Time Complexity of Constraint Satisfaction via Universal Algebra

Peter Jonsson*[1], Victor Lagerkvist†[2], and Biman Roy‡[3]

[1]Department of Computer and Information Science, Linköping University, Linköping, Sweden
[2]Institut für Algebra, TU Dresden, Dresden, Germany
[3]Department of Computer and Information Science, Linköping University, Linköping, Sweden

Abstract

The *exponential-time hypothesis* (ETH) states that 3-SAT is not solvable in subexponential time, i.e. not solvable in $O(c^n)$ time for arbitrary $c > 1$, where n denotes the number of variables. Problems like k-SAT can be viewed as special cases of the *constraint satisfaction problem* (CSP), which is the problem of determining whether a set of constraints is satisfiable. In this paper we study the worst-case time complexity of NP-complete CSPs. Our main interest is in the CSP problem parameterized by a constraint language Γ (CSP(Γ)), and how the choice of Γ affects the time complexity. It is believed that CSP(Γ) is either tractable or NP-complete, and the *algebraic CSP dichotomy conjecture* gives a sharp delineation of these two classes based on algebraic properties of constraint languages. Under this conjecture and the ETH, we first rule out the existence of subexponential algorithms for finite-domain NP-complete CSP(Γ) problems. This result also extends to certain infinite-domain CSPs and structurally restricted CSP(Γ) problems. We then begin a study of the complexity of NP-complete CSPs where one is allowed to arbitrarily restrict the values of individual variables, which is a very well-studied subclass of CSPs. For such CSPs with finite domain D, we identify a relation S_D such that (1) CSP($\{S_D\}$) is NP-complete and (2) if CSP(Γ) over D is NP-complete and solvable in $O(c^n)$ time, then CSP($\{S_D\}$) is solvable in $O(c^n)$ time, too. Hence, the time complexity of CSP($\{S_D\}$) is a lower bound for all CSPs of this particular kind. We also prove that the complexity of CSP($\{S_D\}$) is decreasing when $|D|$ increases, unless the ETH is false. This implies, for instance, that for every $c > 1$ there exists a finite-domain Γ such that CSP(Γ) is NP-complete and solvable in $O(c^n)$ time.

1 Introduction

The *constraint satisfaction problem* over a constraint language Γ (CSP(Γ)) is the computational decision problem of verifying whether a set of constraints over Γ is satisfiable or not. This problem is widely studied from both a theoretical and a practical standpoint. From a practical point of view this problem can be used to model many natural problems occurring in real-world applications. From a more theoretical point of view the CSP problem is (among several other things) of great interest due to its connections with *universal algebra*. It is widely believed that finite-domain CSP problems admit a dichotomy between tractable and NP-complete problems, and the so-called *algebraic approach* has been used to conjecture an exact borderline between

*peter.jonsson@liu.se
†victor.lagerqvist@tu-dresden.de
‡biman.roy@liu.se

tractable and NP-complete problems [15]. This conjectured borderline is sometimes called the *algebraic CSP dichotomy conjecture*. The gist of the algebraic approach is to associate an algebra, a set of functions satisfying a certain closure property, to each constraint language. This associated algebra is usually referred to as the *polymorphisms* of a constraint language, and is known to determine the complexity of a CSP problem up to polynomial-time many-one reductions [26]. However, the mere fact that two CSPs are polynomial-time interreducible does not offer much insight into their relative worst-case time complexity. For example, on the one hand, it has been conjectured that the Boolean satisfiability problem with unrestricted clause length, SAT, is not solvable strictly faster than $O(2^n)$, where n denotes the number of variables [23]. On the other hand, k-SAT is known to be solvable strictly faster than $O(2^n)$ for every $k \geq 1$ [22], and even more efficient algorithms are known for severely restricted satisfiability problems such as 1-in-3-SAT [36]. This discrepancy in complexity stems from the fact that a polynomial time reduction can change the structure of an instance and e.g. introduce a large number of fresh variables. Hence, it is worthwhile to study the complexity of NP-complete CSPs using more fine-grained notions of reductions. To make this a bit more precise, given a constraint language Γ we let

$$\mathsf{T}(\Gamma) = \inf\{c \mid \mathrm{CSP}(\Gamma) \text{ is solvable in time } 2^{cn}\}$$

where n denotes the number of variables. If $\mathsf{T}(\Gamma) = 0$ then $\mathrm{CSP}(\Gamma)$ is said to be solvable in *subexponential time*, and the conjecture that 3-SAT is not solvable in subexponential time is known as the *exponential-time hypothesis* (ETH) [23]. It is worth remarking that no concrete values of $\mathsf{T}(\Gamma)$ are known when $\mathrm{CSP}(\Gamma)$ is NP-complete. Despite this, studying properties of the function T can still be of great interest since such properties can be used to compare and relate the worst-case running times of NP-complete CSP problems. Moreover, for Boolean constraint languages, several properties of the function T are known. For example, it is known that there exists a finite Boolean constraint language Γ such that $\mathrm{CSP}(\Gamma)$ is NP-complete and $\mathsf{T}(\Gamma) = 0$ if and only if $\mathsf{T}(\Gamma) = 0$ for *every* Boolean constraint language Γ [27]. Hence, even though the status of the ETH is unclear at the moment, finding a subexponential time algorithm for one NP-complete Boolean CSP problem is tantamount to being able to solve every Boolean CSP problem in subexponential time. It is also known that there exists a Boolean relation R such that $\mathrm{CSP}(\{R\})$ is NP-complete but $\mathsf{T}(\{R\}) \leq \mathsf{T}(\Gamma)$ for *every* Boolean constraint language Γ such that $\mathrm{CSP}(\Gamma)$ is NP-complete. In Jonsson et al. [27] this problem is referred to as the *easiest NP-complete Boolean CSP problem*. The existence of this relation e.g. rules out the possibility that for each Boolean constraint language Γ there exists Δ such that $\mathsf{T}(\Delta) < \mathsf{T}(\Gamma)$ — a scenario which otherwise would have been compatible with the ETH. These results were obtained by considering more refined algebras than polymorphisms, so-called *partial polymorphisms*. We will describe this algebraic approach in greater detail later on, but the most important property is that the partial polymorphisms of finite constraint languages give rise to a partial order $\sqsubseteq$ with the property that if $\Gamma \sqsubseteq \Delta$, then $\mathsf{T}(\Gamma) \leq \mathsf{T}(\Delta)$. We remark that partial polymorphisms are not only useful when studying CSPs with this very fine-grained notion of complexity, but have also been used to study the classical complexity of many different computational problems where polymorphisms are not applicable [3, 4, 11, 14, 21].

Hence, even though no concrete values are known for $\mathsf{T}(\Gamma)$ when $\mathrm{CSP}(\Gamma)$ is NP-complete, quite a lot is known concerning the relationship between $\mathsf{T}(\Gamma)$ and $\mathsf{T}(\Delta)$ for Boolean Γ and Δ. In this paper we study similar properties of the function T for constraint languages defined over arbitrary finite domains. After having introduced the necessary definitions in Section 2, in Section 3 we consider the existence of subexponential time algorithms for NP-complete CSP problems, in light of the ETH and the algebraic CSP dichotomy conjecture. For this question we obtain a complete understanding and prove that, assuming the algebraic CSP dichotomy conjecture, the ETH is false if and only if (1) there exists a finite constraint language Γ over a finite domain such that $\mathrm{CSP}(\Gamma)$ is NP-complete and $\mathsf{T}(\Gamma) = 0$, if and only if (2) $\mathsf{T}(\Gamma) = 0$ for every finite constraint language Γ defined over a finite domain. In other words, finding a subexponential time algorithm for a single NP-complete, finite-domain CSP problem is tantamount to being able to solve all CSP problems in subexponential time. We also study structurally restricted CSPs where the maximum number of constraints a variable may appear in is bounded by a constant B (CSP(Γ)-B). For problems of this form our results are not as sharp, but we prove that, again assuming the algebraic CSP dichotomy conjecture, that if CSP(Γ) is NP-complete and Γ satisfies an additional algebraic condition, then there exists a constant B such that CSP(Γ)-B is not solvable in subexponential time (unless the ETH is false). We also remark that our proof extends to certain constraint languages defined over infinite domain, and give several examples of infinite-domain NP-complete CSP problems that are not solvable in subexponential time, unless the ETH is false. These results may be interesting to compare to those of De Haan et al. [17], who study subexponential algorithms for structurally restricted CSPs. One crucial difference to our results is that De Haan et al. do not consider constraint language restrictions. For example, it is proven that CSP(Δ)-2, where Δ is the set of all finitary relations of finite cardinality, is not solvable in subexponential time unless the ETH is false. However, a result of this form tells us very little about the complexity of CSP(Γ)-2 for specific constraint languages, since it does not imply that CSP(Γ)-2 is not solvable in subexponential time for every Γ such that CSP(Γ)-2 is NP-complete.

We have thus established that $\mathsf{T}(\Gamma) > 0$ for every NP-complete, finite-domain CSP(Γ), assuming the ETH and the algebraic CSP dichotomy conjecture. This immediately raises the question of which further insights can be gained concerning the behaviour of the function T. For example, for a fixed finite domain, is it possible to construct an infinite chain of NP-complete CSPs with strictly decreasing complexity such that T tends to 0? We study such questions in Section 4 for CSPs where one in an instance is allowed to restrict the values of individual variables arbitrarily. This restricted CSP problem is particularly well-studied, and it is used as *the* definition of CSPs in many cases: see, for instance, the textbook by Russell and Norvig [33, Section 3.7] and the handbook by Rossi et al. [32, Section 2]. This may be viewed as restricting oneself to constraint languages that contain all unary relations. A closely related restriction (that is typically used when studying CSPs from the algebraic viewpoint) is that every unary relation is primitively positively definable in Γ (see Section 2). Such constraint languages are known as *conservative*. These two restrictions are computationally equivalent up to polynomial-time many-one reductions but it is not known whether they are equivalent under reductions that preserve time complexity. Thus, we need to separate them, so we say that a constraint language that contains all unary relations is *ultraconservative*. We note that the algebraic CSP dichotomy conjecture has been verified to hold for the conservative CSPs [12] so it holds for ultraconservative CSPs, too. We show that for every finite domain D there exists a relation S_D such that CSP($\{S_D\}$) is NP-complete and $\mathsf{T}(\{S_D\}) = \mathsf{T}(\{S_D\} \cup 2^D) \leq \mathsf{T}(\Gamma)$ for every ultraconservative and NP-complete CSP(Γ) over D. This relation will be formally defined in Section 4.1, but is worth pointing out that S_D contains only three tuples and that CSP($\{S_D\}$) can be viewed as a higher-domain variant of the monotone 1-in-3-SAT problem. We refer to CSP($\{S_D\} \cup 2^D$) as the *easiest NP-complete ultraconservative CSP problem over D*[1]. Note that the properties of the relation S_D rule out the possibility of an infinite sequence of ultraconservative languages $\Gamma_1, \Gamma_2, \ldots$ such that each CSP(Γ_i) is NP-complete and $\mathsf{T}(\Gamma_i)$ tends to 0, but also have stronger implications, since the value $\mathsf{T}(\{S_D\})$ is a conditional lower bound for the complexity of *all* NP-complete, ultraconservative CSPs over D.

To prove these results we have to overcome several major obstacles. Similar to Jonsson et al. [27]) we use partial polymorphisms instead of total polymorphisms in order to achieve more fine-grained notions of reductions. However, the proof strategy used in Jonsson et al. [27] does not work for arbitrary finite domains since it requires a comprehensive understanding of the polymorphisms of constraint languages resulting in NP-complete CSPs, which is only known for the Boolean domain [29]. Our first observation to tackle this difficulty is that the reformulation

[1]Note that 2^D is the set of all unary relations over D.

of conservative CSP dichotomy theorem making use of *primitive positive interpretations* (pp-interpretations) is useful in our context. At the moment, we may think of a pp-interpretation as a tool which allows us to compare the expressitivity of constraint languages defined over diferent domains, modulo logical formulas consisting of existential quantification, conjunction, and equality constraints. It is well-known that pp-interpretations can be used to obtain polynomial-time reductions between CSPs, and that a conservative CSP(Γ) problem is NP-complete if and only if Γ pp-interprets 3-SAT [1, 12]. However, as already pointed out, such reductions are not useful when studying CSPs with respect to the function T, and it is a priori not evident how the assumption that Γ can pp-interpret 3-SAT can be used to show that $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\Gamma)$. Using properties of conservative constraint languages and quantifier-elimination techniques we in Section 4.1 first show that this assumption can be used to prove there exists a relation R over D of cardinality 3 such that (1) CSP($\{R\}$) is NP-complete and (2) $\mathsf{T}(\{R\}) \leq \mathsf{T}(\Gamma)$. However, this is not enough in order to isolate a unique easiest problem, since there for every finite domain exists a large number of such relations. In Section 4.2, using a combination of partial clone theory and size-preserving reductions, we show that $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\{R\})$ for every such relation R of cardinality 3. We then analyse the time complexity of the problem CSP($\{S_D\}$) and prove that $\mathsf{T}(\{S_D\})$ tends to 0 for increasing values of $|D|$. This also shows, despite the fact that no finite-domain NP-complete CSP(Γ) is solvable in subexponential time (if the algebraic CSP dichotomy conjecture and the ETH are true), that one for every $c > 0$ can find Γ over a finite domain such that CSP(Γ) is NP-complete and solvable in $O(2^{cn})$ time. When all of these results are adjoined, they demonstrate that the function T can indeed be analysed without an extensive knowledge of the polymorphisms related to a constraint language.

2 Preliminaries

Relations and constraint languages. A k-ary *relation* R over a set D is a subset of D^k, and we write $\mathrm{ar}(R) = k$ to denote its arity. A finite set of relations Γ over a set D is called a *constraint language*. Given two tuples s and t we let $s^\frown t$ denote the concatenation of s and t, i.e., if $s = (s_1, \ldots, s_{k_1})$ and $t = (t_1, \ldots, t_{k_2})$ then $s^\frown t = (s_1, \ldots, s_{k_1}, t_1, \ldots, t_{k_2})$. If t is an n-ary tuple we let $t[i]$ denote its ith element and $\mathrm{Proj}_{i_1,\ldots,i_{n'}}(t) = (t[i_1], \ldots, t[i_{n'}])$, $n' \leq n$, denote the *projection* of t on the coordinates $i_1, \ldots, i_{n'} \in \{1, \ldots, n\}$. Similarly, if R is an n-ary relation we let $\mathrm{Proj}_{i_1,\ldots,i_{n'}}(R) = \{\mathrm{Proj}_{i_1,\ldots,i_{n'}}(t) \mid t \in R\}$. We write Eq_D for the equality relation $\{(x, x) \mid x \in D\}$. If there is no risk for confusion we omit the subscript and simply write Eq. For each $d \in D$ we write R^d for the unary, constant relation $\{(d)\}$. We will occasionally represent relations by first-order formulas, and if $\varphi(x_1, \ldots, x_k)$ is a first-order formula with free variables $x_1, \ldots, x_k$ then we write $R(x_1, \ldots, x_k) \equiv \varphi(x_1, \ldots, x_k)$ to define the relation $R = \{(f(x_1), \ldots, f(x_k)) \mid f \text{ is a model of } \varphi(x_1, \ldots, x_k)\}$. As a graphical representation, we will sometimes view a k-ary relation $R = \{t_1, \ldots, t_m\}$ as an $m \times k$ matrix where the columns of the matrix enumerate the arguments of the relation (in some fixed ordering). For example, $\left(\begin{smallmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{smallmatrix}\right)$ represents the relation $\{(0, 0, 1, 1), (0, 1, 0, 1)\}$.

The constraint satisfaction problem. The *constraint satisfaction problem* over a constraint language Γ over D (CSP(Γ)) is the computational decision problem defined as follows.

Instance: A set V of variables and a set C of constraint applications $R(x_1, \ldots, x_k)$ where $R \in \Gamma$, $\mathrm{ar}(R) = k$, and $x_1, \ldots, x_k \in V$.
Question: Does there exist $f : V \to D$ such that $(f(x_1), \ldots, f(x_k)) \in R$ for each $R(x_1, \ldots, x_k)$ in C?

If $\Gamma = \{R\}$ is singleton then we write CSP(R) instead of CSP($\{R\}$), and if Γ is Boolean we typically write SAT(Γ) instead of CSP(Γ). We let $\mathbb{B} = \{0, 1\}$. For example, let $R^{\neq\neq\neq 01}_{1/3} = \{(0, 0, 1, 1, 1, 0, 0, 1), (0, 1, 0, 1, 0, 1, 0, 1), (1, 0, 0, 0, 1, 1, 0, 1)\}$. The SAT problem over $R^{\neq\neq\neq 01}_{1/3}$ can be seen as a variant of 1-in-3-SAT where each variable in each constraint has a complementary variable. We will return to this SAT problem several times in the sequel. For each $k \geq 3$ let

Γ^k_{SAT} be the constraint language which for every $t \in \mathbb{B}^k$ contains the relation $\mathbb{B}^k \setminus \{t\}$. Hence, $\text{SAT}(\Gamma^k_{\text{SAT}})$ can be viewed as an alternative formulation of k-SAT.

Primitive positive definitions and interpretations. Let Γ be a constraint language. A k-ary relation R is said to have a *primitive positive definition* (pp-definition) over Γ if $R(x_1, \ldots, x_k) \equiv \exists y_1, \ldots, y_{k'} . R_1(\mathbf{x_1}) \wedge \ldots \wedge R_m(\mathbf{x_m})$, where each $R_i \in \Gamma \cup \{\text{Eq}\}$ and each $\mathbf{x_i}$ is an $\text{ar}(R_i)$-ary tuple of variables over $x_1, \ldots, x_k, y_1, \ldots, y_{k'}$. In addition, if the primitive positive formula does not contain any existentially quantified variables, we say that it is a *quantifier-free primitive positive formula* (qfpp), and if it does not contain any equality constraints we say that it is a *equality-free primitive positive formula* (efpp). For example, the reader can verify that the textbook reduction from k-SAT to $(k-1)$-SAT, where a clause of length k is replaced by clauses of length $k-1$ making use of one fresh variable, can be formulated as a pp-definition but not as a qfpp-definition. We write $\langle \Gamma \rangle$ (respectively $\langle \Gamma \rangle_{\nexists}$) to denote the smallest set of relations containing Γ and which is closed under pp-definitions (respectively qfpp-definitions). If $\Gamma = \{R\}$ is singleton then we instead write $\langle R \rangle$ and $\langle R \rangle_{\nexists}$. Note that $\langle \Gamma \rangle$ is closed under projections, in the sense that if $R \in \langle \Gamma \rangle$ then $\text{Proj}_{i_1,\ldots,i_n}(R) \in \langle \Gamma \rangle$ for all $i_1, \ldots, i_n \in \{1, \ldots, \text{ar}(R)\}$, but that this does not necessarily hold for $\langle \Gamma \rangle_{\nexists}$. Jeavons [25] proved the following important result.

Theorem 1. *If Γ is a constraint language and Δ is a finite subset of $\langle \Gamma \rangle$, then CSP(Δ) is polynomial-time reducible to CSP(Γ).*

Theorem 1 naturally holds also for relations defined by qfpp- or efpp-formulas. However, there are additional advantages of these more restricted ways of defining relations and we will return to them later on. We are now ready to define the concept of primitive positive interpretations.

Definition 2. *Let D and E be two domains and let Γ and Δ be two constraint languages over D and E, respectively. A* primitive positive interpretation *(pp-interpretation) of Δ over Γ consists of a d-ary relation $F \subseteq D^d$ and a surjective function $f : F \to E$ such that $F, f^{-1}(\text{Eq}_E) \in \langle \Gamma \rangle$ and $f^{-1}(R) \in \langle \Gamma \rangle$ for every $R \in \Delta$, where $f^{-1}(R)$, $\text{ar}(R) = k$, denotes the $(k \cdot d)$-ary relation*

$$\{(x_{1,1}, \ldots, x_{1,d}, \ldots, x_{k,1}, \ldots, x_{k,d}) \in D^{k \cdot d} \mid (f(x_{1,1}, \ldots, x_{1,d}), \ldots, f(x_{k,1}, \ldots, x_{k,d})) \in R\}.$$

The main purpose of pp-interpretations is to relate constraint languages which might be incomparable with respect to pp-definitions. For an example, let us consider the relation $R_{\neq} = \{(x, y) \in \{0, 1, 2\}^2 \mid x \neq y\}$, and observe that $\text{CSP}(\{R_{\neq}\})$ corresponds to the 3-coloring problem. We invite the reader to verify that the standard reduction from 3-coloring to 3-SAT can be phrased as a pp-interpretation of $R_{\neq}$ over Γ^3_{SAT}, but that this reduction cannot be expressed via pp-definitions due to the different domains. Hence, pp-interpretations are generalizations of pp-definitions, and can be used to obtain polynomial-time reductions between CSPs.

Theorem 3 (cf. Theorem 5.5.6 in Bodirsky [5])**.** *If Γ, Δ are constraint languages and there is a pp-interpretation of Δ over Γ, then CSP(Δ) is polynomial-time reducible to CSP(Γ).*

Polymorphisms and partial polymorphisms. Let f be a k-ary function over a finite domain D. We say that f is a *polymorphism* of an n-ary relation R over D if $f(t_1, \ldots, t_k) \in R$ for each k-ary sequence of tuples $t_1, \ldots, t_k \in R$. Here, and in the sequel, we use $f(t_1, \ldots, t_k)$ to denote the componentwise application of the function f to the tuples $t_1, \ldots, t_k$, i.e., $f(t_1, \ldots, t_k)$ is a shorthand for the n-ary tuple $(f(t_1[1], \ldots, t_k[1]), \ldots, f(t_1[n], \ldots, t_k[n]))$. Similarly, if f is a partial function over D, we say that f is a *partial polymorphism* of an n-ary relation R over D if $f(t_1, \ldots, t_k) \in R$ for every sequence $t_1, \ldots, t_k$ such that $f(t_1, \ldots, t_k)$ is defined for each componentwise application. If f is a polymorphism or a partial polymorphism of a relation R then we occasionally also say that R is *invariant* under f. We let $\text{Pol}(R)$ and $\text{pPol}(R)$ denote the set of all polymorphisms, respectively partial polymorphisms, of the relation R. Similarly, for a constraint language Γ, we write $\text{Pol}(\Gamma)$ for the set $\bigcap_{R \in \Gamma} \text{Pol}(R)$, and $\text{pPol}(\Gamma)$ for the set $\bigcap_{R \in \Gamma} \text{pPol}(R)$. We write $\text{Inv}(F)$ to denote the set of all relations invariant under the set of total or partial functions F. It is known that $\mathrm{Inv}(\mathrm{Pol}(\Gamma)) = \langle\Gamma\rangle$ and that $\mathrm{Inv}(\mathrm{pPol}(\Gamma)) = \langle\Gamma\rangle_{\nexists}$, giving rise to the following *Galois connections*.

Theorem 4 ([9, 10, 19, 31])**.** *Let Γ and Γ' be two constraint languages. Then $\Gamma \subseteq \langle\Gamma'\rangle$ if and only if $\mathrm{Pol}(\Gamma') \subseteq \mathrm{Pol}(\Gamma)$ and $\Gamma \subseteq \langle\Gamma'\rangle_{\nexists}$ if and only if $\mathrm{pPol}(\Gamma') \subseteq \mathrm{pPol}(\Gamma)$.*

Time complexity and size-preserving reductions. Given a constraint language Γ we let $\mathsf{T}(\Gamma) = \inf\{c \mid \mathrm{CSP}(\Gamma) \text{ is solvable in time } 2^{cn}\}$ where n denotes the number of variables in a given instance. If $\mathsf{T}(\Gamma) = 0$ then $\mathrm{CSP}(\Gamma)$ is said to be solvable in *subexponential time*. The conjecture that $\mathrm{SAT}(\Gamma^3_{\mathrm{SAT}}) > 0$ is known as the *exponential-time hypothesis* (ETH) [24]. We now introduce a type of reduction useful for studying the complexity of CSPs with respect to the function T.

Definition 5. *Let Γ and Δ be two constraint languages. The function f from the instances of $\mathrm{CSP}(\Gamma)$ to the instances of $\mathrm{CSP}(\Delta)$ is a* many-one linear variable reduction *(LV-reduction) with parameter $d \geq 0$ if (1) f is a polynomial-time many-one reduction from $\mathrm{CSP}(\Gamma)$ to $\mathrm{CSP}(\Delta)$ and (2) $|V'| = d \cdot |V| + O(1)$ where V, V' are the set of variables in I and $f(I)$, respectively.*

The term CV-reduction, short for *constant variable reduction*, is used to denote LV-reductions with parameter 1, and we write $\mathrm{CSP}(\Gamma) \leq^{\mathrm{CV}} \mathrm{CSP}(\Delta)$ when $\mathrm{CSP}(\Gamma)$ has a CV-reduction to $\mathrm{CSP}(\Delta)$. It follows that if $\mathrm{CSP}(\Gamma) \leq^{\mathrm{CV}} \mathrm{CSP}(\Delta)$ then $\mathsf{T}(\Gamma) \leq \mathsf{T}(\Delta)$, and if $\mathrm{CSP}(\Gamma)$ LV-reduces to $\mathrm{CSP}(\Delta)$ then $\mathsf{T}(\Gamma) = 0$ if $\mathsf{T}(\Delta) = 0$. We have the following theorem from Jonsson et al. [27], relating the partial polymorphisms of constraint languages with the existence of CV-reductions.

Theorem 6 ([27])**.** *Let D be a finite domain and let Γ and Δ be two constraint languages over D. If $\mathrm{pPol}(\Delta) \subseteq \mathrm{pPol}(\Gamma)$ then $\mathrm{CSP}(\Gamma) \leq^{\mathrm{CV}} \mathrm{CSP}(\Delta)$.*

We remark that the original proof only concerned Boolean constraint languages but that the same proof also works for arbitrary finite domains. Using Theorem 6 and algebraic techniques from Schnoor and Schnoor [35], Jonsson et al. [27] proved that $\mathsf{T}(\{R^{\neq\neq\neq01}_{1/3}\}) \leq \mathsf{T}(\Gamma)$ for any finite Γ such that $\mathrm{SAT}(\Gamma)$ is NP-complete. This problem was referred to as the *easiest NP-complete* SAT *problem*. We will not go into the details but remark that the proof idea does not work for arbitrary finite domains since it requires a characterisation of every $\mathrm{Pol}(\Gamma)$ such that $\mathrm{CSP}(\Gamma)$ is NP-complete. Such a list is known for the Boolean domain due to Post [29] and Schaefer [34], but not for larger domains.

Complexity of CSP. Let Γ be a constraint language over a finite domain D. We say that Γ is *idempotent* if $R^d \in \langle\Gamma\rangle$ for every $d \in D$, *conservative* if $2^D \subseteq \langle\Gamma\rangle$, and *ultraconservative* if $2^D \subseteq \Gamma$. A unary function $f \in \mathrm{Pol}(\Gamma)$ is said to be an *endomorphism*, and if f in addition is bijective it is said to be an *automorphism*. A constraint language Γ is a *core* if every endomorphism is an automorphism. The following theorem is well-known, see e.g. Barto [1], but is usually expressed in term of polynomial-time many-one reductions instead of CV-reductions.

Theorem 7. *Let Γ be a core constraint language over the domain $\{d_0, \ldots, d_{k-1}\}$. Then $\mathrm{CSP}(\Gamma \cup \{R^{d_0}, \ldots, R^{d_{k-1}}\}) \leq^{\mathrm{CV}} \mathrm{CSP}(\Gamma)$.*

If Γ is a constraint language over $D = \{d_0, \ldots, d_{k-1}\}$, then $\Gamma \cup \{R^{d_0}, \ldots, R^{d_{k-1}}\}$ is both idempotent and a core since its only endomorphism is the identity function on D. The *CSP dichotomy conjecture* states that for any Γ over a finite domain, $\mathrm{CSP}(\Gamma)$ is either tractable or NP-complete [18]. This conjecture was later refined by Bulatov et al. [15] to also induce a sharp characterization of the tractable and intractable cases, expressed in terms of algebraic properties of the constraint language, and is usually called the *algebraic CSP dichotomy conjecture*. We will use the following variant of the conjecture which is expressed in terms of pp-interpretations.

Conjecture 8. *[1, 15] Let Γ be an idempotent constraint language over a finite domain. Then $\mathrm{CSP}(\Gamma)$ is NP-complete if Γ pp-interprets Γ^3_{SAT} and tractable otherwise.*

It is worth remarking that if Γ pp-interprets Γ^3_{SAT} then Γ can pp-interpret every finite-domain relation [5, Theorem 5.5.17].

3 Subexponential Time Complexity

For Boolean constraint languages it has been proven that $\text{SAT}(\Gamma^3_{\text{SAT}})$ is solvable in subexponential time if and only if there exists a finite Boolean constraint language Γ such that $\text{SAT}(\Gamma)$ is NP-complete and solvable in subexponential time [27]. We will strengthen this result to arbitrary domains and prove that $\text{CSP}(\Gamma)$ is never solvable in subexponential time if Γ can pp-interpret Γ^3_{SAT}, unless the ETH is false. The result can also be extended to certain structurally restricted CSPs. The *degree* of a variable $x \in V$ of an instance (V, C) of $\text{CSP}(\Gamma)$ is the number of constraints in C containing x. We let $\text{CSP}(\Gamma)$-B, $B \geq 1$, denote the restricted $\text{CSP}(\Gamma)$ problem where each variable occurring in an instance has degree at most B. We then obtain the following theorem, whose proof can be found in Appendix A.

Theorem 9. *Assume that the ETH is true and let Γ be a finite constraint language over a domain D such that Γ pp-interprets Γ^3_{SAT}. Then $\text{CSP}(\Gamma)$ is not solvable in subexponential time, and if Γ efpp-defines Eq_D then there exists a constant B, depending only on Γ, such that $\text{CSP}(\Gamma)$-B is not solvable in subexponential time.*

We have now obtained a complete understanding of subexponential solvability of finite-domain CSPs modulo the ETH.

Corollary 10. *Assume that the algebraic CSP dichotomy conjecture is true. Then the following statements are equivalent.*

1. *The ETH is false.*
2. $\text{CSP}(\Gamma)$ *is solvable in subexponential time for every finite Γ over a finite domain.*
3. *There exists a finite constraint language Γ over a finite domain D such that $\text{CSP}(\Gamma)$ is NP-complete and subexponential.*

Proof. The implication from (1) to (2) follows from Impagliazzo et al. [24, Theorem 3]. The implication from (2) to (3) is trivial. For the implication from (3) to (1), we first note that $\text{CSP}(\Gamma^c) \leq^{\text{CV}} \text{CSP}(\Gamma)$, where Γ^c is the core of Γ [1, Theorem 3.5]. If Γ^c is expanded with all constants, then Theorem 7 shows that the complexity does not change, and, last, this language can pp-interpret Γ^3_{SAT}, due to the assumption that the algebraic CSP dichotomy conjecture is true, which via Theorem 9 implies that 3-SAT is solvable in subexponential time, and thus that the ETH is false. □

For $\text{CSP}(\Gamma)$-B our results are not as precise since we need the additional assumption that the equality relation is efpp-definable. This is not surprising since the most powerful dichotomy results for CSPs are usually concerned with either constraint language restrictions [12, 15], structural restrictions [17, 20], but rarely both simultaneously. However, in the Boolean domain there are plenty of examples which illustrates how the equality relation may be efpp-defined [16, 27], suggesting that similar techniques may also exist for larger domains.

Theorem 9 also applies to many interesting classes of infinite-domain CSPs. For example, if we consider Γ such that each $R \in \Gamma$ has a first-order definition over the structure $(\mathbb{Q}; <)$, it is known that $\text{CSP}(\Gamma)$ is NP-complete if and only if Γ can pp-interpret Γ^3_{SAT} [5, 7]. Hence, Theorem 9 is applicable, implying that if $\text{CSP}(\Gamma)$ is not solvable in subexponential time if it is NP-complete, unless the ETH fails. More examples of infinite-domain CSPs where Theorem 9 is applicable includes graph satisfiability problems [8] and phylogeny constraints [6]. Note that all of these results hold independently of whether the algebraic CSP dichotomy is true or not. We also remark that the intractable cases of the CSP dichotomy conjecture for certain infinite-domain

CSPs are all based on pp-interpretability of Γ^3_{SAT} [2]. If this conjecture is correct, Theorem 9 and the ETH implies that none of these problems are solvable in subexponential time.

4 The Easiest NP-Complete Ultraconservative CSP Problem

The results from Section 3, assuming the algebraic CSP dichotomy conjecture and the ETH, implies that $\mathsf{T}(\Gamma) > 0$ for any finite-domain and NP-complete CSP(Γ). However, it is safe to say that very little is known about the behaviour of the function T in more general terms. For example, is there for an arbitrary NP-complete CSP(Γ) possible to find an NP-complete CSP(Δ) such that $\mathsf{T}(\Delta) < \mathsf{T}(\Gamma)$? Such a scenario would be compatible with the consequences of Theorem 9. We will show that this is unlikely, and prove that there for every finite domain D exists a relation S_D such that CSP(S_D) is NP-complete but $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\Gamma)$ for any ultraconservative Γ over D such that CSP(Γ) is NP-complete. To prove this we have divided this section into two parts. In Section 4.1 we show that if Γ is ultraconservative and CSP(Γ) is NP-complete, then there exists a relation $R \in \langle\Gamma\rangle_{\nexists}$ which shares certain properties with the relation $R^{\neq\neq\neq01}_{1/3}$. In Section 4.2 we use properties of these relations in order to prove that there for every finite domain D is possible to find a relation S_D such that CSP(S_D) is CV-reducible to any other NP-complete and ultraconservative CSP(Γ) problem.

4.1 $S_{\mathbb{B}}$-Extensions

The columns of the matrix representation of the relation $R^{\neq\neq\neq01}_{1/3}$ from Jonsson et al. [27] (resulting in the easiest NP-complete SAT problem) enumerates all Boolean ternary tuples. We generalize this relation to arbitrary finite domains as follows.

Definition 11. *For each finite D let $S_D = \{t_1, t_2, t_3\}$ denote the $|D|^3$-ary relation such that there for every $(d_1, d_2, d_3) \in D^3$ exists $1 \leq i \leq |D|^3$ such that $(t_1[i], t_2[i], t_3[i]) = (d_1, d_2, d_3)$.*

Hence, similar to $R^{\neq\neq\neq01}_{1/3}$, the columns of the matrix representation of S_D enumerates all ternary tuples over D. For each D the relation S_D is unique up to permutation of arguments, and although we will usually not be concerned with the exact ordering, we sometimes assume that $S_{\mathbb{B}} = R^{\neq\neq\neq01}_{1/3}$ and that $\mathrm{Proj}_{1,\ldots,8}(S_D) = S_{\mathbb{B}}$. The notation S_D is a mnemonic for *saturated*, and the reason behind this will become evident in Section 4.2.1. For example, for $\{0, 1, 2\}$ we obtain a relation $\{t_1, t_2, t_3\}$ with 27 distinct arguments such that $(t_1[i], t_2[i], t_3[i]) \in \{0, 1, 2\}^3$ for each $1 \leq i \leq 27$. Jonsson et al. [27] proved that $S_{\mathbb{B}} \in \langle\Gamma\rangle_{\nexists}$ for every Boolean and idempotent constraint language Γ such that SAT(Γ) is NP-complete. This is not true for arbitrary finite domains, and in order to prove an analogous result we will need the following definition.

Definition 12. *Let R be an n-ary relation of cardinality 3 over a domain D, $|D| \geq 2$. Let $a, b \in D$ be two distinct values. If there exists $i_1, \ldots, i_8 \in \{1, \ldots, n\}$ such that*

$$\mathrm{Proj}_{i_1,\ldots,i_8}(R) = \{(a, a, b, b, b, a, a, b), (a, b, a, b, a, b, a, b), (b, a, a, a, b, b, a, b)\},$$

then we say that R is an $S_{\mathbb{B}}$-extension.

For example, S_D is an $S_{\mathbb{B}}$-extension for every domain D. Note that CSP(R) is always NP-complete when R is an $S_{\mathbb{B}}$-extension. We will now prove that if CSP(Γ) is NP-complete and Γ is ultraconservative, then Γ can pp-define an $S_{\mathbb{B}}$-extension.

Lemma 13. *Let Γ be an ultraconservative constraint language over a finite domain D such that CSP(Γ) is NP-complete. Then there exists a relation $R \in \langle\Gamma\rangle$ which is an $S_{\mathbb{B}}$-extension.*

Proof. Since CSP(Γ) is NP-complete and Γ is ultraconservative, Γ can pp-interpret every Boolean relation. Therefore let $f : F \to \mathbb{B}$, $F \subseteq D^d$ denote the parameters in the pp-interpretation of $S_{\mathbb{B}}$, and note that $f^{-1}(S_{\mathbb{B}}) \in \langle \Gamma \rangle$, but that $f^{-1}(S_{\mathbb{B}})$ is not necessarily an $S_{\mathbb{B}}$-extension since it could be the case that $|f^{-1}(S_{\mathbb{B}})| > 3$. Pick two tuples s and t in F such that $f(s) = 0$ and $f(t) = 1$. Such tuples must exist since f is surjective. Now consider the relation $F_1(x_1, \ldots, x_d) \equiv F(x_1, \ldots, x_d) \wedge \{(s[1]), (t[1])\}(x_1) \wedge \ldots \wedge \{(s[d], t[d])\}(x_d)$. This relation is pp-definable over Γ since Γ is ultraconservative and since $F \in \langle \Gamma \rangle$. By construction, it is clear that $s, t \in F_1$. Assume furthermore than $|F_1| > 2$, i.e., that there exists $u \in F_1 \setminus \{s, t\}$. Assume without loss of generality that $f(u) = 0$, and observe that there for each $i \in \{1, \ldots, d\}$ holds that $u[i] \in \{s[i], t[i]\}$. We claim that there exists some $i \in \{1, \ldots, d\}$ such that $u[i] = t[i] \neq s[i]$. To see this, observe that there must exist i such that $u[i] \neq s[i]$, since otherwise $u = s$, and it then follows that $u[i] = t[i]$. Construct the relation $F_2(x_1, \ldots, x_d) \equiv F_1(x_1, \ldots, x_d) \wedge \{(u[1]), (t[1])\}(x_1) \wedge \ldots \wedge \{(u[d]), (t[d])\}(x_d)$, and note that $F_2 \subset F_1$ since $s \notin F_2$. By repeating this procedure we will obtain a relation $F' \subseteq F$ such that $F' = \{s_0, s_1\}$ and such that $f(s_0) = 0$, $f(s_1) = 1$. Using the relation F' we can then pp-define the relation

$$\begin{aligned} R(x_{1,1}, \ldots, x_{1,d}, \ldots, x_{8,1}, \ldots, x_{8,d}) \equiv & f^{-1}(S_{\mathbb{B}})(x_{1,1}, \ldots, x_{1,d}, \ldots, x_{8,1}, \ldots, x_{8,d}) \wedge \\ & F'(x_{1,1}, \ldots, x_{1,d}) \wedge \ldots \wedge F'(x_{8,1}, \ldots, x_{8,d}). \end{aligned}$$

Clearly, if $(a_{1,1}, \ldots, a_{1,d}, \ldots, a_{8,1}, \ldots, a_{8,d}) \in R$, then $(a_{i,1}, \ldots, a_{i,d}) \in \{s_0, s_1\}$ for each $1 \leq i \leq 8$, and $(f(a_{1,1}, \ldots, a_{1,d}), \ldots, f(a_{8,1}, \ldots, a_{8,d})) \in S_{\mathbb{B}}$ if and only if $(a_{1,1}, \ldots, a_{1,d}, \ldots, a_{8,1}, \ldots, a_{8,d}) \in f^{-1}(S_{\mathbb{B}})$. Since $R \subseteq f^{-1}(S_{\mathbb{B}})$, this implies that $(f(a_{1,1}, \ldots, a_{1,d}), \ldots, f(a_{8,1}, \ldots, a_{8,d})) \in S_{\mathbb{B}}$ if and only if $(a_{1,1}, \ldots, a_{1,d}, \ldots, a_{8,1}, \ldots, a_{8,d}) \in R$ and each $(a_{i,1}, \ldots, a_{i,d}) \in \{s_0, s_1\}$. In other words each element $f(a_{i,1}, \ldots, a_{i,d})$ in a tuple of $S_{\mathbb{B}}$ uniquely correponds to d arguments $a_{i,1}, \ldots, a_{i,d}$ in the corresponding tuple of R, since $(a_{i,1}, \ldots, a_{i,d}) = s_0$ if $f(a_{i,1}, \ldots, a_{i,d}) = 0$, and $(a_{i,1}, \ldots, a_{i,d}) = s_1$ if $f(a_{i,1}, \ldots, a_{i,d}) = 1$. It follows that

$$R = \{s_0 ^\frown s_0 ^\frown s_1 ^\frown s_1 ^\frown s_1 ^\frown s_0 ^\frown s_0 ^\frown s_1, s_0 ^\frown s_1 ^\frown s_0 ^\frown s_1 ^\frown s_0 ^\frown s_1 ^\frown s_0 ^\frown s_1, s_1 ^\frown s_0 ^\frown s_0 ^\frown s_0 ^\frown s_1 ^\frown s_1 ^\frown s_0 ^\frown s_1\},$$

and therefore also that R is an $S_{\mathbb{B}}$-extension. □

Observe that the existence of an $S_{\mathbb{B}}$-extension $R \in \langle \Gamma \rangle$ does not imply that $\mathrm{CSP}(R) \leq^{\mathrm{CV}} \mathrm{CSP}(\Gamma)$. To accomplish this, we need to show that Γ can also qfpp-define an $S_{\mathbb{B}}$-extension.

Lemma 14. *Let Γ be an ultraconservative constraint language over a finite domain D such that* CSP(Γ) *is NP-complete. Then there exists a relation in $\langle \Gamma \rangle_{\nexists}$ which is an $S_{\mathbb{B}}$-extension.*

Proof. We provide a short sketch of the most important ideas. For the full proof the reader may consult Appendix B. Via Lemma 13 there exists an $S_{\mathbb{B}}$-extension $R \in \langle \Gamma \rangle$. It is not necessarily the case that $R \in \langle \Gamma \rangle_{\nexists}$, but it is possible to construct an $S_{\mathbb{B}}$-extension by gradually converting the pp-definition of R over Γ to a qfpp-definition. To do this, let $\mathrm{ar}(R) = n$ and assume e.g. that $R'(x_1, \ldots, x_n) \equiv \exists y. \varphi(x_1, \ldots, x_n, y)$, where $\exists y. \varphi(x_1, \ldots, x_n, y)$ is a pp-formula over Γ. Consider the relation $R'(x_1, \ldots, x_n, y) \equiv \varphi(x_1, \ldots, x_n, y)$. This relation is qfpp-definable over Γ, and if $|R'| > 3$ (and R' is not an $S_{\mathbb{B}}$-extension) one can prove that there either exists a unary constraint $E \in \Gamma$ such that $R''(x_1, \ldots, x_n, y) \equiv \varphi(x_1, \ldots, x_n, y) \wedge E(y)$ is an $S_{\mathbb{B}}$-extension, or that there exists $i \in \{1, \ldots, n\}$ and a relation $F \in \langle \Gamma \rangle_{\nexists}$ such that $R''(x_1, \ldots, x_i, \ldots, x_n, y, z_1, \ldots, z_{\mathrm{ar}(F)}) \equiv \varphi(x_1, \ldots, x_n, y) \wedge F(x_i, y, z_1, \ldots, z_{\mathrm{ar}(F)})$ defines an $S_{\mathbb{B}}$-extension. □

4.2 Properties of and Reductions between $S_{\mathbb{B}}$-Extensions

By Lemma 14, we can completely concentrate on $S_{\mathbb{B}}$-extensions. We will prove that $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\Gamma)$ for every ultraconservative Γ over D such that CSP(Γ) is NP-complete. To prove this, we begin in Section 4.2.1 by investigating properties of $S_{\mathbb{B}}$-extensions, which we use to simplify the total number of distinct cases we need to consider. With the help of these results we in Section 4.2.2 develop techniques in order to show that $\mathrm{CSP}(S_D) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$ for every $S_{\mathbb{B}}$-extension over D.

4.2.1 Saturated $S_{\mathbb{B}}$-Extensions

In this section we simplify the number of cases we need to consider in Section 4.2.2. First note that if $R = \{t_1, t_2, t_3\}$ over D is a relation with $\mathrm{ar}(R) > |D|^3$ then there exists i and j such that $(t_1[i], t_2[i], t_3[i]) = (t_1[j], t_2[j], t_3[j])$. We say that the jth argument is *redundant*, and it is possible to get rid of this by identifying the ith and jth argument with the qfpp-definition

$$R'(x_1, \ldots, x_i, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n) \equiv R(x_1, \ldots, x_i, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_n).$$

This procedure can be repeated until no redundant arguments exist, and we will therefore always implicitly assume that $\mathrm{ar}(R) \leq |D|^3$ and that R has no redundant arguments. If R is an n-ary $S_{\mathbb{B}}$-extension then the argument $i \in \{1, \ldots, n\}$ is said to be *1-choice*, or *constant*, if $|\mathrm{Proj}_i(R)| = 1$, *2-choice* if $|\mathrm{Proj}_i(R)| = 2$, and *3-choice* if $|\mathrm{Proj}_i(R)| = 3$.

Definition 15. *An n-ary $S_{\mathbb{B}}$-extension $R = \{t_1, t_2, t_3\}$ is said to be* saturated *if there for each $1 \leq i \leq n$ and every function $\tau : \{1, 2, 3\} \to \{1, 2, 3\}$, exists $1 \leq j \leq n$ such that $(t_{\tau(1)}[i], t_{\tau(2)}[i], t_{\tau(3)}[i]) = (t_1[j], t_2[j], t_3[j])$.*

Example 16. *The relation S_D is saturated for every D, but if we consider the relations R and R' defined by the matrices $\left(\begin{smallmatrix} 0&0&1&1&1&0&0&0&1&2 \\ 0&1&0&1&0&1&0&0&1&2 \\ 1&0&0&0&1&1&2&0&1&2 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 0&0&1&1&1&0&0&0&1&2 \\ 0&1&0&1&0&1&1&0&1&2 \\ 1&0&0&0&1&1&2&0&1&2 \end{smallmatrix}\right)$ then neither relation is saturated. First, R is not saturated since its matrix representation, for example, does not contain the column $(0, 2, 0)$. Second, R' is not saturated due to the 3-choice argument in position 7.*

We now prove that we without loss of generality may assume that an $S_{\mathbb{B}}$-extension is saturated.

Lemma 17. *Let R be an $S_{\mathbb{B}}$-extension. Then there exists a saturated $S_{\mathbb{B}}$-extension $R' \in \langle R \rangle_{\not\exists}$.*

Proof. We provide a short proof sketch illustrating the most important ideas. See Appendix B for a full proof. Let $n = \mathrm{ar}(R)$ and define R' such that $\mathrm{Proj}_{1,\ldots,n}(R') = R$, and then add the minimum number of arguments which makes R' saturated. Via Theorem 4 it follows that if $R' \notin \langle R \rangle_{\not\exists}$ then this can be witnessed by a partial function f preserving R but not R'. Therefore, there exists tuples $t'_1, t'_2, t'_3 \in R'$ such that $f(t'_1, t'_2, t'_3) \notin R'$, but since $\mathrm{Proj}_{1,\ldots,n}(R') = R$ and since R' is saturated, one can prove that there must exist tuples $t_1, t_2, t_3 \in R$ such that $f(t_1, t_2, t_3) \notin R$, contradicting the assumption that f preserves R. Hence, $R' \in \langle \Gamma \rangle_{\not\exists}$. □

Example 18. *If R is the relation from Example 16 then the saturated relation R' in $\langle R \rangle_{\not\exists}$ from Lemma 17 is given by $R' = \left(\begin{smallmatrix} 0&0&1&1&1&0&0&0&2&2&2&0&0&1&2 \\ 0&1&0&1&0&1&0&2&0&2&0&2&0&1&2 \\ 1&0&0&0&1&1&2&0&0&0&2&2&0&1&2 \end{smallmatrix}\right)$.*

4.2.2 Reductions Between $S_{\mathbb{B}}$-Extensions

The main result of this section (Theorem 23 and Theorem 24) show that $\mathsf{T}(\{S_D\}) = \mathsf{T}(\{S_D\} \cup 2^D) \leq \mathsf{T}(\Gamma)$ whenever Γ is an ultraconservative constraint language over D such that $\mathrm{CSP}(\Gamma)$ is NP-complete. The result is proven by a series of CV-reductions that we present in Lemmas 19–22. Due to space constraints, we only present the proof of Lemma 20 which illustrates several useful techniques, and the remaining proofs can be found in Appendix B. Before we begin, we note that if R is an $S_{\mathbb{B}}$-extension over D then $\{R\}$ is not necessarily a core. For a simple counterexample, $\{S_{\mathbb{B}}\}$ is not a core over $\{0, 1, 2\}$ since the endomorphism $e(0) = 0$, $e(1) = 1$, $e(2) = 0$, is not an automorphism. However, if R is an $S_{\mathbb{B}}$-extension and $E = \{d_1, \ldots, d_m\}$ the set $\bigcup_{1 \leq i \leq \mathrm{ar}(R)} \mathrm{Proj}_i(R)$, every endomorphism $e : E \to E$ of R must be an automorphism. Hence, Theorem 7 is applicable, and we conclude that $\mathrm{CSP}(\{R, R^{d_1}, \ldots, R^{d_m}\}) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$. When working with reductions between $S_{\mathbb{B}}$-extensions we may therefore freely make use of constant relations. Given an instance (V, C) of $\mathrm{CSP}(R)$, where R is an $S_{\mathbb{B}}$-extension, we say that a variable $x \in V$ occurring in a k-choice position in a constraint in C, $1 \leq k \leq 3$, is a *k-choice variable.*

Lemma 19. *Let R be a saturated $S_{\mathbb{B}}$-extension. Then there exists a CV-reduction f from $\mathrm{CSP}(R)$ to $\mathrm{CSP}(R)$ such that for every instance I of $\mathrm{CSP}(R)$, each variable in $f(I)$ occurs as a 3-choice variable in at most one constraint.*

Lemma 20. *Let R be a saturated $S_{\mathbb{B}}$-extension and let R' be R with one or more 3-choice arguments removed, such that R' is still saturated. Then $\mathrm{CSP}(R) \leq^{\mathrm{CV}} \mathrm{CSP}(R')$.*

Proof. Let $R = \{t_1, t_2, t_3\}$, $n = \mathrm{ar}(R)$, $n' = \mathrm{ar}(R')$, and assume that $\mathrm{Proj}_{1,\ldots,n'}(R) = R'$. Let $I = (V, C)$ be an instance of $\mathrm{CSP}(R)$. First apply Lemma 19 in order to obtain an instance $I_1 = (V_1, C_1)$ of $\mathrm{CSP}(R)$ such that each 3-choice variable only occurs in a 3-choice position in a single constraint. Assume there exists $x \in V_1$ and two distinct constraints $c, c' \in C_1$ such that x occurs in positions $i \in \{n'+1, \ldots, n\}$ in c and in a 1- or 2-choice position $j \in \{1, \ldots, n\}$ in c'. Let $S = \mathrm{Proj}_i(R) \cap \mathrm{Proj}_j(R)$, and note that $|S| \leq 2$. Assume first that $|S| = 2$, let $S = \{d_1, d_2\}$, and assume without loss of generality that $t_1[i] = t_1[j] = d_1$, $t_2[i] = t_2[j] = d_2$, and that $t_3[i] \neq t_3[j]$ (the other cases can be treated similarly). Since R is saturated there exists a 2-choice argument $i' \in \{1, \ldots, n\}$ such that $t_1[i'] = t_1[i] = t_1[j]$, $t_2[i'] = t_2[i] = t_2[j]$, and such that $t_3[i'] \neq t_3[i]$. Let y be the variable occurring in the i'th position of c. Create a fresh variable $\hat{x}$, replace x in position i with $\hat{x}$, and for each constraint where x occurs as a 1- or 2-choice variable, replace x with y. Repeat this procedure until every 3-choice variable occurring in position $n'+1, \ldots, n$ only occurs in a single constraint, and let $I_2 = (V_2, C_2)$ be the resulting instance. Assume there exists $x \in V_2$ and a constraint $c \in C_2$ such that x occurs as a 3-choice variable in position $i \in \{n'+1, \ldots, n\}$ and also in a distinct position $j \in \{1, \ldots, n\}$ in c. Let $L = \{t_r \mid 1 \leq r \leq 3, t_r[i] = t_r[j]\}$. Since R does not have any redundant arguments it must be the case that $|L| < 3$. If $|L| = 0$ then the instance is unsatisfiable, in which case we output an arbitrary unsatisfiable instance, and if $|L| = 1$ it is easy to see that any variable occurring in c can be assigned a fixed value, and the constraint may be removed. Therefore, assume that $|L| = 2$, and e.g. that $L = \{t_1, t_2\}$. Since R is saturated there exists a 2-choice argument $j' \in \{1, \ldots, n\}$ such that $t_1[j'] = t_2[j'] \neq t_3[j']$. Let y be the variable occurring in position j' in c and add the constraint $R^{t_1[j']}(y)$. Repeat this for every variable occurring in position $n'+1, \ldots, n$ in a constraint in C_2, and then replace each constraint $R(x_1, \ldots, x'_n, \ldots, x_n)$ by $R'(x_1, \ldots, x_n)$. Note that any variable $\hat{x}$ introduced in the previous step of this reduction is removed in this transformation. Hence, the reduction is a CV-reduction. ☐

Lemma 21. *Let R be an $S_{\mathbb{B}}$-extension and let R' be an $S_{\mathbb{B}}$-extension obtained by adding additional 2-choice arguments to R. Then $\mathrm{CSP}(R') \leq^{\mathrm{CV}} \mathrm{CSP}(R)$.*

Lemma 22. *Let R be a saturated $S_{\mathbb{B}}$-extension over D with 3-choice arguments. Then $\mathrm{CSP}(S_D) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$.*

We have thus proved the main result of this section.

Theorem 23. *Let D be a finite domain and let Γ be a finite, ultraconservative constraint language over D. If $\mathrm{CSP}(\Gamma)$ is NP-complete then $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\Gamma)$.*

Proof. We first observe that if R is an $S_{\mathbb{B}}$-extension over a finite domain D, then $\mathrm{CSP}(S_D) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$. By Lemma 17 we may assume that R is saturated. If R does not contain any 3-choice arguments we use Lemma 20 together with Lemma 21 and obtain a CV-reduction from $\mathrm{CSP}(S_D)$ to $\mathrm{CSP}(R)$. Hence, assume that R contains one or more 3-choice arguments. In this case we use

Lemma 22 and obtain a CV-reduction from $\mathrm{CSP}(S_D)$ to $\mathrm{CSP}(R)$. By Lemma 14 there exists an $S_{\mathbb{B}}$-extension $R \in \langle \Gamma \rangle_{\nexists}$, implying that $\mathrm{CSP}(R) \leq^{\mathrm{CV}} \mathrm{CSP}(\Gamma)$ via Theorem 6, and we know that $\mathrm{CSP}(S_D) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$. We conclude that $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\{R\}) \leq \mathsf{T}(\Gamma)$. ☐

Clearly, $\{S_D\}$ is not an ultraconservative constraint language but the complexity of $\mathrm{CSP}(S_D)$ does not change when we expand the language by adding all unary relations over D (the proof can be found in Appendix B).

Theorem 24. *Let D be a finite domain. Then* $\mathsf{T}(\{S_D\}) = \mathsf{T}(\{S_D\} \cup 2^D)$.

Thus, no NP-complete CSP over an ultraconservative constraint language over D is solvable strictly faster than $\mathrm{CSP}(S_D)$, and, in particular, $\mathsf{T}(\{S_{D'}\}) \leq \mathsf{T}(\{S_D\})$ whenever $D' \supseteq D$. This raises the question of whether $\mathsf{T}(S_D) = \mathsf{T}(S_{D'})$ for all $D, D' \supseteq \{0,1\}$, or if it is possible to find D and D' such that $\mathsf{T}(\{S_{D'}\}) < \mathsf{T}(\{S_D\})$. As the following theorem shows, this is indeed the case, unless $\mathsf{T}(\{S_D\}) = 0$ for every finite D and the ETH fails.

Theorem 25. $\inf\{\mathsf{T}(\{S_D\}) \mid D$ *finite and* $|D| \geq 2\} = 0$.

Proof. Let $D_k = \{0, \ldots, k-1\}$, $k \geq 5$. We will analyse a simple algorithm for $\mathrm{CSP}(S_{D_k})$. Let $I = (V, C)$ be an arbitrary instance of $\mathrm{CSP}(S_{D_k})$. Extend the instance with variables $Z = \{z_0, \ldots, z_{k-1}\}$ and the constraints $R^i(z_i)$, $0 \leq i \leq k-1$. Arbitrarily choose a constraint $c = S_{D_k}(x_1, \ldots, x_{k^3})$ and let $X = \{x_1, \ldots, x_{k^3}\}$. It is straightforward to verify that if a variable x appears in k^2+1 or more positions, then c cannot be satisfied. Thus, $|X| \geq k$. If $X \cap Z = \emptyset$, then we branch on the three tuples in S_{D_k} and in each branch at least k variables in $V \setminus Z$ will be given fixed values. If a variable, say x_i, is given the fixed value d, then we identify x_i with z_d. Thus, at least k variables in $V \setminus Z$ are removed. Assume to the contrary that $X \cap Z \neq \emptyset$. If a variable $z \in Z$ occurs in a 3-choice position, then the variables in $X \setminus Z$ can be assigned fixed values and no branching is needed. If no variable $z \in Z$ occurs in a 3-choice position, then there are $k(k-1)(k-2)$ 3-choice positions in S_{D_k} and they are all covered by variables in $V \setminus Z$. Thus, we perform three branches based on the tuples in S_{D_k}. Recall that a variable can occur in at most k^2 positions in the constraint c since c is otherwise not satisfiable. This implies that at least $\lfloor \frac{k(k-1)(k-2)}{k^2} \rfloor \geq 1$ variables in $V \setminus Z$ are given fixed values (and are removed from $V \setminus Z$) in each branch. When there are no S_{D_k} constraints left, we check whether the remaining set of unary constraints are satisfiable or not. It is straightforward to perform this test in polynomial time. A recursive equation that gives an upper bound on the time complexity of this algorithm is thus $T(1) = \mathit{poly}(||I||), T(n) = 3T(n - \lfloor \frac{k(k-1)(k-2)}{k^2} \rfloor) + \mathit{poly}(||I||)$ (where n denotes the number of variables and $||I||$ the number of bits required to represent I) so $T(n) \in O(3^{n \cdot \frac{k^2}{k(k-1)(k-2)}} \cdot \mathit{poly}(||I||))$. The function $\frac{k^2}{k(k-1)(k-2)}$ obviously tends to 0 with increasing k so the infimum of the set $\{\mathsf{T}(\{S_D\}) \mid D$ is finite and $|D| \geq 2\}$ is equal to 0. ☐

5 Concluding Remarks and Future Research

In this paper we have studied the time complexity of NP-complete CSPs. Assuming the algebraic CSP dichotomy conjecture, we have ruled out subexponential time algorithms for NP-complete, finite-domain CSPs, unless the ETH is false. This proof also extends to degree-bounded CSPs and many classes of CSPs over infinite domains. We then proceeded to study the time complexity of CSPs over ultraconservative constraint languages, and proved that no such NP-complete CSP is solvable strictly faster than $\mathsf{T}(\{S_D\})$. These results raise several directions for future research.

Structurally restricted CSPs and the ETH. Theorem 9 shows that the algebraic approach is viable for analysing the existence of subexponential algorithms for certain structurally restricted $\mathrm{CSP}(\Gamma)$ problems. An interesting continuation would be to try to determine which of the structurally restricted (but not constraint language restricted) CSPs investigated by De Haan et al. [17] could be used to prove similar results. For example, is it the case that CSP(Γ) is not solvable in subexponential time whenever CSP(Γ) is NP-complete and the primal treewidth of an instance is bounded by $\Omega(n)$, unless the ETH fails?

The CSP dichotomy conjecture. Several independent solutions to the algebraic CSP dichotomy conjecture have recently been announced [13, 30, 37]. If any of these proposed proofs is correct, it is tempting to extend Theorem 23 to constraint languages that are not necessarily ultraconservative or conservative. As a starting point, one could try to strengthen the results in Section 4.1, in order to prove that $\langle\Gamma\rangle_{\not\exists}$ contains an $S_{\mathbb{B}}$-extension whenever CSP(Γ) is NP-complete and Γ is conservative (but not ultraconservative).

Acknowledgements

We thank Hannes Uppman for several helpful discussions on the topic of this paper. The second author has received funding from the DFG-funded project "Homogene Strukturen, Bedingungserfüllungsprobleme, und topologische Klone" (Project number 622397). The third author is partially supported by the *National Graduate School in Computer Science* (CUGS), Sweden.

References

[1] L. Barto. Constraint satisfaction problem and universal algebra. *ACM SIGLOG News*, 1(2):14–24, October 2014.

[2] L. Barto and M. Pinsker. The algebraic dichotomy conjecture for infinite domain constraint satisfaction problems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2016)*, pages 615–622, New York, NY, USA, 2016. ACM.

[3] M. Behrisch, M. Hermann, S. Mengel, and G. Salzer. Give me another one! In *Proceedings of the 26th International Symposium on Algorithms and Computation (ISAAC-2015)*, pages 664–676, 2015.

[4] M. Behrisch, M. Hermann, S. Mengel, and G. Salzer. As close as it gets. In *Proceedings of the 10th International Workshop on Algorithms and Computation (WALCOM-2016)*, pages 222–235, 2016.

[5] 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.

[6] M. Bodirsky, P. Jonsson, and T. V. Pham. The complexity of phylogeny constraint satisfaction. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, pages 20:1–20:13, 2016.

[7] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):9:1–9:41, 2010.

[8] M. Bodirsky and M. Pinsker. Schaefer's theorem for graphs. *J. ACM*, 62(3):19:1–19:52, June 2015.

[9] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. I. *Cybernetics*, 5:243–252, 1969.

[10] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. II. *Cybernetics*, 5:531–539, 1969.

[11] E. Böhler, E. Hemaspaandra, S. Reith, and H. Vollmer. Equivalence and isomorphism for boolean constraint satisfaction. In *In Proceedings of the 16th International Workshop on*

Computer Science Logic (CSL-2002), pages 412–426, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[12] A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic*, 12(4):24:1–24:66, July 2011.

[13] A. Bulatov. A dichotomy theorem for nonuniform csps. *CoRR*, abs/1703.03021, 2017.

[14] A. Bulatov and A. Hedayaty. Counting problems and clones of functions. *Multiple-Valued Logic and Soft Computing*, 18(2):117–138, 2012.

[15] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, March 2005.

[16] N. Creignou, U. Egly, and J. Schmidt. Complexity classifications for logic-based argumentation. *ACM Transactions on Computational Logic (TOCL)*, 15(3):19:1–19:20, 2014.

[17] R. de Haan, I. A. Kanj, and S. Szeider. On the subexponential-time complexity of CSP. *Journal of Artificial Intelligence Research (JAIR)*, 52:203–234, 2015.

[18] T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[19] D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27(1):95–100, 1968.

[20] M. Grohe. The structure of tractable constraint satisfaction problems. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS 2006)*, pages 58–72, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[21] L. Ham. Gap theorems for robust satisfiability: Boolean CSPs and beyond. *To appear in Theoretical Computer Science*, 2017.

[22] T. Hertli. 3-SAT faster and simpler - unique-SAT bounds for PPSZ hold in general. *SIAM Journal on Computing*, 43(2):718–729, 2014.

[23] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.

[24] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530, 2001.

[25] P. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

[26] P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, July 1997.

[27] P. Jonsson, V. Lagerkvist, G. Nordh, and B. Zanuttini. Strong partial clones and the time complexity of SAT problems. *Journal of Computer and System Sciences*, 84:52 – 78, 2017.

[28] V. Lagerkvist, M. Wahlström, and B. Zanuttini. Bounded bases of strong partial clones. In *Proceedings of the 45th International Symposium on Multiple-Valued Logic (ISMVL-2015)*, pages 189–194, 2015.

[29] E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.

[30] A. Rafiey, J. Kinne, and T. Feder. Dichotomy for digraph homomorphism problems. *CoRR*, abs/1701.02409, 2017.

[31] B.A. Romov. The algebras of partial functions and their invariants. *Cybernetics*, 17(2):157–167, 1981.

[32] F. Rossi, P. van Beek, and T. Walsh, editors. *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*. Elsevier, 2006.

[33] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010.

[34] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory Of Computing (STOC-78)*, pages 216–226. ACM Press, 1978.

[35] H. Schnoor and I. Schnoor. Partial polymorphisms and constraint satisfaction problems. In N. Creignou, P. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 229–254. Springer Berlin Heidelberg, 2008.

[36] M. Wahlström. *Algorithms, measures and upper bounds for satisfiability and related problems.* PhD thesis, Linköping University, TCSLAB - Theoretical Computer Science Laboratory, The Institute of Technology, 2007.

[37] D. Zhuk. The proof of csp dichotomy conjecture. *CoRR*, abs/1704.01914, 2017.

Appendix

A Additional Proofs for Section 3

Theorem 9. *Assume that the ETH is true and let* Γ *be a finite constraint language over a domain* D *such that* Γ *pp-interprets* Γ^3_{SAT}*. Then* $\text{CSP}(\Gamma)$ *is not solvable in subexponential time, and if* Γ *efpp-defines* Eq_D *then there exists a constant* B*, depending only on* Γ*, such that* $\text{CSP}(\Gamma)$*-*B *is not solvable in subexponential time.*

Proof. Due to the assumption that Γ pp-interprets Γ^3_{SAT}, Γ can pp-interpret any Boolean Δ, as was pointed out in Section 2. In particular, Γ can pp-interpret the constraint language $\{R^{\neq\neq\neq}_{1/3}\}$ from Jonsson et al. [27], where $R^{\neq\neq\neq}_{1/3} = \text{Proj}_{1,\ldots,6}(R^{\neq\neq\neq01}_{1/3})$. It is known that $\text{SAT}(R^{\neq\neq\neq}_{1/3})$-2 is NP-complete and that if it is solvable in subexponential time, then the ETH is false [27]. Hence, we will prove the theorem by giving an LV-reduction from $\text{SAT}(R^{\neq\neq\neq}_{1/3})$-2 to $\text{CSP}(\Gamma)$, respectively to $\text{CSP}(\Gamma)$-B for some $B > 0$.

Let $F \subseteq D^d$ and $f : F \mapsto \mathbb{B}$ denote the parameters in the pp-interpretation of $\{R^{\neq\neq\neq}_{1/3}\}$. Note in particular that $d \in \mathbb{N}$ is a fixed constant. Let

$$f^{-1}(R^{\neq\neq\neq}_{1/3})(x_{1,1}, \ldots, x_{1,d}, \ldots, x_{6,1}, \ldots, x_{6,d}) \equiv \exists y_1, \ldots, y_{k_1}.\varphi_1(x_{1,1}, \ldots, x_{1,d}, \ldots, x_{6,1}, \ldots, x_{6,d}, y_1, \ldots, y_{k_1})$$

and

$$F(x_1, \ldots, x_d) \equiv \exists y_1, \ldots, y_{k_2}.\varphi_2(x_1, \ldots, x_d, z_1, \ldots, z_{k_2})$$

denote efpp-definitions of $f^{-1}(R^{\neq\neq\neq}_{1/3})$ and F over Γ if Eq_D is efpp-definable over Γ, and otherwise pp-definitions of $f^{-1}(R^{\neq\neq\neq}_{1/3})$ and F over Γ. Let L denote the maximum degree of any variable occurring in these pp-definitions, and note that L is a fixed constant depending only on Γ.

Let $I = (V, C)$ be an instance of $\text{SAT}(\{R^{\neq\neq\neq}_{1/3}\})$-2. Since each variable may occur in at most 2 constraints it follows that $|C| \leq 2|V|$. For each variable x_i introduce d fresh variables $x_{i,1}, \ldots, x_{i,d}$, k_2 fresh variables $z_{i,1}, \ldots, z_{i,k_2}$, and introduce the constraint

$$\varphi_2(x_{i,1}, \ldots, x_{i,d}, z_{i,1}, \ldots, z_{i,k_2}).$$

For each constraint $C_i = R^{\neq\neq\neq}_{1/3}(x_i, y_i, z_i, x'_i, y'_i, z'_i)$ introduce k_1 fresh variables $w_{i,1}, \ldots, w_{i,k_1}$ and replace C_i by

$$\varphi_1(x_{i,1}, \ldots, x_{i,d}, y_{i,1}, \ldots, y_{i,d}, z_{i,1}, \ldots, z_{i,d}, x'_{i,1}, \ldots, x'_{i,d}, y'_{i,1}, \ldots, y'_{i,d}, z'_{i,1}, z'_{i,d}, w_{i,1}, \ldots, w_{i,k_1}).$$

If Γ cannot efpp-define Eq_D then we in addition identify any two variables occurring in equality constraints. Let $I' = (V', C')$ denote the resulting instance of CSP(Γ). Clearly, I' can be constructed in polynomial time. We begin by proving that I' has a solution if and only if I has a solution. Let $s' : V' \to D$ be a solution to I'. Recall that every variable x_i in V corresponds to a 'block' of variables $x_{i,1}, \ldots, x_{i,d}$ in V'. Now, consider a subset X of constraints corresponding to

$$\varphi_1(x_{i,1}, \ldots, x_{i,d}, y_{i,1}, \ldots, y_{i,d}, z_{i,1}, \ldots, z_{i,d}, x'_{i,1}, \ldots, x'_{i,d}, y'_{i,1}, \ldots, y'_{i,d}, z'_{i,1}, z'_{i,d}, w_{i,1}, \ldots, w_{i,k_1}).$$

Consider one block of variables $x_{i,1}, \ldots, x_{i,d}$. We know that $(s'(x_{i,1}), \ldots, s'(x_{i,d})) \in F$ due to the constraint $F(x_{i,1}, \ldots, x_{i,d})$ and that s' satisfies X. Since X and the block of variables are arbitrarily chosen, we conclude that the function $s : V \to \mathbb{B}$ defined by

$$s(x) = f(s'(x_1), \ldots, s'(x_d))$$

is a solution to I.

Assume instead that $s : V \to \mathbb{B}$ is a solution to I. Arbitrarily choose $t_0, t_1 \in F$ such that $f(t_0) = 0$ and $f(t_1) = 1$. For each variable $x_i \in V$, let $x_{i,1}, \ldots, x_{i,d}$ denote the corresponding block of variables in V', and let $\hat{V}$ denote the set of all these variables. Define the function $\hat{s} : \hat{V} \to F$ such that $\hat{s}(x_{i,j}) = t_0[j]$ if $s(x_i) = 0$ and $\hat{s}(x_{i,j}) = t_1[j]$ otherwise. The function $\hat{s}$ satisifes every constraint $F(x_{i,1}, \ldots, x_{i,d})$ by definition. Consider a subset X of constraints corresponding to

$$\varphi_1(x_{i,1}, \ldots, x_{i,d}, y_{i,1}, \ldots, y_{i,d}, z_{i,1}, \ldots, z_{i,d}, x'_{i,1}, \ldots, x'_{i,d}, y'_{i,1}, \ldots, y'_{i,d}, z'_{i,1}, z'_{i,d}, w_{i,1}, \ldots, w_{i,k_1}).$$

Recall that φ_1 is a pp-definition of $f^{-1}(R^{\neq\neq\neq}_{1/3})$. Thus, the variables $w_{i,1}, \ldots, w_{i,k_1}$ can be assigned values that in combination with the values provided by $\hat{s}$ satisfies φ_1 and, consequently, X. This implies that there is a solution to I'.

We continue by analysing this reduction. First, observe that if Γ can efpp-define Eq_D then the maximum degree of any variable is $3L$. This implies that I' is in fact an instance of CSP(Γ)-$3L$. Second, note that $|C| \leq 2|V|$, and that we for every constraint in C introduce k_1 fresh variables. This implies that $|V'| \leq |V|d + 2|V|k_1 + k_2$, and, since k_1, k_2 and d are fixed constants, there exists a constant K such that $|V'| = K|V| + O(1)$. Since this reduction is an LV-reduction from SAT($R^{\neq\neq\neq}_{1/3}$)-2 to CSP(Γ)-$3L$ (or to CSP(Γ) if Γ cannot efpp-define Eq_D), it follows that SAT($R^{\neq\neq\neq}_{1/3}$)-2 is solvable in subexponential time if CSP(Γ)-$3L$ (or CSP(Γ)) is solvable in subexponential time. $\square$

B Additional Proofs for Section 4

We will need the following lemma before we can present the proof for Lemma 14.

Lemma 26. *Let Γ be an ultraconservative language over a finite domain D and let $R \in \langle\Gamma\rangle$ be an n-ary relation such that $|R| = 2$. Then there exists $R' \in \langle\Gamma\rangle_{\nexists}$ such that (1) $|R'| = 2$ and (2) $\mathrm{Proj}_{1,\ldots,n}(R') = R$.*

Proof. Let $R(x_1, \ldots, x_n) \equiv \exists y_1, y_2, \ldots, y_m.\varphi(x_1, \ldots, x_n, y_1, y_2, \ldots, y_m)$ denote a pp-definition of R over Γ, and let $R = \{t_1, t_2\}$. We will show that it is possible to remove the existentially quantified arguments $y_1, y_2, \ldots, y_m$ in this pp-definition by gradually adding new arguments to R. First consider the relation $R_1(x_1, \ldots, x_n, y_1) \equiv \exists y_2 \ldots, y_m.\varphi(x_1, \ldots, x_n, y_1, y_2, \ldots, y_m)$. If $|R_1| = 2$ then we move on with the remaining arguments, so instead assume that $|R_1| > 2$. Now note that each tuple $t \in R_1$ in a natural way can be associated with either $t_1 \in R_1$ or $t_2 \in R_2$, depending on whether $t = t_1^\frown t'$ or $t = t_2^\frown t'$. Hence, let $S_1 = \{t[n+1] \mid t \in R_1, t_1^\frown t' = t\}$, and $S_2 = \{t[n+1] \mid t \in R_1, t_2^\frown t' = t\}$. In other words S_1 is the set of values taken by y_1 in the tuples corresponding to t_1, and S_2 the values taken by y_1 in the tuples corresponding to t_2. We consider two cases.

Case 1: $S_1 \cap S_2 = \emptyset$. Arbitrarily choose $d_1 \in S_1$ and $d_2 \in S_2$. Construct the relation $R_1'(x_1, \ldots, x_n, y_1) \equiv R_1(x_1, \ldots, x_n, y_1) \wedge \{(d_1), (d_2)\}(y_1)$, and note that $\{(d_1), (d_2)\} \in \Gamma$ since Γ is ultraconservative. We see that $R_1' = \{s_1^\frown(d_1), s_2^\frown(d_2)\}$.

Case 2: $S_1 \cap S_2 \neq \emptyset$. Arbitrarily choose $d \in S_1 \cap S_2$ and construct the relation $R_1'(x_1, \ldots, x_n, y_1) \equiv R_1(x_1, \ldots, x_n, y_1) \wedge R^d(y_1)$. We see that $R_1' = \{s_1^\frown(d), s_2^\frown(d)\}$. Note that we cannot choose elements as in Case 1 since if (for instance) one element is inside $S_1 \cap S_2$ and one element is outside $S_1 \cap S_2$, then the resulting relation will contain three tuples.

If we repeat this procedure for the remaining arguments $y_2, \ldots, y_m$ we will obtain a relation R' which is qfpp-definable over Γ such that $|R'| = 2$ and $\mathrm{Proj}_{1,\ldots,n}(R') = R$. □

Lemma 14. *Let Γ be an ultraconservative constraint language over a finite domain D such that* $\mathrm{CSP}(\Gamma)$ *is NP-complete. Then there exists a relation in $\langle\Gamma\rangle_{\nexists}$ which is an $S_{\mathbb{B}}$-extension.*

Proof. By Lemma 13 there exists a relation $R \in \langle\Gamma\rangle$ which is an $S_{\mathbb{B}}$-extension. Let

$$R(x_1, \ldots, x_n) \equiv \exists y_1, y_2, \ldots, y_m.\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

denote its pp-definition over Γ. Using this pp-definition we will show that Γ can qfpp-define an $S_{\mathbb{B}}$-extension by gradually removing each existentially quantified variable. First consider the relation $R_1(x_1, \ldots, x_n, y_1) \equiv \exists y_2, \ldots, y_m.\varphi(x_1, \ldots, x_n, y_1, y_2, \ldots, y_m)$. Assume that $|R_1| > 3$, i.e., that R_1 is not an $S_{\mathbb{B}}$-extension. Let $R = \{t_1, t_2, t_3\}$ and for each $1 \leq i \leq 3$ let $S_i = \{t[n+1] \mid t \in R_1, t_i^\frown t' = t\}$, $1 \leq i \leq 3$. In other words S_i contains the possible values taken by the argument y_1 in the tuples of R_1 corresponding to $t_i \in R$. There are now a few cases to consider depending on the sets S_1, S_2, S_3:

1. $|S_1 \cup S_2 \cup S_3| = 1$,
2. $|S_1 \cup S_2 \cup S_3| = 2$, and
3. $|S_1 \cup S_2 \cup S_3| \geq 3$,

The first case implies that the $(n+1)$th argument of R_1 is constant and that R_1 is already an $S_{\mathbb{B}}$-extension. In the third case, first choose $d_1 \in S_1$. If $d_1 \in S_2$ then let $d_2 = s_1$, otherwise choose an arbitrary value in S_2 distinct from d_1. Last, if $d_1 \in S_3$ or $d_2 \in S_3$ then let $d_3 = d_1$ or $d_3 = d_2$; otherwise choose an arbitrary value not occurring in $S_1 \cup S_2$. Note that this is possible since we assumed that $|S_1 \cup S_2 \cup S_3| \geq 3$, which implies that $S_1 \cup S_2 \cup S_3$ contains at least three distinct values. Let E be the unary relation $\{(d_1), (d_2), (d_3)\}$. It is then easy to see (by basically reasoning in the same way as in the proof of Lemma 26) that $\exists y_2, \ldots, y_m.E(y_1) \wedge \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ defines an $S_{\mathbb{B}}$-extension.

Now assume that $|S_1 \cup S_2 \cup S_3| = 2$ and let $\{d_1, d_2\} = S_1 \cup S_2 \cup S_3$. Up to symmetry, we then have the following possible cases:

1. $S_1 = S_2 = S_3 = \{d_1, d_2\}$,
2. $S_1 = S_2 = \{d_1, d_2\}$, $S_3 = \{d_1\}$, or
3. $S_1 = \{d_1\}$, $S_2 = \{d_2\}$, $S_3 = \{d_1, d_2\}$.

The first two cases are easy to handle in a similar way to the case when $|S_1 \cup S_2 \cup S_3| \geq 3$; in both cases, choose the element d_1. This leaves only the case when $S_1 = \{d_1\}$, $S_2 = \{d_2\}$ and that $S_3 = \{d_1, d_2\}$. Since R is an $S_{\mathbb{B}}$-extension there exists $a, b \in D$, $a \neq b$, and indices i_1, i_2, i_3 such that $(t_1[i_1], t_2[i_1], t_3[i_1]) = (b, b, a)$, $(t_1[i_2], t_2[i_2], t_3[i_2]) = (b, a, b)$, and $(t_1[i_3], t_2[i_3], t_3[i_3]) = (a, b, b)$. Define the binary relation F such that

$$F(x, y_1) \equiv \exists x_1, \ldots x_{i_3-1}, x_{i_3+1}, \ldots, x_n.R_1(x_1, \ldots, x_{i_3-1}, x, x_{i_3+1}, \ldots, x_n, y_1) \wedge R^b(x_{i_1}).$$

We claim that $F = \{(a, d_1), (b, d_2)\}$. To see this, observe that the constraint $R^b(x_{i_1})$ rules out the tuple t_3. This implies that if variable x_{i_3} has value a, then the variable y_1 must have value d_1 and if the variable x_{i_3} has value b, then the variable y_1 must have value d_2.

From this observation and Lemma 26, it follows that Γ can qfpp-define a relation F' such that $|F'| = 2$ and such that $\mathrm{Proj}_{1,2}(F') = F$. Let $k + 2$ denote the arity of F' and define a relation

$$R_1'(x_1, \ldots, x_{i_3}, \ldots, x_n, y_1, z_1, \ldots, z_k) \equiv R_1(x_1, \ldots, x_{i_3}, \ldots, x_n, y_1) \wedge F'(x_{i_3}, y_1, z_1, \ldots, z_k).$$

We claim that R_1' is an $S_{\mathbb{B}}$-extension. There are three possible ways of simultaneously choosing variables $x_{i_1}, x_{i_2}, x_{i_3}$. Let us consider the assignment $(x_{i_1}, x_{i_2}, x_{i_3}) = (b, b, a)$. This particular choice gives all variables $x_1, \ldots, x_n$ fixed values (via the constraint $R_1(x_1, \ldots, x_{i_3}, \ldots, x_n, y_1)$). Furthermore, y_1 is assigned the value d_2 (via the constraint $F'(x_{i_3}, y_1, z_1, \ldots, z_k)$) and the variables $z_1, \ldots, z_k$ are given fixed values (since there is only one tuple in F' that allows y_1 to have the value d_2). Thus, there is only one tuple in R_1' that allows $(x_{i_1}, x_{i_2}, x_{i_3}) = (b, b, a)$. The two other cases can be verified similarly and we conclude that $|R_1'| = 3$.

Finally, we see that there are $m - 1$ existentially quantified variables in the definition of R_1' since F' can be qfpp-defined. By repeating the procedure outline above for the remaining arguments we will obtain an $S_{\mathbb{B}}$-extension which is qfpp-definable over Γ. This concludes the proof. $\square$

Before the proof of Lemma 17 we will need the following result from Lagerkvist et al. [28, Lemma 2], restated in slightly simpler terminology.

Lemma 27. *Let R be a relation with m tuples. If $f \notin \mathrm{pPol}(R)$, where f has arity $n > m$, there exists g of arity $n' \leq m$ such that $g \notin \mathrm{pPol}(R)$ and g can be obtained from f by identifying arguments.*

For a k-ary relation R and tuples $t_1, \ldots, t_n \in R$ we write $SetCols(t_1, \ldots, t_n)$ for the set $\{(t_1[1], \ldots, t_n[1]), \ldots, (t_n[k], \ldots, t_n[k])\}$.

Lemma 17. *Let R be an $S_{\mathbb{B}}$-extension. Then there exists a saturated $S_{\mathbb{B}}$-extension $R' \in \langle R \rangle_{\nexists}$.*

Proof. Let $R = \{t_1, t_2, t_3\}$ and let n denote the arity of R. For each $1 \leq i \leq n$ and each function $\tau : \{1, 2, 3\} \to \{1, 2, 3\}$ add a fresh argument taking the values $t_{\tau(1)}[i]$, $t_{\tau(2)}[i]$, $t_{\tau(3)}[i]$. Let R' be the resulting relation and let $R' = \{t'_1, t'_2, t'_3\}$ such that $\text{Proj}_{1,\ldots,n}(t'_i) = t_i$. By construction, R' is a saturated $S_{\mathbb{B}}$-extension, but it remains to prove that $R' \in \langle R \rangle_{\nexists}$. Hence, assume with the aim of reaching a contradiction, that $R' \notin \langle R \rangle_{\nexists}$. Due to the Galois connection in Theorem 4 this implies that $\text{pPol}(R) \not\subseteq \text{pPol}(R')$. Hence, there exists a partial function f preserving R but which does not preserve R', and due to Lemma 27 we may without loss of generality assume that f has arity at most 3. We omit the cases when $\text{ar}(f) \leq 2$ since they are similar, and therefore assume that $f(t'_{\rho(1)}, t'_{\rho(2)}, t'_{\rho(3)}) = t' \notin R'$ for a permutation ρ on $\{1, 2, 3\}$. Note that since $\text{Proj}_{1,\ldots,n}(R') = R$ it must hold that $SetCols(t_{\rho(1)}, t_{\rho(2)}, t_{\rho(3)}) \subseteq SetCols(t'_{\rho(1)}, t'_{\rho(2)}, t'_{\rho(3)})$. Hence, $f(t_{\rho(1)}, t_{\rho(2)}, t_{\rho(3)})$ must be defined, and furthermore $f(t_{\rho(1)}, t_{\rho(2)}, t_{\rho(3)}) \in R$ since we assumed that f preserves R. Assume without loss of generality that $f(t_{\rho(1)}, t_{\rho(2)}, t_{\rho(3)}) = t_{\rho(1)}$, i.e., f restricted to the tuples $t_{\rho(1)}, t_{\rho(2)}, t_{\rho(3)}$ is a projection on the first argument. Since f when applied to $t'_{\rho(1)}, t'_{\rho(2)}, t'_{\rho(3)}$ by assumption is *not* a projection, there exists at least one index $j \in \{n + 1, \ldots, \text{ar}(R')\}$ such that $f(t'_{\rho(1)}[j], t'_{\rho(2)}[j], t'_{\rho(3)}[j]) \neq t'_{\rho(1)}[j]$. Due to the construction of R', there exists $i \in \{1, \ldots, n\}$ and a function $\tau' : \{1, 2, 3\} \to \{1, 2, 3\}$ such that

$$(t_{\tau'(1)}[i], t_{\tau'(2)}[i], t_{\tau'(3)}[i]) = (t'_{\tau(1)}[j], t'_{\tau(2)}[j], t'_{\tau(3)}[j]).$$

In other words it is possible to order the tuples from R in such a way that the values enumerated by these tuples in position i is exactly equal to $(t'_{\tau(1)}[j], t'_{\tau(2)}[j], t'_{\tau(3)}[j])$, where f is not a projection. It follows that $SetCols(t_{\tau'(1)}, t_{\tau'(2)}, t_{\tau'(3)}) \subseteq SetCols(t'_{\tau(1)}, t'_{\tau(2)}, t'_{\tau(3)}) \subseteq \text{dom}(f)$ (since R' is saturated) and therefore also that $f(t_{\tau'(1)}, t_{\tau'(2)}, t_{\tau'(3)}) \notin R$ (since f is not a projection on these tuples). This contradicts the assumption that $f \in \text{pPol}(R)$, and it must therefore be the case that $R' \in \langle R \rangle_{\nexists}$. $\square$

Lemma 19. *Let R be a saturated $S_{\mathbb{B}}$-extension. Then there exists a CV-reduction f from* $\text{CSP}(R)$ *to* $\text{CSP}(R)$ *such that for every instance I of* $\text{CSP}(R)$*, each variable in $f(I)$ occurs as a 3-choice variable in at most one constraint.*

Proof. Let n denote the arity of R and let $\{t_1, t_2, t_3\} = R$. Let $I = (V, C)$ be an instance of $\text{CSP}(R)$. We will create an instance $I' = (V', C')$ of $\text{CSP}(R)$ such that if $x \in V'$ is a 3-choice variable in a constraint then x does not occur as a 3-choice variable in any other constraint. Hence, let $x \in V$ be a 3-choice variable occurring in a constraint $c = R(x_1, \ldots, x_n)$ in position i_1. Assume that x also appears as a 3-choice variable in a constraint $c' = R(x'_1, \ldots, x'_n)$, distinct from c, in position i_2. Let $S = (t_1[i_1], t_2[i_1], t_3[i_1])$ and $S' = (t_1[i_2], t_2[i_2], t_3[i_2])$.

Assume first that $\text{Proj}_{i_1}(R) = \text{Proj}_{i_2}(R)$. Define the function τ such that for each $1 \leq i \leq 3$, $\tau(S[i]) = j$ if and only if $t_j[i_2] = S[i]$ where $1 \leq j \leq 3$. Using the function τ we then define the permutation $\rho : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $\rho(i) = j$ if and only if $(t_1[i], t_2[i], t_3[i]) = (t_{\tau(1)}[j], t_{\tau(2)}[j], t_{\tau(3)}[j])$. This is indeed a well-defined permutation over $\{1, \ldots, n\}$ since R is saturated. Last, identify each variable $x'_{\tau(i)}$ occurring in c' with the variable x_i in c, and remove the constraint c'.

Second, assume that $|\mathrm{Proj}_{i_1}(R) \cap \mathrm{Proj}_{i_2}(R)| = 2$, and let $\mathrm{Proj}_{i_1}(R) \cap \mathrm{Proj}_{i_2}(R) = \{d, d'\}$. Assume without loss of generality that $t_1[i_1] = d$, $t_2[i_1] = d'$, and that $t_3[i_1] \notin \{d, d'\}$. Choose $i \in \{1, \ldots, n\}$, distinct from both i_1 and i_2, such that $t_1[i] = t_1[i_1]$, $t_2[i] = t_2[i_1]$, and $t_3[i] \neq t_3[i_1]$. Such an i must exist since R is saturated. Then identify x with x_i. Define the function τ such that for $1 \leq i \leq 2$, $\tau(S[i]) = j$ if and only if $t_j[i_2] = S[i]$. Using the function τ we then define the permutation $\rho : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $\rho(i) = j$ if and only if $(t_1[i], t_2[i]) = (t_{\tau(1)}[j], t_{\tau(2)}[j])$. Clearly, τ is a well-defined permutation over $\{1, \ldots, n\}$ since R is saturated. Last, identify each variable $x'_{\tau(i)}$ occurring in c' with the variable x_i in c, and remove the constraint c'. The case when $|\mathrm{Proj}_{i_1}(R) \cap \mathrm{Proj}_{i_2}(R)| = 1$, i.e., when x is assigned the same value in any satisfying assignment, is very similar.

Each time this procedure is performed, at least one constraint is removed. Thus, we let I' denote the fixpoint that we will reach in at mots $|C|$ iterations. It is not difficult to verify that I is satisfiable if and only if I' is satisfiable. Furthermore, $|V'| \leq |V|$ and the reduction can be computed in polynomial time. We have thus showed that the reduction is a CV-reduction and therefore proved the lemma. □

Lemma 21. *Let R be an $S_{\mathbb{B}}$-extension and let R' be an $S_{\mathbb{B}}$-extension obtained by adding additional 2-choice arguments to R. Then* $\mathrm{CSP}(R') \leq^{\mathrm{CV}} \mathrm{CSP}(R)$.

Proof. Let $n = \mathrm{ar}(R)$, $n' = \mathrm{ar}(R')$, and $R' = \{t'_1, t'_2, t'_3\}$. By the statement of the lemma we may assume that $\mathrm{Proj}_{1,\ldots,n}(R') = R$, and that $|\mathrm{Proj}_i(R')| = 2$ for every $n' < i \leq n$. We will furthermore assume that $\mathrm{Proj}_i(R')$ for every $n' < i \leq n$ is distinct from $\mathrm{Proj}_j(R')$ for every $1 \leq j \leq n$. To simplify the proof we also assume that $\mathrm{Proj}_{1,\ldots,8}(R') = S_{\mathbb{B}}$. Let $I = (V, C)$ be an instance of $\mathrm{CSP}(R')$. Let x be a variable that appears in two distinct constraints $c_1, c_2 \in C$. Assume that x occurs at position $n + 1 \leq i \leq n'$ in c_1 and at position $1 \leq j \leq n'$ in c_2. We consider a number of cases based on the cardinality of $S = \mathrm{Proj}_i(R') \cap \mathrm{Proj}_j(R')$.

- $|S| = 3$. This is not possible since $|\mathrm{Proj}_i(R')| = 2$.
- $|S| = 2$. Assume that $S = \{a, b\}$ and $\mathrm{Proj}_j(R') = \{a, b, d\}$ (where b, d are not necessarily distinct). Define $f : \{a, b\} \to \{0, 1\}$ such that $f(a) = 0$ and $f(b) = 1$ and $g : \{a, b, d\} \to \{0, 1\}$ such that $g(a) = 0$ and $g(x) = 1$ if $x \neq a$. It follows that there exist indices $l, m \in \{1, \ldots, 6\}$ such that $f(t'_r[i]) = t'_r[l]$ and $g(t'_r[j]) = t'_r[m]$ when $r \in \{1, 2, 3\}$. If $b \neq d$, then we need ensure that x is never assigned d in any satisfying assignment to the resulting instance. For simplicity, assume that $t'_1[j] = d$. Then there exists $p \in \{1, \ldots, 6\}$ such that $t'_1[p] = 1, t'_2[p] = 0, t'_3[p] = 0$. Let w be the variable at position p in c_1, and add the unary relation $R^0(w)$. Now, let y be the variable at position l in c_1 and let z be the variable at position m in c_2. The variable x implies that y, z will always be assigned the same value by a solution to I. Hence, we identify z with y, introduce a fresh variable $\hat{x}$, and replace x at the ith position of c_1 with $\hat{x}$.

- $|S| = 1$. Assume $S = \{a\}$, $\mathrm{Proj}_i(R') = \{a, b\}$ (where a, b are distinct elements), and $\mathrm{Proj}_j(R') = \{a, d, d'\}$ (where a, d, d' are not necessarily distinct). Define $f : \{a, b\} \to \{0, 1\}$ such that $f(a) = 0$ and $f(b) = 1$, and $g : \{a, d, d'\} \to \{0, 1\}$ such that $g(a) = 0$ and $g(x) = 1$ if $x \neq a$. It is not hard to see that there exists $l, m \in \{1, \ldots, 8\}$ such that $f(t'_r[i]) = t'_r[l]$ and $g(t'_r[j]) = t'_r[m]$ when $r \in \{1, 2, 3\}$. Let y be the variable at position l in c_1 and z be the variable at position m in c_2. Add the unary relations $R^0(y)$ and $R^0(z)$, introduce a new variable $\hat{x}$, and replace x at the ith position of c_1 with $\hat{x}$.
- $|S| = 0$. This implies I_1 is unsatisfiable, and we simply output an arbitrary unsatisfiable instance.

By repeating the procedure above until a fixpoint is reached, we will obtain an instance $I_1 = (V_1, C_1)$ such that if $x \in V_1$ and if x appears in a constraint $c \in C_1$ at position $n + 1, \ldots, n'$, then it does not appear in any other constraint. However, it is still possible that $x \in V_1$ appear more than once in a single constraint $c \in C_1$ where (at least) one of the occurrences of x is at position $n + 1, \ldots, n'$. Therefore, assume that x appears in positions i and j in $c \in C_1$ where $i \in \{n + 1, \ldots n'\}$ and $j \in \{1, \ldots n'\}$. Let $L \subseteq \{1, 2, 3\}$ denote the set $\{l \mid t'_l[i] = t'_l[j]\}$.

- $|L| = 3$. This is not possible since there are no redundant arguments in the relation R'.
- $|L| = 2$. Assume (without loss of generality) that $t'_1[i] = t'_1[j]$, $t'_2[i] = t'_2[j]$, and $t'_3[i] \neq t'_3[j]$. Pick $k \in \{1, \ldots, 8\}$ such that $t'_1[k] = t'_2[k] \neq t'_3[k]$. Let y be the variable that appear in the kth position in c. Add a unary constraint $R^{t'_1[k]}(y)$, introduce a fresh variable $\hat{x}$, and replace the x at position i in c with $\hat{x}$.
- $|L| = 1$. Without loss of generality we can assume that $t'_1[i] = t'_1[j]$. For each variable y occurring in the lth position in c add the unary constraint $R^{t'_1[l]}(y)$, and then remove the constraint c.
- $|L| = 0$. This implies that I_1 is unsatisfiable, and we simply output an arbitrary unsatisfiable instance.

Repeat the procedure above until a fixpoint is reached and let $I_2 = (V_2, C_2)$ be the resulting instance. Observe that a variable x that occurs in a constraint at position $n + 1, \ldots, n'$ only occur in a single constraint and in a unique position. Finally, let $I_3 = (V_3, C_3)$ be the instance of $\mathrm{CSP}(R)$ obtained by replacing each constraint $R'(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n'}) \in C_2$ by $R(x_1, \ldots, x_n)$. Note that every fresh variable $\hat{x}$ that were introduced in the previous steps are removed in the conversion of I_2 into I_3. This shows that the reduction is indeed a CV-reduction. □

Lemma 22. *Let R be a saturated $S_{\mathbb{B}}$-extension over D with 3-choice arguments. Then* $\mathrm{CSP}(S_D) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$.

Proof. Let $n = \mathrm{ar}(R)$. Choose three distinct values $d_1, d_2, d_3 \in D$ such that there does not exist any i such that $\mathrm{Proj}_i(R) = \{d_1, d_2, d_3\}$. If no such i exists then $\langle R \rangle_{\not\exists} = \langle S_D \rangle_{\not\exists}$, and we are done. First, construct the relation S such that $\mathrm{Proj}_{1,\ldots,n}(S) = R$, $\mathrm{Proj}_{n+1}(S) = \{d_1, d_2, d_3\}$, and then add the minimum number of arguments to make S saturated. Second, let S' be the relation obtained from S by projecting away every argument i of the form $\mathrm{Proj}_i(S) = \{d_1, d_2, d_3\}$. In other words, S' is equivalent to R, except that it potentially contains more 1-choice and 2-choice arguments. Note that S' is saturated. Via Lemma 21 it then follows that $\mathrm{CSP}(S') \leq^{\mathrm{CV}} \mathrm{CSP}(R)$, and an application of Lemma 20 gives the desired result that $\mathrm{CSP}(S) \leq^{\mathrm{CV}} \mathrm{CSP}(S') \leq^{\mathrm{CV}} \mathrm{CSP}(R)$. This procedure can be repeated arbitrarily many times, which implies that $\mathrm{CSP}(S_D) \leq^{\mathrm{CV}} \mathrm{CSP}(R)$. $\square$

Theorem 24. *Let D be a finite domain. Then* $\mathsf{T}(\{S_D\}) = \mathsf{T}(\{S_D\} \cup 2^D)$.

Proof. $\mathsf{T}(\{S_D\}) \leq \mathsf{T}(\{S_D\} \cup 2^D)$ holds trivially. To prove $\mathsf{T}(\{S_D\} \cup 2^D) \leq \mathsf{T}(\{S_D\})$ we show that $\mathrm{CSP}(\{S_D\} \cup 2^D) \leq^{\mathrm{CV}} \mathrm{CSP}(S_D)$. Since we have already seen many reductions akin to this we only provide a sketch. Let (V, C) be an instance of $\mathrm{CSP}(\{S_D\} \cup 2^D)$. Assume $x \in V$ appears in a unary constraint $E(x) \in C$. If x also appears in another unary constraint $E'(x)$ then these two constraints can be replaced by $E \cap E'(x)$; hence, we may assume that each variable occurs in at most one unary constraint. If x does not occur in any other constraint, then we first check if $E = \emptyset$. If this is the case, the instance is unsatisfiable and we abort the procedure, and otherwise we simply remove the constraint $E(x)$. Now assume that x also appears in the ith position in a constraint $S_D(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_{\mathrm{ar}(S_D)})$. If $E \cap \mathrm{Proj}_i(S_D) = \emptyset$ then the instance is unsatisfiable, and if $E = \mathrm{Proj}_i(S_D)$ then we may safely remove the constraint E. Therefore assume that either $|\mathrm{Proj}_i(S_D) \cap E| = 1$ or that $|\mathrm{Proj}_i(S_D) \cap E| = 2$. The first of these cases is easy to handle since it implies that x is forced a constant value in any satisfying assignment. The second case implies that x appears in a 3-choice position, i.e., $\mathrm{Proj}_i(S_D) = \{d_1, d_2, d_3\}$, for three distinct values d_1, d_2, and d_3. Assume that $E = \{(d_1), (d_2)\}$, and let $t \in S_D$ be the tuple satisfying $t[i] = d_3$. Let $\{s, u\} = S_D \setminus \{t\}$ and choose j such that $s[j] = s[i]$, $u[i] = u[j]$, and $t[j] \in \{s[j], u[j]\}$. Then identify x with the variable x_j throughout the instance. If we repeat this procedure for the remaining constraints containing x, remove the constraint $U(x)$, and then continue with all remaining unary constraints, we will obtain an instance of $\mathrm{CSP}(S_D)$ which is satisfiable if and only if (V, C) is satisfiable. $\square$