A Trichotomy in the Complexity of Propositional Circumscription

Gustav Nordh*

Department of Computer and Information Science
Linköpings Universitet
S-581 83 Linköping, Sweden
Fax: +46 13 28 44 99
gusno@ida.liu.se

Abstract. Circumscription is one of the most important and well studied formalisms in the realm of nonmonotonic reasoning. The inference problem for propositional circumscription has been extensively studied from the viewpoint of computational complexity. We prove that there exists a trichotomy for the complexity of the inference problem in propositional variable circumscription. More specifically we prove that every restricted case of the problem is either Π_2^{P}-complete, coNP-complete, or in P.

1 Introduction

Circumscription, introduced by McCarthy [13], 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 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 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 exists 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 exists no satisfying model β where $\beta <_{(P;Z)} \alpha$.

---

* Supported by the *National Graduate School in Computer Science* (CUGS), Sweden.

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 the fundamental problem of inference. In the case of propositional circumscription the inference problem can be formulated as follows.

– **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.

The formulas T and T' are assumed to be given in conjunctive normal form. It is easy to realize that the inference problem is equivalent (under polynomial-time conjunctive reductions) 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. Moreover we follow the approach in [10, 11, 14] where the clauses of T are allowed to be arbitrary logical relations (sometimes called generalized clauses). This approach was first used by Schaefer to classify the complexity of the satisfiability problem in propositional logic and is sometimes referred to as Schaefer's framework [18].

Circumscription in propositional logic is very well studied from the computational complexity perspective [2, 4, 7, 8, 10, 11, 14]. The inference problem for propositional circumscription has been proved to be Π_2^{P}-complete [8]. This result displays a dramatic increase in the computational complexity compared to the case of ordinary propositional logic, where the inference problem is coNP-complete [3]. This negative result raise the problem of identifying restricted cases in which the 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 [18], who succeeded in obtaining a complete classification of the satisfiability problem in propositional logic. He proved that every special case of the satisfiability problem in propositional logic either is tractable or NP-complete (note that this implies that the inference problem in propositional logic is either tractable or coNP-complete). Recall the result due to Ladner [12] that if $\mathrm{P} \neq \mathrm{NP}$, 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.

Some partial results are known for the complexity of the inference problem. More specifically, both in the general case and in the case where $Q = Z = \emptyset$, it has been proved that every special case of the problem is either Π_2^{P}-complete or lies in coNP [10, 14].

Until now we have lacked a clear picture of the complexity of the inference problem in propositional circumscription, i.e., no complete classification of special cases of the problem with a complexity in coNP as coNP-complete or in P is known. Some cases are known to be coNP-complete, but to the best of our knowledge only one case of the inference problem (where Q and Z need not be empty) is known to be in P [2].

We prove that there exists a trichotomy theorem for the complexity of the inference problem, i.e., for every special case of the problem it is either in P, coNP-complete, or Π_2^{P}-complete. Moreover we discover two new tractable cases. These results are obtained by the use of techniques from universal algebra. These techniques were first applied to the propositional circumscription problem in [14] where dichotomies for the model checking and inference problem for propositional circumscription in 3-valued logic were proved.

Although basic circumscription (where $Q = Z = \emptyset$) is a restricted case of the problem we study, our trichotomy does not imply a trichotomy for basic circumscription. This is because our hardness results do not in general carry over to the restricted case where $Q = Z = \emptyset$. Hence the existence of a trichotomy for the inference problem in basic propositional circumscription is still open.

The paper is organized as follows. In Section 2 we give the necessary background on Constraint Satisfaction Problems (CSPs) and the algebraic techniques that we will use throughout this paper. In Section 3 we prove our trichotomy theorem for the complexity of circumscription in propositional logic and finally in Section 4 we give some conclusions.

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.

2.1 Constraint Satisfaction Problems

The set of all n-tuples of elements from $\{0,1\}$ is denoted by $\{0,1\}^n$. Any subset of $\{0,1\}^n$ is called an n-ary relation on $\{0,1\}$. The set of all finitary relations over $\{0,1\}$ is denoted by BR.

Definition 1. *A constraint language over* $\{0,1\}$ *is an arbitrary set* $\Gamma \subseteq BR$.

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. *The Boolean constraint satisfaction problem (or the generalized satisfiability problem as Schaefer called it) over the constraint language* $\Gamma \subseteq BR$*, denoted* $\textsc{Csp}(\Gamma)$*, is defined to be the decision problem with instance* (V, C)*, where*

– V *is a set of variables, 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 $\{0,1\}$, belonging to Γ, called the constraint relation.*

The question is whether there exists a solution to (V, C), that is, a function from V to $\{0,1\}$ such that, for each constraint in C, the image of the constraint scope is a member of the constraint relation.

Example 1. 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 3-Sat can be expressed as $\textsc{Csp}(\{NAE\})$.

Next we consider operations on $\{0,1\}$. Any operation on $\{0,1\}$ can be extended in a standard way to an operation on tuples over $\{0,1\}$, as follows.

Definition 3. *Let f be a k-ary operation on $\{0,1\}$ and let R be an n-ary relation over $\{0,1\}$. 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 $\textsc{Csp}(\Gamma)$ is that of investigating whether Γ is closed under certain families of operations [9].

Definition 4. *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 5. *For any set $\Gamma \subseteq BR$ the set $\langle \Gamma \rangle$ consists of all relations that can be expressed using relations from $\Gamma \cup \{=\}$ ($=$ is the equality relation on $\{0,1\}$), conjunction, and existential quantification.*

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 referred to as relational clones, or co-clones. An alternative characterization of relational clones is given in the following theorem.

Theorem 1 ([17]). *For every set $\Gamma \subseteq BR$, $\langle \Gamma \rangle = Inv(Pol(\Gamma))$.*

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 logic [18]. Schaefer's result has later been given a much shorter and simplified proof using the algebraic techniques that we will later apply to the inference problem for propositional circumscription. Schaefer's result can be formulated in algebraic terms as follows.

Theorem 2 ([9]). *Let $\Gamma \subseteq BR$ be a constraint language.* $\text{Csp}(\Gamma)$ *is NP-complete if $Pol(\Gamma)$ only contains essentially unary operations, and tractable otherwise. Note that an operation f is essentially unary if and only if $f(d_1, \ldots, d_n) = g(d_i)$ for some non constant unary operation g, and any $d_1, \ldots, d_n \in \{0, 1\}$.*

As we will see later, constraint languages containing the relations $\{(0)\}$ and $\{(1)\}$ will be of particular importance to us.

Definition 6. *Given a constraint language Γ, the idempotent constraint language corresponding to Γ is $\Gamma \cup \{\{(0)\}, \{(1)\}\}$ which is denoted by Γ^{id}.*

2.2 Propositional Circumscription

In this section we make some formal definitions and recall some of the results from [14]. Note that the focus of [14] is on propositional circumscription in many-valued logics, and as a consequence the clause to be inferred is allowed to be a general constraint. Since we only consider circumscription in Boolean logic in this paper, this generalization is no longer necessary and in order to comply with the definitions of the problem in [2, 10] we require that the clause to be inferred is an ordinary clause. The results from [14] still holds.

First we introduce the minimal constraint inference problem. It should be clear that this problem is equivalent to the inference problem for propositional circumscription.

Definition 7. *The minimal constraint inference problem over the constraint language $\Gamma \subseteq BR$, denoted* $\text{Min-Inf-Csp}(\Gamma)$*, is defined to be the decision problem with instance (V, P, Z, Q, C, ψ), 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,*
- *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 $\{0, 1\}$, belonging to Γ, called the constraint relation, and*
- *ψ is a clause such that the set of variables in ψ is a subset of V.*

The question is whether each minimal model α of (V, P, Z, Q, C) is also a model of ψ.

The size of a problem instance of $\text{Min-Inf-Csp}(\Gamma)$ is the length of the encoding of all tuples in all the constraints in C.

We define formally what we mean when we say that a certain special case of a problem is tractable or complete for certain complexity class.

Definition 8. *The problem* MIN-INF-CSP(Γ) *is called tractable if for any finite* $\Gamma' \subseteq \Gamma$ *the problem* MIN-INF-CSP(Γ') *is solvable in polynomial time. The problem* MIN-INF-CSP(Γ) *is called C-complete (for a complexity class C) if* MIN-INF-CSP(Γ') *is C-hard for a certain finite* $\Gamma' \subseteq \Gamma$, *and* MIN-INF-CSP(Γ) $\in C$.

The following theorem forms the basis of the algebraic approach to determine the complexity of the inference problem for circumscription in propositional logic. It states that when investigating the complexity of MIN-INF-CSP(Γ) it is sufficient to consider constraint languages that are relational clones.

Theorem 3 ([14]). MIN-INF-CSP(Γ) *is in P (coNP-complete, Π_2^{P}-complete) if and only if* MIN-INF-CSP($\langle\Gamma\rangle$) *is in P (coNP-complete, Π_2^{P}-complete).*

Proof. Since $\Gamma \subseteq \langle\Gamma\rangle$, any instance of MIN-INF-CSP(Γ) is also an instance of MIN-INF-CSP($\langle\Gamma\rangle$). So MIN-INF-CSP($\langle\Gamma\rangle$) is at least as hard as MIN-INF-CSP(Γ).

To prove the converse, i.e., that MIN-INF-CSP(Γ) is at least as hard as MIN-INF-CSP($\langle\Gamma\rangle$), take a finite set $\Gamma_0 \subseteq \langle\Gamma\rangle$ and an instance $S = (V, P, Z, Q, C, \psi)$ of MIN-INF-CSP(Γ_0). We transform S into an equivalent instance $S' = (V', P', Z', Q', C', \psi')$ of MIN-INF-CSP(Γ_1), where Γ_1 is a finite subset of Γ.

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

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

where $\varrho_1, \dots, \varrho_k \in \Gamma \cup \{=\}$, $v_{m+1}, \dots, v_n$ are new variables not previously present in S, and $v_{11}, \dots, v_{1n_1}, v_{21}, \dots, v_{kn_k} \in \{v_1, \dots, v_n\}$. Replace the constraint C with the constraints $((v_{11}, \dots, v_{1n_1}), \varrho_1), \dots, ((v_{k1}, \dots, v_{kn_k}), \varrho_k)$. Add $v_{m+1}, \dots, v_n$ to V and Z. If we repeat the same reduction for every constraint in C it results in an equivalent instance $S'' = (V'', P, Z'', Q, C'', \psi)$ of MIN-INF-CSP($\Gamma_1 \cup \{=\}$).

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'' and ψ by v_j.
– 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'' and ψ by v_j. 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 without loss of generality that v_i is in Z''. We remove v_i from Z'' and V'', replace all occurrences of v_i in C'' and ψ by v_j.

Finally remove $((v_i, v_j), =)$ from C''.

The resulting instance $S' = (V', P', Z', Q', C', \psi')$ of MIN-INF-CSP(Γ_1) is equivalent to S and has been obtained in polynomial time. Hence MIN-INF-CSP(Γ) is in P (coNP-complete, Π_2^{P}-complete) if and only if MIN-INF-CSP($\langle\Gamma\rangle$) is in P (coNP-complete, Π_2^{P}-complete). □

The following theorem reduces the set of constraint languages that need to be considered even further.

Theorem 4 ([14]). MIN-INF-CSP(Γ) *is in P (coNP-complete, Π_2^{P}-complete) if and only if* MIN-INF-CSP(Γ^{id}) *is in P (coNP-complete, Π_2^{P}-complete).*

Proof. Since $\Gamma \subseteq \Gamma^{id}$), any instance of MIN-INF-CSP(Γ) is an instance of MIN-INF-CSP(Γ^{id}). So MIN-INF-CSP(Γ^{id}) is at least as hard as MIN-INF-CSP(Γ).

To prove the converse, i.e., that MIN-INF-CSP(Γ) is at least as hard as MIN-INF-CSP(Γ^{id}), take a finite set $\Gamma_0 \subseteq \Gamma^{id}$) 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-INF-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 $((x),(0))$ or $((x),(1))$ we do as follows. Remove x from P and Z, add x to Q and remove the constraint. Update ψ as follows. If x occurs in the form $((x),(0))$, then add x to the clause ψ. If x occurs in the form $((x),(1))$, then add $\neg x$ to ψ.

The idea behind the reduction is as follows. If $((x),(0))$ is a constraint in C, we remove it and modify ψ to make sure that every minimal model α of $C \setminus \{((x),(0))\}$ such that $\alpha(x) = 1$, is a model of ψ, and in the case where $\alpha(x) = 0$ we make sure that α is a model of the modified ψ if and only if α was a model of the original ψ. The case where $((x),(1))$ is a constraint in C is handled in the same way. It should be clear that S and S' are equivalent. $\square$

We conclude this section by stating the dichotomy for the complexity of the inference problem in propositional circumscription that was proved in [14].

Theorem 5 ([14]). *Let $\Gamma \subseteq BR$ be a constraint language.* MIN-INF-CSP(Γ) *is Π_2^{P}-complete if $Pol(\Gamma^{id})$ only contains essentially unary operations, and it is in coNP otherwise.*

3 Trichotomy Theorem for the Inference Problem

In this section we prove our trichotomy theorem for the complexity of the inference problem in propositional circumscription. In the light of Theorem 5, what remains to be proved is that every problem MIN-INF-CSP(Γ) in coNP is either coNP-hard or in P. Some important cases like Horn clauses [2] and affine clauses [7] are already known to be coNP-complete. But to the best of our knowledge the only case known to be in P is when the knowledge base only consists of clauses containing at most one positive and negative literal (i.e., clauses that are both Horn and dual-Horn) [2]. Our main results is the discovery of two new tractable classes of knowledge bases (width-2 affine and clauses only containing negative literals) and a proof that for all other classes of knowledge bases the problem is coNP-hard.

We prove this by further exploiting the results obtained in [14], e.g., Theorem 3 that states that to determine the complexity of MIN-INF-CSP(Γ) it is sufficient to consider constraint languages that are relational clones.

Emil Post [16] classified all Boolean clones/relational clones and proved that they form a lattice under set inclusion. Our proofs rely heavily on Post's lattice of Boolean clones/relational clones. An excellent introduction to Post's classification of Boolean clones can be found in the recent survey article [1], for a more complete account, see [15, 17].

See Figure 1 for the lattice of Boolean relational clones. Note that the names for the relational clones in Figure 1 do not agree with Post's names. Post also considered other classes of Boolean functions/relations, so called iterative classes, and this leads to some confusion and inconsistencies if we would use Post's names. The terminology used in Figure 1 was developed by Klaus Wagner in an attempt to construct a consistent scheme of names for clones/relational clones, and was subsequently used in [1].

Now we introduce some relational clones that will be of particular importance to us.

- **Relational Clone $\mathbf{IR}_2$:** For $a \in \{0,1\}$, a Boolean function f is called a-reproducing if $f(a,\ldots,a) = a$. The clones R_a contain all a-reproducing Boolean functions. The clone R_2 contains all functions that are both 0-reproducing and 1-reproducing. Hence $Inv(R_2) = IR_2$ is the relational clone consisting of all relations closed under all functions that are both 0-reproducing and 1-reproducing. Note that functions satisfying $f(a,\ldots,a) = a$ for all a in its domain are usually called idempotent.
- **Relational Clone $\mathbf{ID}_1$:** ID_1 is the relational clone consisting of all relations closed under the affine operation $f(x,y,z) = x \oplus y \oplus z$ and the ternary majority operation $g(x,y,z) = xy \vee yz \vee xz$. It is proved in [5] (Lemma 4.11) that any relation in ID_1 can be represented as a linear equation on at most two variables over the two element field $GF(2)$. Constraint languages $\Gamma \subseteq ID_1$ are usually called width-2 affine in the literature.
- **Relational Clone $\mathbf{IS}_1$:** S_1 is the clone consisting of all 1-separating functions (see [1] for the definition of 1-separating functions). It is proved in [6] (Lemma 39) that $Inv(S_1) = IS_1$ is the relational clone consisting only of relations of the form $\{0,1\}^n \setminus (1,1,\ldots,1)$. That is, IS_1 consists of all relations corresponding to clauses where all literals are negative. Note that [6] uses Post's original names for the Boolean clones and that S_1 is F_8^∞ in Post's notation.

As we have seen in Theorem 4 relational clones of the form $\langle \Gamma^{id} \rangle$ are of particular importance to us (remember that Min-Inf-Csp(Γ) is of the same complexity as Min-Inf-Csp(Γ^{id})). We call relational clones of this form for idempotent relational clones. It can be deduced that a relational clone Γ is idempotent if and only if $IR_2 \subseteq \Gamma$. Hence we have the following lemma.

Lemma 1. *Let Γ_1 be a Boolean relational clone and Γ_2 the relational clone that is the least upper bound of IR_2 and Γ_1 in Post's lattice of relational clones. Then the following holds,* Min-Inf-Csp(Γ_1) *is in P (coNP-complete, Π_2^{P}-complete) if and only if* Min-Inf-Csp(Γ_2) *is in P (coNP-complete, Π_2^{P}-complete).*

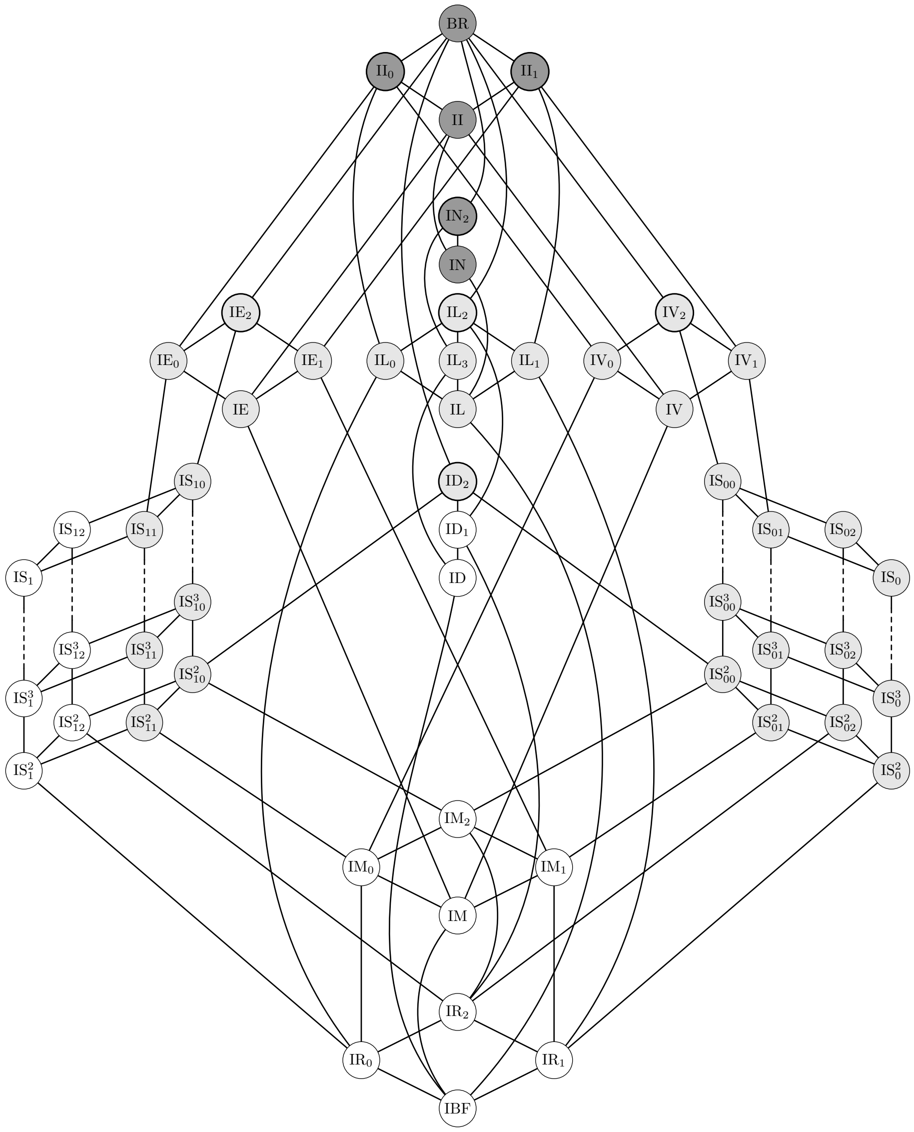

Fig. 1. Lattice (under set inclusion) of all Boolean relational clones (co-clones) and their complexity for the inference problem in propositional circumscription. White means in P, light grey means coNP-complete, and dark grey means Π_2^{P}-complete.

Proof. Remember that IR_2 is the relational clone consisting of all relations that are closed under all functions that are both 0-reproducing and 1-reproducing. Thus, $\{\{(0)\},\{(1)\}\} \subseteq IR_2$. If F is a set of Boolean functions containing a non a-reproducing function f, then $\{\{(0)\},\{(1)\}\} \not\subseteq Inv(F)$. Hence, given a relational clone Γ, then $\{\{(0)\},\{(1)\}\} \subseteq \Gamma$ if and only if $IR_2 \subseteq \Gamma$.

Thus it follows that the least upper bound of IR_2 and Γ_1 in the lattice of relational clones is $\langle \Gamma_1^{id} \rangle = \Gamma_2$. Now by Theorem 4 we get that Min-Inf-Csp(Γ_1) is in P (coNP-complete, Π_2^{P}-complete) if and only if Min-Inf-Csp(Γ_2) is in P (coNP-complete, Π_2^{P}-complete). □

Next we prove our two new tractable cases of Min-Inf-Csp(Γ). First out is the width-2 affine case.

Lemma 2. Min-Inf-Csp(ID_1) *is in P.*

Proof. Remember that the set of constraints that can be expressed by ID_1 can be represented as a system of linear equations over $GF(2)$ where each equation contains at most two variables. Hence each constraint is equivalent to an equation of the following form, $x = c$, $x = y$ or $x = -y$, where $c = 0$ or 1.

Now consider an instance $S = (V, P, Z, Q, C, \psi)$ of Min-Inf-Csp(Γ), where $\Gamma \subseteq ID_1$. We begin by reducing $S = (V, P, Z, Q, C, \psi)$ into an equivalent instance $S' = (V', P', Z', Q', C', \psi')$ such that C' has some special properties. Note that by the symmetry of equations the cases where the roles of x and y are reversed are handled in the same way.

- For all equations of the form $x = c$ we do as follows. If $x = 1$ ($x = 0$) is an equation in C and x ($\neg x$) is a literal in ψ, then every minimal model of C is also a model of ψ and we are done. Otherwise, replace all occurrences of x in C by 1 (0). Remove x from V, P, Z, Q, and remove $\neg x$ (x) from ψ. Finally remove $x = c$ from C.
- For all equations of the form $x = y$ ($x = -y$) where $x \in Q$ and y is present in another equation we do as follows. Replace all occurrences of y in all other equations and ψ by x ($-x$) and remove y from V, P, Z, and Q. Finally remove $x = y$ ($x = -y$) from C.
- For all equations of the form $x = y$ ($x = -y$) where $y \in Z$ and y is present in another equation we do as follows. Replace all occurrences of y in all other equations and ψ by x ($-x$) and remove y from Z and V. Finally remove $x = y$ ($x = -y$) from C.
- For all equations of the form $x = y$ ($x = -y$) where $x \in P$, $y \in P$, and y is present in another equation we do as follows. Replace all occurrences of y in all other equations and ψ by x ($-x$) and remove y from P and V. Finally remove $x = y$ ($x = -y$) from C.

We repeat the above process until no more equations can be removed. It can be realized that in the resulting system of equations C', no variable in Z' occurs in more than one equation, no variable in Q' occurs in an equation together with a variable that occurs in another equation. Moreover, no variable in P' occurs in an equation together with a variable from $P' \cup Z'$ that occurs in another equation.

Now, if ψ' is a tautology, then of course ψ' is true in every minimal model of C', and we are done. So we assume that ψ' is not a tautology. Since ψ' is a clause it is easy to find the (single) assignment of the variables (in ψ') that does not satisfy ψ'. Note that since C' is affine it is easy to decide whether a partial solution can be extended to a total solution, and it is clear that ψ' can be inferred from C' under circumscription if and only if this assignment cannot be extended to a minimal solution to C'. So the question that remains is whether this partial solution (that can be extended to a total solution) can be extended to a minimal solution to C' or not.

Consider the equations of the form $x = y$ or $x = -y$ where neither x nor y is in Q and x is present in ψ'. Then an assignment to x can be extended to a minimal solution to C' if and only if

- x is assigned to 0 in all equations $x = y$ where $x \in P'$ and $y \in P' \cup Z'$ and all equations $x = -y$ where $x \in P'$ and $y \in Z'$, and
- x is assigned to 0 (1) in all equations $x = y$ $(x = -y)$ where $x \in Z'$ and $y \in P'$.

□

Now on to the case of clauses where all literals are negative.

Lemma 3. Min-Inf-Csp(IS_1) *is in P.*

Proof. We recall that IS_1 consists of all relations corresponding to clauses where all literals are negative. That is, relations of the form $\{0,1\}^n \setminus (1,1,\ldots,1)$.

Now consider an instance $S = (V, P, Z, Q, C, \psi)$ of Min-Inf-Csp(Γ), where $\Gamma \subseteq IS_1$. The cases where ψ is a tautology is trivial, so we assume that ψ is not a tautology. Since ψ is a clause it is easy to find the (single) assignment of the variables that does not satisfy ψ. It is clear that ψ can be inferred from C under circumscription if and only if this assignment cannot be extended to a minimal solution to C. Note that since C consists of Horn clauses (with only negative literals) it is easy to decide whether a partial solution can be extended to a total solution, and it should be clear that such a partial solution can be extended to a minimal solution if and only if all variables in P that are assigned by this partial solution are assigned the value 0. Hence Min-Inf-Csp(IS_1) is in P. □

Next we give the complexity of Min-Inf-Csp(Γ) for 8 particular relational clones.

Theorem 6. Min-Inf-Csp(Γ) *is coNP-complete when: 1. $\Gamma = IS_{11}^2$; 2. $\Gamma = IS_0^2$; 3. $\Gamma = IL$; 4. $\Gamma = IV$; or 5. $\Gamma = IE$.* Min-Inf-Csp(Γ) *is in P when: 6. $\Gamma = IS_{12}$; 7. $\Gamma = ID_1$; or 8. $\Gamma = IM_2$.*

Proof. 1. The least upper bound of IS_{11}^2 and IR_2 is IS_{10}^2. IS_{10}^2 contains all Horn clauses with at most 2 variables and it is proved in [2] that Min-Inf-Csp(IS_{10}^2) is coNP-complete, hence by Lemma 1 it follows that Min-Inf-Csp(IS_{11}^2) is coNP-complete.

2. IS_0^2 contains clauses of the form $(x \vee y)$, that is clauses only consisting of two positive literals. It is proved in [2] that MIN-INF-CSP(IS_0^2) is coNP-complete.
3. The least upper bound of IL and IR_2 is IL_2, the set of affine clauses. It is proved in [7] that MIN-INF-CSP(IL_2) is coNP-complete, hence by Lemma 1 it follows that MIN-INF-CSP(IL) is coNP-complete.
4. The least upper bound of IV and IR_2 is IV_2, the set of dual-Horn clauses. By Case 2. and Lemma 1 it follows that MIN-INF-CSP(IV) is coNP-complete.
5. The least upper bound of IE and IR_2 is IE_2, the set of Horn clauses. By Case 1. and Lemma 1 it follows that MIN-INF-CSP(IE) is coNP-complete.
6. It is proved in Lemma 3 that MIN-INF-CSP(IS_1) is in P. The least upper bound of IS_1 and IR_2 is IS_{12}, hence by Lemma 1 it follows that MIN-INF-CSP(IS_{12}) is in P.
7. This is proved in Lemma 2.
8. IM_2 consists of all clauses that are both Horn and dual Horn. It is proved in [2] that MIN-INF-CSP(IM_2) is in P.

□

The previous theorem together with the structure of Post's lattice of relational clones and the results proved in [14] yields a trichotomy for the complexity of MIN-INF-CSP(Γ). The results are summarized in terms of the relational clones in Figure 1. A perhaps more intelligible summary is given in the conclusions below.

4 Conclusions

Only one tractable case of the inference problem for propositional circumscription (where Q and Z need not to be empty) is known, namely when the knowledge base only consists of clauses that are both Horn and dual-Horn [2]. We have found two new tractable classes of knowledge bases (width-2 affine clauses, and Horn clauses only containing negative literals) for the inference problem in propositional circumscription. We have proved that the inference problem is coNP-hard for all other classes of knowledge bases. This together with the results in [14] gives us the following trichotomy for the complexity of the inference problem in propositional circumscription:

- **P:** Horn and dual-Horn, width-2 affine, negative Horn;
- **coNP-complete:** Horn, dual-Horn, affine, bijunctive, (and not Horn and dual-Horn, width-2 affine, or negative Horn);
- Π_2^{P}**-complete:** All that are not Horn, dual-Horn, affine, or bijunctive.

In closing we note that the problem of establishing a trichotomy (as conjectured in [10]) for the complexity of the inference problem for propositional circumscription in the restricted case where all variables must be minimized ($Q = Z = \emptyset$) is still open.

Acknowledgments

The author thanks Steffen Reith for providing the visualization of Post's lattice in Figure 1.

References

1. E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with boolean blocks, part I: Post's lattice with applications to complexity theory. *ACM SIGACT-Newsletter*, 34(4):38–52, 2003.
2. M. Cadoli and M. Lenzerini. The complexity of closed world reasoning and circumscription. *Journal of Computer and System Sciences*, pages 255–301, 1994.
3. S. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.
4. S. Coste-Marquis and P. Marquis. Complexity results for propositional closed world reasoning and circumscription from tractable knoledge bases. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 24–29, 1999.
5. N. Creignou, S. Khanna, and M. Sudan. *Complexity classifications of boolean constraint satisfaction problems*. SIAM, Philadelphia, 2001.
6. V. Dalmau. *Computational Complexity of Problems over Generalized Formulas*. PhD thesis, Department LSI, Universitat Politècnica de Catalanya, Barcelona, 2000.
7. 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.
8. T. Eiter and G. Gottlob. Propositional circumscription and extended closed-world reasoning are Π_2^P-complete. *Theoretical Computer Science*, 114:231–245, 1993.
9. P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.
10. 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.
11. L. Kirousis and P. Kolaitis. The complexity of minimal satisfiability problems. *Information and Computation*, 187(1):20–39, 2003.
12. R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22:155–171, 1975.
13. J. McCarthy. Circumscription - a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
14. G. Nordh and P. Jonsson. An algebraic approach to the complexity of propositional circumscription. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pages 367–376, 2004.
15. N. Pippenger. *Theories of Computability*. Cambridge University Press, Cambridge, 1997.
16. E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.
17. R. Pöschel and L. Kaluznin. *Funktionen- und Relationenalgebren*. DVW, Berlin, 1979.
18. T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226, 1978.