An Algebraic Approach to the Complexity of Propositional Circumscription

Gustav Nordh and Peter Jonsson
Department of Computer and Information Science
Linköpings Universitet, S-581 83 Linköping, Sweden
{gusno, petej}@ida.liu.se

Abstract

Every logical formalism gives rise to two fundamental problems: model checking and inference. Circumscription is one of the most important and well studied formalisms in the realm of nonmonotonic reasoning. The model checking and inference problem for propositional circumscription has been extensively studied from the viewpoint of computational complexity. We use a new approach based on algebraic techniques to study the complexity of the model checking and inference problems for propositional variable circumscription in a unified way. We prove that there exists a dichotomy theorem for the complexity of the inference problem in propositional variable circumscription. We also study the model checking and inference problem for propositional variable circumscription in many-valued logics using the same algebraic techniques. In particular we prove dichotomy theorems for the complexity of model checking and inference for propositional variable circumscription in the case of 3*-valued logic.*

1 Introduction

Circumscription, introduced by McCarthy [17], is perhaps the most well developed and extensively studied formalism in nonmonotonic reasoning. The key intuition behind circumscription is that by focusing on minimal models of formulas we achieve some degree of common sense, because minimal models have as few "exceptions" as possible.

Propositional circumscription in Boolean logic is the basic case of circumscription in which satisfying truth assignments of propositional formulas are partially ordered according to the coordinatewise partial order $\leq$ on Boolean vectors, which extends the order $0 \leq 1$ on $\{0, 1\}$. In propositional variable circumscription only a certain subset of the variables in formulas are subject to minimization, others must maintain a fixed value or are subject to no restrictions at all. Given a propositional Boolean formula T and a partition of the variables in T into three (possibly empty) disjoint subsets $(P; Z; Q)$ where P is the set of variables we want to minimize, Z is the set of variables allowed to vary and Q is the set of variables that must maintain a fixed value, we define the partial order on satisfying models as follows. Let α, β be two models of T, then $\alpha \leq_{(P;Z)} \beta$ if α and β assign the same value to the variables in Q and for every variable p in P, $\alpha(p) \leq \beta(p)$ (moreover if there exist a variable p in P such that $\alpha(p) \neq \beta(p)$, we write $\alpha <_{(P;Z)} \beta$). A minimal model of a formula T is a satisfying model α such that there exist no satisfying model β where $\beta <_{(P;Z)} \alpha$.

We will from now on call the restricted form of propositional circumscription where all variables are subject to minimization (that is $Q = Z = \emptyset$) for basic circumscription, and the more general propositional variable circumscription for propositional circumscription.

Every logical formalism gives rise to two fundamental decision problems: model checking and inference. In the case of propositional circumscription in Boolean logic the model checking and inference problems can be formulated as follows.

- **Model checking:** Given a propositional Boolean formula T, a partition of the variables in T into three disjoint subsets $(P; Z; Q)$ and a truth assignment α, is α a minimal model of T.

- **Inference:** Given two propositional Boolean formulas T and T' and a partition of the variables in T into three disjoint (possibly empty) subsets $(P; Z; Q)$, is T' true in every minimal model of T. If T' is true in every minimal model of T, we denote this by $T \vDash_{CIRC} T'$.

The formulas T and T' are assumed to be given in conjunctive normal form. It is easy to realize that the inference problem is polynomial-time equivalent to the case where T' is a single clause, since T' can be inferred from T under propositional circumscription if and only if each clause of T' can be so inferred.

The inference problem for propositional circumscription is known to have strong connections to the inference problem under closed world assumptions. More specifically the inference problem for propositional Boolean logic under extended closed world assumption (ECWA) is equivalent to the inference problem for propositional circumscription in Boolean logic [12].

The inference and model checking problems for circumscription in propositional Boolean logic are very well studied from the computational complexity perspective [5, 6, 7, 10, 11, 14, 15]. The model checking problem for propositional circumscription in Boolean logic has been proved to be coNP-complete [5]. The inference problem for propositional circumscription in Boolean logic has been proved to be Π_2^{P}-complete [11]. These results display a dramatic increase in the computational complexity compared to the case of ordinary propositional Boolean logic, where the model checking problem is solvable in linear time and the inference problem is coNP-complete [8]. These negative results raise the problem of identifying restricted cases in which the model checking and inference problem for propositional circumscription have computational complexity lower than the general case.

The most natural way to study such restrictions is to study restrictions on the formulas representing knowledge bases, denoted T in above. This is also the approach followed in most of the previous research in the area. Hence in the case of restrictions of the inference problem, we only restrict the formula T while T' are subject to no restrictions.

The ultimate goal of this line of research is to determine the complexity of every restricted special case of the problem. The first result of this type was proved by Schaefer [19], who succeeded in obtaining a complete classification of the satisfiability problem in propositional Boolean logic. He proved that every special case of the satisfiability problem in propositional Boolean logic either is tractable or NP-complete (note that this implies that the inference problem in propositional Boolean logic is either tractable or coNP-complete). Recall the result due to Ladner [16] that if P $\neq$ NP, then there exist decision problems in NP that are neither tractable nor NP-complete. Hence the existence of dichotomy theorems like Schaefer's can not be taken for granted. Schaefer raised the question of the existence of dichotomy theorems for the complexity of the satisfiability problem in propositional many-valued logics. Despite intensive research on this problem, progress has been annoyingly slow. Not until recently a dichotomy theorem for the complexity of the satisfiability problem in 3-valued logics was proved [1].

In the case of the complexity of circumscription in propositional logic, the situation is even more unsatisfying. Recently a dichotomy theorem was proved for the complexity of the model checking problem in the Boolean case [15]. Moreover for the case of basic circumscription (where $Q = Z = \emptyset$) in Boolean logic, it has been proved that every special case of the inference problem is either Π_2^{P}-complete or lies in coNP [14]. In the case of the complexity of circumscription in propositional many-valued logics, to the best of our knowledge, nothing is known neither on the model checking problem nor the inference problem.

In this paper we try to remedy this situation by investigating in a unified way the complexity of circumscription in propositional many-valued logics in general and the complexity of circumscription in propositional 3-valued logics in particular. This investigation is motivated by the fact that in many real world AI applications it is necessary to reason with incomplete or contradicting information. Consider for example the operation of an autonomous agent that has to cope with incomplete information of the world and contradicting information from its sensors. We could represent incomplete or contradicting information by introducing a new truth value, $\frac{1}{2}$. Now if we want to use circumscription in the process of reasoning we just have to give a suitable partial ordering of the truth values. In the case of an autonomous agent a suitable ordering might be $\frac{1}{2} \leq 0$ and $\frac{1}{2} \leq 1$, that is we would like to minimize incomplete and contradicting information. For an interesting example of how circumscription is used in a 3-valued setting, see [9].

We prove dichotomy theorems for the complexity of both the model checking and inference problems for circumscription in propositional 3-valued logics. Moreover we present for the first time a dichotomy theorem for the complexity of the inference problem for circumscription in propositional Boolean logic in the general case where Q and Z need not be empty. In addition all our dichotomy theorems comes with efficiently checkable criteria to decide the complexity of a given subclass of the problem. These results are obtained by the use of novel techniques from universal algebra.

Although basic circumscription (where $Q = Z = \emptyset$) is a restricted case of the problem we study, our dichotomies do not imply dichotomies for basic circumscription. This is because our hardness results does not in general carry over to the restricted case where $Q = Z = \emptyset$. Hence the existence of dichotomies for the inference and model checking problems for basic circumscription in 3-valued logic is still open.

Also note that we do not fix a particular ordering of the truth values, i.e., we only demand that the ordering is a partial order. In particular we say that a special case of the problem is tractable if it is tractable for all possible partial orders of the truth values and that a special case of the problem is hard if there exist a partial order of the truth values such that it is hard. Since previous research has focused on circumscription in Boolean logic where there are only one reasonable ordering of the truth values (namely $0 \leq 1$), this decision has not been necessary before. But even in the case where the ordering of the truth values is fixed, all our results holds if the ordering is a total order. Moreover we follow the approach in [14, 15] where the clauses of T and T' are allowed to be arbitrary logical relations (sometimes called generalized clauses).

The paper is organized as follows. In Section 2 we give the necessary background on Constraint Satisfaction Problems (CSPs) and the algebraic techniques we will use throughout this paper. In Section 3 we prove our results on the complexity of the model checking problem for circumscription in propositional logic. In Section 4 we prove analogous results on the inference problem, and finally we give some ideas for future research in Section 5.

2 Preliminaries

In this section we introduce the notation and basic results on CSPs and the algebraic techniques that we will use in the rest of this paper.

Let D be a finite set. The set of all n-tuples of elements from D is denoted by D^n. Any subset of D^n is called an n-ary relation on D. The set of all finitary relations over D is denoted by R_D. In this paper we will mainly deal with finitary relations over two or three element sets, denoted R_2 and R_3 respectively.

Definition 2.1 *A constraint language over a finite set, D, is an arbitrary set $\Gamma \subseteq R_D$.*

Constraint languages are the way in which we specify restrictions on our problems. For example in the case of the inference problem for propositional circumscription over the constraint language Γ, we demand that all the relations in the knowledge base are present in Γ.

Definition 2.2 *The constraint satisfaction problem over the constraint language Γ, denoted* CSP(Γ)*, is defined to be the decision problem with instance (V, D, C), where*

- *V is a set of variables,*
- *$D = \{d_1, d_2, \ldots, d_k\}$ is a finite set of values (sometimes called a domain), and*
- *C is a set of constraints $\{C_1, \ldots, C_q\}$, in which each constraint C_i is a pair (s_i, ϱ_i) with s_i a list of variables of length m_i, called the constraint scope, and ϱ_i an m_i-ary relation over the set D, belonging to Γ, called the constraint relation.*

The question is whether there exists s solution to (V, D, C), that is, a function from V to D such that, for each constraint in C, the image of the constraint scope is a member of the constraint relation.

Example 2.3 *Let NAE be the following ternary relation on $\{0, 1\}$: $NAE = \{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$. It is easy to see that the well known NP-complete problem* NOT-ALL-EQUAL SAT *can be expressed as* CSP($\{NAE\}$).

Next we consider operations on the domain D. Any operation on D can be extended in a standard way to an operation on tuples over D, as follows.

Definition 2.4 *Let f be a k-ary operation on D and let R be an n-ary relation over D. For any collection of k tuples, $t_1, t_2, \ldots, t_k \in R$, the n-tuple $f(t_1, t_2, \ldots, t_k)$ is defined as follows: $f(t_1, t_2, \ldots, t_k) = (f(t_1[1], t_2[1], \ldots, t_k[1]), f(t_1[2], t_2[2], \ldots, t_k[2]), \ldots, f(t_1[n], t_2[n], \ldots, t_k[n]))$, where $t_j[i]$ is the i:th component in tuple t_j.*

A technique that has shown to be useful in determining the computational complexity of CSP(Γ) is that of investigating whether Γ is closed under certain families of operations [13].

Definition 2.5 *Let $\varrho_i \in \Gamma$. If f is an operation such that for all $t_1, t_2, \ldots, t_k \in \varrho_i$ $f(t_1, t_2, \ldots, t_k) \in \varrho_i$, then ϱ_i is closed under f. If all constraint relations in Γ are closed under f then Γ is closed under f. An operation f such that Γ is closed under f is called a polymorphism of Γ. The set of all polymorphisms of Γ is denoted $Pol(\Gamma)$. Given a set of operations F, the set of all relations that is closed under all the operations in F is denoted $Inv(F)$.*

Definition 2.6 *For any set $\Gamma \subseteq R_D$ the set $\langle\Gamma\rangle$ consists of all relations that can be expressed using relations from $\Gamma \cup \{=_D\}$ ($=_D$ is the equality relation on D), conjunction, and existential quantification.*

A relation belongs to $\langle\Gamma\rangle$ if and only if it can be represented as the projection of the set of all solutions to some instance of CSP(Γ) onto some subset of variables. Intuitively, constraints using relations from $\langle\Gamma\rangle$ are exactly those which can be simulated by constraints using relations from Γ. The sets of relations of the form $\langle\Gamma\rangle$ are often referred to as relational clones. An alternative characterization of relational clones is given in the following theorem.

Theorem 2.7 ([18]) *For every set $\Gamma \subseteq R_D$, $\langle\Gamma\rangle = Inv(Pol(\Gamma))$.*

The following set of operations will be essential in the classification of the complexity of subproblems specified by constraint languages.

Definition 2.8 *Let D be a finite set. An operation f on D is called*

- *a constant operation if there is a $c \in D$ such that $f(d_1, \ldots, d_n) = c$ for any $d_1, \ldots, d_n \in D$;*
- *a near-unanimity operation if it is n-ary and satisfies the identities $f(d_2, d_1, \ldots, d_1) = f(d_1, d_2, d_1, \ldots, d_1) = \ldots = f(d_1, \ldots, d_1, d_2) = d_1$ for any $d_1, d_2 \in D$;*

- *a Mal'tsev operation if it is ternary and* $f(d_1, d_1, d_2) = f(d_2, d_1, d_1) = d_2$*, for any* $d_1, d_2 \in D$*;*
- *a semilattice operation, if it is binary and satisfies the following three conditions:* $f(d_1, f(d_2, d_3)) = f(f(d_1, d_2), d_3)$*,* $f(d_1, d_2) = f(d_2, d_1)$*, and* $f(d_1, d_1) = d_1$*, for any* $d_1, d_2, d_3 \in D$*;*
- *an essentially unary operation if* $f(d_1, \ldots, d_n) = g(d_i)$ *for some non constant unary operation* g*, and any* $d_1, \ldots, d_n \in D$ *(if* g *is the identity operation, then* f *is often called a projection).*

The first dichotomy theorem for a broad class of decision problems was Schaefer's dichotomy theorem for the complexity of the satisfiability problem in propositional Boolean logic.

Theorem 2.9 ([19]) *For any constraint language* $\Gamma \subseteq R_2$*,* $\text{CSP}(\Gamma)$ *is tractable when one of the following conditions holds:*

1. *Every* ϱ *in* Γ *contains the tuple* $(0, 0, \ldots, 0)$*.*
2. *Every* ϱ *in* Γ *contains the tuple* $(1, 1, \ldots, 1)$*.*
3. *Every* ϱ *in* Γ *is definable by a CNF formula in which each conjunct has at most one negated variable.*
4. *Every* ϱ *in* Γ *is definable by a CNF formula in which each conjunct has at most one unnegated variable.*
5. *Every* ϱ *in* Γ *is definable by a CNF formula in which each conjunct has at most two literals.*
6. *Every* ϱ *in* Γ *is definable by a system of linear equations over the field with two elements.*

In all other cases $\text{CSP}(\Gamma)$ *is NP-complete.*

Schaefer's result has later been given a much shorter and simplified proof using the algebraic techniques that we will later apply to the model checking and inference problems for circumscription in propositional logics. Schaefer's result can be reformulated in algebraic terms as follows.

Theorem 2.10 ([13]) *Let* $\Gamma \subseteq R_2$ *be a constraint language.* $\text{CSP}(\Gamma)$ *is NP-complete if* $Pol(\Gamma)$ *only contains essentially unary operations, and tractable otherwise.*

The following property of constraint languages is important when determining the complexity of $\text{CSP}(\Gamma)$. We only state the property in the restricted case where the constraint languages contain all unary singleton relations. The reason for this is that these restricted constraint languages will later be of great importance in our study of the model checking and inference problems in propositional circumscription. Note that this property is more naturally stated in terms of finite algebras [3]. Since we only use this property indirectly and the actual definition of this property is not essential to the understanding of the paper we choose to give a less beautiful and intelligible definition in terms of relational clones (in order to avoid introducing the terminology of finite algebras).

Definition 2.11 *Let* $\Gamma \subseteq R_D$ *be a constraint language.* $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ *is said to satisfy the* G-SET *property if there exist a subset* A *of* D *and an equivalence relation* θ *on* A *such that* $\theta \in Inv(Pol(\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}))$*, and every polymorphism of* $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$*, when restricted to* A *and then factorized by* θ*, is a projection on the factor set* A/θ*.*

Bulatov have recently used algebraic techniques to prove a dichotomy theorem for the complexity of the satisfiability problem in propositional 3-valued logic. We only state the dichotomy theorem in the restricted case where the constraint languages contain all unary singleton relations.

Theorem 2.12 ([1]) *Let* $\Gamma \subseteq R_3$ *be a constraint language.* $\text{CSP}(\Gamma \cup \{(d_1), (d_2), (d_3)\})$ *is NP-complete if* $\Gamma \cup \{(d_1), (d_2), (d_3)\}$ *satisfies the* G-SET *property, and tractable otherwise.*

Moreover a dichotomy conjecture has been proposed for the complexity of the satisfiability problem in propositional many-valued logic. Again, we only state the conjecture in the restricted case where the constraint languages contain all unary singleton relations.

Conjecture 2.13 ([1]) *Let* $\Gamma \subseteq R_D$ *be a constraint language.* $\text{CSP}(\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\})$ *is NP-complete if* $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ *satisfies the* G-SET *property, and tractable otherwise.*

Before we present the definitions of the constraint satisfaction analogous of the model checking and inference problems in propositional circumscription we need to define formally the partial order on functions/vectors that we use to define minimal models.

Definition 2.14 *Let* $\leq$ *denote a partial order on a domain* D*. Let* k *be a positive integer and let* $\alpha = (a_1, \ldots, a_k)$*,* $\beta = (b_1, \ldots, b_k)$ *be two* k*-tuples in* D^k*, let* P, Q, Z *be a partition of* $\{1, 2, \ldots, k\}$ *into disjoint (possibly empty) subsets. We write* $\beta \leq_{(P;Z)} \alpha$ *to denote that* $b_i \leq a_i$ *for all* $i \in P$ *and* $b_j = a_j$ *for all* $j \in Q$*. Let* $\beta <_{(P;Z)} \alpha$ *denote that* $\beta \leq_{(P;Z)} \alpha$ *and that there exist an* $i \in P$ *such that* $b_i \neq a_i$*.*

Next we introduce the minimal constraint satisfaction problem, it should be clear that this problem is equivalent to the model checking problem for circumscription in many-valued logics.

Definition 2.15 *The minimal constraint satisfaction problem over the constraint language* Γ*, denoted* MIN-CSP(Γ)*, is defined to be the decision problem with instance* $(V, P, Z, Q, D, \leq, C, \alpha)$*, where* P, Z, Q *is a partition of* V *into disjoint (possibly empty) subsets and*

- V *is a set of variables,*
- P *represents the variables to minimize,*
- Z *represents the variables that vary,*
- Q *represents the variables that are fixed,*
- D *is a finite set of values (sometimes called a domain),*
- $\leq$ *is a partial order on* D*,*
- C *is a set of constraints* $\{C_1, \ldots, C_q\}$ *in which each constraint* C_i *is a pair* (s_i, ϱ_i) *with* s_i *a list of variables of length* m_i*, called the constraint scope, and* ϱ_i *an* m_i*-ary relation over the set* D*, belonging to* Γ*, called the constraint relation, and*
- α *is a function from* V *to* D *such that, for each constraint in* C *the image of the constraint scope is a member of the constraint relation.*

The question is whether α *is a minimal solution. That is, whether there exists no function* β *from* V *to* D *such that, for each constraint in* C*, the image of the constraint scope is a member of the constraint relation and* $\beta <_{(P;Z)} \alpha$*.*

Next we introduce the minimal constraint inference problem, it should be clear that this problem is equivalent to the inference problem for circumscription in many-valued logics.

Definition 2.16 *The minimal constraint inference problem over the constraint language* Γ*, denoted* MIN-INF-CSP(Γ)*, is defined to be the decision problem with instance* $(V, P, Z, Q, D, \leq, C, \psi)$*, where* $V, P, Z, Q, D, \leq,$ *and* C *are defined as in Definition 2.15 and* ψ *is a constraint such that the set of variables in* ψ*'s constraint scope is a subset of* V*.*

The question is whether each minimal solution α *to* $(V, P, Z, Q, D, \leq, C)$ *is a solution to* ψ*.*

The size of a problem instance of MIN-INF-CSP(Γ) (MIN-CSP(Γ)) is the length of the encoding of all tuples in all the constraints in C.

We conclude this section by defining formally what we mean when we say that a certain special case of a problem is in, or complete for certain complexity class.

Definition 2.17 *The problem* MIN-CSP(Γ) *(*MIN-INF-CSP(Γ)*) is in C (for a complexity class C) if for any finite* $\Gamma' \subseteq \Gamma$ *the problem* MIN-CSP(Γ') *(*MIN-INF-CSP(Γ')*) is in C. The problem* MIN-CSP(Γ) *(*MIN-INF-CSP(Γ)*) is C-complete (for a complexity class C) if* MIN-CSP(Γ') *(*MIN-INF-CSP(Γ')*) is C-hard for a certain finite* $\Gamma' \subseteq \Gamma$*, and* MIN-CSP(Γ) *(*MIN-INF-CSP(Γ)*) is in C.*

3 Model checking

The model checking problem for circumscription in propositional Boolean logic was proved by Cadoli to be coNP-complete [5]. Some special cases of the problem was also shown to be tractable in the same paper. The question of whether a dichotomy theorem could be proved for the computational complexity of the problem was raised at that time, but was left open. This open question was finally resolved by Kirousis and Kolaitis and given an affirmative answer in [15].

We give a simplified proof of the dichotomy theorem for the complexity of the model checking problem for circumscription in propositional Boolean logic. We also prove interesting results on the complexity of the corresponding problem in many-valued logics. In particular we prove a dichotomy theorem for the complexity of the model checking problem for circumscription in propositional 3-valued logics. In addition these dichotomy theorems provides us with efficiently checkable criteria to decide whether a given subclass of the problem is tractable or coNP-complete. We also make explicit the strong connection between this problem and CSP. Note that it is easy to realize that the model checking problem is in coNP.

We begin by stating the dichotomy theorem, due to Kirousis and Kolaitis [15], for the complexity of the model checking problem for circumscription in propositional Boolean logic.

Theorem 3.1 ([15]) *Let* $\Gamma \subseteq R_2$ *be a constraint language.* MIN-CSP(Γ) *is tractable if at least one of the conditions 3,4,5,6 in Theorem 2.9 are satisfied; otherwise it is coNP-complete.*

We will later (in Theorem 3.8) give an alternative proof of the preceding theorem using the algebraic approach.

Example 3.2 *The ternary* NOT-ALL-EQUAL *relation on* $\{0,1\}$*,* NAE*, satisfies none of the conditions 3,4,5,6 in Theorem 2.9, hence* MIN-CSP($\{NAE\}$) *is coNP-complete.*

The following theorem forms the basis of the algebraic approach to determine the complexity of the model checking problem for circumscription in propositional logic. It states that when studying the complexity of MIN-CSP(Γ) it is sufficient to consider constraint languages that are relational clones. Hence the class of constraint languages to be studied can be considerably reduced.

Theorem 3.3 MIN-CSP(Γ) *is tractable (coNP-complete) if and only if* MIN-CSP($\langle\Gamma\rangle$) *is tractable (coNP-complete).*

Proof: Since $\Gamma \subseteq \langle\Gamma\rangle$, if MIN-CSP($\Gamma$) is coNP-complete then so is MIN-CSP($\langle\Gamma\rangle$), and if MIN-CSP($\langle\Gamma\rangle$) is tractable then so is MIN-CSP(Γ).

To prove the converse implications take a finite set $\Gamma_0 \subseteq \langle\Gamma\rangle$ and an instance $S = (V, P, Z, Q, D, \leq, C, \alpha)$ of MIN-CSP(Γ_0). We transform S into an equivalent instance $S' = (V', P', Z', Q', D, \leq, C', \alpha')$ of MIN-CSP(Γ_1), where Γ_1 is a finite subset of Γ.

For every constraint $C = ((v_1, \ldots, v_m), \varrho)$ in S, ϱ can be represented on the form $\varrho(v_1, \ldots, v_m) =$

$$\exists_{v_{m+1}}, \ldots, \exists_{v_n} \varrho_1(v_{11}, \ldots, v_{1n_1}) \wedge \ldots \wedge \varrho_k(v_{k1}, \ldots, v_{kn_k})$$

where $\varrho_1, \ldots, \varrho_k \in \Gamma \cup \{=_D\}$, $v_{m+1}, \ldots, v_n$ are new variables not previously present in S, and $v_{11}, \ldots, v_{1n_1}, v_{21}, \ldots, v_{kn_k} \in \{v_1, \ldots, v_n\}$. Replace the constraint C with the constraints $((v_{11}, \ldots, v_{1n_1}), \varrho_1), \ldots, ((v_{k1}, \ldots, v_{kn_k}), \varrho_k)$. Add $v_{m+1}, \ldots, v_n$ to V and Z. Extend α so that α is a solution to all of the new constraints. We know that this is always possible since α was a solution to $\varrho(v_1, \ldots, v_m)$, and since $\varrho_1 \wedge \ldots \wedge \varrho_k \in \langle\Gamma\rangle$ such an extension can be found in polynomial time by checking each tuple in $\varrho_1 \wedge \ldots \wedge \varrho_k$. If we repeat the same reduction for every constraint in C it results in an equivalent instance $S'' = (V'', P, Z'', Q, D, \leq, C'', \alpha'')$ of MIN-CSP($\Gamma_1 \cup \{=_D\}$).

For each equality constraint $((v_i, v_j), =)$ in S'' we do the following:

- If both v_i and v_j are in P (Z'', Q) we remove v_i from P (Z'', Q) and V'', replace all occurrences of v_i in C'' by v_j and modify α'' so that it is not defined on v_i.

- If v_j is in Q and v_i is in P we remove v_i from P and V'', replace all occurrences of v_i in C'' by v_j and modify α'' so that it is not defined on v_i. The case where v_j is in P and v_i is in Q is handled in the same way.

- The case that remains is when one of v_i and v_j is in Z'', assume w.l.o.g. that v_i is in Z''. We remove v_i from Z'' and V'', replace all occurrences of v_i in C'' by v_j and modify α'' so that it is not defined on v_i.

Finally remove $((v_i, v_j), =)$ from C''. The resulting instance $S' = (V', P', Z', Q', D, \leq, C', \alpha')$ of MIN-CSP(Γ_1) is equivalent to S and has been obtained in polynomial time. Hence if MIN-CSP(Γ) is tractable then so is MIN-CSP($\langle\Gamma\rangle$), and if MIN-CSP($\langle\Gamma\rangle$) is coNP-complete then so is MIN-CSP(Γ). $\Box$

The following theorem states that all the information about the complexity of MIN-CSP(Γ) can be extracted from the set of polymorphisms of Γ.

Theorem 3.4 *Let Γ_1 and Γ_2 be sets of relations over a finite set, such that Γ_1 is finite. If $Pol(\Gamma_2) \subseteq Pol(\Gamma_1)$ then* MIN-CSP(Γ_1) *is polynomial-time reducible to* MIN-CSP(Γ_2).

Proof: $Pol(\Gamma_2) \subseteq Pol(\Gamma_1)$ implies that $Inv(Pol(\Gamma_1)) \subseteq Inv(Pol(\Gamma_2))$ or equivalently $\langle\Gamma_1\rangle \subseteq \langle\Gamma_2\rangle$. Then by Theorem 3.3 it follows that MIN-CSP(Γ_1) is polynomial-time reducible to MIN-CSP(Γ_2). $\Box$

Hence what remains to be done is to determine the complexity of MIN-CSP(Γ) for all possible sets of polymorphisms of Γ. The following theorem will be very useful in this effort.

Theorem 3.5 MIN-CSP(Γ) *is tractable (coNP-complete) if and only if* MIN-CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$) *is tractable (coNP-complete).*

Proof: Since $\Gamma \subseteq \Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$, it follows that MIN-CSP(Γ) is tractable if MIN-CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$) is tractable, and MIN-CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$) is coNP-complete if MIN-CSP(Γ) is coNP-complete.

To prove the converse implications take a finite set $\Gamma_0 \subseteq \Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ and an instance $S = (V, P, Z, Q, D, \leq, C, \alpha)$ of MIN-CSP(Γ_0). We transform S into an equivalent instance $S' = (V', P', Z', Q', D, \leq, C', \alpha')$ of MIN-CSP(Γ_1), where Γ_1 is a finite subset of Γ.

For all variables x occurring in a constraint in S of the type $((x), (d_i))$ two cases emerge.

- In the case where x occurs in no other constraints, remove x from V, P, Z and Q, let α be undefined on x, and remove the constraint.

- In the case where x do occurs in another constraint, remove x from P and Z, add x to Q and remove the constraint.

It is clear that α' is a minimal solution in the resulting instance $S' = (V', P', Z', Q', D, \leq, C', \alpha')$ of MIN-CSP(Γ_1) if and only if α is a minimal solution in the original instance $S = (V, P, Z, Q, D, \leq, C, \alpha)$ of MIN-CSP(Γ_0). $\Box$

There is a strong relation between the complexity of CSP(Γ) and MIN-CSP(Γ). This relation is in part made explicit in the following theorem that will be useful in proving tractability results for MIN-CSP(Γ).

Theorem 3.6 *If* CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$) *is tractable, then* MIN-CSP(Γ) *is tractable.*

Proof: We give a polynomial time Turing reduction from MIN-CSP(Γ) to CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$). Let $min(D)$ denote the set of minimal elements in D (as specified by $\leq$). Given an instance $S = (V, P, Z, Q, D, \leq, C, \alpha)$ of MIN-CSP(Γ), let P^+ denote the set $\{p \in P | \alpha(p)$ is not in $min(D)\}$, and $S' = (V, D, C')$ denote the instance of CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$) where

$$C' = C \cup \bigcup_{p \in P \setminus P^+} \{(p, \alpha(p))\} \cup \bigcup_{q \in Q} \{(q, \alpha(q))\}.$$

Then α is a minimal solution for C if and only if for each $s \in P^+$ and for each $d \in D$ such that $d < \alpha(s)$ the instance $S'' = (V, D, C'')$ of CSP($\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$) where $C'' = C' \cup \{(s, (d))\}$ is unsatisfiable.

We begin by proving the if part. Suppose that α is a minimal solution of C and that there exists an $s \in P^+$ and a $d \in D$ such that $d < \alpha(s)$ and C'' is satisfiable. Let β be a solution of C''. Of course β is also a solution of C, and $\beta <_{(P;Z)} \alpha$ contradicting that α is a minimal solution of C.

Now to the only if part. Suppose that for each $s \in P^+$ and each $d \in D$ such that $d < \alpha(s)$, C'' is unsatisfiable, and suppose that a solution β of C exists such that $\beta <_{(P;Z)} \alpha$. Since $\beta <_{(P;Z)} \alpha$, there exists at least one $t \in P$, such that $\beta(t) < \alpha(t)$. Since $t \in P^+$, it follows that there exists one $s \in P^+$ and a $d \in D$ such that $d < \alpha(s)$ and C'' is satisfiable, contradicting the assumption above. □

All of our coNP-completeness results for the model checking problem can be deduced from the following theorem.

Theorem 3.7 *Let* $\Gamma \subseteq R_D$ *be a constraint language. If* $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ *satisfies the* G-SET *property then* MIN-CSP(Γ) *is coNP-complete.*

Proof: If $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ satisfies the G-SET property, then it follows from the proof of Proposition 7.9 in [4] that there is a relation $\varrho \in \langle \Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\} \rangle$ such that $\varrho = f^{-1}(NAE)$, where NAE is the ternary NOT-ALL-EQUAL relation on $\{0,1\}$, and f is a function from some subset $A \subseteq D$ onto $\{0, 1\}$. Note that by $\varrho = f^{-1}(NAE)$ we mean that ϱ is the following relation $\{(a_1, a_2, a_3) | (c_1, c_2, c_3) \in NAE$, and $a_i \in f^{-1}(c_i)\}$. The intuition behind ϱ is that it is similar to NAE, but we can view more than one element as 0 (1).

We shall prove that MIN-CSP($\{\varrho\}$) is coNP-complete by reduction from the coNP-complete problem MIN-CSP($\{NAE\}$). Let $\leq$ be a total order on D, $A_0 = f^{-1}(0)$, and $A_1 = f^{-1}(1)$. Furthermore let a_0 be the least element of A_0 (w.r.t. $\leq$), a_1 the least element of A_1, and assume w.l.o.g. that $a_0 \leq a_1$. Then map every instance S of MIN-CSP($\{NAE\}$) to an instance S' of MIN-CSP($\{\varrho\}$) which have the same structure, but NAE is replaced by ϱ throughout, and in the given solution α, every 0 is replaced by a_0, and every 1 is replaced by a_1. It should be clear that by the choice of a_0, a_1, and the structure of ϱ, S is equivalent to S'. Since $\varrho \in \langle \Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\} \rangle$ it follows from Theorem 3.3 and Theorem 3.5 that MIN-CSP(Γ) is coNP-complete. □

Next we present an alternative proof of the dichotomy theorem for the complexity of the model checking problem for circumscription in propositional Boolean logic (Theorem 3.1).

Theorem 3.8 *Let* $\Gamma \subseteq R_2$ *be a constraint language.* MIN-CSP(Γ) *is coNP-complete if* $Pol(\Gamma \cup \{(d_1), (d_2)\})$ *only contains essentially unary operations, and tractable otherwise.*

Proof: If $Pol(\Gamma \cup \{(d_1), (d_2)\})$ contains an operation that is not essentially unary then we can conclude from Theorem 2.10 and Theorem 3.6 that MIN-CSP($\Gamma \cup \{(d_1), (d_2)\}$) is tractable. Hence by Theorem 3.5 MIN-CSP(Γ) is tractable when $Pol(\Gamma \cup \{(d_1), (d_2)\})$ contains an operation that is not essentially unary.

What remains to be proved is that MIN-CSP(Γ) is coNP-complete when $Pol(\Gamma \cup \{(d_1), (d_2)\})$ only contains essentially unary operations. First note that if $Pol(\Gamma \cup \{(d_1), (d_2)\})$ only contains essentially unary operations then $Pol(\Gamma \cup \{(d_1), (d_2)\})$ only contains essentially unary operations f whose corresponding unary operation g is the identity operation. This is because if $g(d_i) \neq d_i$, then the relation $\{(d_i)\}$ would not be closed under f. Hence $Inv(Pol(\Gamma \cup \{(d_1), (d_2)\})) = R_2$, the set of all finitary relations over $\{d_1, d_2\}$. Thus we conclude by Cadoli's coNP-completeness result for MIN-CSP(R_2) [5] and Theorem 3.3 that MIN-CSP($\Gamma \cup \{(d_1), (d_2)\}$) is coNP-complete and by Theorem 3.5, MIN-CSP(Γ) is coNP-complete. □

Note that the techniques used in [15] to prove the dichotomy theorem for the complexity of the model checking problem for circumscription in propositional Boolean logic does not seem to be applicable in the case of many-valued propositional logics. On the contrary the algebraic techniques used in this paper are capable of proving interesting results on the complexity of the model checking problem for circumscription in propositional many-valued logics. In particular we are able to present a dichotomy theorem for the complexity of the model checking problem for circumscription in propositional 3-valued logics.

Theorem 3.9 *Let* $\Gamma \subseteq R_3$ *be a constraint language.* MIN-

$\textsc{Csp}(\Gamma)$ *is coNP-complete if* $\Gamma \cup \{(d_1),(d_2),(d_3)\}$ *satisfies the* G-SET *property, and tractable otherwise.*

Proof: If $\Gamma \cup \{(d_1),(d_2),(d_3)\}$ satisfies the G-SET property then by Theorem 3.7 we know that MIN-CSP(Γ) is coNP-complete. Otherwise we know by Theorem 2.12 that CSP($\Gamma \cup \{(d_1),(d_2),(d_3)\}$) is tractable and thus by Theorem 3.6 we get that MIN-CSP(Γ) is tractable. □

Note that the borderline between tractability and coNP-completeness in the preceding theorem coincides with the borderline between tractability and NP-completeness for CSP(Γ) in Theorem 2.12. Hence when $\Gamma \subseteq R_3$ we can use the same polynomial time algorithm as presented in [1] to check whether MIN-CSP(Γ) is coNP-complete or tractable.

By using Theorem 3.6 and known results on the complexity of CSP(Γ) we can identify several tractable cases of the model checking problem for circumscription in propositional many-valued logics.

Theorem 3.10 *If* $Pol(\Gamma)$ *contains a near-unanimity operation, a semilattice operation or a Mal'tsev operation then* MIN-CSP(Γ) *is tractable.*

Proof: It is proved in [2, 13] that CSP(Γ) is tractable if $Pol(\Gamma)$ contains a near-unanimity operation, a semilattice operation or a Mal'tsev operation. Note that if $Pol(\Gamma)$ contains a near-unanimity operation, a semilattice operation or a Mal'tsev operation, then so does $Pol(\Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\})$. Hence the result follows from Theorem 3.6. □

We conclude this section by presenting a conjecture on the complexity of the model checking problem for circumscription in propositional many-valued logics.

Conjecture 3.11 *Let* $\Gamma \subseteq R_D$ *be a constraint language.* MIN-CSP(Γ) *is coNP-complete if* $\Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\}$ *satisfies the* G-SET *property, and tractable otherwise.*

Note that the coNP-completeness part of the conjecture follows from Theorem 3.7. Also note that the tractability part of the conjecture would follow if the conjecture about the structure of the dichotomy for CSP(Γ) (Conjecture 2.13) was proved to be true.

4 Inference

The inference problem for circumscription in propositional Boolean logic was proved by Eiter and Gottlob to be Π_2^{P}-complete [11]. Some special cases of the problem was already at that time known to have complexity lower than the general case, i.e., special cases that are in coNP [7]. Kirousis and Kolaitis managed to prove a dichotomy theorem for the complexity of the inference problem in basic circumscription ($Q = Z = \emptyset$) for propositional Boolean logic [14]. More specifically they prove that every special case of the problem is either in coNP or Π_2^{P}-complete.

We prove a dichotomy theorem for the complexity of the inference problem for circumscription in propositional Boolean logic. We also prove interesting results on the complexity of the corresponding problem in many-valued logics. In particular we prove a dichotomy theorem for the complexity of the inference problem for circumscription in propositional 3-valued logics. In addition these dichotomy theorems provides us with efficiently checkable criteria to decide whether a given subclass of the problem is in coNP or Π_2^{P}-complete. Note that it is easy to realize that the inference problem is in Π_2^{P}.

The following theorem forms the basis of the algebraic approach to determine the complexity of the inference problem for circumscription in propositional logic. The proof is very similar to the proof of Theorem 3.3 and is omitted.

Theorem 4.1 MIN-INF-CSP(Γ) *is in coNP* (Π_2^{P}*-complete) if and only if* MIN-INF-CSP($\langle\Gamma\rangle$) *is in coNP* (Π_2^{P}*-complete).*

The following theorem states that all the information about the complexity of MIN-INF-CSP(Γ) can be extracted from the set of polymorphisms of Γ. The proof is essentially the same as the proof of Theorem 3.4.

Theorem 4.2 *Let* Γ_1 *and* Γ_2 *be sets of relations over a finite set, such that* Γ_1 *is finite. If* $Pol(\Gamma_2) \subseteq Pol(\Gamma_1)$ *then* MIN-INF-CSP(Γ_1) *is polynomial-time reducible to* MIN-INF-CSP(Γ_2).

Hence what remains to be done is to determine the complexity of MIN-INF-CSP(Γ) for all possible sets of polymorphisms of Γ. The following theorem will be very useful in this effort. The proof is substantially different from the proof of the corresponding result (Theorem 3.5) for MIN-CSP(Γ). Here we are forced to make changes to the constraint to be inferred when reducing MIN-INF-CSP($\Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\}$) to MIN-INF-CSP($\Gamma$).

Theorem 4.3 MIN-INF-CSP(Γ) *is in coNP* (Π_2^{P}*-complete) if and only if* MIN-INF-CSP($\Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\}$) *is in coNP* ($\Pi_2^{\mathrm{P}}$*-complete).*

Proof: Since $\Gamma \subseteq \Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\}$, it follows that MIN-INF-CSP(Γ) is in coNP if MIN-INF-CSP($\Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\}$) is in coNP, and MIN-INF-CSP($\Gamma \cup \{(d_1),(d_2),\ldots,(d_{|D|})\}$) is Π_2^{P}-complete if MIN-INF-CSP(Γ) is Π_2^{P}-complete.

To prove the converse implications take a finite set $\Gamma_0 \subseteq \Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ and an instance $S = (V, P, Z, Q, D, \leq, C, \psi)$ of MIN-INF-CSP(Γ_0). We transform S into an equivalent instance $S' = (V', P', Z', Q', D, \leq, C', \psi')$ of MIN-CSP(Γ_1), where Γ_1 is a finite subset of Γ.

For all variables x, y occurring in a constraint in S of the type $((x), (d_i))$ and $((y), (d_i))$, replace the constraint $((y), (d_i))$ by $((x, y), =)$. By using the same technique as in the proof of Theorem 3.3 it is easy to see that MIN-INF-CSP($\Gamma \cup \{=_D\}$) is polynomial time reducible to MIN-INF-CSP(Γ). Hence only $|D|$ constraints of the type $((x), (d_i))$ need to be considered. For all variables x occurring in a constraint in S of the type $((x), (d_i))$ three cases emerge.

- In the case where x occurs neither in ψ nor in any other constraints, remove x from V, P, Z and Q, and remove the constraint.
- In the case where x occurs in another constraint but not in ψ, remove x from P and Z, add x to Q and remove the constraint. Update ψ as follows. Append x to the constraint scope of ψ. Let the constraint relation consist of the following tuples $\{\{D \setminus d_i\} \times D^n\} \cup \{d_i \times \varphi_\psi\}$ where n is the length of ψ's constraint scope and φ_ψ is ψ's constraint relation.
- In the case where x occurs in another constraint and in ψ, remove x from P and Z, add x to Q and remove the constraint. Update ψ as follows. Let the constraint relation consist of the following tuples $\{\{D^j \times \{D \setminus d_i\} \times D^k\} \cup \varphi_\psi\}$ where $j + k + 1$ is the length of ψ's constraint scope, φ_ψ is ψ's constraint relation, and $j + 1$ is the position of x in ψ's constraint scope.

More informally the idea behind the reduction is as follows. If $((x), (d_i))$ is a constraint in C, we remove it and modify ψ to make sure that every minimal model α of $C \setminus \{((x), (d_i))\}$ such that $\alpha(x) \neq d_i$, is a model of ψ, and in the case where $\alpha(x) = d_i$ we make sure that α is a model of the modified ψ if and only if α was a model of the original ψ. It should be clear that S and S' are equivalent. □

There is a strong connection between the model checking and inference problem for circumscription in propositional logics.

Theorem 4.4 ([7]) *If* MIN-CSP(Γ) *is tractable then* MIN-INF-CSP(Γ) *is in coNP.*

Note that the argument in [7] only deals with circumscription in propositional Boolean logic, but it is obvious that the same argument holds also in the case of propositional many-valued logics.

All of our Π_2^{P}-completeness results for the inference problem can be deduced from the following theorem. The proof is similar to the proof of Theorem 3.7 and is omitted.

Theorem 4.5 *Let $\Gamma \subseteq R_D$ be a constraint language. If $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ satisfies the* G-SET *property then* MIN-INF-CSP(Γ) *is Π_2^{P}-complete.*

Next we present the dichotomy theorem for the complexity of the inference problem for circumscription in propositional Boolean logic. Note that previously no dichotomy theorem was known for this problem. The proof is very similar to the proof of Theorem 3.8 and is omitted.

Theorem 4.6 *Let $\Gamma \subseteq R_2$ be a constraint language.* MIN-INF-CSP(Γ) *is Π_2^{P}-complete if $Pol(\Gamma \cup \{(d_1), (d_2)\})$ only contains essentially unary operations, and it is in coNP otherwise.*

Now we are ready to present the dichotomy theorem for the complexity of the inference problem for circumscription in propositional 3-valued logics.

Theorem 4.7 *Let $\Gamma \subseteq R_3$ be a constraint language.* MIN-INF-CSP(Γ) *is Π_2^{P}-complete if $\Gamma \cup \{(d_1), (d_2), (d_3)\}$ satisfies the* G-SET *property, and in coNP otherwise.*

Proof: If $\Gamma \cup \{(d_1), (d_2), (d_3)\}$ satisfies the G-SET property then the Π_2^{P}-completeness part follows from Theorem 4.5. Otherwise we know from Theorem 3.9 that MIN-CSP(Γ) is tractable. Hence by Theorem 4.4 we conclude that MIN-INF-CSP(Γ) is in coNP. □

Note that the borderline between subproblems in coNP and Π_2^{P}-complete subproblems in the preceding theorem coincides with the borderline between tractability and NP-completeness for CSP(Γ) in Theorem 2.12. Hence when $\Gamma \subseteq R_3$ we can use the same polynomial time algorithm as presented in [1] to check whether MIN-INF-CSP(Γ) is in coNP or Π_2^{P}-complete.

We can identify several cases of the inference problem for circumscription in propositional many-valued logics having lower complexity than the general case.

Theorem 4.8 *If $Pol(\Gamma)$ contains a near-unanimity operation, a semilattice operation or a Mal'tsev operation then* MIN-INF-CSP(Γ) *is in coNP.*

Proof: It is proved in Theorem 3.10 that MIN-CSP(Γ) is tractable if $Pol(\Gamma)$ contains a near-unanimity operation, a semilattice operation or a Mal'tsev operation. Thus it follows from Theorem 4.4 that MIN-INF-CSP(Γ) is in coNP. □

We conclude this section by presenting a conjecture on the complexity of the inference problem for circumscription in propositional many-valued logics.

Conjecture 4.9 *Let* $\Gamma \subseteq R_D$ *be a constraint language.* MIN-INF-CSP(Γ) *is* Π_2^{P}*-complete if* $\Gamma \cup \{(d_1), (d_2), \ldots, (d_{|D|})\}$ *satisfies the* G-SET *property, and in coNP otherwise.*

Note that the Π_2^{P}-completeness part of the conjecture follows from Theorem 4.5. Also note that the tractability part of the conjecture would follow if the conjecture about the structure of the dichotomy for CSP(Γ) (Conjecture 2.13) was proved to be true.

5 Future research

The obvious continuation of this work would be to extend the results from 3-valued logics to n-valued logics, but in light of the close relationship here uncovered between these problems and CSP, this will probably have to await the resolution of the dichotomy conjecture for CSP. Another interesting open problem is the existence of dichotomies for basic circumscription (where $Q = Z = \emptyset$) in 3-valued logics. Moreover it is certainly the case that there exist constraint languages (knowledge bases) $\Gamma \subseteq R_2$ such that MIN-INF-CSP(Γ) is tractable. Does there in fact exist a trichotomy (P, coNP-complete, Π_2^{P}-complete) for the complexity of MIN-INF-CSP(Γ)?

Acknowledgements

The authors want to thank Andrei Krokhin for suggesting the proof of Theorem 3.7 and providing many valuable remarks on an earlier version of this paper, and Andrzej Szalas for interesting discussions on the topics in this paper. Gustav Nordh is supported by the *National Graduate School in Computer Science* (CUGS), Sweden. Peter Jonsson is partially supported by the *Center for Industrial Information Technology* (CENIIT) under grant 04.01, and by the *Swedish Research Council* (VR) under grants 221-2000-361 and 621-2003-3421.

References

[1] A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 649–658, 2002.

[2] A. Bulatov. Mal'tsev constraints are tractable. Technical report, PRG-RR-02-05, Computing Laboratory, University of Oxford, 2002.

[3] A. Bulatov, P. Jeavons, and A. Krokhin. Constraint satisfaction problems and finite algebras. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, pages 272–282, 2000.

[4] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the complexity of constraints using finite algebras (revised version). *Submitted for publication, available at* `http://www.dcs.warwick.ac.uk/people/academic/Andrei.Krokhin/papers.html`, 2003.

[5] M. Cadoli. The complexity of model checking for circumscriptive formulae. *Information Processing Letters*, 42:113–118, 1992.

[6] M. Cadoli. *Tractable Reasoning in Artificial Intelligence*. Number 941 in Lecture Notes in Artificial Intelligence. Springer Verlag Berlin Heidelberg, 1995.

[7] M. Cadoli and M. Lenzerini. The complexity of closed world reasoning and circumscription. *Journal of Information and System Sciences*, pages 255–301, 1994.

[8] S. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[9] P. Doherty, J. Kachniarz, and A. Szałas. Using contextually closed queries for local closed-world reasoning in rough knowledge databases. In S. Pal, L. Polkowski, and A. Skowron, editors, *Rough-Neuro Computing: Techniques for Computing with Words*, Cognitive Technologies, pages 219–250. Springer–Verlag, 2003.

[10] A. Durand and M. Hermann. The inference problem for propositional circumscription of affine formulas is coNP-complete. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 451–462, 2003.

[11] T. Eiter and G. Gottlob. Propositional circumscription and extended closed-world reasoning are Π_2^P-complete. *Theoretical Computer Science*, 114:231–245, 1993.

[12] M. Gelfond, H. Przymusinska, and T. Przymusinsky. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38:49–73, 1989.

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

[14] L. Kirousis and P. Kolaitis. A dichotomy in the complexity of propositional circumscription. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science*, pages 71–80, 2001.

[15] L. Kirousis and P. Kolaitis. The complexity of minimal satisfiability problems. *Information and Computation*, 187(1):20–39, 2003.

[16] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[17] J. McCarthy. Circumscription - a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[18] R. Pöschel and L. Kaluznin. *Funktionen- und Relationenalgebren*. DVW, Berlin, 1979.

[19] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226, 1978.