Computing General First-order Parallel and Prioritized Circumscription

Hai Wan[a], Zhanhao Xiao[b], Zhenfeng Yuan[a], Heng Zhang[c], and Yan Zhang[c]
[a]School of Software, Sun Yat-sen University, Guangzhou, China
wanhai@mail.sysu.edu.cn
[b]School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China
[c]School of Computing, Engineering and Mathematics, University of Western Sydney, Sydney, Australia

Abstract

This paper focuses on computing general first-order parallel and prioritized circumscription with varying constants. We propose linear translations from general first-order circumscription to first-order theories under stable model semantics over arbitrary structures, including Tr_v for parallel circumscription and Tr_v^s for conjunction of parallel circumscriptions (further for prioritized circumscription). To improve the efficiency, we give an optimization $\Gamma_\exists$ to reduce logic programs in size when eliminating existential quantifiers during the translations. Based on these results, a general first-order circumscription solver, named cfo2lp, is developed by calling answer set programming (ASP) solvers. Using circuit diagnosis problem and extended stable marriage problem as benchmarks, we compare cfo2lp with a propositional circumscription solver circ2dlp and an ASP solver with complex optimization metasp on efficiency. Experimental results demonstrate that for problems represented by first-order circumscription naturally and intuitively, cfo2lp can compute all solutions over finite structures. We also apply our approach to description logics with circumscription and repairs in inconsistent databases, which can be handled effectively.

Introduction

As an elegant formalism for modelling non-monotonic reasoning, circumscription was introduced by McCarthy to formalize common sense reasoning in (McCarthy 1980; 1986). Lifschitz (1994) presented precise definitions of general first-order (FO) parallel and prioritized circumscription with varying constants, rewritten as a second-order (SO) sentence. Circumscription is theoretically significant because of not only its elegant syntax and semantics, but its expressive power to capture more exact inclusions naturally.

However circumscription has encountered difficulties from a practical viewpoint. So far researchers have developed methods to compute circumscription, however, the computation still remains unsatisfactory. Lifschitz (1994) discussed computational methods for simplifying FO circumscription. Cadoli, Eiter, and Gottlob (1992) eliminated varying predicates in inference, deciding whether a given formula is entailed by a parallel circumscription. Doherty, Łukaszewicz, and Szałas (1997) reduced limited circumscription to FO formulas. Wakaki and Inoue (2004) compiled FO prioritized circumscription without existential quantifiers into logic programs. Oikarinen and Janhunen (2005; 2008) presented a linear transformation to convert prioritized circumscription to logic programs and developed a propositional solver circ2dlp. Lee and Palla (2009; 2012) and Kim, Lee, and Palla (2009) presented a translation to compute a certain class of FO circumscription (named "canonical"). Zhang et al. (2011) embedded FO circumscription without varying constants into theories under stable model semantics. Gebser, Kaminski, and Schaub (2011) developed an implementation of finding inclusion-based minimal answer sets. Up to now, we have not found a solver for computing all models satisfying general FO parallel and prioritized circumscription with varying constants.

The aim of this paper is to achieve a practically usable computational approach and to apply its implementation. First, we propose and prove linear translations from general FO circumscription with varying constants to FO theories under stable model semantics over arbitrary structures: (i) Tr_v for parallel circumscription, (ii) Tr_v^s for conjunction of parallel circumscriptions and further for prioritized circumscription. Secondly, based on these reductions, we can compute general FO circumscription by using existing answer set programming (ASP) solvers, over finite structures. To improve the efficiency, an optimization $\Gamma_\exists$ is given to downsize logic programs when eliminating existential quantifiers during the translations. Thirdly, we develop a solver, named cfo2lp. To compare it with circ2dlp and metasp on efficiency, we use circuit diagnosis problem and extended stable marriage problem as benchmarks. Experimental results demonstrate that our approach can effectively solve problems represented naturally by FO circumscription and find all the solutions. Finally, we apply our approach to description logics (DLs) with circumscription and minimal repairs in inconsistent databases, which can be handled effectively.

Preliminaries

The notions of FO language are defined as usual. A FO formula is in *prenex normal form (PNF)* if it is of the form $Q_1x_1, \ldots, Q_nx_n\varphi$ where $Q_i(i = 1, \ldots, n)$ is $\exists$ or $\forall$ and φ is quantifier-free. A FO formula is in *negation normal form (NNF)* if it is built from literals by $\wedge$, $\vee$, $\forall$, and $\exists$.

Circumscription

We follow the notions of parallel and prioritized circumscription in (Lifschitz 1994). A general FO circumscription is viewed as a FO sentence in circumscription. Let φ be a FO sentence, its vocabulary is divided into three mutually disjoint tuples: *minimized predicate constants* σ_m, *varying constants* σ_v, and *fixed constants*. For each $P \in \sigma_m$, a predicate variable P^* with the same arity is introduced and let σ_m^* be the tuple of such P^*. Similarly, the tuple σ_v^* of individual, function, and predicate variables is introduced for σ_v.

Next, we introduce a comparison relation $<$ between two predicate tuples. Moreover, we use $\sigma_m^* = \sigma_m$ (*resp.* $\sigma_m^* \leq \sigma_m$) as a shorthand for the conjunction of $\forall \bar{x}(P^*(\bar{x}) \leftrightarrow P(\bar{x}))$ (*resp.* $\forall \bar{x}(P^*(\bar{x}) \rightarrow P(\bar{x}))$) for all $P^* \in \sigma_m^*$ and $P \in \sigma_m$. And let the comparison $\sigma_m^* < \sigma_m$ denote the formula $(\sigma_m^* \leq \sigma_m) \wedge \neg(\sigma_m^* = \sigma_m)$. Then *parallel circumscription* of σ_m for φ with σ_v varying is defined by a SO formula:

$$\mathrm{CIRC}[\varphi;\sigma_m;\sigma_v] = \varphi \wedge \forall \sigma_m^* \sigma_v^* (\sigma_m^* < \sigma_m \rightarrow \neg\varphi(\sigma_m^*, \sigma_v^*)) \quad (1)$$

where $\varphi(\sigma_m^*, \sigma_v^*)$ is obtained by substituting variables in σ_m^* (*resp.* σ_v^*) for corresponding constants in σ_m (*resp.* σ_v).

Intuitively, circumscription makes the interpretation of predicates in σ_m minimal under the precondition guaranteeing the validity of φ. A structure $\mathfrak{A}$ is a *σ_m-minimal model* of φ with σ_v varying if it is a model of CIRC$[\varphi; \sigma_m; \sigma_v]$.

If σ_m is decomposed into k disjoint parts $\sigma_1, \ldots, \sigma_k$, and members of σ_i are assigned a higher priority than those of σ_j for $1 \leq i < j \leq k$, then *prioritized circumscription* of such order for φ with σ_v varying is denoted by CIRC$[\varphi; \sigma_1 > \ldots > \sigma_k; \sigma_v]$, also defined by a SO formula. That is, predicates of higher priority are more important than those of lower priority and the former should be minimized further.

Example 1 [Reiter's Example, [McCarthy,1986] Section 7]

$$Quaker(x) \wedge \neg Ab_1(x) \rightarrow Pacifist(x) \quad (2)$$

$$Republican(x) \wedge \neg Ab_2(x) \rightarrow \neg Pacifist(x) \quad (3)$$

If *Nixon* is both a *Quaker* and a *Republican*, there is no conclusion on whether he is a *Pacifist* or not, even circumscribing Ab_1 and Ab_2 in φ. However if it is considered that *Nixon* is more likely to be a normal *Quaker* than a normal *Republican*, we can conclude that *Nixon* is a *Pacifist* by computing CIRC$[(2) \wedge (3); Ab_1 > Ab_2; Pacifist]$.

Stable Model Semantics

Similar to circumscription, a FO theory under stable model semantics (SM-semantics) was generalized in (Ferraris, Lee, and Lifschitz 2007; Lin and Zhou 2011). For a FO sentence ψ and a tuple σ_i of predicate constants, define SM$[\psi; \sigma_i]$ as:

$$\mathrm{SM}[\psi; \sigma_i] = \psi \wedge \forall \sigma_i^* (\sigma_i^* < \sigma_i \rightarrow \neg \mathrm{St}(\psi; \sigma_i)) \quad (4)$$

where $\mathrm{St}(\psi; \sigma_i)$ is defined recursively as follows:

- $\mathrm{St}(P(\bar{x}); \sigma_i) = P^*(\bar{x})$ if $P \in \sigma_i$;
- $\mathrm{St}(F(\bar{x}); \sigma_i) = F(\bar{x})$ if F is a predicate not in σ_i;
- $\mathrm{St}(\psi_1 \circ \psi_2; \sigma_i) = \mathrm{St}(\psi_1; \sigma_i) \circ \mathrm{St}(\psi_2; \sigma_i)$ if $\circ \in \{\wedge, \vee\}$;
- $\mathrm{St}(\psi_1 \rightarrow \psi_2; \sigma_i) = (\mathrm{St}(\psi_1; \sigma_i) \rightarrow \mathrm{St}(\psi_2; \sigma_i)) \wedge (\psi_1 \rightarrow \psi_2)$;
- $\mathrm{St}(Qx\psi_1; \sigma_i) = Qx\mathrm{St}(\psi_1; \sigma_i)$ if $Q \in \{\forall, \exists\}$.

A structure $\mathfrak{A}$ is called a *σ_i-stable model* of φ if it is a model of SM$[\varphi; \sigma_i]$. A (predicate) constant is *intensional* if it occurs in σ_i; otherwise, it is *extensional*.

From Circumscription to SM-semantics

Note that the equivalence between formulas in classical FO logic is still retained in circumscription. So for every FO formula, there always exists a formula in NNF equivalent to it in circumscription. NNF guarantees that $\neg$ only occurs directly ahead of predicates. Here $\neg P$ is treated as $P \rightarrow \bot$, called negative literal conveniently. The implications always follow predicates, so that they are handled easily when taking into account the operator St. Thus the translations in this section take formulas in NNF as inputs.

Parallel Circumscription

Now we pay attention to parallel circumscription with varying predicate constants, a specialization of prioritized circumscription. As formulas (1) and (4) show, parallel circumscription and theories under SM-semantics are similar in mathematical definition, denoted by a SO formula. It is necessary to introduce auxiliary predicates for translations from (1) to (4). There are four challenges in finding a translation:

1. Keep the equivalence between the original sentence φ and the resulting sentence ψ under FO classical logic;
2. Simulate the varying predicate constants σ_v whose corresponding variables σ_v^* can change arbitrarily;
3. Forbid auxiliary predicates from affecting the minimized predicates comparison $\sigma_m^* < \sigma_m$;
4. Make the resulting formula $\mathrm{St}(\psi; \sigma_i)$ equivalent to the original one $\varphi(\sigma_m^*, \sigma_v^*)$.

Fortunately, we find such a linear translation:

Definition 1 *Let φ be any FO sentence in NNF. Let σ_m and σ_v be two disjoint tuples of predicates respectively. Then define $Tr_v(\varphi; \sigma_m; \sigma_v)$ as the conjunction of the following:*

$$\varphi^{\neg\neg} \wedge \tilde{\varphi} \quad (5)$$

$$\gamma \leftrightarrow \bigwedge_{P \in \sigma_m} \forall \bar{x}(P(\bar{x}) \vee \neg P(\bar{x})) \quad (6)$$

$$\bigwedge_{Q' \in \sigma_v'} \forall \bar{x}(\gamma \rightarrow Q'(\bar{x})) \quad (7)$$

where $\varphi^{\neg\neg}$ is obtained from φ by substituting $\neg\neg P(\bar{x})$ for each positive literal $P(\bar{x})$ s.t. $P \in \sigma_m$; $\tilde{\varphi}$ is obtained from φ by substituting $(P(\bar{x}) \rightarrow \gamma)$ for each negative literal $\neg P(\bar{x})$ s.t. $P \in \sigma_m$, and substituting $Q'(\bar{x})$ for each positive literal $Q(\bar{x})$ and $(Q'(\bar{x}) \rightarrow \gamma)$ for each negative literal $\neg Q(\bar{x})$ s.t. $Q \in \sigma_v$; Q' is the corresponding auxiliary predicate not in φ of the same arity for every Q and σ_v' is the tuple of Q'.

Intuitively, translation Tr_v at first guarantees the resulting sentence is equivalent to the original one in FO classical logic. According to the definition of operator St, when $\sigma_m^* < \sigma_m$, γ is true while $\mathrm{St}(\gamma)$ [1] is false and when $\sigma_m^* = \sigma_m$, both are true. Thus, $\mathrm{St}(P(\bar{x}) \rightarrow \gamma)$ is equivalent to $\neg P^*(\bar{x})$.

Formula (7) plays a key role in Tr_v. St((7)) leads to the arbitrariness assignment of predicate variable Q'^* and guarantees $Q'^* \leq Q'$. What's more, the minimized predicates comparison relation $\sigma_m^* < \sigma_m$ is equivalent to that of intensional predicates. Then $\varphi^{\neg\neg}$ remains equivalent to $\mathrm{St}(\varphi^{\neg\neg})$ and $\mathrm{St}(\tilde{\varphi})$ simulates $\varphi(\sigma_m^*, \sigma_v^*)$ in circumscription.

[1] Without confusion, predicates as parameters are omitted.

Proposition 1 *Let φ be any FO sentence in NNF. Let σ_m and σ_v be two disjoint tuples of predicate constants. Then $\exists\sigma'_v\exists\gamma SM[Tr_v(\varphi;\ \sigma_m;\ \sigma_v);\ \sigma_m, \sigma'_v, \gamma]$ [2] is equivalent to CIRC$[\varphi; \sigma_m; \sigma_v]$ by omitting auxiliary predicates, where σ'_v and γ are auxiliary predicates introduced by Tr_v.*

Proof: Let σ be the vocabulary of φ. Suppose $\tau = \sigma \cup \sigma'_v \cup \{\gamma\}$ and $\tau_i = \sigma_m \cup \sigma'_v \cup \{\gamma\}$. Suppose τ-structure $\mathfrak{B}$ and σ-structure $\mathfrak{A}$ have the same interpretation on σ, defined in the same domain D.

$\Leftarrow$ First, we suppose that $\mathfrak{A}$ is a σ-structure satisfying CIRC$[\varphi; \sigma_m; \sigma_v]$. Let $\mathfrak{B}$ interpret γ to $\top$ and every Q' in σ'_v to a relation filled up with D. In other words, for every $Q' \in \sigma'_v$, $Q'(\bar{x})$ and γ are always valid in the interpretation of $\mathfrak{B}$. Now we need to show that $\mathfrak{B}$ satisfies SM$[Tr_v(\varphi; \sigma_m; \sigma_v); \tau_i]$. It is not difficult to check $\mathfrak{B}$ satisfies $Tr_v(\varphi; \sigma_m; \sigma_v)$ (shortly, $\pi(\varphi)$).

To obtain a contradiction, we assume that $\mathfrak{B}$ is not a model of SM$[\pi(\varphi); \tau_i]$. Suppose that an assignment β satisfies both $\tau_i^* < \tau_i$ and $\mathrm{St}(\pi(\varphi); \tau_i)$. In this assumption, we assert that β satisfies $\sigma_m^* < \sigma_m$. Otherwise, $\sigma_m^* = \sigma_m$ implies that for all $P^* \in \sigma_m^*$, $\forall\bar{x}(P^*(\bar{x}) \vee \neg P(\bar{x}))$ are true. Next, β satisfies $\gamma^* \leftrightarrow \forall\bar{x}(P^*(\bar{x}) \vee \neg P(\bar{x}))$, so γ^* should be assigned to $\top$. Then for each $Q'^* \in \sigma'^*_v$, $Q'^*(\bar{x})$ is valid due to $\mathrm{St}((7); \tau_i)$. As mentioned above, $\tau_i^* = \tau_i$ is satisfied by β and it breaks the assumption, so the assertion is true.

Since β satisfies $\sigma_m^* < \sigma_m$, there is at least a predicate variable $P^* \in \sigma_m^*$ dissatisfying $\forall\bar{x}(P^*(\bar{x}) \vee \neg P(\bar{x}))$. Hence γ^* is assigned to $\bot$ by β. Moreover $\forall\bar{x}(\gamma^* \rightarrow Q'^*(\bar{x}))$ are valid and actually Q'^* can change arbitrarily in the domain. Note that β satisfies $\mathrm{St}(P(\bar{x}) \rightarrow \gamma; \tau_i)$ iff β satisfies $\neg P^*(\bar{x})$. According to the substitution method and the NNF of φ, it is clear that if $\beta(\sigma'^*_v) = \beta(\sigma_v^*)$, β satisfies $\mathrm{St}(\tilde{\varphi}; \tau_i)$ iff β satisfies $\varphi(\sigma_m^*, \sigma_v^*)$. Thus β satisfies $\varphi(\sigma_m^*, \sigma_v^*)$ in $\mathfrak{B}$.

Let α be an assignment obtained by restricting β to variables in σ_m^* and let σ_v^* in α be the same as σ'^*_v in β. So α satisfies $\sigma_m^* < \sigma_m$ and $\varphi(\sigma_m^*, \sigma_v^*)$ in $\mathfrak{A}$. No doubt that this conclusion implies that $\mathfrak{A}$ is not a model of CIRC$[\varphi; \sigma_m; \sigma_v]$ and makes a contradiction. So $\mathfrak{B}$ is a model of SM$[\pi(\varphi); \tau_i]$.

$\Rightarrow$ Conversely we suppose that $\mathfrak{B}$ is a τ-structure satisfying SM$[\pi(\varphi); \tau_i]$. Then we want to prove that $\mathfrak{A}$ is a model of CIRC$[\varphi; \sigma_m; \sigma_v]$. Since $\mathfrak{B}$ satisfies the formula $\varphi^{\neg\neg}$ with a vocabulary of σ, φ is satisfied by $\mathfrak{A}$. Because for all $P \in \sigma_m$, the formulas $\forall\bar{x}(P(\bar{x}) \vee \neg P(\bar{x}))$ are always true, γ is interpreted to $\top$ by $\mathfrak{B}$. Furthermore, all $Q'(\bar{x})$ s.t. $Q' \in \sigma'_v$ are valid in $\mathfrak{B}$ in that $\gamma \rightarrow Q'(\bar{x})$ is satisfied.

Using *reductio ad absurdum*, we assume that α, an assignment in $\mathfrak{A}$, satisfies both $\sigma_m^* < \sigma_m$ and $\varphi(\sigma_m^*, \sigma_v^*)$. Let β be an assignment in $\mathfrak{B}$ s.t. $\beta(\sigma_m^*) = \alpha(\sigma_m^*)$ and $\beta(\gamma^*) = \bot$. Due to the interpretation for σ'_v in $\mathfrak{B}$, whatever predicate variables in σ_v^* are assigned to, β satisfies $\sigma'^*_v \leq \sigma'_v$. Obviously it is true that $\tau_i^* < \tau_i$ in β. It is clear that β satisfies $\mathrm{St}(\varphi^{\neg\neg}; \tau_i)$ iff $\mathfrak{B}$ satisfies φ. Thanks to $\sigma_m^* < \sigma_m$, there is at least a predicate variable $P^* \in \sigma_m^*$ letting $\forall\bar{x}(P^*(\bar{x}) \vee \neg P(\bar{x}))$ be false and $\mathrm{St}((6); \tau_i)$ is satisfied by β. In addition, β satisfies $\mathrm{St}((7); \tau_i)$ and predicate variables Q'^* can change arbitrarily in the domain. Now we let β assign Q'^* in σ'^*_v as same as Q^* in σ_v^* assigned by α. As φ is in NNF without implication, by a simple induction on formula structure, it is proved that β satisfies $\mathrm{St}(\tilde{\varphi}; \tau_i)$ iff $\varphi(\sigma_m^*, \sigma_v^*)$ is satisfied by β. According to the above assumption, we conclude that β satisfies $\tau_i^* < \tau_i$ and $\mathrm{St}(\pi(\varphi); \tau_i)$. Since a contradiction is obtained, we can know that $\mathfrak{A}$ is a model of CIRC$[\varphi; \sigma_m; \sigma_v]$. ■

As for translation Tr_v, the number of auxiliary predicates introduced is one more than that of the varying predicates, which is acceptable and reasonable. Compared with the size[3] of the original sentence, the number of minimized and varying predicates is generally so small that the increased size resulting from them is considered insignificant. Thus translation Tr_v actually doubles the original size.

Conjunction of Parallel Circumscriptions

When it comes to encoding problems using the conjunction of parallel circumscriptions, generally we need to compute them one by one. There seems to be no way to substitute a single parallel circumscription for the conjunction of parallel circumscriptions, except for some special cases corresponding to the same FO sentence and the same tuple of varying predicates. Fortunately, based on the Splitting Theorem in SM-semantics (Ferraris et al. 2009), we can integrate conjunction of parallel circumscriptions into a FO theory under SM-semantics over arbitrary structures and then compute it at one time.

Definition 2 *Let $\varphi_1, \ldots, \varphi_k$ be FO sentences in NNF, $\sigma_1, \ldots, \sigma_k$ be mutually disjoint tuples of distinct predicates, and $\sigma_v^1, \ldots, \sigma_v^k$ be tuples of arbitrary predicates. Then let $Tr_v^s(\varphi_o; \sigma_o; \sigma_v^o)$[4] denote the conjunction of below formulas:*

$$\bigwedge_{1\leq j\leq k} (\varphi_j^{\neg\neg} \wedge \tilde{\varphi}_j) \quad (8)$$

$$\bigwedge_{1\leq j\leq k} (\gamma_j \leftrightarrow \bigwedge_{P\in\sigma_j} \forall\bar{x}(P(\bar{x}) \vee \neg P(\bar{x}))) \quad (9)$$

$$\bigwedge_{1\leq j\leq k} (\gamma_j \rightarrow \bigwedge_{Q^j\in\delta_j} \forall\bar{x}\, Q^j(\bar{x})) \quad (10)$$

where $\varphi_j^{\neg\neg}$ is obtained from $\varphi^{\neg\neg}$ in definition 1 by additionally substituting $\neg\neg P(\bar{x})$ for each positive literal $P(\bar{x})$ if P in $\sigma_i (i \neq j)$ occurs in φ_j, $\tilde{\varphi}_j$ is obtained from φ_j by substituting: (i) $(P(\bar{x}) \rightarrow \gamma_j)$ for each negative literal $\neg P(\bar{x})$ s.t. $P \in \sigma_j$, (ii) $\neg\neg P(\bar{x})$ for each positive literal $P(\bar{x})$ if P in $\sigma_i (i \neq j)$, and (iii) $Q^j(\bar{x})$ for each positive literal $Q(\bar{x})$ and $(Q^j(\bar{x}) \rightarrow \gamma_j)$ for each negative literal $\neg Q(\bar{x})$ s.t. $Q \in \sigma_v^j$; and δ_j denotes the tuple of Q^j for the j-th sentence.

Translation Tr_v^s actually applies translation Tr_v to each FO sentence respectively with a little modification. To avoid the strictly positive[5] occurrences of intensional predicates, translation Tr_v^s adds $\neg\neg$ in front of those minimized predicates occurring in other sentences. Intuitively, γ_j reflects the minimization of the j-th circumscription.

[2] To take no account of the interpretation of auxiliary predicates, we use $\exists P\psi$ represent the formula obtained from ψ by substituting predicate variable P for predicate constant P.

[3] We define size to be the number of connectives and atoms.

[4] We use φ_o as a shorthand for $\varphi_1, \varphi_2, \ldots, \varphi_k$ if no confusion occurs. σ_o, σ_v^o, γ_o, and δ_o are similar shorthands.

[5] For space limitations, please refer to (Ferraris et al. 2009).

Proposition 2 *Let φ_o be k FO sentences in NNF, σ_o be k mutually disjoint tuples of distinct predicates, and σ_v^o be k tuples of arbitrary predicates. Then $\exists\delta_o\exists\gamma_o SM[Tr_v^s(\varphi_o;\sigma_o;\sigma_v^o);\sigma_o,\delta_o,\gamma_o]$ is equivalent to $\bigwedge_{1\leq j\leq k} CIRC[\varphi_j;\sigma_j;\sigma_v^j]$ by omitting auxiliary predicates.*

Proof:[sketch] Using translation Tr_v with a little modification, we can translate each CIRC$[\varphi_j;\sigma_j;\sigma_v^j]$ into a FO sentence under SM-semantics, shortly denoted by $Tr_v(\varphi_j)$. Because φ_j is in NNF without implication, in the predicate dependency graph[5] of $Tr_v(\varphi_j)$, there is no edge being introduced by $\varphi_j^{\neg\neg}$ and $\tilde{\varphi}_j$ except for subformulas in form of $(P \rightarrow \gamma_j)$. According to the construction rule of predicate dependence graph, formula (6) in Tr_v also introduces no edge. Next, those subformulas $(P \rightarrow \gamma_j)$ in $\tilde{\varphi}_j$ only introduce edges from γ_j to P. In addition, formula (7) in Tr_v only introduces edges from auxiliary predicates in δ_j to γ_j. Therefore, strongly connected component can only occur in $\delta_j \cup \{\gamma_j\}$. In accordance with the substitutions of $\neg\neg$ in Tr_v^s, predicates in σ_j have no strictly positive occurrence in $\varphi_i^{\neg\neg}$ and $\tilde{\varphi}_i$ s.t. $i \neq j$. Additionally, all auxiliary predicates of φ_j are newly introduced, so they cannot occur in other sentences. Indeed, with the precondition of the Splitting Theorem satisfied, $\bigwedge_{1\leq j\leq k}$ SM$[Tr_v(\varphi_j);\sigma_j,\delta_j,\gamma_j]$ is equivalent to SM$[Tr_v^s(\varphi_o;\sigma_o;\sigma_v^o);\sigma_o,\delta_o,\gamma_o]$. So this proposition is proved and shows translation Tr_v^s is faithful. ■

Prioritized Circumscription

Next, we further consider the computation of prioritized circumscription via converting it into a FO theory under SM-semantics. Indeed, a prioritized circumscription can be represented by the conjunction of parallel circumscriptions, which was presented by Proposition 15 in (Lifschitz 1994): CIRC$[\varphi;\sigma_1 > \ldots > \sigma_k;\sigma_v]$ is equivalent to the conjunction of CIRC$[\varphi;\sigma_j;\bigcup_{j<i\leq k}\sigma_i,\sigma_v]$ s.t. $1 \leq j \leq k$.

Based on translation Tr_v^s, we can integrate parallel circumscriptions as conjuncts to compute a prioritized circumscription at a time rather than at k times. Because all priorities of minimized predicates are disjoint, translation Tr_v^s can be applied to compute prioritized circumscription, shown in Example 2. Besides, these parallel circumscriptions share the same FO sentence φ, so there is only one $\varphi^{\neg\neg}$ in the resulting sentence. Further we can adapt translation Tr_v^s to compute the conjunction of prioritized circumscriptions.

Next we analyze translation Tr_v^s applied to prioritized circumscription. Suppose that there are m_j minimized predicates in j-th priority and n varying predicates. Then the number of auxiliary predicates introduced is $\sum_{j=1}^{k}(j-1)m_j + k(n+1)$. Because the size of the original sentence is much greater than the number of minimized and varying predicates, translation Tr_v^s expands into $k+1$ times in size.

Example 2 *For CIRC$[\varphi;Ab_1 > Ab_2;Pacifist]$ in Example 1, by applying Tr_v^s, we can get a sentence under SM-semantics. Here only show the result for $1st$ priority $\{Ab_1\}$:*

$$\neg Quaker(x) \vee Ab_1(x) \vee Pacifist^1(x) \quad (11)$$

$$\neg Republican(x) \vee Ab_2^1(x) \vee (Pacifist^1(x) \rightarrow \gamma_1) \quad (12)$$

$$\gamma_1 \leftrightarrow \forall x(Ab_1(x) \vee \neg Ab_1(x)) \quad (13)$$

$$\gamma_1 \rightarrow Pacifist^1(x) \wedge Ab_2^1(x) \quad (14)$$

Optimization and Computation

In the above section, we introduce translations from general FO circumscription into FO theories under stable model semantics over arbitrary structures. According to (Cabalar and Ferraris 2007), every sentence in PNF without existential quantifiers under SM-semantics can be translated easily into ASP. However, because of the arbitrariness of FO sentence, it is essential to eliminate existential quantifiers.

Optimization in Elimination Existential Quantifiers

With Zhang's reduction (Zhang et al. 2011), existential quantifiers in SM-semantics can be eliminated over finite structures. Based on his approach, we propose an optimization to introduce fewer auxiliary predicates and downsize logic programs when eliminating existential quantifiers during translating general circumscription to ASP.

Definition 3 *Let φ be a FO sentence in PNF of the form $\forall\bar{x}\exists\bar{y}\vartheta(\bar{x},\bar{y})$ where ϑ is quantifier-free in NNF, and we define optimization $\Gamma_\exists(\varphi;\sigma_m;\sigma_v)$ to be the conjunction of below formulas with universal quantifiers omitted:*

$$\neg\neg S(\bar{x},\overline{\min}) \quad (15)$$

$$(succ(\bar{y},\bar{y}') \wedge S(\bar{x},\bar{y}')) \vee \vartheta^{\neg\neg}(\bar{x},\bar{y}) \rightarrow S(\bar{x},\bar{y}) \quad (16)$$

$$(succ(\bar{y},\bar{y}') \wedge W(\bar{x},\bar{y}')) \vee \vartheta^{\top}(\bar{x},\bar{y}) \rightarrow W(\bar{x},\bar{y}) \quad (17)$$

$$T(\bar{x},\overline{\min}) \vee \tilde{\vartheta}(\bar{x},\overline{\min}) \quad (18)$$

$$succ(\bar{y},\bar{y}') \rightarrow (T(\bar{x},\bar{y}) \leftrightarrow \tilde{\vartheta}(\bar{x},\bar{y}') \vee T(\bar{x},\bar{y}')) \quad (19)$$

$$\begin{aligned}&\{[succ(\bar{y},\bar{y}')\wedge\neg W(\bar{x},\bar{y}')] \vee \bar{y}=\overline{\max}\}\wedge W(\bar{x},\bar{y})\\&\quad\rightarrow (T(\bar{x},\overline{\max}) \leftrightarrow \tilde{\vartheta}(\bar{x},\bar{y}))\end{aligned} \quad (20)$$

$$\gamma \leftrightarrow \bigwedge_{P\in\sigma_m} \forall\bar{x}(P(\bar{x}) \vee \neg P(\bar{x})) \quad (21)$$

$$\bigwedge_{Q'\in\sigma_v'} \forall\bar{x}(\gamma \rightarrow Q'(\bar{x})) \quad (22)$$

where $\vartheta^{\neg\neg}$ and $\tilde{\vartheta}$ are obtained from $\varphi^{\neg\neg}$ and $\tilde{\varphi}$ respectively in Definition 1; $\vartheta^{\top}$ is obtained from ϑ by substituting: (i) $\top$ for each positive literal $Q(\bar{t})$ s.t. $Q \in \sigma_v$ and for each negative literal $\neg P(\bar{t})$ s.t. $P \in \sigma_m \cup \sigma_v$, (ii)$\neg\neg P(\bar{t})$ for each positive literal $P(\bar{t})$ s.t. $P \in \sigma_m$; succ is a successor relation on the domain based on a total order; $\overline{\max}$ and $\overline{\min}$ denote the maximum and minimum tuple on the successor relation respectively; S, T, W, γ, and σ_v' are auxiliary predicates.

Optimization $\Gamma_\exists$ integrates translation Tr_v and Zhang's reduction from general FO parallel circumscription to FO theories under SM-semantics. Now there are two ways which differ in the first step to eliminate the first block of continuous existential quantifiers. The first way calls Zhang's reduction after applying Tr_v while the second uses $\Gamma_\exists$. Next both ways use Zhang's reduction repeatedly till eliminating all existential quantifiers.

After the first step, the second way introduces one fewer auxiliary predicate than those introduced by the first way. Besides, suppose the size of the original sentence is n, then the result of the first way is $8\times n$, while that of the second is $5\times n$. With iterations increasing, the size grows rapidly. Eventually, $\Gamma_\exists$ reduces logic programs in size by one third.

We can easily generalize $\Gamma_\exists$ with Tr_v^s for prioritized circumscription, which is analyzed in the next section.

Computing Circumscription via ASP

The above translations and optimization actually can be applied to each conjunct of the original sentences, so they can be applied more flexibly. For example, CIRC$[\exists y\vartheta(y) \wedge \varphi]$ is equivalent to SM$[\Gamma_\exists(\exists y\vartheta(y)) \wedge Tr_v(\varphi)]$.

Eventually, fixed and varying predicates are treated as extensional predicates in SM-semantics rather than being removed from the formula. Extensional predicates can be handled easily in ASP, by introducing $\forall\bar{x}(P(\bar{x}) \vee \neg P(\bar{x}))$(Ferraris, Lee, and Lifschitz 2011). To sum up, we can compute parallel circumscription with varying predicates by 4 steps:

1. Turn the input into the sentence in both PNF and NNF;
2. Apply Tr_v or $\Gamma_\exists$ to get a theory under SM-semantics;
3. Use Zhang's reduction repeatedly till obtaining ASP;
4. Add $\forall\bar{x}(P(\bar{x})\vee\neg P(\bar{x}))$ for each fixed/varying predicate.

With the similar method, we can generalize the computation to suit prioritized circumscription.

Remark 1 *As for the above translations, we have not mentioned function constants and individual constants, because for each n-arity function we can introduce a n+1-arity predicate to represent it. Precisely, $\exists y P_f(\bar{x}, y)$ can simulate $f(\bar{x})$ with a restriction of $\neg(P_f(\bar{x}, y) \wedge P_f(\bar{x}, z) \wedge y \neq z)$. We can use such predicates rather than functions as varying constants. Particularly, varying individual constants can be simulated by existential quantifiers.*

Some Experimental Results

We developed a general FO circumscription solver cfo2lp[6]. cfo2lp firstly accepts a circumscription, then translates it into a logic program, and finally invokes an ASP solver with a finite extensional database[7]. To compare cfo2lp with a propositional circumscription solver circ2dlp[8] and an ASP solver supporting inclusion-based minimization metasp[9], we use *circuit diagnosis problem (CDP)* and *extended stable marriage problem (ESMP)* as benchmarks.

Circuit Diagnosis Problem

According to (Reiter 1987; Besnard and Cordier 1994), CDP is stated: given a circuit and its observation, find a minimal explanatory diagnosis, which is a set of error components.

We use a *n*-bit ripple adder as our circuit consisting of *n* full adders which include *and* gate, *xor* gate, and *or* gate. Let I_1 and I_2 be the inputs and *O* be the output of the gate. Then we can use FO sentence φ to represent the total ripple adder and the observation of its inputs and outputs. Based on engineering experiences, different kinds of circuit component have different error probability. Intuitively, We can assign a higher priority in circumscription to the components of lower error probability. Suppose that *and* gate is more likely in error than *xor* and so is *xor* than *or*. We can obtain a minimal explanatory diagnosis Δ iff Δ is a model of CIRC$[\varphi; Ab_o > Ab_x > Ab_a; I_1, I_2, O]$.

Extended Stable Marriage Problem

According to (Mairson 1992), the stable marriage problem is the problem of finding a stable match between men and women given their respective preference list. Let *block pair* represent a pair of man and woman who are not partners, both prefer each other to their current partners in the *match*. A match is stable if it contains no block pair. (Iwama et al. 1999) extended it with both *incomplete* lists and *ties*, making the complexity become NP-hard.

Based on Iwama *et al.*'s extension, we further extend it with "satisfaction" to make it more realistic in practice. With this notion, we aim to find optimal stable marriages which there are as few as possible people unsatisfying. Next, we represent our extension with prioritized circumscription. To represent the preference lists, let $Grade(x, p, y)$ denote person x grades person y with a natural number p. Let $H_1(x)$ denote that the current partner of a person x is graded by x with a grade not exceeding one third of the number of pairs, which is considered that x is unsatisfied with the match extremely. Similarly, we define $H_2(x)$ to represent x is somewhat unsatisfied, if the grade is in the intermediate third. To get the minimality of extremely unsatisfied people and secondly somewhat unsatisfied people, we compute the circumscription with the priority of $H_1 > H_2$. We can use a FO sentence φ with existential quantifiers to describe ESMP naturally. By computing CIRC$[\varphi; H_1 > H_2; Partner]$, the interpretation of $Partner$ is regarded as a solution.

Experimental Results

Table 1 (Table 2) shows the performance comparisons among cfo2lp, circ2dlp and metasp scale up, when the number n of gates in the ripple adder (persons in ESMP) grows. Real numbers in the tables figure the run time (in seconds) of solvers to compute circumscription. If the time exceeds one hour, we simply write it as "–". All experiments run on a PC with AMD A10-5800K 3.8GHz CPU on Linux Ubuntu 13.04. Each instance was randomly generated, which was computed five times and taken the average by calling the same ASP solver claspD[10] as back-end.

Table 1: Experiments for Circuit Diagnosis Problem

	Instance 1			Instance 2			Instance 3		
n	cfo2lp	metasp	circ2dlp	cfo2lp	metasp	circ2dlp	cfo2lp	metasp	circ2dlp
5	0.000	0.000	0.000	0.020	0.000	0.000	0.012	0.000	0.000
10	0.018	0.018	0.000	0.052	0.012	0.000	0.018	0.004	0.000
20	0.182	0.080	0.012	0.120	0.016	0.016	0.124	0.080	0.088
40	0.384	0.142	9.860	1.094	0.184	9.722	0.342	0.146	9.646
80	1.988	2.588	—	8.484	1.924	—	4.112	2.782	—
160	33.562	35.240	—	9.062	24.642	—	9.288	23.626	—
200	54.404	79.682	—	22.484	95.024	—	62.740	73.944	—

Table 2: Experiments for Extended Stable Marriage Problem

	Instance 1				Instance 2			
n	cfo2lp*	cfo2lp	metasp	circ2dlp	cfo2lp*	cfo2lp	metasp	circ2dlp
6	0.020	0.012	0.012	39.582	0.056	0.036	0.012	11.122
12	0.662	0.442	0.052	—	0.682	0.488	0.040	—
18	8.420	6.668	0.264	—	12.240	8.224	0.566	—
24	49.506	43.124	3.166	—	38.988	32.682	4.582	—
30	59.582	44.912	4.360	—	30.112	12.088	1.344	—
36	253.284	173.760	66.762	—	233.448	201.780	24.440	—
42	369.146	152.180	6.360	—	757.820	537.024	49.224	—

Actually, cfo2lp and metasp have comparative performances which are better than those of circ2dlp in CDP.

[6] cfo2lp. http://ss.sysu.edu.cn/%7ewh/cfo2lp.html

[7] An extensional database is a structure consisting of extensional predicate and function constants under SM-semantics.

[8] circ2dlp. http://www.tcs.hut.fi/Software/circ2dlp/

[9] metasp. http://www.cs.uni-potsdam.de/wv/metasp/

[10] claspD. http://www.cs.uni-potsdam.de/claspD/

While in ESMP, metasp has the best performance and cfo2lp is ranked second because of the big size of logic program resulting from existential quantifiers.

To analyze optimization $\Gamma_\exists$ for prioritized circumscription, we remove the optimization from cfo2lp and implement cfo2lp* (in Table 2). As its performances show, optimized logic programs can save 10-60% time cost.

Indeed, experimental results show that our approach can solve problems represented by circumscription effectively.

Applications

This section applies cfo2lp to DLs with circumscription and finding minimal repairs in inconsistent databases, both of which can be handled effectively.

Description Logics with Circumscription

To extend DLs with non-monotonic features, (Bonatti, Lutz, and Wolter 2006) proposed that parallel and prioritized circumscription can be used in a straightforward and transparent way for modelling defeasible inheritance.

Example 3 [Example in (Bonatti, Lutz, and Wolter 2006)]

$$Mammal \sqsubseteq \exists habitat.Land \sqcup Ab_M \tag{23}$$
$$Whale \sqsubseteq Mammal \sqcap \neg\exists habitat.Land \tag{24}$$

Indeed, concept Ab_M can be regarded as abnormality of mammals. Intuitively, mammals normally live on land and whales are mammals not inhabiting land. We can use different circumscription policies to obtain different assertions of concepts. One is only circumscribing the abnormality predicates by computing CIRC$[(23) \wedge (24); Ab_M;]$. However, this policy is too strong, because all concepts except abnormality predicates, such as *Whale*, have no any change actually and it is not natural and intuitive for finding more exact and smaller inclusions. When we vary concept *Land* and role *Habitat* freely and compute CIRC$[(23) \wedge (24); Ab_M; Habitat, Land]$, we can conclude that mammals are almost likely to live on land except for whales.

By applying our approach to such DLs, we can obtain assertions of all concepts and roles of all minimal models as the computation result, which provides a reference in Herbrand models over a fixed set of individuals.

Repairs in Inconsistent Databases

Barceló and Bertossi (2003) represented a repair, which minimally modifies inconsistent database instances, by an ASP program. A repair program encodes function dependency with the restriction on a FO sentence without existential quantifiers. Bertossi (2011) further generalized repairs with prioritized circumscription, in such order: database predicates, predicates with annotation, and query predicates. Using our approach, we can relax the restriction of function dependency to allow true FO quantifiers. By our approach, we can find all minimal repairs in inconsistent databases.

Related Work and Discussions

In (Cadoli, Eiter, and Gottlob 1992), varying predicates can be complied away only in inference which decides whether a formula is entailed by a circumscription, but we are more interested in finding all minimal models. To find all models, we consider all interpretations of varying predicates because they would make structures different.

An embedding of FO circumscription in SM-semantics has been shown in (Zhang et al. 2011), but it forbids constants to vary. However, the policy of circumscription forbidding varying predicates is often too strong for many applications, such as Example 3. It is more natural and intuitive to allow varying constants when finding more exact and smaller models. Besides, our approach can integrate parallel circumscriptions to compute them at one time, even further generalized to prioritized circumscription.

There is a choice between encoding naturalness and computation efficiency. circ2dlp (Oikarinen and Janhunen 2008) can compute prioritized circumscription in the propositional case and metasp (Gebser, Kaminski, and Schaub 2011) can find inclusion-based minimal answer sets in extended logic programs under a priority order. They both may need an unintuitive and complicated input. However, we focus on encoding naturalness rather than computation efficiency. With the true FO quantifiers, we can represent problems naturally and succinctly. Besides, for problems which need to be encoded with existential quantifiers, such as ESMP, an exponential expansion in size probably occurs so that the computation may become intractable in circ2dlp. For these problems, metasp represents them with constraints instead of existential quantifiers and computes more efficiently. While cfo2lp may cause a logic program of big size resulting from existential quantifiers which affects the efficiency.

In many applications, we can take advantage of FO quantifiers and represent problems more naturally and succinctly. For an ASP layman, compared with other efficient ASP-based solvers, it is easier to use FO circumscription to represent problems. Our approach can close the gap between representation naturalness and computation efficiency.

Conclusion

The relationship between FO circumscription and FO theories under SM-semantics has been clarified in this paper. Furthermore, we proposed and proved linear translations from general FO parallel and prioritized circumscription to FO theories under SM-semantics over arbitrary structures. Based on the translations, all minimal models of FO theories can be computed effectively via ASP solver. Our approach is not only theoretically interesting but of practical relevance. We can apply it to compute minimal models in DLs with circumscription and minimal repairs in inconsistent databases.

Now we summarize the contributions of this paper. First, we proposed a practically available framework of computation for general FO parallel and prioritized circumscription with varying constants over finite structures, which builds a bridge between circumscription and theories under SM-semantics. Secondly, based on a total order, we optimized the elimination of existential quantifiers to figure out practical problems represented by general circumscription and developed a solver cfo2lp. Finally, allowing true FO quantifiers, varying constants, and priorities in circumscription provides a flexible and natural way to represent problems so that our approach can be applied in more fields.

Acknowledgments

The authors would like to thank Yongmei Liu and her research group for their helpful and informative discussions. We are also grateful to Kewen Wang for his useful suggestions. Hai Wan thanks Research Fund for the Doctoral Program of Higher Education of China (No. 20110171120041), Natural Science Foundation of Guangdong Province of China (No. S2012010009836), Guangzhou Science and Technology Project (No. 2013J4100058), and Natural Science Foundation of China (No. 61073053) for the support of this research.

References

Barceló, P., and Bertossi, L. 2003. Logic programs for querying inconsistent databases. In *Practical Aspects of Declarative Languages*. Springer. 208–222.

Bertossi, L. 2011. Database repairing and consistent query answering. *Synthesis Lectures on Data Management* 3(5):1–121.

Besnard, P., and Cordier, M. 1994. Explanatory diagnoses and their characterization by circumscription. *Annals of Mathematics and Artificial Intelligence* 11(1-4):75–96.

Bonatti, P.; Lutz, C.; and Wolter, F. 2006. Description logics with circumscription. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR-06)*, 400–410.

Cabalar, P., and Ferraris, P. 2007. Propositional theories are strongly equivalent to logic programs. *Theory and Practice of Logic Programming* 7(6):745–759.

Cadoli, M.; Eiter, T.; and Gottlob, G. 1992. An efficient method for eliminating varying predicates from a circumscription. *Artificial Intelligence* 54:397–410.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 1997. Computing circumscription revisited: a reduction algorithm. *Journal of Automated Reasoning* 18(3):297–336.

Ferraris, P.; Lee, J.; Lifschitz, V.; and Palla, R. 2009. Symmetric splitting in the general theory of stable models. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 797–803.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2007. A new perspective on stable models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 372–379.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2011. Stable models and circumscription. *Artificial Intelligence* 175(1):236–263.

Gebser, M.; Kaminski, R.; and Schaub, T. 2011. Complex optimization in answer set programming. *Theory and Practice of Logic Programming* 11(4-5):821–839.

Iwama, K.; Manlove, D.; Miyazaki, S.; and Morita, Y. 1999. Stable marriage with ties and incomplete lists. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP-99)*, 443–452.

Kim, T.; Lee, J.; and Palla, R. 2009. Circumscriptive event calculus as answer set programming. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 823–829.

Lee, J., and Palla, R. 2009. System F2LP–computing answer sets of first-order formulas. In *Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-09)*, 515–521.

Lee, J., and Palla, R. 2012. Reformulating the situation calculus and the event calculus in the general theory of stable models and in answer set programming. *Journal of Artificial Intelligence Research* 43(1):571–620.

Lifschitz, V. 1994. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming-Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford: Clarendon Press. 297–352.

Lin, F., and Zhou, Y. 2011. From answer set logic programming to circumscription via logic of GK. *Artificial Intelligence* 175(1):264–277.

Mairson, H. 1992. The stable marriage problem. *The Brandeis Review* 12(1):37–41.

McCarthy, J. 1980. Circumscription – a form of nonmonotonic reasoning. *Artificial Intelligence* 13(1,2):27–39,171–172.

McCarthy, J. 1986. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence* 28(1):89–116.

Oikarinen, E., and Janhunen, T. 2005. CIRC2DLP-translating circumscription into disjunctive logic programming. In *Proceedings of the 8th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-05)*, 405–409.

Oikarinen, E., and Janhunen, T. 2008. Implementing prioritized circumscription by computing disjunctive stable models. In *Artificial Intelligence: Methodology, Systems, and Applications*. Springer. 167–180.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32(1):57–96.

Wakaki, T., and Inoue, K. 2004. Compiling prioritized circumscription into answer set programming. In *Proceedings of the 20th International Conference on Logic Programming (ICLP-04)*, 356–370.

Zhang, H.; Zhang, Y.; Ying, M.; and Zhou, Y. 2011. Translating first-order theories into logic programs. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, 1126–1131.