for cellular automata and for non-deterministic linear space Turing machines. Interestingly, for such CSPs of constant width, the complexity of satisfiability is directly correlated with the complexity of a prospective satisfying assignment. In particular, our main result immediately gives rise to an algorithm running in time $g(w) \cdot k$ that determines whether a CSP instance encoded by an ODD of length k has a satisfying assignment of width at most w, where $g : \mathbb{N} \to \mathbb{N}$ is a suitable function depending only on w.

It is worth highlighting some fundamental differences in the way in which ODDs are used in our work and in the way in which other notions of ordered decision diagrams have been used in previous works (Bollig 2012; Bollig 2014; Hachtel and Somenzi 1993; Woelfel 2006; Sawitzki 2004). The main difference is that in previous works each path in an ODD (or analogue formalisms) encodes a whole assignment for the variables of the input instance. Therefore, in these contexts, an ODD encodes a *set of solutions*. On the other hand, in our approach, a path in an ODD encodes the assignment of a *single variable*, and the whole ODD is used to encode a single assignment. As a consequence, in our approach we can encode assignments for CSP instances containing exponentially many variables.

This conceptual difference creates several challenges. For instance, even the process of characterizing whether a given ODD encodes a valid solution of a given CSP instance is a challenging task that required the development of new machinery for the manipulation of ODDs which may be of independent interest.

It is worth noting that our results cannot be obtained from techniques developed in other contexts which study CSPs from the perspective of structural graph theory such as the techniques developed in (Alekhnovich and Razborov 2002; Allender et al. 2014; Courcelle 1990; Courcelle, Makowsky, and Rotics 2000) to tackle CSPs of bounded treewidth or cliquewidth. Indeed, as we have shown in Section 6, CSPs whose constraint graphs are grids may have constant ODD width. On the other hand, grids have treewidth $\Omega(n)$ and clique-width $\Omega(n)$. Actually, as shown in (de Melo and de Oliveira Oliveira 2019), ODDs of constant width can be used to represent hypercubes of arbitrary size. Besides having unbounded treewidth and clique-width, (Chandran and Kavitha 2006; Bonomo et al. 2016), hypercubes are not nowhere dense, and therefore these graphs may have unbounded genus, unbounded local treewidth, unbounded expansion, etc. Therefore, techniques developed to study computational problems on classes of graphs in which these parameters are bounded (Grohe 2008; Kreutzer 2008; Grohe 2014) do not apply to our framework either.

2 Preliminaries

We denote by $\mathbb{N} \doteq \{0, 1, \ldots\}$ the set of natural numbers (including zero), and by $\mathbb{N}_+ \doteq \mathbb{N} \setminus \{0\}$ the set of positive natural numbers. For each $c \in \mathbb{N}_+$, we let $[c] \doteq \{1, 2, \ldots, c\}$ and $[\![c]\!] \doteq \{0, 1, \ldots, c-1\}$.

Given a set S and a number $\mathfrak{a} \in \mathbb{N}_+$, we let $S^{\times \mathfrak{a}}$ denote the set of all $\mathfrak{a}$-tuples of elements from S. An *alphabet* is any finite, non-empty set Σ. A *string* over an alphabet Σ is any finite sequence of symbols from Σ. We denote by Σ^+ the set of all (non-empty) strings over Σ. A *language* over Σ is any subset L of Σ^+. In particular, for each $k \in \mathbb{N}_+$, we let Σ^k be the language of all strings of length k over Σ. If $s_1, \ldots, s_{\mathfrak{a}}$ are strings in Σ^k, where $s_i = \sigma_{i,1} \ldots \sigma_{i,k}$ for each $i \in [\mathfrak{a}]$, then the *tensor product* of $s_1, \ldots, s_{\mathfrak{a}}$ is defined as the string

$$s_1 \otimes \cdots \otimes s_{\mathfrak{a}} \doteq (\sigma_{1,1}, \ldots, \sigma_{\mathfrak{a},1}) \cdots (\sigma_{k,1}, \ldots, \sigma_{k,\mathfrak{a}})$$

of length k over the alphabet $\Sigma^{\times \mathfrak{a}}$. The tensor product of strings of length k over the alphabet Σ can be extended to the tensor product of languages $L_1, \ldots, L_{\mathfrak{a}} \subseteq \Sigma^k$ by setting

$$L_1 \otimes \cdots \otimes L_{\mathfrak{a}} \doteq \{s_1 \otimes \cdots \otimes s_{\mathfrak{a}} \colon s_i \in L_i \text{ for each } i \in [\mathfrak{a}]\}.$$

Ordered Decision Diagrams. Let Σ be an alphabet and $w \in \mathbb{N}_+$. A (Σ, w)*-layer* is a tuple $B \doteq (\ell, r, T, I, F, \iota, \phi)$, where $\ell \subseteq [\![w]\!]$ is a set of *left states*, $r \subseteq [\![w]\!]$ is a set of *right states*, $T \subseteq \ell \times \Sigma \times r$ is a set of *transitions*, $I \subseteq \ell$ is a set of *initial states*, $F \subseteq r$ is a set of *final states* and $\iota, \phi \in \{0, 1\}$ are Boolean flags satisfying the two following conditions:

1. if $\iota = 0$ then $I = \emptyset$;
2. if $\phi = 0$ then $F = \emptyset$.

In what follows, we may write $\ell(B)$, $r(B)$, $T(B)$, $I(B)$, $F(B)$, $\iota(B)$ and $\phi(B)$ to refer to the sets ℓ, r, T, I and F and to the Boolean flags ι and ϕ, respectively.

We let $\mathcal{B}(\Sigma, w)$ denote the set of all (Σ, w)-layers. Note that, $\mathcal{B}(\Sigma, w)$ is non-empty and has at most $2^{\mathcal{O}(|\Sigma| \cdot w^2)}$ elements.

Let $k \in \mathbb{N}_+$. A (Σ, w)*-ordered decision diagram* (or simply, (Σ, w)-*ODD*) of *length* k is a string $D \doteq B_1 \cdots B_k \in \mathcal{B}(\Sigma, w)^k$ of length k over the alphabet $\mathcal{B}(\Sigma, w)$ satisfying the following conditions:

1. for each $i \in [k-1]$, $\ell(B_{i+1}) = r(B_i)$;
2. $\iota(B_1) = 1$ and, for each $i \in \{2, \ldots, k\}$, $\iota(B_i) = 0$;
3. $\phi(B_k) = 1$ and, for each $i \in [k-1]$, $\phi(B_i) = 0$.

We note that Condition 2 guarantees that only the first layer of an ODD is allowed to have initial states. Analogously, Condition 3 guarantees that only the last layer of an ODD is allowed to have final states.

For each $k \in \mathbb{N}_+$, we denote by $\mathcal{B}(\Sigma, w)^{\circ k}$ the set of all (Σ, w)-ODDs of length k.

The *width* of an ODD $D = B_1 \cdots B_k \in \mathcal{B}(\Sigma, w)^{\circ k}$ is defined as

$$\mathsf{w}(D) \doteq \max\{|\ell(B_1)|, \ldots, |\ell(B_k)|, |r(B_k)|\}.$$

We remark that $\mathsf{w}(D) \leq w$.

Let $D = B_1 \cdots B_k \in \mathcal{B}(\Sigma, w)^{\circ k}$ and $s = \sigma_1 \cdots \sigma_k \in \Sigma^k$. A *valid sequence* for s in D is a sequence of transitions $\langle (\mathfrak{p}_1, \sigma_1, \mathfrak{q}_1), \ldots, (\mathfrak{p}_k, \sigma_k, \mathfrak{q}_k) \rangle$ such that $\mathfrak{p}_{i+1} = \mathfrak{q}_i$ for each $i \in [k-1]$, and $(\mathfrak{p}_i, \sigma_i, \mathfrak{q}_i) \in T(B_i)$ for each $i \in [k]$. Such a valid sequence is called *accepting* for s if, additionally, $\mathfrak{p}_1 \in I(B_1)$ and $\mathfrak{q}_k \in F(B_k)$. We say that D *accepts* s if there exists an accepting sequence for s in D. The language of D, denoted by $\mathcal{L}(D)$, is defined as the set of all strings accepted by D, i.e. $\mathcal{L}(D) \doteq \{s \in \Sigma^k \colon s \text{ is accepted by } D\}$.

A (Σ, w)-layer B is called *deterministic* if the following conditions are satisfied:

1. for each $\mathfrak{p} \in \ell(B)$ and each $\sigma \in \Sigma$, there exists at most one right state $\mathfrak{q} \in r(B)$ such that $(\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B)$;
2. if $\iota(B) = 1$, then $I(B) = \ell(B)$ and $|\ell(B)| = 1$.

On the other hand, a (Σ, w)-layer B is called *complete* if, for each $\mathfrak{p} \in \ell(B)$ and each $\sigma \in \Sigma$, there exists at least one right state $\mathfrak{q} \in r(B)$ such that $(\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B)$. We let $\widehat{\mathcal{B}}(\Sigma, w)$ be the subset of $\mathcal{B}(\Sigma, w)$ comprising all deterministic, complete (Σ, w)-layers.

An ODD $D = B_1 \cdots B_k \in \mathcal{B}(\Sigma, w)^{\circ k}$ is called *deterministic* (*complete*, resp.) if, for each $i \in [k]$, B_i is a deterministic (complete, resp.) layer. We remark that, if D is deterministic, then there exists at most one valid sequence in D for each string in Σ^k. On the other hand, if D is complete, then there exists at least one valid sequence in D for each string in Σ^k. For each $k \in \mathbb{N}_+$, we let $\widehat{\mathcal{B}}(\Sigma, w)^{\circ k}$ be the subset of $\mathcal{B}(\Sigma, w)^{\circ k}$ comprising all deterministic, complete (Σ, w)-ODDs of length k.

3 A Symbolic Representation of Constraint Graphs

Let Σ be an alphabet, and k be a positive integer. A (Σ, k)-*syntactic constraint satisfaction problem*, or (Σ, k)-syntactic CSP for short, is a pair $\mathcal{C} = (G, \lambda)$ where G is a graph with vertex set $V(G) \subseteq \Sigma^k$, and λ is a function that maps each edge $(u, v) \in E(G)$ to a set $\lambda(u, v) \subseteq \Sigma^k \times \Sigma^k$. An *assignment* for $\mathcal{C}$ is a function $\alpha\colon V(G) \to \Sigma^k$ that maps each vertex $v \in V(G)$ to a string $\alpha(v) \in \Sigma^k$. We say that such an assignment α is a *solution for* $\mathcal{C}$ if for each edge $(u, v) \in E(G)$, the pair $(\alpha(u), \alpha(v))$ belongs to $\lambda(u, v)$.

Let $\mathcal{C} = (G, \lambda)$ be a (Σ, k)-CSP, for some alphabet Σ and some $k \in \mathbb{N}$. We represent $\mathcal{C}$ as a language $\mathbb{L}(\mathcal{C}) \subseteq (\Sigma^{\times 4})^k$, called the *language of* $\mathcal{C}$, which is defined as follows.

$$\mathbb{L}(\mathcal{C}) \doteq \{u \otimes v \otimes a \otimes b\colon (u, v) \in E(G),\ (a, b) \in \lambda(u, v)\}.$$

Similarly, we represent an assignment $\alpha\colon V(G) \to \Sigma^k$ for $\mathcal{C}$ as language $\mathbb{L}(\alpha) \subseteq (\Sigma^{\times 2})^k$, called the *language of* α, which is defined as follows.

$$\mathbb{L}(\alpha) \doteq \{v \otimes \alpha(v)\colon v \in V(G)\}.$$

The width of a (Σ, k)-syntactic CSP $\mathcal{C}$, denoted by $\mathsf{w}(\mathcal{C})$, is defined as the minimum width of an ODD over the alphabet $\Sigma^{\times 4}$ whose language is equal to $\mathbb{L}(\mathcal{C})$.

$$\mathsf{w}(\mathcal{C}) = \min\{w \in \mathbb{N}_+ : \exists D \in \mathcal{B}(\Sigma^{\times 4}, w)^{\circ k}, \mathcal{L}(D) = \mathbb{L}(\mathcal{C})\}.$$

Similarly, the width of an assignment α is defined as the mininum width of an ODD over the alphabet $\Sigma^{\times 2}$ whose language is equal to $\mathbb{L}(\alpha)$.

$$\mathsf{w}(\alpha) = \min\{w \in \mathbb{N}_+ : \exists D \in \mathcal{B}(\Sigma^{\times 2}, w)^{\circ k}, \mathcal{L}(D) = \mathbb{L}(\alpha)\}.$$

The main result of this work (Theorem 1) states that the satisfiability problem for syntactic CSPs can be solved in fixed-parameter linear-time when parameterized by the width of the input CSP and by the minimum width of a sought solution.

Theorem 1 (Main Theorem). *Let Σ be an alphabet. There is a computable function $f_\Sigma : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, such that for each $k, w, w' \in \mathbb{N}$, one can determine in time $f_\Sigma(w, w') \cdot k$ whether a given (Σ, k)-syntactic CSP $\mathcal{C}$ of width w has a solution of width w'.*

The proof of Theorem 1 is presented in Section 5. Before that, we introduce in Section 4 some machinery for operating with ODDs at a local level, layer by layer. These operations are primarily used in Section 5 so as to reduce the instances of the satisfiability problem of implicitly-represented CSPs to particular instances of the directed reachability problem in DAGs.

4 Local Operations with ODDs

Let Σ_1 and Σ_2 be two alphabets and $g\colon \Sigma_1 \to \Sigma_2$ be a mapping from Σ_1 to Σ_2. Such a mapping can be homomorphically extended to strings over Σ_1 by simply setting, for each $k \in \mathbb{N}_+$ and each string $s = \sigma_1\sigma_2 \cdots \sigma_k \in \Sigma_1^k$,

$$g(s) \doteq g(\sigma_1)g(\sigma_2) \cdots g(\sigma_k).$$

Furthermore, g can be also extended to languages $L \subseteq \Sigma_1^k$ by setting $g(L) \doteq \{g(s)\colon s \in L\}$.

In the other direction, if $L \subseteq \Sigma_2^k$, then we let

$$g^{-1}(L) \doteq \{s \in \Sigma_1^k\colon g(s) \in L\}$$

be the set of strings in Σ_1^k that are mapped by g to some string in L. We call $g^{-1}(L)$ the *inverse* of L under g.

Let $k, w, w' \in \mathbb{N}_+$, $D = B_1 \cdots B_k \in \mathcal{B}(\Sigma_1, w)^{\circ k}$ and $D' = B'_1 \cdots B'_k \in \mathcal{B}(\Sigma_2, w')^{\circ k}$. We let

$$D \otimes D' \doteq (B_1, B'_1) \cdots (B_k, B'_k)$$

be the string over $\mathcal{B}(\Sigma_1, w) \times \mathcal{B}(\Sigma_2, w)$ obtained by pairing layers of D with layers of D' position-wise.

In Lemma 2 below we list several statements concerning the manipulation of ordered decision diagrams. More precisely, Lemma 2 states that operations such as union, intersection, complementation, tensor products, mapping and inverse mapping can be realized on ODDs by manipulating their layers in in a position-wise way, in such a way that layers in distinct positions are manipulated independently from each other. We note that all statements in Lemma 2 follow from well known results from automata theory and from the theory of ordered-binary decision diagrams. For this reason, in we omit the proof of this lemma in this extended abstract. For completeness, the proof can be found in the full version of the paper.

Lemma 2 (Simulation Lemma). *Let Σ_1 and Σ_2 be two alphabets and $g\colon \Sigma_1 \to \Sigma_2$ be a mapping from Σ_1 to Σ_2. There exist mappings*

1. $\boldsymbol{f}_\cup\colon \mathcal{B}(\Sigma_1, w) \times \mathcal{B}(\Sigma_2, w') \to \mathcal{B}(\Sigma_1 \cup \Sigma_2, w + w')$;
2. $\boldsymbol{f}_\cap\colon \mathcal{B}(\Sigma_1, w) \times \mathcal{B}(\Sigma_2, w') \to \mathcal{B}(\Sigma_1 \cap \Sigma_2, w \cdot w')$;
3. $\boldsymbol{f}_\otimes\colon \mathcal{B}(\Sigma_1, w) \times \mathcal{B}(\Sigma_2, w') \to \mathcal{B}(\Sigma_1 \times \Sigma_2, w \cdot w')$;
4. $\boldsymbol{f}_g\colon \mathcal{B}(\Sigma_1, w) \to \mathcal{B}(\Sigma_2, w)$;
5. $\boldsymbol{f}_g^{-1}\colon \mathcal{B}(\Sigma_2, w') \to \mathcal{B}(\Sigma_1, w')$;
6. $\boldsymbol{f}_{pw}\colon \mathcal{B}(\Sigma_1, w) \to \widehat{\mathcal{B}}(\Sigma_1, 2^w)$;
7. $\widehat{\boldsymbol{f}}_\neg\colon \widehat{\mathcal{B}}(\Sigma_1, w) \to \widehat{\mathcal{B}}(\Sigma_1, w)$;

such that, for each $k \in \mathbb{N}_+$, each $D \in \mathcal{B}(\Sigma_1, w)^{\circ k}$ and each $D' \in \mathcal{B}(\Sigma_2, w')^{\circ k}$, the following statements hold:

1. *$\boldsymbol{f}_\cup(D, D')$ is an ODD in $\mathcal{B}(\Sigma_1 \cup \Sigma_2, w + w')^{\circ k}$ with*
$$\mathcal{L}(\boldsymbol{f}_\cup(D, D')) = \mathcal{L}(D) \cup \mathcal{L}(D').$$
2. *$\boldsymbol{f}_\cap(D \otimes D')$ is an ODD in $\mathcal{B}(\Sigma_1 \cap \Sigma_2, w \cdot w')^{\circ k}$ with*
$$\mathcal{L}(\boldsymbol{f}_\cap(D \otimes D')) = \mathcal{L}(D) \cap \mathcal{L}(D').$$
3. *$\boldsymbol{f}_\otimes(D \otimes D')$ is an ODD in $\mathcal{B}(\Sigma_1 \times \Sigma_2, w \cdot w')^{\circ k}$ with*
$$\mathcal{L}(\boldsymbol{f}_\otimes(D \otimes D')) = \mathcal{L}(D) \otimes \mathcal{L}(D').$$
4. *$\boldsymbol{f}_g(D)$ is an ODD in $\mathcal{B}(\Sigma_2, w)^{\circ k}$ with*
$$\mathcal{L}(\boldsymbol{f}_g(D)) = g(\mathcal{L}(D)).$$
5. *$\boldsymbol{f}_g^{-1}(D')$ is an ODD in $\mathcal{B}(\Sigma_1, w')^{\circ k}$ with*
$$\mathcal{L}(\boldsymbol{f}_g^{-1}(D')) = g^{-1}(\mathcal{L}(D')).$$

6. *$\boldsymbol{f}_{pw}(D)$ is a deterministic, complete ODD in $\widehat{\mathcal{B}}(\Sigma_1, 2^w)^{\circ k}$ with*
$$\mathcal{L}(\boldsymbol{f}_{pw}(D)) = \mathcal{L}(D).$$

7. *If D is a deterministic, complete ODD, then $\widehat{\boldsymbol{f}}_{\neg}(D)$ is a deterministic, complete ODD in $\widehat{\mathcal{B}}(\Sigma_1, w)^{\circ k}$ with*
$$\mathcal{L}(\widehat{\boldsymbol{f}}_{\neg}(D)) = \Sigma_1^k \setminus \mathcal{L}(D).$$

4.1 Sublayers

Let Σ be an alphabet and $w \in \mathbb{N}_+$. A layer $B' \in \mathcal{B}(\Sigma, w)$ is called a *sublayer* of a layer $B \in \mathcal{B}(\Sigma, w)$ if $\ell(B') \subseteq \ell(B)$, $r(B') \subseteq r(B)$, $T(B') \subseteq T(B)$, $I(B') \subseteq I(B)$, $F(B') \subseteq F(B)$, $\iota(B') = \iota(B)$ and $\phi(B') = \phi(B)$.

For each $k \in \mathbb{N}_+$, we say that an ODD $D' = B'_1 \cdots B'_k \in \mathcal{B}(\Sigma, w)^{\circ k}$ is a *sub-ODD* of an ODD $D = B_1 \cdots B_k \in \mathcal{B}(\Sigma, w)^{\circ k}$ if, for each $i \in [k]$, B'_i is a sublayer of B_i. We remark that if D' is a sub-ODD of D then $\mathcal{L}(D') \subseteq \mathcal{L}(D)$. The converse does not hold. More precisely, there may exist subsets of $\mathcal{L}(D)$ which are not equal to the language of any sub-ODD of D. We also note that if D is a deterministic ODD, then any sub-ODD of D is also deterministic.

Let B be a layer in $\mathcal{B}(\Sigma, w)$ and B' be a sublayer of B. We say that B' is a *reachable sublayer* of B if the following additional conditions are satisfied:

1. if $\iota(B) = 1$, then $\ell(B') = I(B)$;
2. if $\mathfrak{p} \in \ell(B')$ and there exist $\sigma \in \Sigma$ and $\mathfrak{q} \in r(B)$ such that $(\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B)$, then $\mathfrak{q} \in r(B')$ and $(\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B')$;
3. if $\mathfrak{q} \in r(B')$, then there exist $\sigma \in \Sigma$ and $\mathfrak{p} \in \ell(B')$ such that $(\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B')$;
4. $I(B') = I(B)$;
5. $F(B') = r(B') \cap F(B)$.

Let $k \in \mathbb{N}_+$, $D = B_1 \cdots B_k$ be an ODD in $\mathcal{B}(\Sigma, w)^{\circ k}$ and $D' = B'_1 \cdots B'_k$ be a sub-ODD of D. We say that D' is a *reachable sub-ODD* of D if B'_i is a reachable sublayer of B_i for each $i \in [k]$.

From the very definition of reachable sublayers, one can verify that each layer $B \in \mathcal{B}(\Sigma, w)$ has at least one reachable sublayer. Analogously, it is easy to see that for each $k \in \mathbb{N}_+$ and each ODD $D \in \mathcal{B}(\Sigma, w)^{\circ k}$, D has at least one reachable sub-ODD D'. Lemma 4 below states that this reachable sub-ODD is actually unique, and that it has the same language as the original one. A useful consequence of this fact, which we will explore in the proof of our main theorem, is that in order to determine whether the language of D is empty, we just need to determine whether the final state set of the last layer of D' is empty.

Let $B \in \mathcal{B}(\Sigma, w)$, $X \subseteq \ell(B)$ and $\Sigma' \subseteq \Sigma$. We let $\mathbf{N}(B, X, \Sigma')$ be the set of all right states of B that are reachable from some left state in X by reading some symbol in Σ'. More formally, $\mathbf{N}(B, X, \Sigma')$ is defined as the set $\{\mathfrak{q} \in r(B) \colon \exists\, \mathfrak{p} \in X, \exists\, \sigma \in \Sigma', (\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B)\}$.

Lemma 3. *Let Σ be an alphabet, $w \in \mathbb{N}_+$ and B be a layer in $\mathcal{B}(\Sigma, w)$. If $\iota(B) = 1$, then B has exactly one reachable sublayer. Otherwise, for each $X \subseteq \ell(B)$, B has exactly one reachable sublayer B' such that $\ell(B') = X$.*

Proof. Let B' be a reachable sublayer of B. By Condition 2, $r(B') \supseteq \mathbf{N}(B, \ell(B'), \Sigma)$. On the other hand, by Condition 3, $r(B') \subseteq \mathbf{N}(B, \ell(B'), \Sigma)$. As a result, we obtain that $r(B') = \mathbf{N}(B, \ell(B'), \Sigma)$. Moreover, one can verify that $T(B') = \{(\mathfrak{p}, \sigma, \mathfrak{q}) \in T(B) \colon \mathfrak{p} \in \ell(B'), \sigma \in \Sigma, \mathfrak{q} \in r(B')\}$. Finally, we have that $I(B') = I(B)$, $F(B') = r(B') \cap F(B)$, $\iota(B') = \iota(B)$ and $\phi(B') = \phi(B)$. Thus, if $\iota(B) = 1$, then $\ell(B') = I(B)$ by Condition 1, and consequently B' is uniquely determined from B. Otherwise, provided that $\ell(B') = X$ for some $X \subseteq \ell(B)$, B' is uniquely determined from B and the set X. □

Lemma 4. *Let Σ be an alphabet and $k, w \in \mathbb{N}_+$. Every ODD $D = B_1 \cdots B_k \in \mathcal{B}(\Sigma, w)^{\circ k}$ has exactly one reachable sub-ODD $D' = B'_1 \cdots B'_k$. Additionally, $\mathcal{L}(D') = \mathcal{L}(D)$, and $\mathcal{L}(D) \neq \emptyset$ if and only if $F(B'_k) \neq \emptyset$.*

Proof. Let $D' = B'_1 \cdots B'_k$ be a reachable sub-ODD of D. Since $\iota(B_1) = 1$, $\ell(B'_1) = I(B_1)$. Moreover, it follows from the fact that D' is an ODD that $\ell(B'_{i+1}) = r(B'_i)$ for each $i \in [k-1]$. Consequently, we obtain by Lemma 3 that, for each $i \in [k]$, the layer B'_i is uniquely determined. Therefore, $D' = B'_1 \cdots B'_k$ is the unique reachable sub-ODD of D.

Now, we prove that $\mathcal{L}(D') = \mathcal{L}(D)$. Since D' is a sub-ODD of D, $\mathcal{L}(D') \subseteq \mathcal{L}(D)$. Thus, it just remains to show that $\mathcal{L}(D') \supseteq \mathcal{L}(D)$. Let $s = \sigma_1 \cdots \sigma_k \in \mathcal{L}(D)$ and $\zeta = \langle (\mathfrak{p}_1, \sigma_1, \mathfrak{q}_1), \ldots, (\mathfrak{p}_k, \sigma_k, \mathfrak{q}_k) \rangle$ be an accepting sequence for s in D. By Condition 1, $\mathfrak{p}_1 \in I(B'_1)$. As a result, we have by Condition 2 that, for each $i \in [k]$, $\mathfrak{q}_i \in r(B'_i)$ and $(\mathfrak{p}_i, \sigma_i, \mathfrak{q}_i) \in T(B'_i)$. Finally, it follows from Condition 5 that $\mathfrak{q}_k \in F(B'_k)$. Therefore, ζ is also an accepting sequence for s in D', and consequently $\mathcal{L}(D') = \mathcal{L}(D)$.

Finally, we prove $\mathcal{L}(D) \neq \emptyset$ if and only if $F(B'_k) \neq \emptyset$. Note that, since $\mathcal{L}(D') = \mathcal{L}(D)$, $\mathcal{L}(D) \neq \emptyset$ implies $F(B'_k) \neq \emptyset$. Thus, it just remains to prove the converse. Suppose that $F(B'_k) \neq \emptyset$, and let $\mathfrak{q}_k \in F(B'_k)$. By Condition 3, there exit $\sigma_k \in \Sigma$ and $\mathfrak{p}_k \in \ell(B'_k)$ such that $(\mathfrak{p}_k, \sigma_k, \mathfrak{q}_k) \in T(B'_k)$. Since $r(B'_{k-1}) = \ell(B'_k)$, $\mathfrak{p}_k \in r(B'_{k-1})$. As a result, it follows from Condition 3 that there exit $\sigma_{k-1} \in \Sigma$ and $\mathfrak{p}_{k-1} \in \ell(B'_{k-1})$ such that $(\mathfrak{p}_{k-1}, \sigma_{k-1}, \mathfrak{q}_{k-1}) \in T(B'_{k-1})$, where $\mathfrak{q}_{k-1} = \mathfrak{p}_k$. More generally, one can verify that, for each $i \in [k]$, there exit $\sigma_i \in \Sigma$ and $\mathfrak{p}_i \in \ell(B'_i)$ such that $(\mathfrak{p}_i, \sigma_i, \mathfrak{q}_i) \in T(B'_i)$, where $\mathfrak{q}_j = \mathfrak{p}_{j+1}$ for each $j \in [k-1]$. Additionally, it follows from Condition 1 that, if $\mathfrak{p}_1 \in \ell(B'_1)$, then $\mathfrak{p}_1 \in I(B'_1)$. Therefore, there exists in D' an accepting sequence $\zeta = \langle (\mathfrak{p}_1, \sigma_1, \mathfrak{q}_1), \ldots, (\mathfrak{p}_k, s_k, \mathfrak{q}_k) \rangle$ for some string $s = \sigma_1 \cdots \sigma_k \in \Sigma^k$. Since $\mathcal{L}(D') = \mathcal{L}(D)$, ζ is also an accepting sequence for s in D, i.e. $s \in \mathcal{L}(D)$, which implies $\mathcal{L}(D) \neq \emptyset$.

□

5 Proof of Theorem 1

In this section, we prove the main result of this work, Theorem 1, which states that satisfiability of syntactic CSPs can be solved in fixed-parameter linear time when parameterized by the width of the input CSP and by the minimum width of a prospective solution.

CSPs and Assignments From Languages. Let Σ be an alphabet $k \in \mathbb{N}_+$, and $L \subseteq (\Sigma^{\times 4})^k$. We let $\mathbb{C}(L) \doteq (G_L, \lambda_L)$ be the (Σ, k)-syntactic CSP defined by setting

1. $V(G_L) \doteq \{s : \exists s', t, t' \in \Sigma^k, s \otimes s' \otimes t \otimes t' \in L \vee s' \otimes s \otimes t \otimes t' \in L\}$.
2. $E(G_L) \doteq \{(s, s') : \exists t, t' \in \Sigma^k,\ s \otimes s' \otimes t \otimes t' \in L\}$.
3. For each edge $(s, s') \in E(G_L)$, $\lambda_L(s, s') \doteq \{(t, t') : s \otimes s' \otimes t \otimes t' \in L\}$.

Now, let $S \subseteq (\Sigma^{\times 2})^k$. We say that S is a *functional language* if for each string $s \in \Sigma^k$, there exists at most one string $t \in \Sigma^k$ such that $s \otimes t \in S$. Such a functional language may be regarded as the encoding of a function $\alpha_S : \mathrm{Dom}(S) \to \mathrm{Im}(S)$ where

$$\mathrm{Dom}(S) \doteq \{s \in \Sigma^k : \exists t \in \Sigma^k,\ s \otimes t \in S\},$$

$$\mathrm{Im}(S) \doteq \{t \in \Sigma^k : \exists s \in \Sigma^k,\ s \otimes t \in S\},$$

and for each $s \in \mathrm{Dom}(S)$, $\alpha_S(s) = t$ if and only if $s \otimes t \in S$. Since for each $s \in \mathrm{Dom}(S)$ there is a unique t such that $s \otimes t \in S$, the function α_S is well defined.

Satisfiability for Languages. Let $\mathcal{E} \colon \Sigma^{\times 4} \to \Sigma^{\times 2}$ be the function that sends each 4-tuple $(\sigma, \sigma', \tau, \tau') \in \Sigma^{\times 4}$ to the pair $\mathcal{E}(\sigma, \sigma', \tau, \tau') \doteq (\sigma, \sigma')$. For each $k \in \mathbb{N}_+$, we extend this function to strings $s \otimes s' \otimes t \otimes t'$ in $(\Sigma^{\times 4})^k$, by setting $\mathcal{E}(s \otimes s' \otimes t \otimes t') \doteq s \otimes s'$. Going further, for each $k \in \mathbb{N}_+$, we extend $\mathcal{E}$ to languages $L \subseteq (\Sigma^{\times 4})^k$, by setting

$$\mathcal{E}(L) \doteq \{s \otimes s' \colon \exists t, t' \in \Sigma^k,\ s \otimes s \otimes t \otimes t' \in L\}.$$

Now, let $\mathsf{sw} \colon \Sigma^{\times 4} \to \Sigma^{\times 4}$ be the function that sends each 4-tuple $(\sigma, \tau, \sigma', \tau') \in \Sigma^{\times 4}$ to the 4-tuple $\mathsf{sw}(\sigma, \tau, \sigma', \tau') \doteq (\sigma, \sigma', \tau, \tau')$. In other words, the function sw swaps the second and third entries of each tuple in $\Sigma^{\times 4}$. For each $k \in \mathbb{N}_+$, we extend this function to strings $s \otimes t \otimes s' \otimes t' \in (\Sigma^{\times 4})^k$, by setting

$$\mathsf{sw}(s \otimes t \otimes s' \otimes t') \doteq s \otimes s' \otimes t \otimes t',$$

and to languages $L \subseteq (\Sigma^{\times 4})^k$ by setting

$$\mathsf{sw}(L) \doteq \{s \otimes s' \otimes t \otimes t' \colon s \otimes t \otimes s' \otimes t' \in L\}.$$

Definition 5. *Let $k \in \mathbb{N}_+$, $L \subseteq (\Sigma^{\times 4})^k$ and $S \subseteq (\Sigma^{\times 2})^k$. We say that S is a* solution *for L if the following conditions are satisfied:*

1. *S is a functional language;*
2. *$\mathcal{E}(L) \subseteq \mathcal{E}(\mathsf{sw}(S \otimes S))$;*
3. *$\mathsf{sw}(S \otimes S) \cap \mathcal{E}(L) \otimes \Sigma^k \otimes \Sigma^k \subseteq L$.*

Intuitively, Condition 1 requires that S is a legitimate representation of a function from Σ^k to Σ^k. Condition 2 requires that the function represented by S assigns a value to both endpoints of each edge of the constraint graph $\mathbb{C}(L)$. Indeed, note that $S \otimes S$ consists of all strings $(s, t, s', t') \in (\Sigma^{\times 4})^k$ such that both $s \otimes t$ and $s' \otimes t'$ belong to S. Consequently, $\mathcal{E}(\mathsf{sw}(S \otimes S))$ consists of all strings $s \otimes s' \in (\Sigma^{\times 2})^k$ such that both s and s' belong to $\mathrm{Dom}(S)$. Therefore, by requiring that $\mathcal{E}(L) \subseteq \mathcal{E}(S \otimes S)$ we ensure that, for each edge (s, s') in the edge set $E(G_L)$ of the graph G_L of the k-syntactic CSP $\mathbb{C}(L) = (G_L, \lambda_L)$, both vertices s and s' belong to the domain of the function α_S. Finally, Condition 3 guarantees that the solution represented by S satisfies all the constraints specified by L. In other words, if (s, s') is an edge of G_L, $\alpha_S(s) = t$ and $\alpha_S(s') = t'$, then the pair (t, t') belongs to the list $\lambda_L(s, s')$ of possible pairs of values associated to the edge (s, s'). That is to say, the 4-tuple (s, s', t, t') must belong to L. In view of the discussion above, we have the following lemma.

Lemma 6. *Let $k \in \mathbb{N}_+$, $L \subseteq (\Sigma^{\times 4})^k$ and $S \subseteq (\Sigma^{\times 2})^k$. Then S is a solution for L if and only if α_S is a solution for the k-syntactic CSP $\mathbb{C}(L)$.*

Therefore, given an ODD $D \in \mathcal{B}(\Sigma^{\times 4}, w)^{\circ k}$, representing a k-syntactic CSP $\mathbb{C}(\mathcal{L}(D))$, in order to determine whether $\mathbb{C}(\mathcal{L}(D))$ has a solution of width at most w', it is enough to search for an ODD $D' \in \mathcal{B}(\Sigma^{\times 2}, w)^{\circ k}$ such that the language $\mathcal{L}(D')$ is a solution for $\mathcal{L}(D)$. Using the tools for manipulation of ODDs developed in Section 4, we will reduce this latter problem to the reachability problem in a suitably defined graph. Before doing so, we will prove some auxiliary results.

Lemma 7. *Let Σ be an alphabet, $k, w, w' \in \mathbb{N}_+$, $D = B_1 \dots B_k$ be an ODD in $\mathcal{B}(\Sigma, w)^{\circ k}$ and $D' = B'_1 \dots B'_k$ be a deterministic, complete ODD in $\widehat{\mathcal{B}}(\Sigma, w')^{\circ k}$. Then $\mathcal{L}(D) \subseteq \mathcal{L}(D')$ if and only if there exists an ODD $D'' = B''_1 \dots B''_k$ in $\mathcal{B}(\Sigma, w \cdot w')^{\circ k}$ such that*

1. *for each $i \in [k]$, B''_i is a reachable sub-layer of $\boldsymbol{f}_\cap(B_i, \widehat{\boldsymbol{f}}_\neg(B'_i))$, and*
2. *$F(B''_k) = \emptyset$.*

Proof. Let $\widehat{\boldsymbol{f}}_\neg(D') = \widehat{\boldsymbol{f}}_\neg(B'_1) \dots \widehat{\boldsymbol{f}}_\neg(B'_k)$. By Lemma 2.7, $\widehat{\boldsymbol{f}}_\neg(D')$ is a deterministic, complete ODD in $\widehat{\mathcal{B}}(\Sigma, w')^{\circ k}$ with language $\mathcal{L}(\widehat{\boldsymbol{f}}_\neg(D')) = \Sigma^k \setminus \mathcal{L}(D')$. Now, let $\boldsymbol{f}_\cap(D \otimes \widehat{\boldsymbol{f}}_\neg(D')) = \boldsymbol{f}_\cap(B_1, \widehat{\boldsymbol{f}}_\neg(B'_1)) \dots \boldsymbol{f}_\cap(B_k, \widehat{\boldsymbol{f}}_\neg(B'_k))$. Then, it follows from Lemma 2.2 that $\boldsymbol{f}_\cap(D \otimes \widehat{\boldsymbol{f}}_\neg(D'))$ is an ODD in $\mathcal{B}(\Sigma, w \cdot w')^{\circ k}$ with language $\mathcal{L}(\boldsymbol{f}_\cap(D \otimes \widehat{\boldsymbol{f}}_\neg(D'))) = \mathcal{L}(D) \cap \mathcal{L}(\widehat{\boldsymbol{f}}_\neg(D')) = \mathcal{L}(D) \cap (\Sigma^k \setminus \mathcal{L}(D'))$. Note that, $\mathcal{L}(D) \subseteq \mathcal{L}(D')$ if and only if $\mathcal{L}(D) \cap (\Sigma^k \setminus \mathcal{L}(D')) = \emptyset$. Thus, $\mathcal{L}(D) \subseteq \mathcal{L}(D')$ if and only if $\mathcal{L}(\boldsymbol{f}_\cap(D \otimes \widehat{\boldsymbol{f}}_\neg(D'))) = \emptyset$. Therefore, it follows from Lemma 4 that $\mathcal{L}(D) \subseteq \mathcal{L}(D')$ if and only if there exists an ODD $D'' = B''_1 \dots B''_k \in \mathcal{B}(\Sigma, w \cdot w')^{\circ k}$ such that for each $i \in [k]$, B''_i is a reachable sub-layer of $\boldsymbol{f}_\cap(B_i, \widehat{\boldsymbol{f}}_\neg(B'_i))$, and $F(B''_k) = \emptyset$. □

Let $w \in \mathbb{N}_+$ and B be a deterministic layer in $\mathcal{B}(\Sigma^{\times 2}, w)$. A *functional coloring* for B is a pair (ξ, δ) of symmetric[1] functions $\xi \colon \ell(B) \times \ell(B) \to \{0, 1\}$ and $\delta \colon r(B) \times r(B) \to \{0, 1\}$, such that the following conditions are satisfied.

1. If $\iota(B) = 1$ and $I(B) = \{\mathfrak{p}\}$, then $\xi(\mathfrak{p}, \mathfrak{p}) = 0$;
2. for each $\mathfrak{p} \in \ell(B)$ and each $\mathfrak{q}, \mathfrak{q}' \in r(B)$, if there exist symbols $\sigma, \tau, \tau' \in \Sigma$ such that $\tau \neq \tau'$ and $(\mathfrak{p}, (\sigma, \tau), \mathfrak{q})$

[1]By symmetric, we mean that $\xi(\mathfrak{p}, \mathfrak{p}') = \xi(\mathfrak{p}', \mathfrak{p})$ for each $\mathfrak{p} \in \ell(B)$ and $\delta(\mathfrak{q}, \mathfrak{q}') = \delta(\mathfrak{q}', \mathfrak{q})$ for each $\mathfrak{q} \in r(B)$.

and $(\mathfrak{p}, (\sigma, \tau'), \mathfrak{q}')$ are transitions in $T(B)$, then $\delta(\mathfrak{q}, \mathfrak{q}') = 1$;

3. for each $\mathfrak{p}, \mathfrak{p}' \in \ell(B)$, if $\xi(\mathfrak{p}, \mathfrak{p}') = 1$ and there exist symbols $\sigma, \tau, \tau' \in \Sigma$ and right states $\mathfrak{q}, \mathfrak{q}' \in r(B)$ such that $(\mathfrak{p}, (\sigma, \tau), \mathfrak{q})$ and $(\mathfrak{p}', (\sigma, \tau'), \mathfrak{q}')$ are transitions in $T(B)$, then $\delta(\mathfrak{q}, \mathfrak{q}') = 1$;
4. if $\phi(B) = 1$, then $\delta(\mathfrak{q}, \mathfrak{q}') = 0$ for each $\mathfrak{q}, \mathfrak{q}' \in F(B)$.

We note that in Condition 3, the symbols τ and τ' are not necessarily distinct. Moreover, it is worth noting that Condition 3 implies that for each $\mathfrak{p} \in \ell(B)$, if $\xi(\mathfrak{p}, \mathfrak{p}) = 1$ and there exist symbols $\sigma, \tau \in \Sigma$ and right state $\mathfrak{q} \in r(B)$ such that $(\mathfrak{p}, (\sigma, \tau), \mathfrak{q})$ is a transition in $T(B)$, then $\delta(\mathfrak{q}, \mathfrak{q}) = 1$. Indeed, this fact follows simply by restricting Condition 3 to the case in which $\mathfrak{p} = \mathfrak{p}'$, $\mathfrak{q} = \mathfrak{q}'$ and $\tau = \tau'$.

Next, we define the notion of functional coloring for deterministic ODDs. This notion will be used to provide a syntactic characterization of ODDs whose language is a functional language.

Definition 8. *Let $k, w \in \mathbb{N}_+$ and $D = B_1 \cdots B_k$ be a deterministic ODD in $\mathcal{B}(\Sigma^{\times 2}, w)^{\circ k}$. A* functional coloring *for D is a sequence $(\xi_1, \delta_1) \cdots (\xi_k, \delta_k)$ of pairs of functions satisfying the following conditions:*

1. *for each $i \in [k]$, (ξ_i, δ_i) is a functional coloring of B_i, and*
2. *for each $i \in [k-1]$, $\delta_i = \xi_{i+1}$.*

As mentioned above, functional colorings provide us with a syntactic way of determining whether the language accepted by an ODD is a functional language. Indeed, the next lemma states that deterministic ODDs accepting a functional language are precisely those deterministic ODDs whose reachable sub-ODD admits a functional coloring.

Lemma 9. *Let $k, w \in \mathbb{N}_+$, D be a deterministic ODD in $\mathcal{B}(\Sigma^{\times 2}, w)^{\circ k}$ and D' be the reachable sub-ODD of D. Then $\mathcal{L}(D)$ is a functional language if and only if D' admits a functional coloring.*

Proof. Let D be a deterministic ODD in $\mathcal{B}(\{0,1\}^{\times 2}, w)^{\circ k}$ and $D' = B_1' B_2' \ldots B_k'$ be the reachable sub-ODD of D. We prove that $\mathcal{L}(D)$ is a functional language if and only if its unique reachable sub-ODD D' admits a functional coloring.

First, assume that $\mathcal{L}(D)$ is a functional language. Let $(\xi_1, \delta_1) \ldots (\xi_k, \delta_k)$ be the unique sequence of pairs of symmetric functions where for each $i \in [k-1]$, $\delta_i = \xi_{i+1}$, and for each $i \in [k]$, $\xi_i \colon \ell(B_i') \times \ell(B_i') \to \{0,1\}$ and $\delta_i \colon r(B_i') \times r(B_i') \to \{0,1\}$ satisfy the following conditions.

1A. If $i = 1$ and $I(B_1') = \{\mathfrak{p}\}$, then $\xi(\mathfrak{p}, \mathfrak{p}) = 0$.

2A. For each $\mathfrak{p} \in \ell(B_i')$ and each $\mathfrak{q}, \mathfrak{q}' \in r(B_i')$, if there exist symbols $\sigma, \tau, \tau' \in \{0,1\}$ such that $\tau \neq \tau'$ and $(\mathfrak{p}, (\sigma, \tau), \mathfrak{q})$ and $(\mathfrak{p}, (\sigma, \tau'), \mathfrak{q}')$ are transitions in $T(B_i')$, then set $\delta_i(\mathfrak{q}, \mathfrak{q}') = 1$.

3A. For each $\mathfrak{p}, \mathfrak{p}' \in \ell(B_i')$, if $\xi_i(\mathfrak{p}, \mathfrak{p}') = 1$ and there exist symbols $\sigma, \tau, \tau' \in \{0,1\}$ and right states $\mathfrak{q}, \mathfrak{q}' \in r(B_i')$ such that $(\mathfrak{p}, (\sigma, \tau), \mathfrak{q})$ and $(\mathfrak{p}', (\sigma, \tau'), \mathfrak{q}')$ are transitions in $T(B_i')$, then set $\delta_i(\mathfrak{q}, \mathfrak{q}') = 1$.

4A. For each $\mathfrak{q}, \mathfrak{q}' \in r(B_i')$, if $\delta_i(\mathfrak{q}, \mathfrak{q}')$ has not been defined in any of the above cases, then set $\delta_i(\mathfrak{q}, \mathfrak{q}') = 0$.

Note that Conditions 1A-3A above are just restatements of Conditions 1-3 of the definition of functional coloring. We claim that the sequence $(\xi_1, \delta_1) \cdots (\xi_k, \delta_k)$ is a functional coloring of D'. For the sake of contradiction, assume that this is not the case. Then the only condition that can fail is Condition 4 of the definition of functional coloring. In other words, there exist final states $\hat{\mathfrak{q}}_k, \hat{\mathfrak{q}}_k' \in F(B_k')$ such that $\delta_k(\hat{\mathfrak{q}}_k, \hat{\mathfrak{q}}_k') = 1$. This implies that, for some $j \in [k]$, the following statements hold.

- There exist $\mathfrak{p}_j \in \ell(B_j')$ and $\sigma_j, \tau_j, \tau_j' \in \{0,1\}$, with $\tau_j \neq \tau_j'$, such that $(\mathfrak{p}_j, (\sigma_j, \tau_j), \mathfrak{q}_j)$ and $(\mathfrak{p}_j, (\sigma_j, \tau_j'), \mathfrak{q}_j')$ are transitions in $T(B_j')$, where $\mathfrak{q}_j, \mathfrak{q}_j' \in r(B_j')$.
- For each $i \in \{j+1, \ldots, k\}$, there exist transitions $(\mathfrak{p}_i, (\sigma_i, \tau_i), \mathfrak{q}_i)$ and $(\mathfrak{p}_i, (\sigma_i, \tau_i'), \mathfrak{q}_i')$ in $T(B_i')$ such that $\mathfrak{p}_i = \mathfrak{q}_{i-1}$, and $\mathfrak{q}_k = \hat{\mathfrak{q}}_k$ and $\mathfrak{q}_k' = \hat{\mathfrak{q}}_k'$.

Furthermore, since D' is the reachable sub-ODD of D, which is deterministic, there exists a sequence $\langle(\mathfrak{p}_1, (\sigma_1, \tau_1), \mathfrak{q}_1), \ldots, (\mathfrak{p}_{j-1}, (\sigma_{j-1}, \tau_{j-1}), \mathfrak{q}_{j-1})\rangle$ of transitions in D such that $\mathfrak{p}_1 \in I(B_1')$ and, for each $i \in [j-1]$, $\mathfrak{q}_i = \mathfrak{p}_{i+1}$ and $(\mathfrak{p}_i, (\sigma_i, \tau_i), \mathfrak{q}_i) \in T(B_i')$, otherwise $\mathfrak{p}_j \notin \ell(B_j')$. As a result, we obtain that

$$\langle(\mathfrak{p}_1, (\sigma_1, \tau_1), \mathfrak{q}_1), \ldots, (\mathfrak{p}_j, (\sigma_j, \tau_j), \mathfrak{q}_j), \\ (\mathfrak{p}_{j+1}, (\sigma_{j+1}, \tau_{j+1}), \mathfrak{q}_{j+1}), \ldots, (\mathfrak{p}_k, (\sigma_k, \tau_k), \mathfrak{q}_k)\rangle$$

and

$$\langle(\mathfrak{p}_1, (\sigma_1, \tau_1), \mathfrak{q}_1), \ldots, (\mathfrak{p}_j, (\sigma_j, \tau_j'), \mathfrak{q}_j), \\ (\mathfrak{p}_{j+1}', (\sigma_{j+1}, \tau_{j+1}'), \mathfrak{q}_{j+1}'), \ldots, (\mathfrak{p}_k', (\sigma_k, \tau_k'), \mathfrak{q}_k')\rangle$$

are accepting sequences in D. Moreover, since $\tau_j \neq \tau_j'$, these sequences are necessarily distinct. Therefore, there exist strings $s, t, t' \in \{0,1\}^k$ such that $t \neq t'$ and $s \otimes t, s \otimes t' \in \mathcal{L}(D)$. Indeed, consider $s = \sigma_1 \cdots \sigma_k$, $t = \tau_1 \cdots \tau_k$ and $t' = \tau_1' \cdots \tau_k'$. However, this contradicts the assumption that $\mathcal{L}(D)$ is a functional language. As a consequence, we must conclude that $(\xi_1, \delta_1) \ldots (\xi_k, \delta_k)$ is a functional coloring of D'.

For the converse, let $(\xi_1, \delta_1) \cdots (\xi_k, \delta_k)$ be a functional coloring of D'. For the sake of contradiction, suppose that $\mathcal{L}(D) = \mathcal{L}(D')$ is not a functional language. Then, there exist strings $s, t, t' \in \Sigma^k$ such that $t \neq t$ and $s \otimes t, s \otimes t' \in \mathcal{L}(D)$. Assume that $s = \sigma_1 \cdots \sigma_k$, $t = \tau_1 \cdots \tau_k$ and $t' = \tau_1' \cdots \tau_k'$. Let j be the least integer in $[k]$ such that $\tau_j \neq \tau_j'$, and let $\langle(\mathfrak{p}_1, (\sigma_1, \tau_1), \mathfrak{q}_1), \ldots, (\mathfrak{p}_k, (\sigma_k, \tau_k), \mathfrak{q}_k)\rangle$ and $\langle(\mathfrak{p}_1', (\sigma_1, \tau_1'), \mathfrak{q}_1'), \ldots, (\mathfrak{p}_k', (\sigma_k, \tau_k'), \mathfrak{q}_k')\rangle$ be the accepting sequences for $s \otimes t$ and $s \otimes t'$ in D, respectively. Since D is deterministic, these sequences are well defined. Furthermore, these sequences are also accepting sequences for $s \otimes t$ and $s \otimes t'$ in D', respectively, which implies $\mathfrak{q}_k, \mathfrak{q}_k' \in F(B_k')$. Moreover, by the minimality of j, $\mathfrak{p}_i = \mathfrak{p}_i'$ for each $i \in [j]$. In particular, $\mathfrak{p}_j = \mathfrak{p}_j'$. Thus, since $\tau_j \neq \tau_j'$, it follows from Condition 2 of the definition of functional coloring for layers that $\delta_j(\mathfrak{q}_j, \mathfrak{q}_j') = 1$. Consequently, $\xi_{j+1}(\mathfrak{p}_{j+1}, \mathfrak{p}_{j+1}') = \delta_j(\mathfrak{q}_j, \mathfrak{q}_j') = 1$ and, by Condition 3, $\delta_{j+1}(\mathfrak{q}_{j+1}, \mathfrak{q}_{j+1}') = 1$.

More generally, one can verify that, for each $i \in \{j, \dots, k\}$, $\delta_i(\mathsf{q}_i, \mathsf{q}'_i) = 1$. Therefore, $\delta_k(\mathsf{q}_k, \mathsf{q}'_k) = 1$. However, this contradicts the assumption that $(\xi_1, \delta_1) \cdots (\xi_k, \delta_k)$ is a functional coloring of D'. Therefore, we must conclude that $\mathcal{L}(D)$ is a functional language. □

We let $\boldsymbol{f}_{\mathcal{E}}$ and $\boldsymbol{f}_{\mathsf{sw}}$ be instantiations of the operator $\boldsymbol{f}_g$ from Lemma 2 when g is set to the maps $\mathcal{E}\colon \Sigma^{\times 4} \to \Sigma^{\times 2}$ and $\mathsf{sw}\colon \Sigma^{\times 4} \to \Sigma^{\times 4}$, respectively. Analogously, we let $\boldsymbol{f}_{\mathcal{E}}^{-1}$ be the instantiation of the operator $\boldsymbol{f}_g^{-1}$ from Lemma 2 when g is set to $\mathcal{E}$. We note that, if $S \subseteq (\Sigma^{\times 2})^k$, then $\boldsymbol{f}_{\mathcal{E}}^{-1}(S) = S \otimes \Sigma^k \otimes \Sigma^k$.

For each $B \in \mathcal{B}(\Sigma^{\times 4}, w)$ and each $B' \in \mathcal{B}(\Sigma^{\times 2}, w')$, we let $X_0(B')$ and $X(B, B')$ be the layers over the alphabet $\Sigma^{\times 2}$ defined as follows:

$$X_0(B') = \boldsymbol{f}_{\mathcal{E}}(\boldsymbol{f}_{\mathsf{sw}}(\boldsymbol{f}_{\otimes}(B', B'))),$$

and

$$X(B, B') \doteq \boldsymbol{f}_{\cap}(\boldsymbol{f}_{\mathcal{E}}(B), \widehat{\boldsymbol{f}}_{\neg}(\boldsymbol{f}_{pw}(X_0(B')))).$$

We also let $Y_0(B, B')$ and $Y(B, B')$ be the layers over the alphabet $\Sigma^{\times 4}$ defined as follows:

$$Y_0(B, B') \doteq \boldsymbol{f}_{\cap}(\boldsymbol{f}_{\mathsf{sw}}(\boldsymbol{f}_{\otimes}(B', B')), \boldsymbol{f}_{\mathcal{E}}^{-1}(\boldsymbol{f}_{\mathcal{E}}(B)))$$

and

$$Y(B, B') \doteq \boldsymbol{f}_{\cap}(Y_0(B, B'), \boldsymbol{f}_{\neg}(\boldsymbol{f}_{pw}(B))).$$

We remark that, by Lemma 2, the width of the layer $X(B, B')$ is at most $w \cdot 2^{(w')^2}$ and the width of the layer $Y(B, B')$ is at most $w \cdot (w')^2 \cdot 2^w$.

Lemma 10. *Let $k, w, w' \in \mathbb{N}_+$, $D = B_1 \cdots B_k$ be an ODD in $\mathcal{B}(\Sigma^{\times 4}, w)^{\circ k}$ and $D' = B'_1 \cdots B'_k$ be an ODD in $\mathcal{B}(\Sigma^{\times 2}, w')^{\circ k}$. Then $\mathcal{L}(D')$ is a solution for $\mathcal{L}(D)$ if and only if the reachable sub-ODD of $\boldsymbol{f}_{pw}(D')$ admits a functional coloring, and there exist ODDs*

- *$D_1 = B_1^1 \cdots B_k^1 \in \mathcal{B}(\Sigma^{\times 2}, w \cdot 2^{(w')^2})^{\circ k}$ and*
- *$D_2 = B_1^2 \cdots B_k^2 \in \mathcal{B}(\Sigma^{\times 4}, w \cdot (w')^2 \cdot 2^w)^{\circ k}$,*

satisfying the following conditions:

1. *for each $i \in [k]$, B_i^1 is a reachable sublayer of $X(B_i, B'_i)$, and $F(B_k^1) = \emptyset$;*
2. *for each $i \in [k]$, B_i^2 is a reachable sublayer of $Y(B_i, B'_i)$, and $F(B_k^2) = \emptyset$.*

Proof. By Lemma 2, $\boldsymbol{f}_{pw}(D') = \boldsymbol{f}_{pw}(B'_1) \cdots \boldsymbol{f}_{pw}(B'_k)$ is a deterministic, complete ODD with language $\mathcal{L}(D') = \mathcal{L}(\boldsymbol{f}_{pw}(D'))$. Furthermore, it follows from Lemma 9 that $\mathcal{L}(D')$ is a functional language if and only if the reachable sub-ODD of $\boldsymbol{f}_{pw}(D')$ admits a functional coloring.

Now, by combining Lemma 2 with Lemma 7, we have that the inclusion

$$\mathcal{E}(\mathcal{L}(D)) \subseteq \mathcal{E}(\mathsf{sw}(\mathcal{L}(D') \otimes \mathcal{L}(D'))) \tag{1}$$

holds if and only if Condition 1 is satisfied.

Similarly, by combining Lemma 2 with Lemma 7, we have that the inclusion

$$\mathsf{sw}(\mathcal{L}(D') \otimes \mathcal{L}(D')) \cap \mathcal{E}(\mathcal{L}(D)) \otimes \Sigma^k \otimes \Sigma^k \subseteq \mathcal{L}(D), \tag{2}$$

holds if and only if Condition 2 is satisfied. Finally, by Definition 5, $\mathcal{L}(D')$ is a solution for $\mathcal{L}(D)$ if and only if $\mathcal{L}(D')$ is a functional language and the inclusions described in (1) and (2) are satisfied. □

Now, we are finally in a position to prove our main result (Theorem 1).

Proof of Theorem 1. Let $k, w, w' \in \mathbb{N}_+$ and $D \in \mathcal{B}(\Sigma^{\times 4}, w)^{\circ k}$ be an ODD of length k over the alphabet $\Sigma^{\times 4}$. We will show that one can determine in time $f_\Sigma(w, w') \cdot k$, for some computable function f_Σ, whether the k-syntactic CSP $\mathbb{C}(\mathcal{L}(D))$ defined by D has a solution of width at most w'. Additionally, our algorithm constructs such a solution in case it exists.

In order to prove our result, we will show how to reduce the problem of searching for a solution of width at most w' to the reachability problem on a suitably constructed graph. Let $\mathcal{U}$ be the directed graph defined as follows. The vertex set of $\mathcal{U}$ is constituted by all tuples of the form

$$\left[i, B'_i, B_i^1, B_i^2, \xi_i, \delta_i\right],$$

where

1. $B'_i \in \mathcal{B}(\Sigma^{\otimes 2}, w')$,
2. (ξ_i, δ_i) is a functional coloring of a reachable sub-layer of $\boldsymbol{f}_{pw}(B'_i)$,
3. B_i^1 is a reachable sub-layer of $X(B_i, B'_i)$ with $F(B_i^1) = \emptyset$,
4. B_i^2 is a reachable sub-layer of $Y(B_i, B'_i)$ with $F(B_i^2) = \emptyset$,
5. $\iota(B'_i) = \iota(B_i^1) = \iota(B_i^2) = 1$ if $i = 1$, and
6. $\phi(B'_i) = \phi(B_i^1) = \phi(B_i^2) = 1$ if $i = k$.

And, the edge set of $\mathcal{U}$ is constituted by all ordered pairs of vertices of the form

$$\left(\left[i, B'_i, B_i^1, B_i^2, \xi_i, \delta_i\right], \left[i{+}1, B'_{i+1}, B_{i+1}^1, B_{i+1}^2, \xi_{i+1}, \delta_{i+1}\right]\right)$$

such that $i \in [k-1]$, $r(B'_i) = \ell(B'_{i+1})$, $r(B_i^1) = \ell(B_{i+1}^1)$, $r(B_i^2) = \ell(B_{i+1}^2)$, and $\delta_i = \xi_{i+1}$. Then, one can verify that a sequence of vertices of the form

$$\left[1, B'_1, B_1^1, B_1^2, \xi_1, \delta_1\right], \dots, \left[k, B'_k, B_k^1, B_k^2, \xi_k, \delta_k\right] \tag{3}$$

is a path in $\mathcal{U}$ if and only if the following statements hold:

1. $D' = B'_1 \cdots B'_k$ is an ODD in $\mathcal{B}(\Sigma^{\times 2}, w')^{\circ k}$;
2. $(\xi_1, \delta_1) \cdots (\xi_k, \delta_k)$ is a functional coloring of the (unique) reachable sub-ODD of $\boldsymbol{f}_{pw}(D')$;
3. $D_1 = B_1^1 \dots B_k^1$ is an ODD in $\mathcal{B}(\Sigma^{\times 2}, w \cdot 2^{(w')^2})$ such that $F(B_k^1) = \emptyset$ and, for each $i \in [k]$, B_i^1 is a reachable sublayer of $X(B_i, B'_i)$;
4. $D_2 = B_1^2 \dots B_k^2$ is an ODD in $\mathcal{B}(\Sigma^{\times 4}, w \cdot (w')^2 \cdot 2^w)$ such that $F(B_k^2) = \emptyset$ and, for each $i \in [k]$, B_i^2 is a reachable sublayer of $Y(B_i, B'_i)$.

Therefore, we obtain by Lemma 10 that a path as described in (3) exists if and only if $\mathcal{L}(D')$ is a solution for $\mathcal{L}(D)$.

Now, we estimate the running time of our algorithm. First, we note that there are at most $2^{\mathcal{O}(|\Sigma|\cdot(w')^2)}$ distinct possible configurations for the layer B'. Furthermore, since the width of B' is upped bounded by w', there are at most $2^{\mathcal{O}(w')}$ distinct reachable sub-layers of $\boldsymbol{f}_{pw}(B')$, and there are at most $2^{\mathcal{O}(2^{w'})}$ distinct functional colorings for each reachable sub-layer of $\boldsymbol{f}_{pw}(B')$. Since the width of $X(B,B')$ is at most $w\cdot 2^{(w')^2}$, we obtain that $X(B,B')$ has at most $2^{\mathcal{O}(w\cdot 2^{(w')^2})}$ distinct reachable sub-layers. Analogously, it follows from the fact that the width of $Y(B,B')$ is at most $w\cdot(w')^2\cdot 2^w$ that $Y(B,B')$ has at most $2^{\mathcal{O}(w\cdot(w')^2\cdot 2^w)}$ distinct reachable sub-layers. This implies that the number of vertices of $\mathcal{U}$ is upper bounded by $k\cdot 2^{\mathcal{O}(|\Sigma|\cdot w\cdot(w')^2\cdot 2^w + w2^{(w')^2})}$. Since the vertex set of $\mathcal{U}$ is split into levels, and each level has at most $2^{\mathcal{O}(|\Sigma|\cdot w\cdot(w')^2\cdot 2^w + w2^{(w')^2})}$ vertices, one can determine in time $k\cdot 2^{\mathcal{O}(|\Sigma|\cdot w\cdot(w')^2\cdot 2^w + w2^{(w')^2})}$ if there exists a path from a vertex in the first level to a vertex in the last level of $\mathcal{U}$. In case such path exists, we can construct it in the same amount of time. Finally, in this case, the sought ODD $D' = B'_1\cdots B'_k$ is uniquely determined and can be read right away from the constructed path. □

6 Satisfiability of Constant Width CSPs is PSPACE-Hard

In this section, we show that satisfiability of symbolic CSPs is already PSPACE-hard for CSPs of constant width. For such CSPs the complexity of satisfiability is directly correlated with the complexity of a witness. Interestingly, for such CSPs, our main result immediately gives rise to an algorithm that determines the existence of solutions of with at most w in time $f(w)\cdot n$ where $f:\mathbb{N}\to\mathbb{N}$ is a suitable computable function and n is the size of the input CSP. More precisely, the following theorem is the main result of this section.

Theorem 11. *There exist constants $c,w\in\mathbb{N}$ such that the following problem is PSPACE-hard. The input consists of a $([\![c]\!],n)$-syntactic CSP $\mathcal{C}$ of width w for some $n\in\mathbb{N}$. The goal is to determine whether $\mathcal{C}$ is satisfiable.*

We dedicate the remainder of this section to the proof of Theorem 11. Let $c\in\mathbb{N}$, and let $\mathcal{V},\mathcal{H}\subseteq[\![c]\!]\times[\![c]\!]$ be binary relations over $[\![c]\!]$. An $m\times n$ $(c,\mathcal{V},\mathcal{H})$-matrix is a matrix $M\in[\![c]\!]^{m\times n}$ such that

1. $(M_{i,j},M_{i+1,j})\in\mathcal{V}$ for each $(i,j)\in[m-1]\times[n]$, and
2. $(M_{i,j},M_{i,j+1})\in\mathcal{H}$ for each $(i,j)\in[m]\times[n-1]$.

Intuitively, each pair of vertically consecutive entries of M belong to $\mathcal{V}$ and each pair of horizontally consecutive entries of M belong to $\mathcal{H}$. Now, consider the following reachability problem.

Problem Name: $(c,\mathcal{V},\mathcal{H})$-REACHABILITY
Input: A pair of strings (s,t) where $s,t\in[c]^k$.
Question: Is there some $(c,\mathcal{V},\mathcal{H})$-matrix M whose first row is equal to s and whose last row is equal to t?

It can be shown that there are fixed $c\in\mathbb{N}$, and relations $\mathcal{V},\mathcal{H}\in[\![c]\!]\times[\![c]\!]$ for which the $(c,\mathcal{V},\mathcal{H})$-reachability problem is PSPACE-complete. Indeed, the configuration reachability problems for 1-dimensional cellular automata, and for nondeterministic linear-space Turing machines, which are known to be PSPACE-complete problems under polynomial time reductions, can be reduced to $(c,\mathcal{V},\mathcal{H})$-REACHABILITY for suitable c, $\mathcal{V}$ and $\mathcal{H}$ which depend only on the cellular automaton or Turing machine in question, but not on the input configurations.

Proposition 12. *There exists some $c\in\mathbb{N}$ and some $\mathcal{V},\mathcal{H}\subseteq[\![c]\!]\times[\![c]\!]$ such that $(c,\mathcal{V},\mathcal{H})$-REACHABILITY is PSPACE-complete.*

Below, we will show how to reduce $(c,\mathcal{V},\mathcal{H})$-REACHABILITY to satisfiability of syntactic CSPs of constant width. This shows that the latter problem is PSPACE-hard.

Let $n\in\mathbb{N}$. For each $i\in[\![c^n]\!]$, we let $b^c_n(i)$ denote the base-c representation of the number i with n digits. For each $j\in[n]$, we let $u_n(j)=0^{j-1}10^{n-j}$ be the binary string that has 1 at the j-th position and 0 everywhere else. For each $i\in[\![c^n]\!]$ and each $j\in[c]$ we let $\rho[i,j]=b^c_n(i)\odot u_n(j)$ be the string representation of the pair (i,j).

Consider the $([2c],n)$-syntactic CSP $\mathcal{C}(c,\mathcal{V},\mathcal{H},s,t)=(G,\lambda)$ defined as follows. We set $V(G)=\{\rho[i,j] : i\in[\![c^n]\!], j\in[n]\}$. and

$$\begin{aligned}E(G) &= \{(\rho[i,j],\rho[i+1,j]) : i\in[\![c^n-1]\!],\ j\in[n]\}\ \cup\\ &\qquad \{(\rho[i,j],\rho[i,j+1]) : i\in[\![c^n]\!],\ j\in[n-1]\}.\end{aligned}$$

For each number $a\in[\![c]\!]$, we let $\tau(a)=a^n$ be the string of length n where all entries are equal to a. Below, we let

$$\tilde{\mathcal{V}}=\{(\tau(a),\tau(b)) : (a,b)\in\mathcal{V}\}.$$

For each $(i,j)\in[\![c^n-1]\!]\times[n]$, we let

$$\begin{aligned}\lambda(\rho[i,j],\rho[i+1,j]) = \tilde{\mathcal{V}}\cup\{(\tau(a),\tau(b+c)) : (a,b)\in\mathcal{V}\}\\ \cup\{(\tau(b+c),\tau(b+c)) : b\in[\![c]\!]\}.\end{aligned} \tag{4}$$

Below, we let $\tilde{\mathcal{H}}=\{(\tau(a),\tau(b)) : (a,b)\in\mathcal{H}\}$ and $\overline{\mathcal{H}}=\{(\tau(a+c),\tau(b+c)) : (a,b)\in\mathcal{H}\}$. For each $(i,j)\in[\![c^n]\!]\times[n-1]$, we let

$$\lambda(\rho[i,j],\rho[i,j+1]) =$$

$$\begin{cases}\{(\tau(s_j),\tau(s_{j+1}))\}\cap\tilde{\mathcal{H}} & \text{if } i=0,\\ \tilde{\mathcal{H}}\cup\{(\tau(t_j+c),\tau(t_{j+1}+c))\} & \text{if } 1\le i\le c^n-2,\\ \{(\tau(t_j+c),\tau(t_{j+1}+c))\}\cap\overline{\mathcal{H}} & \text{if } i=c^n-1.\end{cases} \tag{5}$$

Intuitively, the graph G of the CSP is (isomorphic to) a $c^n\times n$ grid whose vertices are indexed by numbers in $[\![c^n]\!]\times$

$[n]$. A solution for the CSP is a labeling of vertices of this graph with strings from the set $\{\tau(a) : a \in [\![2c]\!]\}$ in such a way that Conditions (4) and (6) are satisfied. For each $a \in [\![c]\!]$, the string $\tau(a+c)$ should be considered as a "marked" copy of the string $\tau(a)$. Condition 4 states that for each pair of vertically consecutive vertices (i,j) and $(i+1,j)$, if a solution for the CSP labels the vertex (i,j) with the string $\tau(a)$ for some $a \in [\![c]\!]$, then there is some $b \in [\![c]\!]$, such that $(\tau(a), \tau(b)) \in \mathcal{V}$ and $(i+1,j)$ is either labeled with $\tau(b)$ or with its marked version $\tau(b+c)$. On the other hand, if (i,j) is labeled with $\tau(b+c)$, for some $b \in [\![c]\!]$, then $(i+1,j)$ should also be labeled with $\tau(b+c)$. As a consequence, in a valid solution, if a vertex (i,j) is labeled with $\tau(b+c)$, then all vertices vertically below (i,j) are also labeled with $\tau(b+c)$. Now, Condition (6) guarantees that for each j in $[c]$, the label of vertex $(0,j)$ is equal to $\tau(s_j)$ and the label of vertex (c^n-1, j) is equal to $\tau(t_j + c)$. Additionally, for each $i \in \{1, \dots, c^n-2\}$, and each $j \in [n-1]$, either (i,j) is labeled with $\tau(a)$ and $(i, j+1)$ is labeled with $\tau(b)$, for some $(a,b) \in \mathcal{H}$, or (i,j) is labeled with $\tau(t_j + c)$ and $(i, j+1)$ is labeled with $\tau(t_{j+1}+c)$. This, in particular, guarantees that if a position (i,j) is labeled with $\tau(b+c)$ for some $b \in [\![c]\!]$, then for each $j' \in [n]$, (i,j') is labeled with $\tau(t_j)$.

From the above discussion, we have the following claim.

Claim 13. *Let $c \in \mathbb{N}$, and let $\mathcal{V}, \mathcal{H} \subseteq [\![c]\!] \times [\![c]\!]$, and $s, t \in [\![c]\!]^n$. Then (s,t) is an* YES *instance of $(c, \mathcal{V}, \mathcal{H})$-*REACHABILITY *if and only if the $(2c, n)$-syntactic CSP $\mathcal{C}(c, \mathcal{V}, \mathcal{H}, s, t)$ is satisfiable.*

In view of Proposition 12 and of Claim 13, proving Theorem 11 amounts to showing that for some constant w, the language there is an ODD $D(c, \mathcal{V}, \mathcal{H}, s, t) \in \mathcal{B}(\Sigma^{\times 4}, w)^{\circ n}$ accepting the language $L(\mathcal{C}(c, \mathcal{V}, \mathcal{H}, s, t))$. The construction of this ODD follows from the following sequence of statements, each of which can be proved using standard techniques for the manipulation of ODDs, together with Lemma 2.

Proposition 14. *Let $c \in \mathbb{N}$, and $\mathcal{V} \subseteq [\![c]\!] \times [\![c]\!]$.*

1. *There exists a constant $w_1 \in \mathbb{N}$ (depending on $\mathcal{V}$) such that for each $n \in \mathbb{N}$, there exists an ODD $D_1 \in \mathcal{B}([\![c]\!]^{\times 4}, w_1)^{\circ n}$ with*
$$\mathcal{L}(D_1) = \{\rho[i,j] \otimes \rho[i+1,j] \otimes \tau(a) \otimes \tau(b) : i \in [\![c^n - 1]\!], j \in [n], \text{ and } (a,b) \in \mathcal{V}\}.$$
2. *There exists a constant $w_2 \in \mathbb{N}$ (depending on $\mathcal{V}$) such that for each $n \in \mathbb{N}$, there exists an ODD $D_2 \in \mathcal{B}([\![c]\!]^{\times 4}, w_2)^{\circ n}$ with*
$$\mathcal{L}(D_2) = \{\rho[i,j] \otimes \rho[i+1,j] \otimes \tau(a) \otimes \tau(b+c) : i \in [\![c^n - 1]\!], j \in [n], \text{ and } (a,b) \in \mathcal{V}\}.$$
3. *There exists a constant w_3 (depending on c) such that for each $n \in \mathbb{N}$, there exists an ODD $D_3 \in \mathcal{B}([\![c]\!]^{\times 4}, w_2)^{\circ n}$ such that*
$$\mathcal{L}(D_3) = \{\rho[i,j] \otimes \rho[i+1,j] \otimes \tau(b+c) \otimes \tau(b+c) : i \in [\![c^n - 1]\!], j \in [n], \text{ and } b \in [\![c]\!]\}.$$

We note that the language $\mathcal{L}(D_1) \cup \mathcal{L}(D_2) \cup \mathcal{L}(D_3)$ encode all vertical constraints defined in Equation 4. The following proposition takes care of encodings of horizontal constraints.

Proposition 15. *Let $c \in \mathbb{N}$, and $\mathcal{H} \subseteq [\![c]\!] \times [\![c]\!]$.*

1. *There exists a constant $w_4 \in \mathbb{N}$ (depending on $\mathcal{H}$) such that for each $n \in \mathbb{N}$, there exists an ODD $D_4 \in \mathcal{B}([\![c]\!]^{\times 4}, w_4)^{\circ n}$ with*
$$\mathcal{L}(D_4) = \{\rho[i,j] \otimes \rho[i,j+1] \otimes \tau(a) \otimes \tau(b) : i \in [\![c^n - 2]\!], j \in [n-1], \text{ and } (a,b) \in \mathcal{H}\}.$$
2. *There exists a constant $w_5 \in \mathbb{N}$ (depending on $\mathcal{H}$) such that for each $n \in \mathbb{N}$, and each string $s \in [\![c]\!]^n$, there exists an ODD $D_5(s) \in \mathcal{B}([\![c]\!]^{\times 4}, w_5)^{\circ n}$ with*
$$\mathcal{L}(D_5(s)) = \{\rho[0,j] \otimes \rho[0,j+1] \otimes \tau(s_j) \otimes \tau(s_{j+1}) : j \in [n-1], \text{ and } (s_j, s_{j+1}) \in \mathcal{H}\}.$$
3. *There exists a constant $w_6 \in \mathbb{N}$ (depending on $\mathcal{H}$) such that for each $n \in \mathbb{N}$, and each string $t \in [\![c]\!]^n$, there exists an ODD $D_6(t) \in \mathcal{B}([\![c]\!]^{\times 4}, w_6)^{\circ n}$ with*
$$\mathcal{L}(D_6(t)) = \{\rho[i,j] \otimes \rho[i,j+1] \otimes \tau(t_j + c) \otimes \tau(t_{j+1} + c) : i \in [c^n], \ j \in [n-1], \text{ and } (t_j, t_{j+1}) \in \mathcal{H}\}.$$

Now, we have that
$$\mathcal{L}(D(c, \mathcal{V}, \mathcal{H}, s, t)) = \mathcal{L}(D_1(s)) \cup \mathcal{L}(D_2(s)) \cup \mathcal{L}(D_3(s)) \cup \mathcal{L}(D_4(s)) \cup \ \mathcal{L}(D_5(s)) \ \cup \ \mathcal{L}(D_6(t)).$$

Therefore, by Lemma 2, one can assume that the width of $D(c, \mathcal{V}, \mathcal{H}, s, t))$ is at most $w_1 + w_2 + w_3 + w_4 + w_5 + w_6$. This completes the proof of Theorem 11 □.

7 Conclusion

In this work, we have introduced a new symbolic approach for the study of constraint satisfaction problems. Intuitively, in our approach we use ordered decision diagrams to represent both CSP instances and prospective solutions for these instances. In our main result, we showed that the problem of determining whether a (Σ, k)-syntactic CSP of width w has a solution of width w' can be solved in time $f_\Sigma(w, w') \cdot k$. Additionally, we have shown that CSPs of constant width are PSPACE-hard. On the other hand, our main result indicates that the complexity of the satisfiability problem for such PSPACE-hard CSPs of constant width is directly correlated with the complexity of a prospective solution. In particular, in this case, our main result immediately implies the existence of an algorithm running in time $g_\Sigma(w') \cdot k$, where w' is the minimum width of a solution for the CSP in question.

Given that the satisfiability problem for CSPs specified by ODDs of constant width is PSPACE-hard, our main result can be used to provide a parameterized attack to each problem in PSPACE. More precisely, for each such problem X there is some constant c_X and a polynomial-time computable map M from instances of X to ODDs of width at most c_X such that for each instance I, we have that I is a YES instance of X if and only if the CSP encoded by the ODD $M(I)$ has a solution. Note that the length of the ODD $M(I)$ is at most $n^{O(1)}$, where n is the size of the input instance I. Therefore, using our main result, we have that the problem of determining whether $M(I)$ has a solution of width at most w can be solved in time $f_X(w) \cdot n^{O(1)}$ for a suitable computable function f_X.

Acknowledgements

Alexsander Andrade de Melo acknowledges support from the Brazilian agencies CNPq/GD 140399/2017-8 and CAPES/PDSE 88881.187636/2018-01.

Mateus de Oliveira Oliveira acknowledges support from the Bergen Research Foundation and from the Research Council of Norway (Grant Nr. 288761).

References

Alekhnovich, M., and Razborov, A. A. 2002. Satisfiability, branch-width and tseitin tautologies. In *Proc. of the 43rd Symposium on Foundations of Computer Science*, 593–603.

Allender, E.; Chen, S.; Lou, T.; Papakonstantinou, P. A.; and Tang, B. 2014. Width-parametrized SAT: Time–space tradeoffs. *Theory of Computing* 10(12):297–339.

Balabanov, V.; Widl, M.; and Jiang, J.-H. R. 2014. Qbf resolution systems and their proof complexities. In *International Conference on Theory and Applications of Satisfiability Testing*, 154–169. Springer.

Bollig, B. 2012. On symbolic obdd-based algorithms for the minimum spanning tree problem. *Theoretical Computer Science* 447:2–12.

Bollig, B. 2014. On the width of ordered binary decision diagrams. In *International Conference on Combinatorial Optimization and Applications*, 444–458. Springer.

Bonomo, F.; Grippo, L. N.; Milanič, M.; and Safe, M. D. 2016. Graph classes with and without powers of bounded clique-width. *Discrete Applied Mathematics* 199:3–15.

Chandran, L. S., and Kavitha, T. 2006. The treewidth and pathwidth of hypercubes. *Discrete Mathematics* 306(3):359–365.

Courcelle, B.; Makowsky, J. A.; and Rotics, U. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Th. of Comp. Syst.* 33(2):125–150.

Courcelle, B. 1990. The monadic second-order logic of graphs I. Recognizable sets of finite graphs. *Information and computation* 85(1):12–75.

Cygan, M.; Fomin, F. V.; Kowalik, Ł.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized algorithms*, volume 4. Springer.

de Melo, A. A., and de Oliveira Oliveira, M. 2019. On the width of regular classes of finite structures. In *Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings*, volume 11716 of *Lecture Notes in Computer Science*, 18–34. Springer.

Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Monographs in Computer Science. Springer.

Grohe, M. 2008. Algorithmic meta theorems. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, 30–30. Springer.

Grohe, M. 2014. Algorithmic meta theorems for sparse graph classes. In *International Computer Science Symposium in Russia*, 16–22. Springer.

Hachtel, G. D., and Somenzi, F. 1993. A symbolic algorithm for maximum flow in 0-1 networks. In *Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*, 403–406. IEEE Computer Society Press.

Kreutzer, S. 2008. Algorithmic meta-theorems. In *International Workshop on Parameterized and Exact Computation*, 10–12. Springer.

Narizzano, M.; Peschiera, C.; Pulina, L.; and Tacchella, A. 2009. Evaluating and certifying qbfs: A comparison of state-of-the-art tools. *AI Commun.* 22(4):191–210.

Sawitzki, D. 2004. Implicit flow maximization by iterative squaring. In *International Conference on Current Trends in Theory and Practice of Computer Science*, 301–313. Springer.

Woelfel, P. 2006. Symbolic topological sorting with obdds. *Journal of Discrete Algorithms* 4(1):51–71.