Upper and Lower Bounds on the Time Complexity of Infinite-domain CSPs

Peter Jonsson[1] and Victor Lagerkvist[1]

Department of Computer and Information Science, Linköping University, Sweden
{peter.jonsson, victor.lagerkvist}@liu.se

Abstract. The constraint satisfaction problem (CSP) is a widely studied problem with numerous applications in computer science. For infinite-domain CSPs, there are many results separating tractable and NP-hard cases while upper bounds on the time complexity of hard cases are virtually unexplored. Hence, we initiate a study of the worst-case time complexity of such CSPs. We analyse backtracking algorithms and show that they can be improved by exploiting sparsification. We present even faster algorithms based on enumerating finite structures. Last, we prove non-trivial lower bounds applicable to many interesting CSPs, under the assumption that the strong exponential-time hypothesis is true.

1 Introduction

The *constraint satisfaction problem* over a constraint language Γ (CSP(Γ)) is the problem of finding a variable assignment which satisfies a set of constraints, where each constraint is constructed from a relation in Γ. This problem is a widely studied computational problem and it can be used to model many classical problems such as k-colouring and the Boolean satisfiability problem. In the context of artificial intelligence, CSPs have been used for formalizing a wide range of problems, cf. Rossi et al. [30]. Efficient algorithms for CSP problems are hence of great practical interest. If the domain D is finite, then a CSP(Γ) instance I with variable set V can be solved in $O(|D|^{|V|} \cdot poly(||I||))$ time by enumerating all possible assignments. Hence, we have an obvious *upper bound* on the time complexity. This bound can, in many cases, be improved if additional information about Γ is known, cf. the survey by Woeginger [36] or the textbook by Gaspers [14]. There is also a growing body of literature concerning *lower bounds* [16, 20, 21, 33].

When it comes to CSPs over infinite domains, there is a large number of results that identify polynomial-time solvable cases, cf. Ligozat [23] or Rossi et al. [30]. However, almost nothing is known about the time complexity of solving NP-hard CSP problems. One may conjecture that a large number of practically relevant CSP problems do not fall into the tractable cases, and this motivates a closer study of the time complexity of hard problems. Thus, we initiate such a study in this paper. Throughout the paper, we measure time complexity in the number of variables. Historically, this has been the most common way of measuring time complexity. One reason is that an instance may be massively larger than the number of variables — a SAT instance with n variables may contain up to 2^{2n} distinct clauses if repeated literals are disallowed — and measuring in the instance size may give far too optimistic figures, especially since naturally appearing test examples tend to contain a moderate number of constraints. Another reason is that in the finite-domain case, the size of the search space is very closely related to the number of variables. We show that one can reason in a similar way when it comes to the complexity of many infinite-domain CSPs.

The relations in finite-domain CSPs are easy to represent by simply listing the allowed tuples. When considering infinite-domain CSPs, the relations need to be implicitly represented. A natural way is to consider disjunctive formulas over a finite set of basic relations. Let $\mathcal{B}$ denote some finite set of basic relations such that CSP($\mathcal{B}$) is tractable. Let $\mathcal{B}^{\vee\infty}$ denote the closure of $\mathcal{B}$ under disjunctions, and let $\mathcal{B}^{\vee k}$ be the subset of $\mathcal{B}^{\vee\infty}$ containing only disjunctions of length at most k. Consider the following example: let $D = \{true, false\}$ and let $\mathcal{B} = \{B_1, B_2\}$ where $B_1 = \{true\}$ and $B_2 = \{false\}$. It is easy to see that CSP($\mathcal{B}^{\vee\infty}$) corresponds to the Boolean SAT problem while CSP($\mathcal{B}^{\vee k}$) corresponds to the k-SAT problem.

CSPs in certain applications such as AI are often based on binary basic relations and unions of them (instead of free disjunctive formulas). Clearly, such relations are a subset of the relations in $\mathcal{B}^{\vee k}$ and we let $\mathcal{B}^{\vee =}$ denote this set of relations. We do not explicitly bound the length of disjunctions since they are bounded by $|\mathcal{B}|$. The literature on such CSPs is voluminous and we refer the reader to Renz and Nebel [29] for an introduction. The languages $\mathcal{B}^{\vee\infty}$ and $\mathcal{B}^{\vee k}$ have been studied to a smaller extent in the literature. There are both works studying disjunctive constraints from a general point of view [9, 11] and application-oriented studies; examples include temporal reasoning [19, 31], interactive graphics [27], rule-based reasoning [25], and set constraints (with applications in descriptive logics) [4]. We also note (see Section 2.2 for details) that there is a connection to constraint languages containing first-order definable relations. Assume Γ is a finite constraint language containing relations that are first-order definable in $\mathcal{B}$, and that the first order theory of $\mathcal{B}$ admits quantifier elimination. Then, upper bounds on CSP(Γ) can be inferred from results such as those that will be presented in Sections 3 and 4. This indicates that studying the time complexity of CSP($\mathcal{B}^{\vee\infty}$) is worthwhile, especially since our understanding of first-order definable constraint languages is rapidly increasing [3].

To solve infinite-domain CSPs, backtracking algorithms are usually employed. Unfortunately, such algorithms can be highly inefficient in the worst case. Let p denote the maximum arity of the relations in $\mathcal{B}$, let $m = |\mathcal{B}|$, and let $|V|$ denote the number of variables in a given CSP instance. We show (in Section 3.1) that the time complexity ranges from $O(2^{2^{m \cdot |V|^p} \cdot \log(m \cdot |V|^p)} \cdot poly(||I||))$ (which is doubly exponential with respect to the number of variables) for CSP($\mathcal{B}^{\vee\infty}$) to $O(2^{2^m \cdot |V|^p \cdot \log m} \cdot poly(||I||))$ time for $\mathcal{B}^{\vee =}$ (and the markedly better bound of $O(2^{|V|^p \log m} \cdot poly(||I||))$ if $\mathcal{B}$ consists of pairwise disjoint relations.) The use of heuristics can probably improve these figures in some cases, but we have not been able to find such results in the literature and it is not obvious how to analyse backtracking combined with heuristics. At this stage, we are mostly interested in obtaining a baseline: we need to know the performance of simple algorithms before we start studying more sophisticated ones. However, some of these bounds can be improved by combining backtracking search with methods for reducing the number of constraints. We demonstrate this with *sparsification* [18] in Section 3.2.

In Section 4 we switch strategy and show that disjunctive CSP problems can be solved significantly more efficiently via a method we call *structure enumeration*. This method is inspired by the enumerative method for solving finite-domain CSPs. With this algorithm, we obtain the upper bound $O(2^{|V|^p \cdot m} \cdot poly(||I||))$ for CSP($\mathcal{B}^{\vee\infty}$). If we additionally assume that $\mathcal{B}$ is jointly exhaustive and pairwise disjoint then the running time is improved further to $O(2^{|V|^p \cdot \log m} \cdot poly(||I||))$. This bound beats or equals every bound presented in Section 3. We then proceed to show even better bounds for certain choices of $\mathcal{B}$. In Section 4.2 we consider equality constraint languages over a countably infinite domain and show that such CSP problems are solvable in $O(|V|B_{|V|} \cdot poly(||I||)$ time, where $B_{|V|}$ is the $|V|$-th Bell number. In Section 4.3 we focus on three well-known temporal reasoning problems and obtain significantly improved running times.

We tackle the problem of determining lower bounds for CSP($\mathcal{B}^{\vee\infty}$) in Section 5, i.e. identifying functions f such that no algorithm for CSP($\mathcal{B}^{\vee\infty}$) has a better running time than $O(f(|V|))$. We accomplish this by relating CSP problems and certain complexity-theoretical conjectures, and obtain strong lower bounds for the majority of the problems considered in Section 4. As an example, we show that the temporal CSP($\{<,>,=\}^{\vee\infty}$) problem is solvable in time $O(2^{|V|\log|V|} \cdot poly(||I||))$ but, assuming a conjecture known as the *strong exponential time hypothesis* (SETH), not solvable in $O(c^{|V|})$ time for *any* $c > 1$. Hence, even though the algorithms we present are rather straightforward, there is, in many cases, very little room for improvement, unless the SETH fails.

2 Preliminaries

We begin by defining the constraint satisfaction problem and continue by discussing first-order definable relations.

2.1 Constraint Satisfaction

Definition 1. *Let Γ be a set of finitary relations over some set D of values. The constraint satisfaction problem over Γ (CSP(Γ)) is defined as follows:*

Instance: *A set V of variables and a set C of* constraints *of the form $R(v_1, \ldots, v_k)$, where k is the arity of R, $v_1, \ldots, v_k \in V$ and $R \in \Gamma$.*
Question: *Is there a function $f : V \to D$ such that $(f(v_1), \ldots, f(v_k)) \in R$ for every $R(v_1, \ldots, v_k) \in C$?*

The set Γ is referred to as the *constraint language*. Observe that we do not require Γ or D to be finite. Given an instance I of CSP(Γ) we write $||I||$

for the number of bits required to represent I. We now turn our attention to constraint languages based on disjunctions. Let D be a set of values and let $\mathcal{B} = \{B_1, \ldots, B_m\}$ denote a finite set of relations over D, i.e. $B_i \subseteq D^j$ for some $j \geq 1$. Let the set $\mathcal{B}^{\vee\infty}$ denote the set of relations defined by disjunctions over $\mathcal{B}$. That is, $\mathcal{B}^{\vee\infty}$ contains every relation $R(x_1, \ldots, x_p)$ such that $R(x_1, \ldots, x_p)$ if and only if $B_{i_1}(\mathbf{x}_1) \vee \cdots \vee B_{i_t}(\mathbf{x}_t)$ where $\mathbf{x}_1, \ldots, \mathbf{x}_t$ are sequences of variables from $\{x_1, \ldots, x_p\}$ such that the length of $\mathbf{x}_j$ equals the arity of B_{i_j}. We refer to $B_{i_1}(\mathbf{x}_1), \ldots, B_{i_t}(\mathbf{x}_t)$ as the *disjuncts* of R. We assume, without loss of generality, that a disjunct occurs at most once in a disjunction. We define $\mathcal{B}^{\vee k}$, $k \geq 1$, as the subset of $\mathcal{B}^{\vee\infty}$ where each relation is defined by a disjunction of length at most k. It is common, especially in qualitative temporal and spatial constraint reasoning, to study a restricted variant of $\mathcal{B}^{\vee k}$ when all relations in $\mathcal{B}$ has the same arity p. Define $\mathcal{B}^{\vee =}$ to contain every relation R such that $R(\mathbf{x})$ if and only if $B_{i_1}(\mathbf{x}) \vee \cdots \vee B_{i_t}(\mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_p)$. For examples of basic relations, we refer the reader to Sections 4.2 and 4.3.

We adopt a simple representation of relations in $\mathcal{B}^{\vee\infty}$: every relation R in $\mathcal{B}^{\vee\infty}$ is represented by its defining disjunctive formula. Note that two objects $R, R' \in \mathcal{B}^{\vee\infty}$ may denote the same relation. Hence, $\mathcal{B}^{\vee\infty}$ is not a constraint language in the sense of Definition 1. We avoid tedious technicalities by ignoring this issue and view constraint languages as multisets. Given an instance $I = (V, C)$ of CSP($\mathcal{B}^{\vee\infty}$) under this representation, we let $\mathrm{Disj}(I) = \{B_{i_1}(\mathbf{x}_1), \ldots, B_{i_t}(\mathbf{x}_t) \mid B_{i_1}(\mathbf{x}_1) \vee \cdots \vee B_{i_t}(\mathbf{x}_t) \in C\}$ denote the set of all disjuncts appearing in C.

We close this section by recapitulating some terminology. Let $\mathcal{B} = \{B_1, \ldots, B_m\}$ be a set of relations (over a domain D) such that all $B_1, \ldots, B_m$ have arity p. We say that $\mathcal{B}$ is *jointly exhaustive* (JE) if $\bigcup \mathcal{B} = D^p$ and that $\mathcal{B}$ is *pairwise disjoint* (PD) if $B_i \cap B_j = \emptyset$ whenever $i \neq j$. If $\mathcal{B}$ is both JE and PD we say that it is JEPD. Observe that if $B_1, \ldots, B_m$ have different arity then these properties are clearly not relevant since the intersection between two such relations is always empty. These assumptions are common in for example qualitative spatial and temporal reasoning, cf. [24]. Let Γ be an arbitrary set of relations with arity $p \geq 1$. We say that Γ is *closed under intersection* if $R_1 \cap R_2 \in \Gamma$ for all choices of $R_1, R_2 \in \Gamma$. Let R be an arbitrary binary relation. We define the *converse* $R^{\smile}$ of R such that $R^{\smile} = \{(y, x) \mid (x, y) \in R\}$. If Γ is a set of binary relations, then we say that Γ is *closed under converse* if $R^{\smile} \in \Gamma$ for all $R \in \Gamma$.

2.2 First-order Definable Relations

Languages of the form $\mathcal{B}^{\vee\infty}$ have a close connection with languages defined over first-order structures admitting quantifier elimination, i.e. every first-order definable relation can be defined by an equivalent formula without quantifiers. We have the following lemma.

Lemma 2. *Let Γ be a finite constraint language first-order definable over a relational structure $(D, R_1, \ldots, R_m)$ admitting quantifier elimination, where $R_1, \ldots, R_m$ are JEPD. Then there exists a k such that (1) CSP(Γ) is polynomial-time reducible to CSP($\{R_1, \ldots, R_m\}^{\vee k}$) and (2) if CSP($\{R_1, \ldots, R_m\}^{\vee k}$) is solvable in $O(f(|V|) \cdot poly(||I||))$ time then CSP(Γ) is solvable in $O(f(|V|) \cdot poly(||I||))$ time.*

Proof. Assume that every relation $R \in \Gamma$ is definable through a quantifier-free first-order formula ϕ_i over $R_1, \ldots, R_m$. Let ψ_i be ϕ_i rewritten in conjunctive normal form. We need to show that every disjunction in ψ_i can be expressed as a disjunction over $R_1, \ldots, R_m$. Clearly, if ψ_i only contains positive literals, then this is trivial. Hence, assume there is at least one negative literal. Since $R_1, \ldots, R_m$ are JEPD it is easy to see that for any negated relation in $\{R_1, \ldots, R_m\}$ there exists $\Gamma \subseteq \{R_1, \ldots, R_m\}$ such that the union of Γ equals the complemented relation. We can then reduce CSP(Γ) to CSP($\{R_1, \ldots, R_m\}^{\vee k}$) by replacing every constraint by its conjunctive normal formula over $R_1, \ldots, R_m$. This reduction can be done in polynomial time with respect to $||I||$ since each such definition can be stored in a table of fixed size. Moreover, since this reduction does not increase the number of variables, it follows that CSP(Γ) is solvable in $O(f(|V|) \cdot \mathit{poly}(||I||))$ time whenever CSP($\mathcal{B}^{\vee k}$) is solvable in $O(f(|V|) \cdot \mathit{poly}(||I||))$ time. □

As we will see in Section 4, this result is useful since we can use upper bounds for CSP($\mathcal{B}^{\vee k}$) to derive upper bounds for CSP(Γ), where Γ consists of first-order definable relations over $\mathcal{B}$. There is a large number of structures admitting quantifier elimination and interesting examples are presented in every standard textbook on model theory, cf. Hodges [15]. A selection of problems that are highly relevant for computer science and AI are discussed in Bodirsky [3].

3 Fundamental Algorithms

In this section we investigate the complexity of algorithms for CSP($\mathcal{B}^{\vee \infty}$) and CSP($\mathcal{B}^{\vee k}$) based on branching on the disjuncts in constraints (Section 3.1) and the sparsification method (Section 3.2.) Throughout this section we assume that $\mathcal{B}$ is a set of basic relations such that CSP($\mathcal{B}$) is in P.

3.1 Branching on Disjuncts

Let $\mathcal{B} = \{B_1, \ldots, B_m\}$ be a set of basic relations with maximum arity $p \geq 1$. It is easy to see that CSP($\mathcal{B}^{\vee \infty}$) is in NP. Assume we have an instance I of CSP($\mathcal{B}^{\vee \infty}$) with variable set V. Such an instance contains at most $2^{m \cdot |V|^p}$ distinct constraints. Each such constraint contains at most $m \cdot |V|^p$ disjuncts so the instance I can be solved in

$$O((m \cdot |V|^p)^{2^{m \cdot |V|^p}} \cdot \mathit{poly}(||I||)) = O(2^{2^{m \cdot |V|^p} \cdot \log(m \cdot |V|^p)} \cdot \mathit{poly}(||I||))$$

time by enumerating all possible choices of one disjunct out of every disjunctive constraint. The satisfiability of the resulting sets of constraints can be checked in polynomial time due to our initial assumptions. How does such an enumerative approach compare to a branching search algorithm? In the worst case, a branching algorithm without heuristic aid will go through all of these cases so the bound above is valid for such algorithms. Analyzing the time complexity of branching algorithms equipped with powerful heuristics is a very different (and presumably very difficult) problem.

Assume instead that we have an instance I of CSP($\mathcal{B}^{\vee k}$) with variable set V. There are at most $m \cdot |V|^p$ different disjuncts which leads to at most $\sum_{i=0}^{k}(m|V|^p)^i \leq k \cdot (m|V|^p)^k$ distinct constraints. We can thus solve instances with $|V|$ variables in $O(k^{k\cdot(m|V|^p)^k} \cdot poly(||I||)) = O(2^{k \cdot \log k \cdot (m|V|^p)^k} \cdot poly(||I||))$ time.

Finally, let I be an instance of CSP($\mathcal{B}^{\vee=}$) with variable set V. It is not hard to see that I contains at most $2^m \cdot |V|^p$ distinct constraints, where each constraint has length at most m. Non-deterministic guessing gives that instances of this kind can be solved in

$$O(m^{2^m \cdot |V|^p} \cdot poly(||I||)) = O(2^{2^m \cdot |V|^p \cdot \log m} \cdot poly(||I||))$$

time. This may appear to be surprisingly slow but this is mainly due to the fact that we have not imposed any additional restrictions on the set $\mathcal{B}$ of basic relations. Hence, assume that the relations in $\mathcal{B}$ are PD. Given two relations $R_1, R_2 \in \mathcal{B}^{\vee=}$, it is now clear that $R_1 \cap R_2$ is a relation in $\mathcal{B}^{\vee=}$, i.e. $\mathcal{B}^{\vee=}$ is closed under intersection. Let $I = (V, C)$ be an instance of CSP($\mathcal{B}^{\vee=}$). For any sequence of variables $(x_1, \ldots, x_p)$, we can assume that there is at most one constraint $R(x_1, \ldots, x_p)$ in C. This implies that we can solve CSP($\mathcal{B}^{\vee=}$) in $O(m^{|V|^p} \cdot poly(||I||)) = O(2^{|V|^p \log m} \cdot poly(||I||))$ time. Combining everything so far we obtain the following upper bounds.

Lemma 3. *Let $\mathcal{B}$ be a set of basic relations with maximum arity p and let $m = |\mathcal{B}|$. Then*

- *CSP($\mathcal{B}^{\vee\infty}$) is solvable in $O(2^{2^{m\cdot|V|^p}\cdot\log(m\cdot|V|^p)} \cdot poly(||I||))$ time,*
- *CSP($\mathcal{B}^{\vee k}$) is solvable in $O(2^{k\cdot\log k\cdot(m|V|^p)^k} \cdot poly(||I||))$ time,*
- *CSP($\mathcal{B}^{\vee=}$) is solvable in $O(2^{2^m\cdot|V|^p\cdot\log m} \cdot poly(||I||))$ time, and*
- *CSP($\mathcal{B}^{\vee=}$) is solvable in $O(2^{|V|^p \log m} \cdot poly(||I||))$ time if $\mathcal{B}$ is PD.*

A bit of fine-tuning is often needed when applying highly general results like Lemma 3 to concrete problems. For instance, Renz and Nebel [29] show that the RCC-8 problem can be solved in $O(c^{\frac{|V|^2}{2}})$ for some (unknown) $c > 1$. This problem can be viewed as CSP($\mathcal{B}^{\vee=}$) where $\mathcal{B}$ contains JEPD binary relations and $|\mathcal{B}| = 8$. Lemma 3 implies that CSP($\mathcal{B}^{\vee=}$) can be solved in $O(2^{3|V|^2})$ which is significantly slower if $c < 8^2$. However, it is well known that $\mathcal{B}$ is closed under converse. Let $I = (\{x_1, \ldots, x_n\}, C)$ be an instance of CSP($\mathcal{B}^{\vee=}$). Since $\mathcal{B}$ is closed under converse, we can always assume that if $R(x_i, x_j) \in C$, then $i \leq j$. Thus, we can solve CSP($\mathcal{B}^{\vee=}$) in $O(m^{\frac{|V|^2}{2}} \cdot poly(||I||)) = O(2^{\frac{|V|^2}{2}\log m} \cdot poly(||I||))$ time. This figure matches the bound by Renz and Nebel better when c is small.

3.2 Sparsification

The complexity of the algorithms proposed in Section 3 is dominated by the number of constraints. An idea for improving these running times is therefore to reduce the number of constraints within instances. One way of accomplishing this is by using *sparsification* [18]. Before presenting this method, we need a few additional definitions. An instance of the *k-Hitting Set problem* consists of a finite set U (the *universe*) and a collection $\mathcal{C} = \{S_1, \ldots, S_m\}$ where $S_i \subseteq U$ and $|S_i| \leq k$, $1 \leq i \leq m$. A *hitting set* for $\mathcal{C}$ is a set $C \subseteq U$ such that $C \cap S_i \neq \emptyset$ for each $S_i \in \mathcal{C}$. Let $\sigma(\mathcal{C})$ be the set of all hitting sets of $\mathcal{C}$. The k-Hitting Set problem is to find a minimal size hitting set. $\mathcal{T}$ is a *restriction* of $\mathcal{C}$ if for each $S \in \mathcal{C}$ there is a $T \in \mathcal{T}$ with $T \subseteq S$. If $\mathcal{T}$ is a restriction of $\mathcal{C}$, then $\sigma(\mathcal{T}) \subseteq \sigma(\mathcal{C})$. We then have the following result[1].

Theorem 4 (Impagliazzo et al. [18]). *For all $\varepsilon > 0$ and positive k, there is a constant K and an algorithm that, given an instance $\mathcal{C}$ of k-Hitting Set on a universe of size n, produces a list of $t \leq 2^{\varepsilon \cdot n}$ restrictions $\mathcal{T}_1, \ldots, \mathcal{T}_t$ of $\mathcal{C}$ so that $\sigma(\mathcal{C}) = \bigcup_{i=1}^{t} \sigma(\mathcal{T}_i)$ and so that for each $\mathcal{T}_i$, $|\mathcal{T}_i| \leq Kn$. Furthermore, the algorithm runs in time $poly(n) \cdot 2^{\varepsilon \cdot n}$.*

Lemma 5. *Let $\mathcal{B}$ be a set of basic relations with maximum arity p and let $m = |\mathcal{B}|$. Then CSP($\mathcal{B}^{\vee k}$) is solvable in $O(2^{(\varepsilon + K \log k) \cdot |V|^p \cdot m} \cdot poly(||I||))$ time for every $\varepsilon > 0$, where K is a constant depending only on ε and k.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathcal{B}^{\vee k}$). We can easily reduce CSP($\mathcal{B}^{\vee k}$) to k-Hitting set by letting $U = \text{Disj}(I)$ and $\mathcal{C}$ be the set corresponding to all disjunctions in C. Then choose some $\varepsilon > 0$ and let $\{\mathcal{T}_1, \ldots, \mathcal{T}_t\}$ be the resulting sparsification. Let $\{\mathcal{T}'_1, \ldots, \mathcal{T}'_t\}$ be the corresponding instances of CSP($\mathcal{B}^{\vee k}$). Each instance $\mathcal{T}'_i$ contains at most $K \cdot |U| \leq K \cdot |V|^p \cdot m$ distinct constraints, where K is a constant depending on ε and k, and can therefore be solved in time $O(poly(||I||) \cdot k^{K \cdot |V|^p \cdot m})$ by exhaustive search à la Section 3.1. Last, answer yes if and only if some $\mathcal{T}'_i$ is satisfiable. This gives a total running time of

$$poly(|V|^p \cdot m) \cdot 2^{\varepsilon \cdot |V|^p \cdot m} + 2^{\varepsilon \cdot |V|^p \cdot m} \cdot k^{K \cdot |V|^p \cdot m} \cdot poly(||I||) \in$$
$$O(2^{\varepsilon \cdot |V|^p \cdot m} \cdot 2^{K \cdot |V|^p \cdot m \cdot \log k} \cdot poly(||I||)) = O(2^{(\varepsilon + K \log k) \cdot |V|^p \cdot m} \cdot poly(||I||))$$

since $t \leq 2^{\varepsilon \cdot n}$. □

This procedure can be implemented using only polynomial space, just as the enumerative methods presented in Section 3.1. This follows from the fact that the restrictions $\mathcal{T}_1, \ldots, \mathcal{T}_t$ of $\mathcal{C}$ can be computed one after another with polynomial delay [10, Theorem 5.15]. Although this running time still might seem excessively slow observe that it is significantly more efficient than the $2^{k \cdot \log k \cdot (m|V|^p)^k}$ algorithm for CSP($\mathcal{B}^{\vee k}$) in Lemma 3.

4 Improved Upper Bounds

In this section, we show that it is possible to obtain markedly better upper bounds than the ones presented in Section 3. In Section 4.1 we first consider general

[1] We remark that Impagliazzo et al. [18] instead refer to the k-Hitting set problem as the k-Set cover problem.

algorithms for CSP($\mathcal{B}^{\vee\infty}$) based on structure enumeration, and in Sections 4.2 and 4.3, based on the same idea, we construct even better algorithms for equality constraint languages and temporal reasoning problems.

4.1 Structure Enumeration

We begin by presenting a general algorithm for CSP($\mathcal{B}^{\vee\infty}$) based on the idea of enumerating all variable assignments that are implicitly described in instances. As in the case of Section 3 we assume that $\mathcal{B}$ is a set of basic relations such that CSP($\mathcal{B}$) is solvable in $O(poly(||I||))$ time.

Theorem 6. *Let $\mathcal{B}$ be a set of basic relations with maximum arity p and let $m = |\mathcal{B}|$. Then CSP($\mathcal{B}^{\vee\infty}$) is solvable in $O(2^{m|V|^p} \cdot poly(||I||))$ time.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathcal{B}^{\vee\infty}$). Let $S = \text{Disj}(I)$ and note that $|S| \leq m|V|^p$. For each subset S_i of S first determine whether S_i is satisfiable. Due to the initial assumption this can be done in $O(poly(||I||))$ time since this set of disjuncts can be viewed as an instance of CSP($\mathcal{B}$). Next, check whether S_i satisfies I by, for each constraint in C, determine whether at least one disjunct is included in S_i. Each such step can determined in time $O(poly(||I||))$ time. The total time for this algorithm is therefore $O(2^{m|V|^p} \cdot poly(||I||))$. ☐

The advantage of this approach compared to the branching algorithm in Section 3 is that enumeration of variable assignments is much less sensitive to instances with a large number of constraints. We can speed up this result even further by making additional assumptions on the set $\mathcal{B}$. This allows us to enumerate smaller sets of constraints than in Theorem 6.

Theorem 7. *Let $\mathcal{B}$ be a set of basic relations with maximum arity p and let $m = |\mathcal{B}|$. Then*

1. *CSP($\mathcal{B}^{\vee\infty}$) solvable in $O(2^{|V|^p \cdot \log m} \cdot poly(||I||)))$ time if $\mathcal{B}$ is JEPD, and*
2. *CSP($\mathcal{B}^{\vee\infty}$) is solvable in $O(2^{|V|^p \cdot \log(m+1)} \cdot poly(||I||)))$ time if $\mathcal{B}$ is PD.*

Proof. First assume that $\mathcal{B}$ is JEPD and let $I = (V, C)$ be an instance of CSP($\mathcal{B}^{\vee\infty}$). Observe that every basic relation has the same arity p since $\mathcal{B}$ is JEPD. Let F be the set of functions from $|V|^p$ to $\mathcal{B}$. Clearly $|F| \leq 2^{|V|^p \log m}$. For every $f_i \in F$ let $S_{f_i} = \{B_j(\mathbf{x_j}) \mid \mathbf{x_j} \in V^p, f_i(\mathbf{x_j}) = B_j\}$. For a set S_{f_i} one can then determine in $O(poly(||I||))$ time whether it satisfies I by, for every constraint in C, check if at least one disjunct in every constraint is included in S_{f_i}. Hence, the algorithm is sound. To prove completeness, assume that g is a satisfying assignment of I and let S_g be the set of disjuncts in C which are true in this assignment. For every $B_i(\mathbf{x_i}) \in S_g$ define the function f as $f(\mathbf{x_i}) = B_i$. Since $\mathcal{B}$ is PD it cannot be the case that $f(\mathbf{x_i}) = B_i = B_j$ for some $B_j \in \mathcal{B}$ distinct from B_i. Next assume that there exists $\mathbf{x_i} \in V^p$ but no $B_i \in \mathcal{B}$ such that $B_i(\mathbf{x_i}) \in S_g$. Let $\mathcal{B} = \{B_1, \ldots, B_m\}$ and let $f_1, \ldots, f_m$ be functions agreeing with f for every value for which it is defined and such that $f_i(\mathbf{x_i}) = B_i$. Since $\mathcal{B}$ is JE it holds that f satisfies I if and only if some f_i satisfies I.

Next assume that $\mathcal{B}$ is PD but not JE. In this case we use the same construction but instead consider the set of functions F' from V^p to $\mathcal{B} \cup \{D^p\}$. There are $2^{|V|^p \cdot \log(m+1)})$ such functions, which gives the desired bound $O(2^{|V|^p \cdot \log(m+1)} \cdot \mathit{poly}(||I||))$. The reason for adding the additional element D^p to the domains of the functions is that if $f \in F'$, and if $f(\mathbf{x}) = D^p$ for some $\mathbf{x} \in V^p$, then this constraint does not enforce any particular values on $\mathbf{x}$. □

4.2 Equality Constraint Languages

Let $\mathcal{E} = \{=, \neq\}$ over some countably infinite domain D. The language $\mathcal{E}^{\vee\infty}$ is a particular case of an *equality constraint language* [5], i.e. sets of relations definable through first-order formulas over the structure $(D, =)$. Such languages are of fundamental interest in complexity classifications for infinite domain CSPs, since a classification of CSP problems based on first-order definable relations over some fixed structure, always includes the classification of equality constraint language CSPs. We show that the $O(2^{|V|^2} \cdot \mathit{poly}(||I||))$ time algorithm in Theorem 7 can be improved upon quite easily. But first we need some additional machinery. A *partition* of a set X with n elements is a pairwise disjoint set $\{X_1, \ldots, X_m\}$, $m \leq n$ such that $\bigcup_{i=1}^{m} X_i = X$. A set X with n elements has B_n partitions, where B_n is the n-th *Bell number*. Let $L(n) = \frac{0.792n}{\ln(n+1)}$. It is known that $B_n < L(n)^n$ [1] and that all partitions can be enumerated in $O(nB_n)$ time [13, 32].

Theorem 8. *CSP($\mathcal{E}^{\vee\infty}$) is solvable in $O(|V|2^{|V| \cdot \log L(|V|)} \cdot \mathit{poly}(||I||))$ time.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathcal{E}^{\vee\infty}$). For every partition $S_1 \cup \ldots \cup S_n$ of V we interpret the variables in S_i as being equal and having the value i, i.e. a constraint $(x = y)$ holds if and only if x and y belong to the same set and $(x \neq y)$ holds if and only if x and y belong to different sets. Then check in $\mathit{poly}(||I||)$ time if this partition satisfies I using the above interpretation. The complexity of this algorithm is therefore $O(|V|B_{|V|} \cdot \mathit{poly}(||I||)) \subseteq O(|V|L(|V|)^{|V|} \cdot \mathit{poly}(||I||)) = O(|V|2^{|V| \cdot \log L(|V|)} \cdot \mathit{poly}(||I||))$. □

Observe that this algorithm is much more efficient than the $O(2^{|V|^2} \cdot \mathit{poly}(||I||))$ algorithm in Theorem 7. It is well known that equality constraint languages admit quantifier elimination [5]. Hence, we can use Lemma 2 to extend Theorem 8 to cover arbitrary equality constraint languages.

Corollary 9. *Let Γ be a finite set of relations first-order definable over $(D, =)$. Then CSP(Γ) is solvable in $O(|V|2^{|V| \cdot \log L(|V|)} \cdot \mathit{poly}(||I||))$ time.*

4.3 Temporal Constraint Reasoning

Let $\mathcal{T} = \{<, >, =\}$ denote the JEPD order relations on $\mathbb{Q}$ and recall that CSP($\mathcal{T}$) is tractable [34]. Theorem 7 implies that CSP($\mathcal{T}^{\vee\infty}$) can be solved in $O(2^{|V|^2 \cdot \log 3} \cdot \mathit{poly}(||I||))$ time. We improve this as follows.

Theorem 10. *CSP($\mathcal{T}^{\vee\infty}$) is solvable in $O(2^{|V| \log |V|} \cdot \mathit{poly}(||I||))$ time.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathcal{T}^{\vee\infty}$). Assume $f : V \to \mathbb{Q}$ satisfies this instance. It is straightforward to see that there exists some $g : V \to \{1, \ldots, |V|\}$ which satisfies I, too. Hence, enumerate all $2^{|V| \log |V|}$ functions from V to $\{1, \ldots, |V|\}$ and answer yes if any of these satisfy the instance. □

It is well known that the first-order theory of $(\mathbb{Q}, <)$ admits quantifier elimination [6, 15]. Hence, we can exploit Lemma 2 to obtain the following corollary.

Corollary 11. *Let Γ be a finite temporal constraint language over $(\mathbb{Q}, <)$. If CSP(Γ) is NP-complete, then it is solvable in $O(2^{|V| \log |V|} \cdot poly(||I||))$ time.*

We can also obtain strong bounds for Allen's interval algebra, which is a well-known formalism for temporal reasoning. Here, one considers relations between intervals of the form $[x, y]$, where $x, y \in \mathbb{R}$ is the starting and ending point, respectively. Let *Allen* be the $2^{13} = 8192$ possible unions of the set of the thirteen relations in Table 1. For convenience we write constraints such as $(\mathsf{p} \vee \mathsf{m})(x, y)$ as $x\{\mathsf{p}, \mathsf{m}\}y$, using infix notation and omitting explicit disjunction signs. The problem CSP(*Allen*) is NP-complete and all tractable fragments have been identified [22].

Basic relation		Example	Endpoints
x precedes y	p	xxx	$x^e < y^s$
y preceded by x	p^{-1}	yyy	
x meets y	m	xxxx	$x^e = y^s$
y met-by x	m^{-1}	yyyy	
x overlaps y	o	xxxx	$x^s < y^s < x^e$,
y overl.-by x	o^{-1}	yyyy	$x^e < y^e$
x during y	d	xxx	$x^s > y^s$,
y includes x	d^{-1}	yyyyyyy	$x^e < y^e$
x starts y	s	xxx	$x^s = y^s$,
y started by x	s^{-1}	yyyyyyy	$x^e < y^e$
x finishes y	f	xxx	$x^e = y^e$,
y finished by x	$\mathsf{f}^{-}1$	yyyyyyy	$x^s > y^s$
x equals y	$\equiv$	xxxx	$x^s = y^s$,
		yyyy	$x^e = y^e$

Table 1. The thirteen basic relations in Allen's interval algebra. The endpoint relations $x^s < x^e$ and $y^s < y^e$ that are valid for all relations have been omitted.

Given an instance $I = (V, C)$ of CSP(*Allen*) we first create two fresh variables x_i^s and x_i^e for every $x \in V$, intended to represent the startpoint and endpoint of the interval x. Then observe that a constraint $x\{r_1, \ldots, r_m\}y \in C$, where each r_i is a basic relation, can be represented as a disjunction of temporal constraints over x^s, x^e, y^s and y^e by using the definitions of each basic relation in Table 1. Applying Theorem 10 to the resulting instance gives the following result.

Corollary 12. *CSP(Allen) is solvable in $O(2^{2|V|(1+\log |V|)} \cdot poly(||I||))$ time.*

Finally, we consider *branching time*. We define the following relations on the set of all points in the forest containing all oriented, finite trees where the in-degree of each node is at most one.

1. $x = y$ if and only if there is a path from x to y and a path from y to x,
2. $x < y$ if and only if and there is a path from x to y but no path from y to x,
3. $x > y$ if and only if there is a path from y to x but no path from x to y,
4. $x||y$ if and only if there is no path from x to y and no path from y to x,

These four basic relations are known as the *point algebra for branching time*. We let $\mathcal{P} = \{||, <, >, =\}$. The problem CSP($\mathcal{P}^{\vee\infty}$) is NP-complete and many tractable fragments have been identified [8].

Theorem 13. *CSP($\mathcal{P}^{\vee\infty}$) is solvable in $O(2^{|V|+2|V|\log|V|} \cdot poly(||I||))$ time.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathcal{P}^{\vee\infty}$). We use the following algorithm.

1. enumerate all directed forests over V where the in-degree of each node is at most one,
2. for every forest F, if at least one disjunct in every constraint in C is satisfied by F, answer yes,
3. answer no.

It is readily seen that this algorithm is sound and complete for CSP($\mathcal{P}^{\vee\infty}$). As for the time complexity, recall that the number of directed labelled trees with $|V|$ vertices is equal to $|V|^{|V|-2}$ by Cayley's formula. These can be efficiently enumerated by e.g. enumerating all *Prüfer sequences* [28] of length $|V| - 2$. To enumerate all forests instead of trees, we can enumerate all labelled trees with $|V| + 1$ vertices and only consider the trees where the extra vertex is connected to all other vertices. By removing this vertex we obtain a forest with $|V|$ vertices. Hence, there are at most $2^{|V|}|V|^{|V|-1}$ directed forests over V. The factor $2^{|V|}$ stems from the observation that each forest contains at most $|V|$ edges, where each edge has two possible directions. We then filter out the directed forests containing a tree where the in degree of any vertex is more than one. Last, for each forest, we enumerate all $|V|^{|V|}$ functions from V to the forest, and check in $poly(||I||)$ time whether it satisfies I. Put together this gives a complexity of $O(2^{|V|}|V|^{|V|-1}|V|^{|V|} \cdot poly(||I||)) \subseteq O(2^{|V|+2|V|\log|V|} \cdot poly(||I||))$. □

Branching time does not admit quantifier elimination [3, Section 4.2] so Lemma 2 is not applicable. However, there are closely connected constraint languages on trees that have this property. Examples include the *triple consistency problem* with important applications in bioinformatics [7].

5 Lower Bounds

The algorithms presented in Section 4 give new upper bounds for the complexity of CSP($\mathcal{B}^{\vee\infty}$). It is natural to also ask, given reasonable complexity theoretical assumptions, how much room there is for improvement. This section is divided into Section 5.1, where we obtain lower bounds for CSP($\mathcal{B}^{\vee\infty}$) and CSP($\mathcal{B}^{\vee k}$) for $\mathcal{B}$ that are JEPD, and in Section 5.2, where we obtain lower bounds for Allen's interval algebra.

5.1 Lower Bounds for JEPD Languages

One of the most well-known methods for obtaining lower bounds is to exploit the *exponential-time hypothesis* (ETH). The ETH states that there exists a $\delta > 0$ such that 3-SAT is not solvable in $O(2^{\delta|V|})$ time by any deterministic algorithm, i.e. it is not solvable in subexponential time [16]. If the ETH holds, then there is an increasing sequence $s_3, s_4, \ldots$ of reals such that k-SAT cannot be solved in time $2^{s_k|V|}$ but it can be solved in $2^{(s_k+\epsilon)|V|}$ time for arbitrary $\epsilon > 0$. The *strong exponential-time hypothesis* (SETH) is the conjecture that the limit of the sequence $s_3, s_4, \ldots$ equals 1, and, as a consequence, that SAT is not solvable in time $O(2^{\delta|V|})$ for any $\delta < 1$ [16]. These conjectures have in recent years successfully been used for proving lower bounds of many NP-complete problems [26].

Theorem 14. *Let $\mathcal{B} = \{R_1, R_2, \ldots, R_m\}$ be a JEPD set of nonempty basic relations. If the SETH holds then CSP($\mathcal{B}^{\vee\infty}$) cannot be solved in $O(2^{\delta|V|})$ time for any $\delta < 1$.*

Proof. If the SETH holds then SAT cannot be solved in $O(2^{\delta|V|})$ time for any $\delta < 1$. We provide a polynomial-time many-one reduction from SAT to CSP($\mathcal{B}^{\vee\infty}$) which only increases the number of variables by a constant — hence, if CSP($\mathcal{B}^{\vee\infty}$) is solvable in $O(2^{\delta|V|})$ time for some $\delta < 1$ then SAT is also solvable in $O(2^{\delta|V|})$ time, contradicting the original assumption.

Let $I = (V, C)$ be an instance of SAT, where V is a set of variables and C a set of clauses. First observe that since $m \geq 2$ and since $\mathcal{B}$ is JEPD, $\mathcal{B}$ must be defined over a domain with two or more elements. Also note that the requirement that $\mathcal{B}$ is JEPD implies that complement of $R_1(\mathbf{x})$ can be expressed as $R_2(\mathbf{x}) \vee \ldots \vee R_m(\mathbf{x})$. Now, let p denote the arity of the relations in $\mathcal{B}$. We introduce $p-1$ fresh variables $T_1, \ldots, T_{p-1}$ and then for every clause $(\ell_1 \vee \ldots \vee \ell_k) \in C$ create the constraint $(\phi_1(x_1, T_1, \ldots, T_{p-1}) \vee \ldots \vee \phi_k(x_k, T_1, \ldots, T_{p-1}))$, where $\phi_i(x_i, T_1, \ldots, T_{p-1}) = R_1(x_i, T_1, \ldots, T_{p-1})$ if $\ell_i = x_i$ and $\phi_i(x_i, T_1, \ldots, T_{p-1}) = R_2(x_i, T_1, \ldots, T_{p-1}) \vee \ldots \vee R_m(x_i, T_1, \ldots, T_{p-1})$ if $\ell_i = \neg x_i$. Hence, the resulting instance is satisfiable if and only if I is satisfiable. Since the reduction only introduces $p-1$ fresh variables it follows that SAT is solvable in time $O(2^{\delta(|V|+p-1)}) = O(2^{\delta|V|})$. □

Even though this theorem does not rule out the possibility that CSP($\mathcal{B}^{\vee k}$) can be solved significantly faster for some k it is easy to see that CSP($\mathcal{B}^{\vee k}$) cannot be solved in subexponential time for any $k \geq 3(|\mathcal{B}| - 1)$. First assume that the ETH holds. By following the proof of Theorem 14 we can reduce 3-SAT to CSP($\mathcal{B}^{\vee 3(|\mathcal{B}|-1)}$), which implies that CSP($\mathcal{B}^{\vee 3(|\mathcal{B}|-1)}$) cannot be solved in $2^{\delta n}$ time either. The bound $k = 3(|\mathcal{B}| - 1)$ might obviously feel a bit unsatisfactory and one might wonder if this can be improved. We can in fact make this much more precise by adding further restrictions to the set $\mathcal{B}$. As in the case of the equality constraint languages in Section 4.2 we let $=$ denote the equality relation on a given countably infinite domain.

Theorem 15. *Let $\mathcal{B} = \{=, R_1, \ldots, R_m\}$ be a set of binary PD, nonempty relations. If the ETH holds then CSP($\mathcal{B}^{\vee k}$) cannot be solved in $O(2^{s_k|V|})$ time.*

Proof. We prove this result by reducing k-SAT to CSP($\mathcal{B}^{\vee k}$) in such a way that we at most introduce one fresh variable. Let $I = (V, C)$ be an instance of k-SAT, where V is a set of variables and C a set of clauses. We know that $R_1 \subseteq \{(a, b) \mid a, b \in D \text{ and } a \neq b\}$ since $\mathcal{B}$ is PD. Introduce one fresh variable T. For every clause $(\ell_1 \vee \ldots \vee \ell_k) \in C$ create the constraint $(\phi_1 \vee \ldots \vee \phi_k)$, where $\phi_i := x_j = T$ if $\ell_i = x_j$ and $\phi_i = R_1(x_j, T)$ if $\ell_i = \neg x_j$. Let (V', C') be the resulting instance of CSP($\mathcal{B}^{\vee k}$). We show that (V', C') has a solution if and only if (V, C) has a solution.

Assume first that (V, C) has a solution $f : V \to \{0, 1\}$. Arbitrarily choose a tuple $(a, b) \in R_1$. We construct a solution $f' : V' \to \{a, b\}$ for (V', C'). Let $f'(T) = b$, and for all $v \in V$ let $f'(v) = b$ if $f(v) = 1$ and let $f'(v) = a$ if $f(v) = 0$. Arbitrarily choose a clause $(\ell_1 \vee \ldots \vee \ell_k) \in C$ and assume for instance that ℓ_1 evaluates to 1 under the solution f. If $\ell_1 = x_i$, then $f(x_i) = 1$ and the corresponding disjunct in the corresponding disjunctive constraint in C' is $x_i = T$. By definition, $(f'(x_i), f'(T)) = (b, b)$. If $\ell_1 = \neg x_i$, then $f(x_i) = 0$ and the corresponding disjunct in the corresponding disjunctive constraint in C' is $R_1(x_i, T)$. By definition, $(f'(x_i), f'(T)) = (a, b)$ and $(a, b) \in R_1$.

Assume instead that $f' : V' \to D$ is a solution to (V', C'), and that $f'(T) = c$. We construct a solution $f : V \to \{0, 1\}$ to (V, C) as follows. Arbitrarily choose a disjunctive constraint $(d_1 \vee \cdots \vee d_k) \in C'$ and let $(\ell_1 \vee \cdots \vee \ell_k)$ be the corresponding clause in C'. Assume that $\ell_1 = x_i$. If d_1 is true under f', then let $f(x_i) = 1$ and, otherwise, $f(x_i) = 0$. If $\ell_1 = \neg x_i$, then do the opposite: $f(x_i) = 0$ if d_1 is true and $f(x_i) = 1$ otherwise. If the function f is well-defined, then f is obviously a solution to (V, C). We need to prove that there is no variable that is simultaneously assigned 0 and 1. Assume this is the case. Then there is some variable x_i such that the constraints $x_i = T$ and $R_1(x_i, T)$ are simultaneously satisfied by f'. This is of course impossible due to the fact that R_1 contains no tuple of the form (a, a). □

If we in addition require that $\mathcal{B}$ is JE we obtain substantially better lower bounds for CSP($\mathcal{B}^{\vee\infty}$).

Theorem 16. *Let $\mathcal{B} = \{=, R_1, \ldots, R_m\}$ be a set of binary JEPD relations over a countably infinite domain. If the SETH holds then CSP($\mathcal{B}^{\vee\infty}$) cannot be solved in $O(c^{|V|})$ time for any $c > 1$.*

Proof. First observe that the binary inequality relation $\neq$ over D can be defined as $\bigcup_{i=1}^{m} R_i$ since $\mathcal{B}$ is JEPD. In the the proof we therefore use $\neq$ as an abbreviation for $\bigcup_{i=1}^{m} R_i$. Let $I = (V, C)$ be an instance of SAT with variables $V = \{x_1, \ldots, x_n\}$ and the set of clauses C. Let K be an integer such that $K > \log c$. Assume without loss of generality that n is a multiple of K. We will construct an instance of CSP($\mathcal{B}^{\vee\infty}$) with $\frac{n}{K} + 2^K = \frac{n}{K} + O(1)$ variables. First, introduce 2^K fresh variables $v_1, \ldots, v_{2^K}$ and make them different by imposing $\neq$ constraints. Second, introduce $\frac{n}{K}$ fresh variables $y_1, \ldots, y_{\frac{n}{K}}$, and for each $i \in \{1, \ldots, \frac{n}{K}\}$ impose the constraint $(y_i = v_1 \vee y_i = v_2 \vee \cdots \vee y_i = v_{2^k})$. Let $V_1, \ldots, V_{\frac{n}{K}}$ be a partition of V such that each $|V_i| = K$. We will represent each set V_i of Boolean variables by one y_i variable over D. To do this we will interpret each auxiliary variable z_i as a K-ary Boolean tuple. Let $h : \{v_1, \ldots, v_{2^K}\} \rightarrow \{0,1\}^K$ be an injective function which assigns a Boolean K-tuple for every variable v_i. Let g_+ be a function from $\{1, \ldots, K\}$ to subsets of $\{v_1, \ldots, v_{2^K}\}$ such that $v_i \in g(j)$ if and only if the j-th element in $h(v_i)$ is equal to 1. Define g_- in the analogous way. Observe that $|g_+(j)| = |g_-(j)| = 2^{K-1}$ for each $j \in \{1, \ldots, K\}$.

For the reduction, let $(\ell_{i_1} \vee \ldots \vee \ell_{i_{n'}})$, $\ell_{i_j} = x_{i_j}$ or $\ell_{i_j} = \neg x_{i_j}$, be a clause in C. We assume that $n' \leq n$ since the clause contains repeated literals otherwise. For each literal ℓ_{i_j} let $V_{i'} \subseteq V$ be the set of variables such that $x_{i_j} \in V_{i'}$. Each literal ℓ_{i_j} is then replaced by $\bigvee_{z \in g_+(i_j)} y_{i'} = z$ if $\ell_{i_j} = x_{i_j}$, and with $\bigvee_{z \in g_-(i_j)} y_{i'} = z$ if $\ell_{i_j} = \neg x_{i_j}$. This reduction can be done in polynomial time since a clause with n' literals is replaced by a disjunctive constraint with $n' 2^{K-1}$ disjuncts (since K is a constant depending only on c). It follows that SAT can be solved in

$$O(c^{\frac{n}{K}+O(1)} \cdot poly(||I||)) = O(2^{(\frac{n}{K}+O(1)) \cdot \log c} \cdot poly(||I||)) = O(2^{\delta \cdot n} \cdot poly(||I||))$$

for some $\delta < 1$, since $K > \log c$. Thus, the SETH does not hold. □

As an illustrative use of the theorem we see that the temporal problem CSP($\mathcal{T}^{\vee\infty}$) is solvable in $O(2^{|V| \log |V|} \cdot poly(||I||))$ time but not in $O(c^{|V|})$ time for any $c > 1$ if the SETH holds. Lower bounds can also be obtained for the branching time problem CSP($\mathcal{P}^{\vee\infty}$) since there is a trivial reduction from CSP($\mathcal{T}$)$^{\vee\infty}$ which does not increase the number of variables: simply add a constraint $(x < y \vee x > y \vee x = y)$ for every pair of variables in the instance. Similarly, the equality constraint satisfaction problem CSP($\mathcal{E}^{\vee\infty}$) is not solvable in $O(c^{|V|})$ time for any $c > 1$ either, unless the SETH fails. Hence, even though the algorithms in Sections 4.2 and 4.3 might appear to be quite simple, there is very little room for improvement.

5.2 Lower Bounds for Allen's Interval Algebra

Theorems 14, 15 and 16 gives lower bounds for all the problems considered in Sections 4.2 and 4.3 except for CSP(*Allen*) since unlimited use of disjunction is not allowed in this language. It is however possible to relate the complexity of CSP(*Allen*) to the Chromatic Number problem, i.e. the problem of computing the number of colours needed to colour a given graph.

Theorem 17. *If CSP(Allen) can be solved in $O(\sqrt{c}^{|V|})$ time for some $c < 2$, then* Chromatic Number *can be solved in $O((c+\epsilon)^{|V|})$ time for arbitrary $\epsilon > 0$.*

Proof. We first present a polynomial-time many-one reduction from k-Colourability to CSP(*Allen*) which introduces k fresh variables. Given an undirected graph $G = (\{v_1, \ldots, v_n\}, E)$, introduce the variables $z_1, \ldots, z_k$ and $v_1, \ldots, v_n$, and:

1. impose the constraints $z_1\{\mathsf{m}\}z_2\{\mathsf{m}\} \ldots \{\mathsf{m}\}z_k$,
2. for each v_i, $1 \leq i \leq n$, add the constraints $v_i\{\equiv, \mathsf{s}^{-1}\}z_1$, $v_i\{\mathsf{p}, \mathsf{m}, \mathsf{f}^{-1}, \mathsf{d}^{-1}\}z_j$ $(2 \leq j \leq k-1)$, and $v_i\{\mathsf{p}, \mathsf{m}, \mathsf{f}^{-1}\}z_k$,
3. for each $(v_i, v_j) \in E$, add the constraint $v_i\{\mathsf{s}, \mathsf{s}^{-1}\}v_j$.

Consulting Table 1, we see that for each v_i, it holds that its right endpoint must equal the right endpoint of some z_i, and its left endpoint must equal the left endpoint of z_1. It is now obvious that the resulting instance has a solution if and only if G is k-colourable. The result then follows since there is a polynomial-time Turing reduction from Chromatic Number to CSP(*Allen*) by combining binary search (that will evaluate $\log n$ Allen instances) with the reduction above (recall that $O(\log n \cdot c^n) \subseteq O((c+\epsilon)^n)$ for every $\epsilon > 0$) . Observe that if $k = n$ then the reduction introduces n fresh variables, which is where the constant $\sqrt{c}$ in the expression $O(\sqrt{c}^{|V|})$ stems from. CSP(*Allen*). □

The exact complexity of Chromatic Number has been analysed and discussed in the literature. Björklund et al. [2] have shown that the problem is solvable in $2^{|V|} \cdot poly(||I||)$ time. Impagliazzo and Paturi [17] poses the following question: "Assuming SETH, can we prove a $2^{n-o(n)}$ lower bound for Chromatic Number?". Hence, an $O(\sqrt{c}^{|V|})$, $c < 2$, algorithm for CSP(*Allen*) would also be a major breakthrough for Chromatic Number.

6 Discussion

We have investigated several novel algorithms for solving disjunctive CSP problems, which, with respect to worst-case time complexity, are much more efficient than e.g. backtracking algorithms without heuristics. These bounds can likely be improved, but, due to the lower bounds in Section 5, probably not to a great degree. Despite this, algorithms for solving infinite domain constraint satisfaction problems are in practice used in many non-trivial applications. In light of this the following research direction is particularly interesting: *how to formally analyse the time complexity of branching algorithms equipped with (powerful) heuristics?* In the case of finite-domain CSPs and, in particular, DPLL-like algorithms for the k-SAT problem there are numerous results to be found in the literature, cf. the survey by Vsemirnov et al. [35]. This is not the case for infinite-domain CSPs, even though there is a considerable amount of empirical evidence that infinite-domain CSPs can be efficiently solved by such algorithms, so one ought to be optimistic about the chances of actually obtaining non-trivial bounds. Yet, sharp formal analyses appear to be virtually nonexistent in the literature.

Another research direction is to strengthen the lower bounds in Section 5 even further. It would be interesting to prove stronger lower bounds for CSP($\mathcal{B}^{\vee k}$) for some concrete choices of $\mathcal{B}$ and k. As an example, consider the temporal problem CSP($\mathcal{T}^{\vee 4}$). From Theorem 15 we see that CSP($\mathcal{T}^{\vee 4}$) is not solvable in $O(2^{s_4|V|})$ time for some $s_4 < \log 1.6$, assuming the ETH holds, since the currently best deterministic algorithm for 4-SAT runs in $O(1.6^{|V|})$ time [12]. On the other hand, if CSP($\mathcal{T}^{\vee 4}$) is solvable in $O(\sqrt{c}^{|V|})$ time for some $c < 2$, then Chromatic Number can be solved in $O((c+\epsilon)^{|V|})$ time for arbitrary $\epsilon > 0$. This can be proven similar to the reduction in Theorem 17 but by making use of temporal constraints instead of interval constraints. Hence, for certain choices of $\mathcal{B}$ and k it might be possible to improve upon the general bounds given in Section 5.

References

1. D. Berend and T. Tassa. Improved bounds on Bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.
2. A. Björklund, T. Husfeldt, and M. Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009.
3. M. Bodirsky. Complexity classification in infinite-domain constraint satisfaction. Mémoire d'habilitation à diriger des recherches, Université Diderot – Paris 7. Available at arXiv:1201.0856, 2012.
4. M. Bodirsky and M. Hils. Tractable set constraints. *Journal of Artificial Intelligence Research*, 45:731–759, 2012.
5. M. Bodirsky and J. Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 43(2):136–158, 2008.
6. M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):9:1–9:41, 2010.
7. M. Bodirsky and J. K. Mueller. The complexity of rooted phylogeny problems. *Logical Methods in Computer Science*, 7(4), 2011.
8. M. Broxvall and P. Jonsson. Point algebras for temporal reasoning: Algorithms and complexity. *Artificial Intelligence*, 149(2):179 – 220, 2003.
9. M. Broxvall, P. Jonsson, and J. Renz. Disjunctions, independence, refinements. *Artificial Intelligence*, 140(1-2):153 – 173, 2002.
10. C. Calabro. *The Exponential Complexity of Satisfiability Problems*. PhD thesis, University of California, San Diego, CA, USA, 2009.
11. D. Cohen, P. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.
12. E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. M. Kleinberg, C. H. Papadimitriou, P. Raghavan, and U. Schöning. A deterministic $(2 - 2/(k + 1))^n$ algorithm for k-SAT based on local search. *Theoretical Computer Science*, 289(1):69–83, 2002.
13. B. Djokic, M. Miyakawa, S. Sekiguchi, I. Semba, and I. Stojmenovic. A fast iterative algorithm for generating set partitions. *The Computer Journal*, 32(3):281–282, June 1989.
14. S. Gaspers. *Exponential Time Algorithms - Structures, Measures, and Bounds*. VDM, 2010.
15. W. Hodges. *A Shorter Model Theory*. Cambridge University Press, New York, NY, USA, 1997.
16. R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.
17. R. Impagliazzo and R. Paturi. Exact complexity and satisfiability. In G. Gutin and S. Szeider, editors, *Parameterized and Exact Computation*, volume 8246 of *Lecture Notes in Computer Science*, pages 1–3. Springer International Publishing, 2013.
18. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
19. P. Jonsson and C. Bäckström. A unifying approach to temporal constraint reasoning. *Artificial Intelligence*, 102(1):143–155, 1998.
20. P. Jonsson, V. Lagerkvist, G. Nordh, and B. Zanuttini. Complexity of SAT problems, clone theory and the exponential time hypothesis. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-2013)*, pages 1264–1277, 2013.

21. I. Kanj and S. Szeider. On the subexponential time complexity of CSP. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)*, 2013.
22. A. Krokhin, P. Jeavons, and P. Jonsson. Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra. *Journal of the ACM*, 50(5):591–640, September 2003.
23. G. Ligozat. *Qualitative Spatial and Temporal Reasoning*. Wiley-ISTE, 2011.
24. G. Ligozat and J. Renz. What is a qualitative calculus? A general framework. In C. Zhang, H. W. Guesgen, and W. K. Yeap, editors, *PRICAI 2004: Trends in Artificial Intelligence*, volume 3157 of *Lecture Notes in Computer Science*, pages 53–64. Springer Berlin Heidelberg, 2004.
25. B. Liu and J. Jaffar. Using constraints to model disjunctions in rule-based reasoning. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI-96, Portland, Oregon, 1996, Volume 2.*, pages 1248–1255, 1996.
26. D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of EATCS*, 3(105), 2013.
27. K. Marriott, P. Moulder, P. J. Stuckey, and A Borning. Solving disjunctive constraints for interactive graphical applications. In *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming (CP-2001)*, pages 361–376, 2001.
28. H. Prüfer. Neuer beweis eines satzes über permutationen. *Archiv der Mathematik und Physik*, 27:742–744, 1918.
29. J. Renz and B. Nebel. Efficient methods for qualitative spatial reasoning. *Journal of Artificial Intelligence Research*, 15(1):289–318, 2001.
30. F. Rossi, P. van Beek, and T. Walsh, editors. *Handbook of Constraint Programming*. Elsevier, 2006.
31. K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120(1):81–117, 2000.
32. I. Stojmenović. An optimal algorithm for generating equivalence relations on a linear array of processors. *BIT Numerical Mathematics*, 30(3):424–436, 1990.
33. P. Traxler. The time complexity of constraint satisfaction. In *Proceeding of the Third International Workshop on Parameterized and Exact Computation (IWPEC-2008)*, pages 190–201, 2008.
34. Marc B. Vilain and Henry A. Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI-86)*, pages 377–382, 1986.
35. M. Vsemirnov, E. Hirsch, E. Dantsin, and S. Ivanov. Algorithms for SAT and upper bounds on their complexity. *Journal of Mathematical Sciences*, 118(2):4948–4962, 2003.
36. G. Woeginger. Exact algorithms for NP-hard problems: a survey. In M. Juenger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial Optimization – Eureka! You Shrink!*, pages 185–207, 2000.