Electronic Colloquium on Computational Complexity, Report No. 100 (2004)

The Complexity of Satisfiability Problems: Refining Schaefer's Theorem

Eric Allender* Michael Bauland† Neil Immerman‡
Henning Schnoor† Heribert Vollmer†

November 18, 2004

Abstract

Schaefer proved in 1978 that the Boolean constraint satisfaction problem for a given constraint language is either in P or is NP-complete, and identified all tractable cases. Schaefer's dichotomy theorem actually shows that there are at most two constraint satisfaction problems, up to polynomial-time isomorphism (and these isomorphism types are distinct if and only if $\mathrm{P} \neq \mathrm{NP}$). We show that if one considers logspace isomorphisms, then there are exactly five isomorphism types (assuming that the complexity classes $\mathrm{NP}, \mathrm{P}, \mathrm{NL}, \oplus\mathrm{L}$, and L are all distinct). We also consider AC^0 reductions, which provide a more detailed picture of the structure of P. We show that for constraint satisfaction problems *that include the equality relation*, there are exactly six isomorphism types under AC^0 isomorphisms (under the same assumption). Our work leaves open the question of whether there is a finite number of isomorphism types of constraint satisfaction problems under AC^0 isomorphisms.

1 Introduction

In 1978, Schaefer classified the Boolean constraint satisfaction problem and showed that, depending on the allowed relations in a propositional formula, the problem is

*Department of Computer Science, Rutgers University, Piscataway, NJ 08855, allender@cs.rutgers.edu.
†Theoretische Informatik, Universität Hannover, Appelstraße 4, D-30167 Hannover, Germany, {bauland|schnoor|vollmer}@thi.uni-hannover.de. Supported in part by DFG grant Vo 630/5-1.
‡Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003, immerman@cs.umass.edu.

ISSN 1433-8092

either in P or is NP-complete [Sch78]. This famous "dichotomy theorem" overlooks the fact that different problems in P have quite different complexity, and there is now a well-developed complexity theory to classify different problems in P. Furthermore, in Schaefer's original work (and in the many subsequent simplified presentations of his theorem [CKS01]) it is already apparent that certain classes of constraint satisfaction problems are either trivial (the 0-valid and 1-valid relations) or are solvable in NL (the bijunctive relations) or $\oplus$L (the affine relations), whereas for other problems (the Horn and anti-Horn relations) he provides only a reduction to problems that are complete for P. Is this a complete list of complexity classes that can arise in the study of constraint satisfaction problems? Given the amount of attention that the dichotomy theorem has received, it is surprising that no paper has addressed the question of how to refine Schaefer's classification beyond some steps in this direction in Schaefer's original paper (see [Sch78, Theorem 5.1]).

Our own interest in this question grew out of the observation that there is at least one other fundamental complexity class that arises naturally in the study of Boolean constraint satisfaction problems that does *not* appear in the list $(\mathrm{AC}^0, \mathrm{NL}, \oplus\mathrm{L}, \mathrm{P})$ of feasible cases identified by Schaefer. This is the class SL (symmetric logspace) that has very recently been shown by Reingold to coincide with deterministic logspace [Rei04]. (Theorem 5.1 of [Sch78] does already present examples of constraint satisfaction problems that are complete for SL.) Are there other classes that arise in this way?

The answer is more complex than we anticipated.

If we examine constraint satisfaction problems using logspace reducibility $\leq_m^{\log}$, then we are able to show that this list of complexity classes is exhaustive. Every constraint satisfaction problem is isomorphic to the standard complete set for one of the classes $\mathrm{NP}, \mathrm{P}, \oplus\mathrm{L}, \mathrm{NL}$, or L under isomorphisms computable and invertible in logspace.

However, this overlooks the fact that there is a rich collection of complexity classes lying within logspace; the correct tool to investigate these classes is AC^0 reducibility $\leq_m^{\mathrm{AC}^0}$. We are able to show that all constraint satisfaction problems that are complete for $\mathrm{NP}, \mathrm{P}, \oplus\mathrm{L}$, and NL under $\leq_m^{\log}$ reductions are already complete under $\leq_m^{\mathrm{AC}^0}$ reductions, and hence by [Agr01] these problems are equivalent to the standard complete sets for those classes under AC^0 isomorphisms $\equiv_{\mathrm{iso}}^{\mathrm{AC}^0}$. The situation becomes more complex when we examine constraint satisfaction problems in L.

All constraint satisfaction problems in L *that include the equality predicate* are either trivially solvable in AC^0 or they are complete for L under $\leq_m^{\mathrm{AC}^0}$ and hence are $\equiv_{\mathrm{iso}}^{\mathrm{AC}^0}$-isomorphic to the standard L-complete set. Our tools break down when investigating constraint satisfaction problems in L when the equality predicate is not present. We have some partial results for this case, but we leave open the question of whether there is any constraint satisfaction problem outside of AC^0 that is not complete for L, or even whether there are are infinitely many $\equiv_{\text{iso}}^{\text{AC}^0}$ types.

The proofs use a connection between complexity of constraint languages and universal algebra which has been very useful in analyzing complexity issues of constraints. An introduction to this connection can be found in [Pip97].

2 Preliminaries

An n-ary Boolean relation is a subset of $\{0,1\}^n$. For a set V of variables, a constraint application C is an application of an n-ary Boolean relation R to an n-tuple of variables $(x_1,\ldots,x_n)$ from V. An assignment $I\colon V \to \{0,1\}$ *satisfies* the constraint application $R(x_1,\ldots,x_n)$ iff $(I(x_1),\ldots,I(x_n)) \in R$. In this paper we use the standard correspondence between Boolean relations and propositional formulas: A formula $\varphi(x_1,\ldots,x_n)$ defines the relation $R_\varphi = \{(\alpha_1,\ldots,\alpha_n) \mid \varphi(\alpha_1,\ldots,\alpha_n) = 1\}$. The meaning should always be clear from the context.

A *constraint language* is a finite set of Boolean relations. The Boolean *Constraint Satisfaction Problem* over a constraint language Γ ($\mathrm{CSP}(\Gamma)$) is the question if a given set φ of Boolean constraint applications using relations from Γ is simultaneously satisfiable, i.e. if there exists an assignment $I\colon V \to \{0,1\}$, such that I satisfies every $C \in \varphi$. It is easy to see that the Boolean CSP over some language Γ is the same as satisfiability of conjunctive Γ-formulas. A well-known restriction of the general satisfiability problem is 3SAT, which can be seen as the CSP problem over the language $\Gamma_{3\mathrm{SAT}} := \{(x_1 \vee x_2 \vee x_3), (\overline{x_1} \vee x_2 \vee x_3), (\overline{x_1} \vee \overline{x_2} \vee x_3), (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})\}$.

There is a very useful connection between the complexity of the CSP problem and universal algebra, which requires a few definitions:

Definition A class of Boolean functions is called *closed* or a *clone*, if it is closed under superposition. (As explained in the survey articles [BCRV03, BCRV04] being closed under superposition is essentially the same thing as being closed under arbitrary composition.) Since the intersection of clones is again a clone, we can define, for a set B of Boolean functions, $\langle B \rangle$ as the smallest clone containing B.

It is clear that $\langle B \rangle$ is the set of Boolean functions that can be calculated by Boolean circuits using only gates for functions from B [BCRV03, BCRV04].

It is easy to see that the set of clones forms a lattice. For the Boolean case, Emil Post identified all clones (Table 1) and their inclusion structure (Figure 1). The clones are interesting for the study of the complexity of CSPs, because the complexity of $\mathrm{CSP}(\Gamma)$ depends on the closure properties of the relations in Γ, which we will define next.

Name	Definition	Base
BF	All Boolean functions	$\{\vee, \wedge, \neg\}$
R_0	$\{f \in \mathrm{BF} \mid f$ is 0-reproducing $\}$	$\{\wedge, \oplus\}$
R_1	$\{f \in \mathrm{BF} \mid f$ is 1-reproducing $\}$	$\{\vee, \leftrightarrow\}$
R_2	$\mathrm{R}_1 \cap \mathrm{R}_0$	$\{\vee, x \wedge (y \leftrightarrow z)\}$
M	$\{f \in \mathrm{BF} \mid f$ is monotonic $\}$	$\{\vee, \wedge, 0, 1\}$
M_1	$\mathrm{M} \cap \mathrm{R}_1$	$\{\vee, \wedge, 1\}$
M_0	$\mathrm{M} \cap \mathrm{R}_0$	$\{\vee, \wedge, 0\}$
M_2	$\mathrm{M} \cap \mathrm{R}_2$	$\{\vee, \wedge\}$
S_0^n	$\{f \in \mathrm{BF} \mid f$ is 0-separating of degree $n\}$	$\{\rightarrow, \mathrm{dual}(h_n)\}$
S_0	$\{f \in \mathrm{BF} \mid f$ is 0-separating $\}$	$\{\rightarrow\}$
S_1^n	$\{f \in \mathrm{BF} \mid f$ is 1-separating of degree $n\}$	$\{x \wedge \overline{y}, h_n\}$
S_1	$\{f \in \mathrm{BF} \mid f$ is 1-separating $\}$	$\{x \wedge \overline{y}\}$
S_{02}^n	$\mathrm{S}_0^n \cap \mathrm{R}_2$	$\{x \vee (y \wedge \overline{z}), \mathrm{dual}(h_n)\}$
S_{02}	$\mathrm{S}_0 \cap \mathrm{R}_2$	$\{x \vee (y \wedge \overline{z})\}$
S_{01}^n	$\mathrm{S}_0^n \cap \mathrm{M}$	$\{\mathrm{dual}(h_n), 1\}$
S_{01}	$\mathrm{S}_0 \cap \mathrm{M}$	$\{x \vee (y \wedge z), 1\}$
S_{00}^n	$\mathrm{S}_0^n \cap \mathrm{R}_2 \cap \mathrm{M}$	$\{x \vee (y \wedge z), \mathrm{dual}(h_n)\}$
S_{00}	$\mathrm{S}_0 \cap \mathrm{R}_2 \cap \mathrm{M}$	$\{x \vee (y \wedge z)\}$
S_{12}^n	$\mathrm{S}_1^n \cap \mathrm{R}_2$	$\{x \wedge (y \vee \overline{z}), h_n\}$
S_{12}	$\mathrm{S}_1 \cap \mathrm{R}_2$	$\{x \wedge (y \vee \overline{z})\}$
S_{11}^n	$\mathrm{S}_1^n \cap \mathrm{M}$	$\{h_n, 0\}$
S_{11}	$\mathrm{S}_1 \cap \mathrm{M}$	$\{x \wedge (y \vee z), 0\}$
S_{10}^n	$\mathrm{S}_1^n \cap \mathrm{R}_2 \cap \mathrm{M}$	$\{x \wedge (y \vee z), h_n\}$
S_{10}	$\mathrm{S}_1 \cap \mathrm{R}_2 \cap \mathrm{M}$	$\{x \wedge (y \vee z)\}$
D	$\{f \mid f$is self-dual$\}$	$\{x\overline{y} \vee x\overline{z} \vee (\overline{y} \wedge \overline{z})\}$
D_1	$\mathrm{D} \cap \mathrm{R}_2$	$\{xy \vee x\overline{z} \vee y\overline{z}\}$
D_2	$\mathrm{D} \cap \mathrm{M}$	$\{xy \vee yz \vee xz\}$
L	$\{f \mid f$is linear$\}$	$\{\oplus, 1\}$
L_0	$\mathrm{L} \cap \mathrm{R}_0$	$\{\oplus\}$
L_1	$\mathrm{L} \cap \mathrm{R}_1$	$\{\leftrightarrow\}$
L_2	$\mathrm{L} \cap \mathrm{R}$	$\{x \oplus y \oplus z\}$
L_3	$\mathrm{L} \cap \mathrm{D}$	$\{x \oplus y \oplus z \oplus 1\}$
V	$\{f \mid$ There is a formula of the form $c_0 \vee c_1x_1 \vee \cdots \vee c_nx_n$ such that c_i are constants for $1 \leq i \leq n$ that describes $f\}$	$\{\vee, 1, 0\}$
V_0	$[\{\vee\}] \cup \{0\}$	$\{\vee, 0\}$
V_1	$[\{\vee\}] \cup \{1\}$	$\{\vee, 1\}$
V_2	$[\{\vee\}]$	$\{\vee\}$
E	$\{f \mid$ There is a formula of the form $c_0 \wedge (c_1 \vee x_1) \wedge \cdots \wedge (c_n \vee x_n)$ such that c_i are constants for $1 \leq i \leq n$ that describes $f\}$	$\{\wedge, 1, 0\}$
E_0	$[\{\wedge\}] \cup \{0\}$	$\{\wedge, 0\}$
E_1	$[\{\wedge\}] \cup \{1\}$	$\{\wedge, 1\}$
E_2	$[\{\wedge\}]$	$\{\wedge\}$
N	$[\{\neg\}] \cup \{0\} \cup \{1\}$	$\{\neg, 1\}$
N_2	$[\{\neg\}]$	$\{\neg\}$
I	$[\{\mathrm{id}\}] \cup \{0\} \cup \{1\}$	$\{\mathrm{id}, 0, 1\}$
I_0	$[\{\mathrm{id}\}] \cup \{0\}$	$\{\mathrm{id}, 0\}$
I_1	$[\{\mathrm{id}\}] \cup \{1\}$	$\{\mathrm{id}, 1\}$
I_2	$[\{\mathrm{id}\}]$	$\{\mathrm{id}\}$

Table 1: List of all closed classes of Boolean functions, and their bases. The function h_n is defined as: $h_n(x_1, \ldots, x_{n+1}) := \bigvee_{i=1}^{n+1} x_1 \wedge x_2 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_{n+1}$

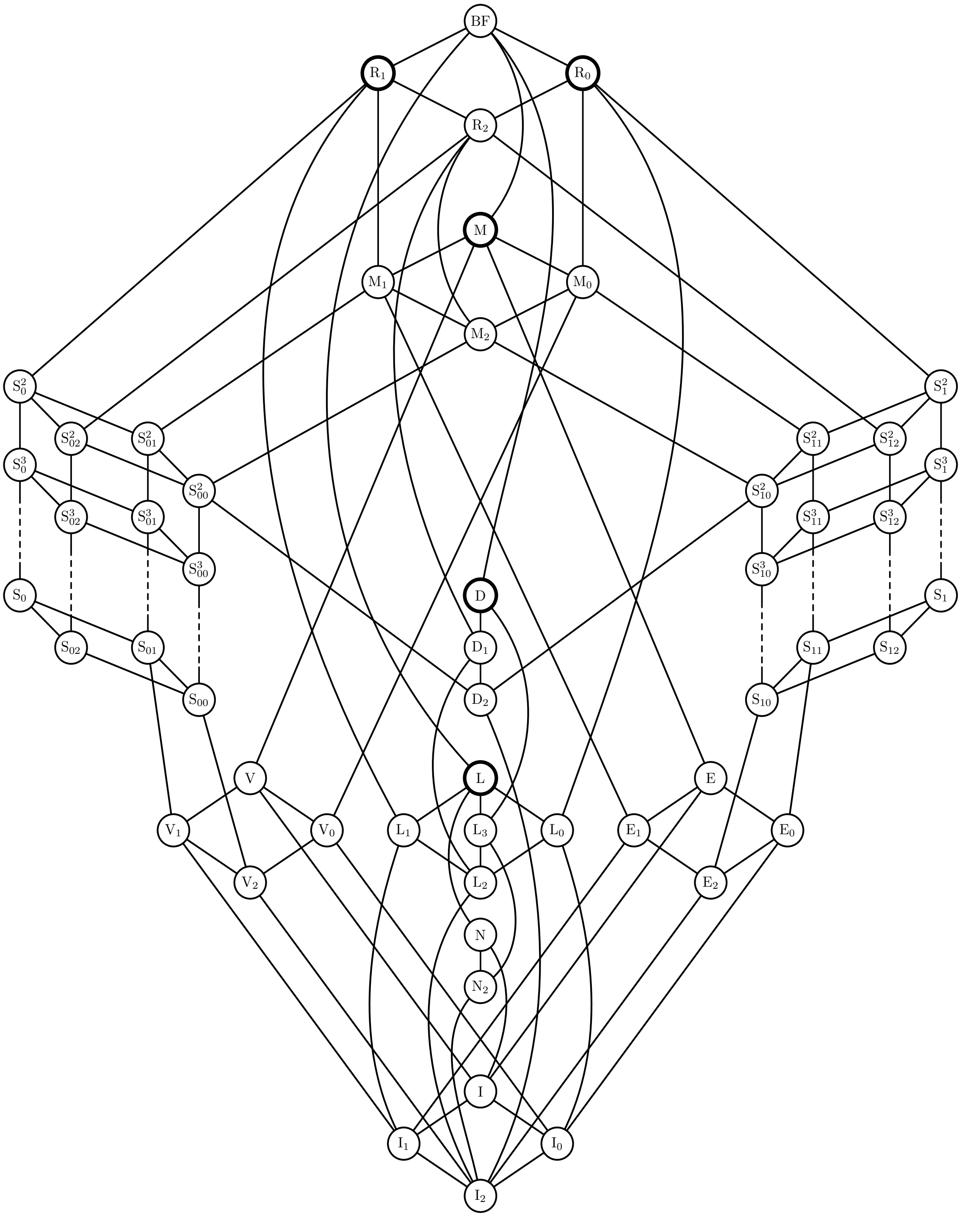

Figure 1: Graph of all closed classes of Boolean functions

Definition A k-ary relation R is *closed* under an n-ary Boolean function f, or f is a *polymorphism* of R, if for all $x_1, \dots, x_n \in R$ with $x_i = (x_i[1], x_i[2], \dots, x_i[k])$, we have

$$(f(x_1[1], \dots, x_n[1]), f(x_1[2], \dots, x_n[2]), \dots, f(x_1[k], \dots, x_n[k])) \in R.$$

We denote the set of all polymorphisms of R by $\mathrm{Pol}(R)$, and for a set Γ of Boolean relations we define $\mathrm{Pol}(\Gamma) := \{f \mid f \in \mathrm{Pol}(R) \text{ for every } R \in \Gamma\}$. For a set B of Boolean functions, $\mathrm{Inv}(B) := \{R \mid B \subseteq \mathrm{Pol}(R)\}$ is the set of *invariants* of B.

It is easy to see that every set of the form $\mathrm{Pol}(\Gamma)$ is a clone. As discussed in the surveys [BCRV03, BCRV04], the operators Pol and Inv form a "Galois connection" between the lattice of clones and certain sets of Boolean relations, which is very useful for complexity analysis of the CSP problem. The concept of relations closed under certain Boolean functions is interesting, because many properties of Boolean relations can be expressed using this terminology. For example, a set of relations can be expressed by Horn-formulas if and only if every relation in the set is closed under the binary AND function. Horn is one of the properties that ensures the corresponding satisfiability problem to be tractable. More generally, it holds that tractability of formulas over a given set of relations only depends on the set of its polymorphisms. A proof of the following theorem can be found in e.g. [JCG97] and [Dal00]:

Theorem 2.1 *If* $\mathrm{Pol}(\Gamma_2) \subseteq \mathrm{Pol}(\Gamma_1)$*, then every* $R \in \Gamma_1$ *can be expressed using relations from* Γ_2*, equality, and introduction of new existentially quantified variables.*

Therefore:

Theorem 2.2 *Let* Γ_1 *and* Γ_2 *be sets of Boolean relations such that* Γ_1 *is finite and* $\mathrm{Pol}(\Gamma_2) \subseteq \mathrm{Pol}(\Gamma_1)$*. Then* $\mathrm{CSP}(\Gamma_1) \leq_m^{\mathrm{P}} \mathrm{CSP}(\Gamma_2)$*.*

Trivially, the binary equality predicate $=$ is closed under every Boolean function. Thus, $=$ is contained in every set $\mathrm{Inv}(B)$ for a clone B (these sets often are called *co-clones*). On the other hand, every relation is closed under the projection function, $\phi_i^n(x_1, \dots, x_n) = x_i$. It is clear that when a set of relations is "big", the set of its polymorphisms is "small". So the most general case is a constraint language Γ such that $\mathrm{Pol}(\Gamma)$ only contains the projections, and these cases of the CSP are NP-complete. An example for this is the language $\Gamma_{3\mathrm{SAT}}$ from above: It can be shown that $\mathrm{Pol}(\Gamma_{3\mathrm{SAT}})$ only contains the projections, and therefore 3SAT is NP-complete.

As we have seen in the above theorem, the complexity of the CSP problem for a given constraint language is determined by the set of its polymorphisms. However, this only holds when we are just interested in the question if a given CSP is in P or

NP-complete (as Schaefer showed, one of these cases always holds). For complexity classes below P, the Galois connection has its limits: For constraint languages Γ such that $\mathrm{CSP}(\Gamma)$ is polynomial time solvable, the complexity of $\mathrm{CSP}(\Gamma)$ is not completely determined by the set of polymorphisms of Γ, as can easily be seen in the following important example:

Example 2.3 Let $\Gamma_1 := \{\{(0)\}, \{(1)\}\}$, $\Gamma_2 := \Gamma_1 \cup \{=\}$. It is obvious that $\mathrm{Pol}(\Gamma_1) = \mathrm{Pol}(\Gamma_2)$; the set of polymorphisms is the clone R_2. Formulas over Γ_1 only contain clauses of the form x or $\overline{x}$ for some variable x, whereas in Γ_2, we additionally have the binary equality predicate. We will now see that $\mathrm{CSP}(\Gamma_1)$ has very different complexity than $\mathrm{CSP}(\Gamma_2)$.

Satisfiability of a Γ_1-formula φ can be decided in AC^0. (For every variable x, check if both x and $\overline{x}$ are clauses in φ. If this is the case, φ is not satisfiable. If for all variables this does not happen, then φ is satisfiable. One can see that the complexity of this problem lies in coNLOGTIME $\subseteq \mathrm{AC}^0$.)

In contrast, $\mathrm{CSP}(\Gamma_2)$ is complete for SL under $\leq_m^{\mathrm{AC}^0}$ reductions. (Recall that SL = L [Rei04].) We show that the complement of the graph accessibility problem (GAP) for undirected graphs, which is known to be complete for SL, can be reduced to $\mathrm{CSP}(\Gamma_2)$. Let $G = (V, E)$ be a finite, undirected graph, and s, t vertices in V. For every edge $(v_1, v_2) \in E$, add a constraint $v_1 = v_2$. Also add $\overline{s}$ and t. It is obvious that there exists a path in G from s to t if and only if the resulting formula is not satisfiable. In fact, it is easy to see that $\mathrm{CSP}(\Gamma_2)$ is not only hard for SL, but it also lies within SL so it is complete for L under $\leq_m^{\mathrm{AC}^0}$ reductions.

The lesson to learn from this example is that the usual reduction among constraint satisfaction problems arising from the same co-clone is not an $\leq_m^{\mathrm{AC}^0}$ reduction. The following lemma summarizes the main relationships.

Lemma 2.4 *Let Γ_1 and Γ_2 be sets of relations over a finite set, where Γ_1 is finite and $\mathrm{Pol}(\Gamma_2) \subseteq \mathrm{Pol}(\Gamma_1)$. Then* $\mathrm{CSP}(\Gamma_1) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma_2 \cup \{=\}) \leq_m^{\log} \mathrm{CSP}(\Gamma_2)$

Proof Any relation R from Γ_1 can be expressed as

$$R(x_1, \dots, x_n) \iff R_1(x_{1,1}, \dots, x_{1,n_1}) \wedge \cdots \wedge R_m(x_{m,1}, \dots, x_{m,n_m})$$
$$\wedge (x_{i_1} = x_{i_2}) \wedge (x_{i_3} = x_{i_4}) \cdots \wedge (x_{i_{n-1}} = x_{i_n})$$

for some $R_i \in \Gamma_2$. (The variables x_{i_k} are not necessarily pairwise distinct, and not all of the variables x_{i_k} and $x_{i,k}$ are necessarily from $\{x_1, \dots, x_n\}$.) Since this local replacement can be computed in AC^0, this establishes the first reducibility relation.

For the second reduction, we need to eliminate all of the $=$-constraints. We do this by identifying variables x_{i_1} and x_{i_2} if and only if there is a $=$-path from x_{i_1} to x_{i_2} in the formula. By [Rei04], this can be computed in logspace. $\square$

3 Classification

Theorem 3.1 *Let* Γ *be a finite set of Boolean relations.*

- *If* $\mathrm{I}_0 \subseteq \mathrm{Pol}(\Gamma)$ *or* $\mathrm{I}_1 \subseteq \mathrm{Pol}(\Gamma)$*, then every constraint formula over* Γ *is satisfiable, and therefore* $\mathrm{CSP}(\Gamma)$ *is trivial.*
- *If* $\mathrm{Pol}(\Gamma) \in \{\mathrm{I}_2, \mathrm{N}_2\}$*, then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-complete for* NP*.*
- *If* $\mathrm{Pol}(\Gamma) \in \{\mathrm{V}_2, \mathrm{E}_2\}$*, then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-complete for* P*.*
- *If* $\mathrm{Pol}(\Gamma) \in \{\mathrm{L}_2, \mathrm{L}_3\}$*, then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-complete for* $\oplus\mathrm{L}$*.*
- *If* $\mathrm{S}_{00} \subseteq \mathrm{Pol}(\Gamma) \subseteq \mathrm{S}_{00}^2$ *or* $\mathrm{S}_{10} \subseteq \mathrm{Pol}(\Gamma) \subseteq \mathrm{S}_{10}^2$ *or* $\mathrm{Pol}(\Gamma) \in \{\mathrm{D}_2, \mathrm{M}_2\}$*, then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-complete for* NL*.*
- *If* $\mathrm{Pol}(\Gamma) \in \{\mathrm{D}_1, \mathrm{D}\}$*, then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-complete for* L*.*
- *If* $\mathrm{S}_{02} \subseteq \mathrm{Pol}(\Gamma) \subseteq \mathrm{R}_2$ *or* $\mathrm{S}_{12} \subseteq \mathrm{Pol}(\Gamma) \subseteq \mathrm{R}_2$*, then* $\mathrm{CSP}(\Gamma)$ *is in* L*, and* $\mathrm{CSP}(\Gamma \cup \{=\})$ *is complete for* L *under* $\leq_m^{\mathrm{AC}^0}$*.*

Theorem 3.1 is a refinement of Theorem 5.1 from [Sch78] and Theorem 6.5 from [CKS01]. The proof follows from the lemmas in the following subsections. First, we mention some corollaries.

Corollary 3.2 *For any set* Γ*,* $\mathrm{CSP}(\Gamma)$ *is logspace-isomorphic to the standard complete set for one of the following complexity classes:* $\mathrm{NP}, \mathrm{P}, \mathrm{NL}, \oplus\mathrm{L}, \mathrm{L}$*.*

Proof It is immediate from Theorem 3.1 that if $\mathrm{CSP}(\Gamma)$ is not in L, then it is complete for one of $\mathrm{NP}, \mathrm{P}, \mathrm{NL}$ and $\oplus\mathrm{L}$ under $\leq_m^{\mathrm{AC}^0}$ reductions. By [Agr01] each of these problems is AC^0-isomorphic to the standard complete set for its class. On the other hand, if $\mathrm{CSP}(\Gamma)$ is solvable in L then it is an easy matter to reduce any problem $A \in \mathrm{L}$ to $\mathrm{CSP}(\Gamma)$ via a length-squaring, invertible logspace reduction (by first checking if $x \in A$, and then using standard padding techniques to map x to a long satisfiable instance if $x \in A$, and mapping x to a long syntactically incorrect input if $x \notin A$). Logspace isomorphism to the standard complete set now follows by [Har78] (since the standard complete set is complete under invertible, length-squaring reductions). $\square$

Corollary 3.3 *For any set* Γ *such that* $= \in \Gamma$*,* $\mathrm{CSP}(\Gamma)$ *is* AC^0*-isomorphic either to* $0\Sigma^*$ *or to the standard complete set for one of the following complexity classes:* $\mathrm{NP}, \mathrm{P}, \mathrm{NL}, \oplus\mathrm{L}, \mathrm{L}$*.*

Proof The proof is nearly identical. For the case when $\mathrm{CSP}(\Gamma)$ is trivial, as above we can provide a length-squaring reduction that is a first-order projection. Isomorphism now follows via [ABI97]. $\square$

3.1 Upper Bounds: Algorithms

First, we state results that are well known; see e.g. [Sch78],[BCRV04]:

Proposition 3.4 *Let Γ be a Boolean constraint language.*

1. *If $\mathrm{Pol}(\Gamma) \in \mathrm{I}_2, \mathrm{N}_2$, then $\mathrm{CSP}(\Gamma)$ is NP-complete. Otherwise, $\mathrm{CSP}(\Gamma) \in \mathrm{P}$.*
2. *$\mathrm{L}_2 \subseteq \mathrm{Pol}(\Gamma)$ implies $\mathrm{CSP}(\Gamma) \in \oplus\mathrm{L}$.*
3. *$\mathrm{D}_2 \subseteq \mathrm{Pol}(\Gamma)$ implies $\mathrm{CSP}(\Gamma) \in \mathrm{NL}$.*
4. *$\mathrm{I}_0 \subseteq \mathrm{Pol}(\Gamma)$ or $\mathrm{I}_1 \subseteq \mathrm{Pol}(\Gamma)$ implies every instance of $\mathrm{CSP}(\Gamma)$ is satisfiable by the all-0 or the all-1 tuple, and therefore $\mathrm{CSP}(\Gamma)$ is trivial.*

Lemma 3.5 *Let Γ be a constraint language such that $\mathrm{S}_{02} \subseteq \mathrm{Pol}(\Gamma)$ or $\mathrm{S}_{12} \subseteq \mathrm{Pol}(\Gamma)$. Then $\mathrm{CSP}(\Gamma) \in \mathrm{SL}$.*

Proof Let $\mathrm{Pol}(\Gamma) \supseteq \mathrm{S}_{12}$. Observe $\mathrm{Inv}(\mathrm{S}_{12}) = \bigcup_{k\geq 2} \mathrm{Inv}(\mathrm{S}_{12}^k)$, and since Γ is finite, we have $\mathrm{Pol}(\Gamma) \supseteq \mathrm{S}_{12}^k$ for some k. A base for $\mathrm{Inv}(\mathrm{S}_{12}^k)$ is $\{\mathrm{NAND}^k, \{0\}, \{1\}, =\}$. Let φ be a conjunction of constraint applications over this language. Membership in SL follows with this outline of an L^{SL}-algorithm:

– For every $\mathrm{NAND}(x_1, \ldots, x_k)$-clause: For every occurring variable x_i, check with an SL-GAP-algorithm if there is a =-path in the input formula from x_i to a variable y such that y is a clause. If this is true for every variable, then $\varphi \notin \mathrm{SAT}$.

– For every x clause: Check if there is a =-path in the input formula from x to a variable y such that $\overline{y}$ is a clause (i.e., $y = 0$ is a clause). If this is the case, then $\varphi \notin \mathrm{SAT}$.

– Otherwise, $\varphi \in \mathrm{SAT}$ holds.

For the S_{02}^k-classes, the algorithm works analogously (here we have OR instead of NAND and therefore we search for a =-path to a $\overline{y}$-gate in step 1). $\square$

Lemma 3.6 *Let Γ be a constraint language such that $\mathrm{Pol}(\Gamma) \supseteq \mathrm{S}_{00}$ or $\mathrm{Pol}(\Gamma) \supseteq \mathrm{S}_{10}$. Then $\mathrm{CSP}(\Gamma) \in \mathrm{NL}$.*

Proof The following algorithm is based on the proof for Theorem 6.5 in [CKS01]. Observe that there is no finite set Γ such that $\mathrm{Pol}(\Gamma) = \mathrm{S}_{00}$. Therefore, $\mathrm{Pol}(\Gamma) \supseteq \mathrm{S}_{00}^k$ for some $k \geq 2$ holds. Note that $\{\mathrm{OR}^k, x, \overline{x}, x \to y\}$ is a base for $\mathrm{Inv}(\mathrm{S}_{00})^k \supseteq \Gamma$.

Now the algorithm works as follows: For a given formula φ over the relations mentioned above plus equality, consider every positive clause $x_{i_1} \vee \cdots \vee x_{i_k}$. The clause is satisfiable if and only if there is one variable in $\{x_{i_1}, \ldots, x_{i_k}\}$ which can be set to 1 without violating any of the $\overline{x}$ and $x \to y$ clauses. For a variable $y \in \{x_{i_1}, \ldots, x_{i_k}\}$, this can be checked as follows:

For each clause $\overline{x}$, check with an NL-GAP-algorithm if there is an $\to$-$=$-path from y to x, that is a sequence $yR_1z_1, z_1R_2z_2, \ldots, z_{m-1}R_mx$ for $R_i \in \{\to, =\}$. If one of these is the case, then y cannot be set to 1. Otherwise, we can set y to 1, and the clause is satisfiable. If a clause is shown to be unsatisfiable, reject. If no clause is shown to be unsatisfiable in this way, accept.

The S_{10}-case is analogous, in this case we have NAND instead of OR. □

Our final upper bound in this section is combined with a hardness result, and thus serves as a bridge to the next two sections.

Lemma 3.7 *Let Γ be a constraint language. If $\mathrm{Pol}(\Gamma) \in \{\mathrm{D}_1, \mathrm{D}\}$, then $\mathrm{CSP}(\Gamma)$ is $\leq_m^{\mathrm{AC}^0}$-complete for* SL.

Proof Note that $\mathrm{Pol}(\oplus) = \mathrm{D}$ and $\mathrm{Pol}(R) = \mathrm{D}_1$, where $R = x_1 \wedge (x_2 \oplus x_3)$. The satisfiability problem for formulas that are conjunctions of clauses of the form x or $y \oplus z$ is complete for SL by Problem 4.1 in Section 7 of [AG00], which proves completeness for the case $\mathrm{Pol}(\Gamma) = \mathrm{D}_1$ and thus proves membership in $\oplus$L for the case $\mathrm{Pol}(\Gamma) = \mathrm{D}$. It suffices to prove hardness in the case $\mathrm{Pol}(\Gamma) = \mathrm{D}$.

This can easily be shown: For every given clause x, introduce $x \oplus f$ for a new variable f, so we only have $x \oplus y$-clauses. If the original formula holds, the new one holds with the same assignment plus $f = 0$. If the new formula φ' holds, there is some I such that $I \models \varphi'$. We know that $\overline{I} \models \varphi'$ as well, because $\oplus$ is closed under N_2. Therefore, without loss of generality, $I(f) = 0$. Then $I \setminus \{f = 0\} \models \varphi$.

Thus, the problem for formulas allowing x-clauses can be reduced to one not allowing them. Therefore, both cases are SL-complete. Note that this shows that the $\leq_m^{\log}$ reduction of Lemma 2.4 can be replaced by an $\leq_m^{\mathrm{AC}^0}$ reduction in the case when $\mathrm{Pol}(\Gamma_2) \in \{\mathrm{D}_1, \mathrm{D}\}$. This could also be shown directly by expressing equality over Γ_2. In the next subsection, we will see additional cases where the $\leq_m^{\log}$ reduction in Lemma 2.4 can be replaced by an $\leq_m^{\mathrm{AC}^0}$ reduction. □

3.2 Removing the Equality Relation

The hardness proofs in the upcoming section make use of Lemma 2.4, and therefore require the equality relation is present in the considered constraint language Γ. The next two lemmas prove that in most cases this implies hardness for the general case.

Lemma 3.8 *Let* Γ *be a finite set of Boolean relations, where* $\mathrm{Pol}(\Gamma) \subseteq \mathrm{M}_2$*. Then* $\mathrm{CSP}(\Gamma \cup \{=\}) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma)$*.*

Proof The relation "$x \to y$" is invariant under M_2. Thus given any such Γ, we can construct "$x \to y$" with help of new existentially quantified variables that do not appear anywhere else in the formula. Hence we can express $x = y$ with $x \to y \wedge y \to x$. $\square$

Lemma 3.9 *Let* Γ *be a finite set of Boolean relations, where* $\mathrm{Pol}(\Gamma) \subseteq \mathrm{L}$*. Then* $\mathrm{CSP}(\Gamma \cup \{=\}) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma)$*.*

Proof For any such set Γ, the relation R^4_{even} can be defined (where this relation consists of all 4-tuples with an even number of 1's). Note that $x = y$ is equivalent to $\exists z R^4_{\mathrm{even}}(z, z, x, y)$. $\square$

3.3 Lower Bounds: Hardness Results

One technique of proving hardness for constraint satisfaction problems is to reduce certain Boolean circuit related problems to CSPs. In [Rei01], many decision problems regarding circuits were discussed. In particular, the "Satisfiability Problem for B Circuits" ($\mathrm{SAT}^{\mathrm{C}}(B)$) is very useful for our purposes here. $\mathrm{SAT}^{\mathrm{C}}(B)$ is the problem of determining if a given Boolean circuit with gates from B has an input vector on which it computes output "1".

Lemma 3.10 *Let* Γ *be a constraint language such that* $\mathrm{Pol}(\Gamma) \in \{\mathrm{E}_2, \mathrm{V}_2\}$*. Then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-hard for* P*.*

Proof It is well-known that the satisfiability problems for Horn and anti-Horn formulas is P-complete under $\leq_m^{\log}$ reductions. We include a proof for the anti-Horn case showing hardness under $\leq_m^{\mathrm{AC}^0}$ reductions. (Membership in P follows directly from Schaefer's work.) The proof uses the standard idea of simulating each gate in a Boolean circuit with Boolean constraints expressing the function of each gate. We show $\mathrm{SAT}^C(\mathrm{S}_{11}) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma)$. The result then follows from [Rei01] plus the observation that his hardness result holds under $\leq_m^{\mathrm{AC}^0}$. Let C be a $\{(x \wedge (y \vee z), c_0\}$-circuit. For each gate $g \in C$, introduce a new variable x_g. Now, introduce constraint clauses as follows:

1. Let g be a c_0-gate. Then add a constraint $\overline{x_g}$ (i.e., $x_g = 0$).
2. Let g be a $x \vee (y \wedge z)$-gate, and let g_x, g_y, g_z be the predecessor gates of g. Then introduce a constraint $x_g \rightarrow (x_{g_x} \wedge (x_{g_y} \vee x_{g_z}))$ (this can be expressed as a conjunction of two anti-Horn clauses as follows: $(\overline{x_g} \vee x_{g_x}) \wedge (\overline{x_g} \vee x_{g_y} \vee x_{g_z})$).
3. For the output-gate g, add a constraint x_g.

By construction, the resulting constraint φ is an anti-Horn-formula. Thus all relations are closed under V_2.

We claim $C \in \mathrm{SAT}$ if and only if $\varphi \in \mathrm{SAT}$.

Let $C \in \mathrm{SAT}$. Now, assign all variables in the constraint the value the corresponding gate in the circuit has when given the satisfying assignment to the input gates. That is, we are assuming that $C(\alpha_1, \ldots, \alpha_n) = 1$. Assign to any x_g in φ the value $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n)$ (which is the value of the gate g when $(\alpha_1, \ldots, \alpha_n)$ is given as input for C). Obviously, all introduced constraint clauses are satisfied with this variable assignment.

Let $\varphi \in \mathrm{SAT}$. Assign to all input gates of the circuit the corresponding value of the satisfying assignment for φ. It can easily be shown that for all $g \in C$, $\mathrm{val}(g) \geq x_g$ holds. Since this is true for the output gate as well, and the clause x_g (for $g \in C$ the output-gate of the circuit) exists in φ, the circuit value is 1.

To complete the proof, note that all occurrences of $=$ can be removed, by Lemma 3.8. $\square$

Lemma 3.11 *Let* Γ *be a constraint language such that* $\mathrm{Pol}(\Gamma) \in \{\mathrm{L}_2, \mathrm{L}_3\}$. *Then* $\mathrm{CSP}(\Gamma)$ *is* $\leq_m^{\mathrm{AC}^0}$*-hard for* $\oplus\mathrm{L}$.

Proof Assume without loss of generality that Γ contains $=$. The proof of the general case now follows from Lemma 3.9.

For the L_2-case, this can be shown in a straightforward manner similar to the proof of Lemma 3.10: We show $\mathrm{SAT}^C(\mathrm{L}_0) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma)$ for a constraint language Γ with $\mathrm{Pol}(\Gamma) = \mathrm{L}_2$. The result then follows with [Rei01]. Since we can express x_{out} and $x_1 = x_2 \oplus x_3$ as L_2-invariant relations, we can directly reproduce the given L_0-circuit. This does not work for L_3, since we cannot express x or $\overline{x}$ in L_3. However, since L_3 is basically L_2 plus negation, we can "extend" a given relation from $\mathrm{Inv}(\mathrm{L}_2)$ so that it is invariant under negation, by simply doubling the truth-table: We show for Γ_1, Γ_2 constraint languages such that $\mathrm{Inv}(\Gamma_1) = \mathrm{L}_2$ and

$\mathrm{Inv}(\Gamma_2) = \mathrm{L}_3$, $\mathrm{CSP}(\Gamma_1) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma_2)$ holds: For an n-ary relation $R \in \mathrm{Inv}(L_2)$, let $\overline{R} := \{(\overline{x_1}, \ldots, \overline{x_n}) \mid (x_1, \ldots, x_n) \in R\}$, and let R' be the $(n+1)$-ary relation

$$R' := \{0\} \times R \cup \{1\} \times \overline{R}.$$

It is obvious that R' is closed under N_2 and under L_2, and hence under L_3. Let φ be an instance of $\mathrm{CSP}(\Gamma_1)$. Let $\Gamma_1' := \{R' \mid R \in \Gamma\}$. Let $\varphi = \bigwedge_{i=1}^{n} R_n(x_{i_1}, \ldots, x_{i_{n_i}})$. We set $\varphi' := \bigwedge_{i=1}^{n} R_n'(t, x_{i_1}, \ldots, x_{i_{n_i}})$ for a new variable t.

Let $\varphi \in \mathrm{SAT}$, $I \models \varphi$. Then $I \cup \{t = 0\} \models \varphi'$.

Let $\varphi' \in \mathrm{SAT}$, $I' \models \varphi'$. Without loss of generality, let $I'(t) = 0$ (otherwise, observe $\overline{I'} \models \varphi'$ holds as well), therefore $I'\{t = 0\} \models \varphi$. Because of Lemma 2.4, $\mathrm{CSP}(\Gamma_1) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma_1') \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma_2)$ follows. □

With the same technique, we can also examine the complexity of CSPs invariant under M_2:

Lemma 3.12 *Let Γ be a constraint language such that* $\mathrm{Pol}(\Gamma) \subseteq \mathrm{M}_2$. *Then* $\mathrm{CSP}(\Gamma)$ *is $\leq_m^{\mathrm{AC}^0}$-hard for* NL.

Proof As in the preceding lemma, we assume without loss of generality that Γ contains $=$. The general case follows from Lemma 3.8.

We show $\mathrm{SAT}^C(\mathrm{E}_0) \leq_m^{\mathrm{AC}^0} \mathrm{CSP}(\Gamma)$. The result then follows with [Rei01] for the case $\mathrm{Pol}(\Gamma) = \mathrm{M}_2$, and with Lemma 2.4 for classes below M_2. Let C be a $\{\wedge, 0\}$-circuit. For each gate $g \in C$, we introduce a variable x_g and constraint applications as follows:

1. Let g be a constant 0-gate. Then add a constraint application $\overline{x_g}$.
2. Let g be an $\wedge$-gate, $g = g_1 \wedge g_2$. Add two constraint applications $x_g \rightarrow x_{g_1}$ and $x_g \rightarrow x_{g_2}$.
3. Let g be the output-gate. Add a constraint application x_g.

Note that the needed relations are all closed under $\wedge$ and $\vee$, thus closed under M_2. Let φ be the conjunction of the constructed constraint applications. We claim $C \in \mathrm{SAT}^C \Leftrightarrow \varphi \in \mathrm{SAT}$.

Let $C \in \mathrm{SAT}^C$. Thus, there exist $\alpha_1, \ldots, \alpha_n$, such that $C(\alpha_1, \ldots, \alpha_n) = 1$. Now, for any $g \in C$, let $I(x_g) := \mathrm{val}_g(\alpha_1, \ldots, \alpha_n)$. We claim I is a satisfying assignment for φ.

Let f be a constraint application in φ. Then there exists a gate $g \in C$ such that f was introduced for gate g.

Case 1: g is a 0-gate. Then the constraint is of the form $\overline{x_g}$. This constraint application is satisfied by I, since the value of the gate g is 0 in the circuit, thus $I(x_g) = \mathrm{val}_g(\alpha_1, \ldots, \alpha_n) = 0$.

Case 2: g is an $\wedge$-gate: $g = g_1 \wedge g_2$. Then $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) = \mathrm{val}_{g_1}(\alpha_1, \ldots, \alpha_n)$ $\wedge \mathrm{val}_{g_2}(\alpha_1, \ldots, \alpha_n)$, thus $I(x_g) = I(x_{g_1}) \wedge I(x_{g_2})$, and I satisfies the constraint applications $x_g \rightarrow x_{g_1}$ and $x_g \rightarrow x_{g_2}$.

Case 3: For the output-gate $g \in C$ $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) = 1$ holds, since $(\alpha_1, \ldots, \alpha_n)$ is a satisfying argument for C. Thus, the constraint application x_g is satisfied by I.

Hence, the constraint φ is satisfied by I.

Now, let $\varphi \in \mathrm{SAT}$. Let $g_1, \ldots, g_n$ be the input gates for C, I a satisfying assignment for the variables in φ, and $\alpha_i := I(x_{g_i})$ for $i = 1, \ldots, n$. We claim $C(\alpha_1, \ldots, \alpha_n) = 1$. To show this, it is sufficient to prove $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) \geq I(x_g)$ for all $g \in C$: Since for the output-gate g, x_g is a constraint application in φ, we know $C(\alpha_1, \ldots, \alpha_n) = \mathrm{val}_g(\alpha_1, \ldots, \alpha_n) \geq I(x_g) = 1$. We prove $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) \geq I(x_g)$ by induction:

Let g be a gate in C.

Case 1: g is an input-gate. In this case, $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) = I(x_g)$ by construction.

Case 2: g is a 0-gate. Then $\overline{x_g}$ is a constraint application in φ, and thus $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) \geq I(x_g) = 0$.

Case 3: g is an $\wedge$-gate. Let $g = g_1 \wedge g_2$. Without loss of generality, let $I(x_g) = 1$. Since $x_g \rightarrow x_{g_1}$ and $x_g \rightarrow x_{g_2}$ are constraint applications in φ, we know $I(g_1) = I(g_2) = 1$. From the induction hypothesis follows $\mathrm{val}_{g_1}(\alpha_1, \ldots, \alpha_n) = \mathrm{val}_{g_2}(\alpha_1, \ldots, \alpha_n) = 1$, and thus $\mathrm{val}_g(\alpha_1, \ldots, \alpha_n) = \mathrm{val}_{g_1}(\alpha_1, \ldots, \alpha_n)$ $\wedge \mathrm{val}_{g_2}(\alpha_1, \ldots, \alpha_n) = 1$.

$\square$

The case of R_2-invariant relations is already covered in Example 2.3.

Corollary 3.13 *Let Γ be a finite set of relations such that $\mathrm{Pol}(\Gamma) \subseteq \mathrm{R}_2$. Then $\mathrm{CSP}(\Gamma \cup \{=\})$ is $\leq_m^{\mathrm{AC}^0}$-hard for* SL.

Note that the lemmas in this section cover all classes in Post's lattice, and therefore Theorem 3.1 is proven.

4 Conclusion and Further Research

We have obtained a complete classification for constraint satisfaction problems under logspace isomorphisms, and identified five isomorphism types corresponding to the complexity classes $\mathrm{NP}, \mathrm{P}, \mathrm{NL}, \oplus\mathrm{L}$, and L. Considering only the problems complete for $\mathrm{NP}, \mathrm{P}, \mathrm{NL}$ or $\oplus\mathrm{L}$, we obtained a classification for AC^0-isomorphisms. We also saw that for complexity classes below L, the complexity of the constraint satisfaction problem is not completely determined by the set of polymorphisms of a given constraint language. We saw that for constraint languages Γ such that $\mathrm{Pol}(\Gamma) = \mathrm{R}_2$, the base does make a difference for the complexity of $\mathrm{CSP}(\Gamma)$. The same is true for the case $\mathrm{Pol}(\Gamma) = \mathrm{S}_{\alpha 2}^m$ for $\alpha \in \{0,1\}, m \in \mathbb{N}, m \geq 2$: In the general case, this problem is hard for L under $\leq_m^{\mathrm{AC}^0}$-reductions. But there exist bases for these co-clones which make the constraint satisfaction problem easier: Let $\Gamma := \{x_1 \vee \cdots \vee x_m, \overline{x}\}$. Then $\mathrm{Pol}(\Gamma) = \mathrm{S}_{02}^m$. It can be shown (see the proposition below) that the CSP for this constraint language is in coNLOGTIME, and so we know for these co-clones, having equality as an element of the constraint language does make a difference. As a consequence of Lemmas 3.7, 3.8, and 3.9, the question of whether or not equality is contained in some constraint language Γ thus only makes a difference if $\mathrm{Pol}(\Gamma)$ is one of the $\mathrm{S}_{\alpha 2}$ classes or R_2.

Proposition 4.1 *Let Γ be a constraint language such that every relation in Γ can be either written as conjunction of literals, or it is monotone increasing (decreasing). Then* $\mathrm{CSP}(\Gamma) \in \mathrm{coNLOGTIME}$.

Proof Without loss of generality, let every relation be a conjunction of literals or monotone increasing. Let φ be an instance of $\mathrm{CSP}(\Gamma)$ and N be the set of variables x for which there is a constraint application in Γ which is a conjunction of literals containing $\overline{x}$. Now, let I be an assignment to the variables in φ as follows:

$$I(x) := \begin{cases} 0, & x \in N \\ 1, & \text{otherwise.} \end{cases}$$

Obviously, I satisfies all constraint applications in φ which are a conjunction of literals, and φ is satisfiable if and only if it is satisfied by I. If φ is not satisfiable, then there is a constraint which is not satisfied by I, and since relations in Γ are of bounded arity, this can be verified in coNLOGTIME. $\square$

We think these problems are either trivial (when all relations are monotone increasing (decreasing)) or complete for coNLOGTIME under uniform projections ($\leq_m^{\mathrm{dlt}}$, see [RV97])—for a given instance of 1^*, write some application of a monotone increasing relation for every occurring 1, and a clause which contradicts this one for every other symbol. We believe these are the only non-trivial cases for which the CSP is not complete for L.

We think is is worthwhile to identify criteria for bases for these classes to decide whether a certain base gives rise to a CSP in coNLOGTIME, or to a CSP that is complete for L, or perhaps to a problem of intermediate complexity.

Acknowledgments

The first and third authors thank Denis Thérien for organizing a workshop at Bellairs research institute where Phokion Kolaitis lectured at length about constraint satisfiability problems. We thank Phokion Kolaitis for his lectures and for stimulating discussions. We also thank Nadia Creignou for helpful hints.

References

[ABI97] E. Allender, J. Balcazar, and N. Immerman. A first-order isomorphism theorem. *SIAM Journal on Computing*, 26:557–567, 1997.

[AG00] C. Alvarez and R. Greenlaw. A compendium of problems complete for symmetric logarithmic space. *Computational Complexity*, 9(2):123–145, 2000.

[Agr01] M. Agrawal. The first-order isomorphism theorem. In *Foundations of Software Technology and Theoretical Computer Science: 21st Conference, Bangalore, India, December 13-15, 2001. Proceedings*, Lecture Notes in Computer Science, pages 58–69, Berlin Heidelberg, 2001. Springer Verlag.

[BCRV03] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.

[BCRV04] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part II: Constraint satisfaction problems. *SIGACT News*, 35(1):22–35, 2004.

[CKS01] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems.* Monographs on Discrete Applied Mathematics. SIAM, 2001.

[Dal00] V. Dalmau. *Computational complexity of problems over generalized formulas.* PhD thesis, Department de Llenguatges i Sistemes Informàtica, Universitat Politécnica de Catalunya, 2000.

[Har78] J. Hartmanis. On the logtape isomorphism of complete sets. *Theoretical Computer Science*, 7:273–286, 1978.

[JCG97] P. G. Jeavons, D. A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.

[Pip97] N. Pippenger. *Theories of Computability.* Cambridge University Press, Cambridge, 1997.

[Rei01] S. Reith. *Generalized Satisfiability Problems.* PhD thesis, Fachbereich Mathematik und Informatik, Universität Würzburg, 2001.

[Rei04] O. Reingold. Undirected st-connectivity in log-space. Technical Report TR04-094, ECCC Reports, 2004.

[RV97] K. Regan and H. Vollmer. Gap-languages and log-time complexity classes. *Theoretical Computer Science*, 188:101–116, 1997.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proccedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'