Point-Sensitive Circumscription Computation via Answer Set Programming and Applications
(Preliminary Report)

Hai Wan[1] **and Zhanhao Xiao**[2] **and Rui Yang**[2] **and Pu Wang**[2]
[1]Software School, Sun Yat-Sen University, P.R. China 510275
[2]School of Information Science and Technology, Sun Yat-Sen University, P.R. China 510275
wanhai@mail.sysu.edu.cn, zhanhaoxiao@gmail.com, yangrui_yr90@163.com, wangpu96@gmail.com

Abstract

As a generalization of both McCarthy's circumscription and Lifschitz's pointwise circumscription, Amir's point-sensitive circumscription is a useful tool for formalizing solutions for Theories of Action. Point-sensitive circumscription can maintain the control of the minimization process over selective fine-grained variance of predicates and functions, whereas the computation and applications still remain unsatisfactory. Thus, the practical value of point-sensitive circumscription has been severely restricted. In this paper, we propose translation Tr^{ps} from point-sensitive circumscription to first-order stable model semantics over arbitrary structures. Based on the reduction from stable model semantics to answer set programming over finite structures, a point-sensitive circumscription solver, named psc2lp, is developed. We also provide a situation calculus style encoding method by examining a variant of the Yale Shooting Scenario, to solve the Frame Problem and the Qualification Problem. These problems in situation calculus represented naturally by point-sensitive circumscription, can be handled by our approach using existing answer set solvers effectively.

1 Introduction

Point-sensitive circumscription, devised by Amir in [Amir, 1997; 1998], is a nonmonotonic method defined along the intuitions of pointwise circumscription [Lifschitz, 1986; 1987a; 1987b] with control of the minimization process over selective fine-grained variance of predicates and functions. Amir's point-sensitive circumscription can be considered as a generalization of both McCarthy's circumscription [McCarthy, 1980; 1986] and Lifschitz's pointwise circumscription, and is a useful tool for finding the Frame Problem and the Qualification Problem solutions for Theories of Action.

Pointwise circumscription has been used in formalizing some Entailment Classes in the theory of Features and Fluents [Sandewall and Shoham, 1995; Iwanuma and Oota, 1996; Iwanuma *et al.*, 2009], and in [Amir, 1997]. It was argued in [Lifschitz, 1987a] and [Doherty and Łukaszewicz, 1994] that pointwise circumscription has the power to be a tool for formalizing solutions for the Frame Problem.

The unintended minimal model which pointwise circumscription can capture is the insistence on minimizing one fluent at a time, while not allowing other changes for other fluents for the same time point in situation calculus. Compared with pointwise circumscription, point-sensitive circumscription might be to minimize changes for all the fluents once for a given time point. However, despite the progresses on theoretical aspects, the computation of point-sensitive circumscription still remains unsatisfactory, which encounters difficulties from a practical viewpoint. On the other hand, the basic case (circumscribing predicates with no other constants varied) for pointwise circumscription can be rewrite as a first-order sentence, whereas this property of generalized pointwise circumscription and point-sensitive circumscription disappears, which makes the computation of point-sensitive circumscription difficult.

This paper intends to address this issue by translating point-sensitive circumscription into Answer Set Programming (ASP) [Gelfond and Lifschitz, 1988], a promising approach that has been successfully implemented by a number of sophisticated solvers [Drescher *et al.*, 2008; Leone *et al.*, 2006]. To the authors' knowledge, we have found no solvers for computing point-sensitive circumscription.

In this paper, we propose a computation approach and implement such a solver. First, we present a translation from point-sensitive circumscription to first-order stable model semantics over arbitrary structures: Tr^{ps}. The translation has been proved faithful. Over finite structures, point-sensitive circumscription under stable model semantics can be translated into answer set programming. Based on this translation, we can compute point-sensitive circumscription by using existing ASP solvers, over finite structures. Secondly, based on the first-order stable model solver - T2LP [Zhang *et al.*, 2011; Zhang, 2011], we implement a solver, named psc2lp, for computing arbitrary point-sensitive circumscription. Thirdly, we also provide a situation calculus style encoding method for Reiter's basic action theories by examining a variant of the Yale Shooting Scenario, which demonstrates the Frame Problem and the Qualification Problem in situation calculus represented naturally by point-sensitive circumscription can be handled by our approach effectively.

2 Preliminaries

2.1 McCarthy's Circumscription

McCarthy's circumscription [1980] is one of the first and major nonmonotonic reasoning tools. We follow the notions of *parallel circumscription* in [Lifschitz, 1994]. Logic symbols used in this paper are defined as usual. Let φ be a first-order sentence, σ_i be a tuple of $minimized\ predicate\ constants$ and σ_v be a tuple of $individual$, $function$, or $predicate\ constants$ totally differing from σ_i. The rest of the vocabulary of φ are called $fixed\ constants$. Let σ_i^* be a tuple of predicate variables with the same arity as predicate constants in σ_i respectively; similarly, let σ_v^* be a tuple of corresponding variables of same arity as constants in σ_v respectively.

Before defining circumscription, we introduce a comparison relation $<$ between two predicate tuples. Moreover, we abbreviate 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 $P^* \in \sigma_i^*$ and $P \in \sigma_i$, to $\sigma_i^* = \sigma_i$ (*resp.* $\sigma_i^* \leq \sigma_i$). Therefore, we let the comparison $\sigma_i^* < \sigma_i$ stand for the formula $(\sigma_i^* \leq \sigma_i) \wedge \neg(\sigma_i^* = \sigma_i)$. Then parallel circumscription of σ_i for φ with σ_v varied is defined by a second-order formula:

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

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

Intuitively, circumscription makes the extension of some predicates minimal under the precondition guaranteing the validity of φ, where the extension of predicate P is a set of elements in the domain letting P be true. A structure μ is a σ_i-*minimal model* of φ with σ_v varied if it is a model of $\mathrm{CIRC}[\varphi; \sigma_i; \sigma_v]$.

2.2 Lifschitz's Pointwise Circumscription

Pointwise circumscription (basic case) was first proposed in [Lifschitz, 1986] and then expanded in [Lifschitz, 1987a; 1987b], called *generalized pointwise circumscription*. The basic case for pointwise circumscription is the formula:

$$\varphi(P) \wedge \forall x \neg[P(x) \wedge \varphi(\lambda y(P(y) \wedge x \neq y))]$$

where we minimize the predicate P with no other constants varied. Intuitively, it shows that it is impossible to make the extension of exactly one minimized predicate smaller by changing it at exactly one point. One of the benefits of such an approach involves general first-order circumscriptive theories. This property disappears in generalized pointwise circumscription and point-sensitive circumscription.

Let $\varphi(P, Z)$ be a sentence, where P represents a predicate constant and Z denotes a list of predicate constants or function constants Z_i(in particular, a 0-arity function constant is an individual constant). We write $EQ_V(P, Q)$ for

$$EQ_V(P, Q) \stackrel{\mathrm{def}}{=} \forall x(\neg V(x) \rightarrow (P(x) \leftrightarrow Q(x))) \quad (2)$$

where P, Q, V are predicates with same arity. If P, Q are function constants of the same arity as predicate constant V, then $EQ_V(P, Q)$ stands for $\forall x(\neg V(x) \rightarrow (P(x) = Q(x)))$. Intuitively, the formula $EQ_V(P, Q)$ denotes that P and Q are equal outside V.

The generalized pointwise circumscription of P in φ with Z_i varied on V_i is, by definition,

$$\begin{aligned} &\mathrm{C_{PW}}[\varphi; P; Z_1/V_1, \ldots, Z_n/V_n] = \\ &\qquad \varphi(P, Z) \wedge \forall x P^* Z^* \neg[P(x) \wedge \neg P^*(x) \wedge \\ &\qquad\qquad \bigwedge_{1 \leq i \leq k} EQ_{V_i x}(Z_i^*, Z_i) \wedge \varphi(P^*, Z^*)] \end{aligned} \quad (3)$$

where P^* is an auxiliary predicate corresponding to minimized predicate P, Z^* is a list $Z_1^*, \ldots, Z_n^*$ of predicate and function variables corresponding to the predicate and function constants Z, and $\lambda x u V_i(x, u)(i = 1, \ldots, n)$ is a predicate without parameters which does not contain $Z_1, \ldots, Z_n$ and whose arity is the arity of P plus the arity of Z_i.

For a model $\mathcal{M}$ of $\varphi(P, Z)$, let $|\mathcal{M}|$ be the associated universe, and for every term, function or predicate a, $a^{\mathcal{M}}$ is the realization of a in $\mathcal{M}$.

Definition 1. (Definition 2.1 in [Amir, 1998]) *Let $\mathcal{M}_1$, $\mathcal{M}_2$ have the same universe U, and let $\xi \in U^k$, where k is the arity of P. We say that $\mathcal{M}_1 \leq^\xi \mathcal{M}_2$ iff:*

1. *$K^{\mathcal{M}_1} = K^{\mathcal{M}_2}$ for every function or predicate constant K that is neither P nor in Z,*
2. *for any $i = 1, \ldots, n$, $Z_i^{\mathcal{M}_1}$ and $Z_i^{\mathcal{M}_2}$ coincide on $\{\eta | \neg V_i^{\mathcal{M}_1}(\xi, \eta)\}$,*
3. *$P^{\mathcal{M}_1}(\xi) \rightarrow P^{\mathcal{M}_2}(\xi)$.*

Let $[\varphi(P, Z)]$ be the set of models of $\varphi(P, Z)$. The following proposition 1 says that every model of circumscription for $\varphi(P, Z)$ is minimal in $[\varphi(P, Z)]$ according to the orders $\leq^\xi$.

Proposition 1. ([Lifschitz, 1987a]) *Let $\mathcal{M}$ be a model of $\varphi(P, Z)$. $\mathcal{M} \models \mathrm{C_{PW}}[\varphi; P; Z_1/V_1, \ldots, Z_n/V_n]$ iff for each $\xi \in \mathcal{M}^k$, $\mathcal{M}$ is minimal relative to $\leq^\xi$.*

$$\begin{aligned} &\mathcal{M} \models \mathrm{C_{PW}}[\varphi; P; Z_1/V_1, \ldots, Z_n/V_n] \equiv \\ &\qquad \forall \mathcal{M}' \in [\varphi(P, Z)]\, \forall \xi \in |\mathcal{M}|^k \\ &\qquad\qquad \neg(\mathcal{M}' \leq^\xi \mathcal{M} \wedge \mathcal{M} \not\leq^\xi \mathcal{M}') \end{aligned}$$

2.3 Amir's Point-Sensitive Circumscription

Amir [1998] presented *point-sensitive circumscription*, a modified version of pointwise circumscription in which the minimized predicate is minimized according to a minimization domain. This minimization domain may be a point and may be the complete set of elements, which preserve the ability to select/vary parts of the theory/domain dynamically.

We use similar notations to those used in generalized pointwise circumscription. We want to minimize a predicate constant P in a appropriate region. In addition to the definition of EQ_V in (2), let us write $LS_R(P, Q)$ for

$$\begin{aligned} LS_R(P, Q) \stackrel{\mathrm{def}}{=} &\forall x(R(x) \rightarrow (P(x) \rightarrow Q(x)) \\ &\wedge \exists x(R(x) \wedge \neg P(x) \wedge Q(x))) \end{aligned} \quad (4)$$

where P, Q predicates or functions, and R a predicate, all with same arity. Intuitively, the formula $LS_R(P, Q)$ denotes that $P \cap R \subsetneqq Q \cap R$ or in the other word, the predicate P is smaller than Q in the region R.

The point-sensitive circumscription of P in φ with Z_i varied on V_i and P minimized using R is defined below,

$$\begin{aligned}\mathrm{C_{PS}}[\varphi; P/R; Z_1/V_1, \ldots, Z_n/V_n] = \\ \varphi(P,Z) \wedge \forall x P^* Z^* \neg[LS_{Rx}(P^*, P) \wedge \\ \bigwedge_{i=1}^{n} EQ_{V_i x}(Z_i^*, Z_i) \wedge \varphi(P^*, Z^*)]\end{aligned} \tag{5}$$

where P^* is an auxiliary predicate corresponding to minimized predicate P, Z^* is a list $Z_1^*, \ldots, Z_n^*$ of predicate and function variables corresponding to the predicate and function constants Z, and $\lambda xuR(x,u)$, $\lambda xuV_i(x,u)(i = 1,\ldots,n)$ are predicates without parameters which do not contain $Z_1,\ldots,Z_n$ and whose arity both are the arity of P plus the arity Z_i.

For a model $\mathcal{M}$ of $\varphi(P,Z)$, let $|\mathcal{M}|$ be the associated universe, and for every term, function or predicate a, $a^{\mathcal{M}}$ is the realization of a in $\mathcal{M}$.

Definition 2. (Definition 4.1 in [Amir, 1998]) *Let $\mathcal{M}_1$, $\mathcal{M}_2$ have the same universe U, and let $\xi \in U^k$, where k is the arity of P. We say $\mathcal{M}_1 \ll^{\xi} \mathcal{M}_2$ (a strict partial order) iff:*

1. *$K^{\mathcal{M}_1} = K^{\mathcal{M}_2}$ for every function or predicate constant K that is neither P nor in Z,*
2. *for any $i = 1,\ldots,n$, $Z_i^{\mathcal{M}_1}$ and $Z_i^{\mathcal{M}_2}$ coincide on $\{\eta | \neg V_i^{\mathcal{M}_1}(\xi,\eta)\}$,*
3. *$LS_{R(\xi)}(P^{\mathcal{M}_1}, P^{\mathcal{M}_2})(R(\xi) = \lambda u R(\xi,u))$.*

The following proposition says that every model of the circumscription formula for $\varphi(P,Z)$ is minimal in $[\varphi(P,Z)]$ according to all of the orders $\ll^{\xi}$, and vice versa.

Proposition 2. (Proposition 4.2 in [Amir, 1998]) *Let $\mathcal{M}$ be a model of $\varphi(P,Z)$.*

$$\begin{aligned}\mathcal{M} \models \mathrm{C_{PS}}[\varphi; P/R; Z_1/V_1,\ldots,Z_n/V_n] \equiv \\ \forall \mathcal{M}' \in [\varphi(P,Z)]\, \forall \xi \in |\mathcal{M}|^k \neg(\mathcal{M}' \ll^{\xi} \mathcal{M})\end{aligned}$$

Example 1. Let $\varphi(P) \equiv (P(a) \vee P(b)) \wedge (a \neq b)$, let $U = \{1,2\}$ be the set of elements in the universe. Let M_a, M_b, M_{ab}, and M_ϕ be the models with universe U, with a, b interpreted to 1, 2, respectively, and the following interpretations for the predicate P: $P^{M_a} = \{1\}$, $P^{M_b} = \{2\}$, $P^{M_{ab}} = \{1,2\}$, $P^{M_\phi} = \phi$.

When it comes to pointwise circumscription of P, it becomes counter-intuitive since $C_{PW}[\varphi; P; P/\lambda x.True]$ is unsatisfiable. According to Definition 1, we can obtain $M_a \leq^2 M_b$ and $M_b \not\leq^2 M_a$. As $M_a \in [\varphi(P)]$, M_b is not a model of $C_{PW}[\varphi; P; P/\lambda x.True]$. Similarly, M_a and M_{ab} do not satisfy $C_{PW}[\varphi; P; P/\lambda x.True]$. Because M_ϕ is not a model of $\varphi(P)$, pointwise circumscription of P in $\varphi(P)$ has no model. But when it is applied to point-sensitive circumscription, we can get two models: M_a and M_b, with same models of McCarthy's parallel circumscription, based on Proposition 2.

Compared to pointwise circumscription, point-sensitive circumscription tends to minimize predicates in a minimum view, where predicates be required to be smaller than all other predicates, by controlling the minimization domain.

2.4 Stable Model Semantics

Similar to Parallel circumscription's definition in (1), stable model semantics was recently generalized to first-order language in [Ferraris *et al.*, 2007; Lin and Zhou, 2011]. Given a first-order sentence φ and a tuple σ_i of predicate constants, let SM$[\varphi;\sigma_i]$ stand for the second-order sentence:

$$\mathrm{SM}[\varphi;\sigma_i] = \varphi \wedge \forall \sigma_i^* (\sigma_i^* < \sigma_i \rightarrow \neg \mathrm{St}(\varphi;\sigma_i)) \tag{6}$$

where $\mathrm{St}(\varphi;\sigma_i)$ 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 $F \notin \sigma_i$;
- $\mathrm{St}(\psi \circ \chi;\sigma_i) = \mathrm{St}(\psi;\bar{P}) \circ \mathrm{St}(\chi;\bar{P})$ if $\circ \in \{\wedge,\vee\}$;
- $\mathrm{St}(\psi \rightarrow \chi;\sigma_i) = (\mathrm{St}(\psi;\sigma_i) \rightarrow \mathrm{St}(\chi;\sigma_i)) \wedge (\psi \rightarrow \chi)$;
- $\mathrm{St}(Qx\psi;\sigma_i) = Qx\mathrm{St}(\psi;\sigma_i)$ if $Q \in \{\forall,\exists\}$.

A structure μ 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*. According to [Cabalar and Ferraris, 2007], every universal formula without existential quantifiers under stable model semantics is equivalent to a logic program, which is a foundation of computing stable model semantics via existing ASP solvers.

Building on the result of [Cabalar *et al.*, 2005], Lee and Palla [2009; 2012] defined a translation that turns an "almost universal" formula under the stable model semantics into a logic program under the assumption that every positive (negative, respectively) occurrence of a formula $\exists x\varphi(x)$ ($\forall x\varphi(x)$, respectively) in the original formula ϑ belongs to a subformula ϕ of ϑ such that ϕ contains no strictly positive occurrence of any intensional predicates. System F2LP[1] is an implementation of the translation above [Lee and Palla, 2009].

Zhang $et\ al.$ showed an embedding of first-order circumscription in first-order stable model semantics and also introduce a translation that turns arbitrary first-order formulas into logic programs under finite structures, implemented as system T2LP[2] [Zhang *et al.*, 2011; Zhang, 2011].

3 Translating Point-Sensitive Circumscription into Stable Model Semantics

An embedding of first-order parallel circumscription without varied constants in first-order stable model semantics has been shown in Section 4 of [Zhang *et al.*, 2011]. In this section, we propose translation from point-sensitive circumscription into stable model semantics over arbitrary structures.

Note that the equivalence between formulas in classical first-order logic is still retained in circumscription. So for every first-order formula, there always exists a formula in negation normal form equivalent to it in circumscription. Negation normal form guarantees that $\neg$ only occurs directly ahead of predicates. Here $\neg P$ is treated as $P \rightarrow \bot$, which is called negative literal conveniently. The implication always follows predicates, so that it is handled easily when taking into account the operator St. Thus the translations in this section take formulas in negation normal form as inputs.

[1] http://reasoning.eas.asu.edu/f2lp

[2] http://ss.sysu.edu.cn/~wh/T2LP.html

Our main idea is in brief to introduce auxiliary predicates to simulate the varied predicate constants and their corresponding varying domains $Z_1/V_1, \ldots, Z_n/V_n$, as well as minimization domain R in the second-order sentence.

Definition 3. *Let φ be any first-order sentence in negation normal form. Then we define $Tr^{ps}(\varphi; P/R; Z_1/V_1, \ldots, Z_n/V_n)$ be the conjunction of below formulas with omitting universal quantifiers:*

$$\varphi^{\neg\neg} \wedge \tilde{\varphi} \tag{7}$$

$$\gamma \leftrightarrow \forall \bar{x}(P(\bar{x}) \vee \neg P(\bar{x})) \tag{8}$$

$$(\gamma \to P_R(\bar{x})) \wedge (\gamma \to T(\bar{x}, \bar{y})) \wedge \bigwedge_{1 \leq i \leq k} \gamma \to Q_i(\bar{x}) \tag{9}$$

$$R(\bar{x}, \bar{y}) \wedge P(\bar{y}) \to P_R(\bar{y}) \tag{10}$$

$$R(\bar{x}, \bar{y}) \wedge (P(\bar{y}) \to \gamma) \to (P_R(\bar{y}) \to \gamma) \tag{11}$$

$$\begin{aligned} T(\bar{x}, \overline{\min}) \vee (R(\bar{x}, \overline{\min}) \wedge (P(\overline{\min}) \to \gamma) \\ \wedge \neg\neg P(\overline{\min})) \end{aligned} \tag{12}$$

$$\begin{aligned} succ(\bar{x}, \bar{z}) \to [T(\bar{x}, \bar{y}) \leftrightarrow (R(\bar{x}, \bar{\bar{z}}) \\ \wedge (P(\bar{z}) \to \gamma) \wedge \neg\neg P(\bar{z})) \vee T(\bar{x}, \bar{z})] \end{aligned} \tag{13}$$

$$\bigwedge_{1 \leq i \leq n} \neg V_i(\bar{x}, \bar{y}) \wedge \neg\neg Z_i(\bar{y}) \to Q_i(\bar{y}) \tag{14}$$

$$\bigwedge_{1 \leq i \leq n} \neg V_i(\bar{x}, \bar{y}) \wedge \neg Z_i(\bar{y}) \to (Q_i(\bar{y}) \to \gamma) \tag{15}$$

where $\varphi^{\neg\neg}$ is obtained from φ by substituting $\neg\neg P(\bar{x})$ for each positive literal $P(\bar{x})$; $\tilde{\varphi}$ is obtained from φ by substituting $P_R(\bar{x})$ for each positive literal $P(\bar{x})$, $(P_R(\bar{x}) \to \gamma)$ for each negative literal $\neg P(\bar{x})$, $Q_i(\bar{x})$ for each positive literal $Z_i(\bar{x})$ and $(Q_i(\bar{x}) \to \gamma)$ for each negative literal $\neg Z_i(\bar{x})$ such that $1 \leq i \leq n$; succ describes the successor relation on the domain based on a total order and $\overline{\min}$ is the minimal tuple in the order; P_R, T, Q_i and γ are auxiliary predicates without occurrence in φ.

Actually, Definition 3 provides a syntactic translation from a first-order sentence to another one, which can be achieved in polynomial time. Next, the soundness and completeness of the translation are illustrated by Proposition 3 and then it is proved to be faithful.

Proposition 3. *Let φ be any first-order sentence in negation normal form. Let ψ denote $Tr^{ps}(\varphi; P/R; Z_1/V_1, \ldots, Z_n/V_n)$. Then, over finite structures, $\mathrm{SM}[\psi; P, P_R, T, \gamma, Q_1, \ldots, Q_n]$ is equivalent to $\mathrm{C_{PW}}[\varphi; P/R; Z_1/V_1, \ldots, Z_n/V_n]$, where P_R, T, γ and $Q_1, \ldots, Q_n$ are auxiliary predicates introduced by the translation.*

Proof(sketch). Intuitively, the translation resulting ψ under stable model semantics simulates φ in point-sensitive circumscription. As it is defined above, $\mathrm{SM}[\psi; \sigma_i]$ is equivalent to the formula $\psi \wedge \forall \sigma_i^* \neg(\sigma_i^* < \sigma_i \wedge St(\psi; \sigma_i))$. Compared with the definition of point-sensitive circumscription (5), that of S-M is similar on the second-order part. Based on the similarity, we propose a translation from point-sensitive circumscription into SM with auxiliary predicates.

Indeed, Formula (7) is equivalent to φ because each substitution in $\tilde{\varphi}$ must be true on account of Formulas (8) and (9). In addition, the validation of auxiliary predicates makes formulas (10)-(15) be always true and guarantees their corresponding predicate variables within the range of them, such as $T^* \leq T$. Thus, the conjunction of these formulas, *i.e.*, ψ, is equivalent to φ and it describes the equivalence of translation in the first-order part.

Next, these formulas change after applying the operator St. Specifically, $St(\gamma; \sigma_i) \equiv \gamma^* = \bot$ if $P^* < P$ and otherwise $\gamma^* = \gamma$. Actually, $P^* < P$ must be true or it will make all predicate variables equal to their corresponding predicate constants. Then $\sigma_i^* < \sigma_i$ is false and it makes the second-order part be true which has no influence on the translation. When $P^* < P$, the implication with an antecedent of γ^* is true and the implication in from of $P^*(\bar{x}) \to \gamma^*$ is equivalent to $\neg P^*(\bar{x})$.

Additionally, $St((10) \wedge (11); \sigma_i)$ describes the property that P_R^* is equivalent to P^* in the region Rx. Because of $P^* < P$, it equals to $\forall \bar{y}(Rx(\bar{y}) \to (P_R^*(\bar{y}) \to P(\bar{y})))$. In $St((12) \wedge (13); \sigma_i)$, predicate variable T^* actually describes the existential quantifier over successor structure, according to the idea of Eiter $et\ al.$ and Zhang $et\ al.$. So it is equivalent to $\exists \bar{y}(Rx(\bar{y}) \wedge P_R^*(\bar{y}) \wedge P(\bar{y}))$. Consequently the conjunction of them is equivalent to $LS_R(P_R^*, P)$.

Similarly, $St((14) \wedge (15); \sigma_i)$ describes the property that Q_i^* and Z_i are equal outside $V_i x$ such that $1 \leq i \leq k$. Then each its conjunctive simulates $EQ_i x(Q_i^*, Z_i)$. Since predicate constants P_R and Q_i are always valid, corresponding predicate variables P_R^* and Q_i^* can change arbitrarily in the domain and it actually describes the second-order universal quantifier. Here predicate variables P_R^* and Q_i^* simulate p and z_i in point-sensitive circumscription respectively. According to the substitution rule, $St(\tilde{\varphi}; \sigma_i)$ is equivalent to φ^*. Because of $St(\varphi^{\neg\neg}; \sigma_i) = \varphi$, the second-order part of $\mathrm{SM}[\psi; \sigma_i]$ is equal to that of $\mathrm{C_{PW}}[\varphi; P/R; Z_1/V_1, \ldots, Z_n/V_n]$. So far the faithfulness of the translation Tr^{ps} is proved. □

Remark 1. *For the special form of point-sensitive circumscription $\mathrm{C_{PS}}[\varphi; P; Z_1/V_1, \ldots, Z_n/V_n]$, whose R actually is treated as $True$, Formulas (10)-(13) can be omitted, which can make the translation Tr^{ps} more efficient by reducing the number of rules generated.*

Remark 2. *As for Tr^{ps}, we have not mentioned function and individual constants because predicates can simulate functions easily. For each n-arity function, we can introduce a n+1-arity predicate to represent it. Particularly, individual constants varied actually can be simulated by individual variables in scope of first-order existential quantifiers.*

4 Application in Situation Calculus

So far, in this paper we have shown how we can translate point-sensitive circumscription into first-order stable model semantics. In this section we will use point-sensitive circumscription representation to reformulate the Frame Problem and the Qualification Problem in situation calculus.

4.1 Situation Calculus

The situation calculus [Reiter, 2001; Lin, 2008] is one of the most well-known formalisms for reasoning about actions. The situation calculus is a many-sorted first-order language (with some second-order ingredients) suitable for representing changes. Prolog can be used to implement the situation calculus, based on the fact that Clark's completion semantics accounts for definitional axioms.

The basic special sorts in situation calculus are $situations$, $actions$, and $fluents$ ($relational$ fluents and $functional$ fluents), situations and actions are represented as individuals that can be quantified over. There could be other sorts, some of them domain dependent like $block$ for blocks in the blocks world and others domain independent like $truth$ for truth values. Assume the following special domain independent predicates and functions: a binary predicate $Holds(P(x), s)$ denoting fluent P is true in situation s, usually using $P(x, s)$ as shorthand for $Holds(P(x), s)$; a binary function $do(a, s)$ denoting the successor situation to s resulting from performing action a; a binary predicate $Poss(a, s)$ meaning that action a is possible in situation s.

We assume that a description $\mathcal{D}$ consists of a finite number of the following sets of axioms. We often identify $\mathcal{D}$ with the conjunction of the universal closures of all axioms in $\mathcal{D}$. In the following, F, F_i are fluent names, A is an action name, V , V_i are truth values, s, s' are situation variables, $\phi(s)$ is a simple state formula about s, constants a, a' are action variables, f is a variable of sort fluent, v is a variable of sort truth value, and x, x_i, y, y_i are lists of variables.

Reiter's $basic\ action\ theory$ (BAT) is of the form

$$\Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0} \tag{16}$$

where

- Σ: the set of the foundational axioms;
- $\mathcal{D}_{ss}$: a set of successor state axioms of the form

  $$F(x, do(a, s)) \leftrightarrow \Phi_F(x, a, s),$$

  where $\Phi_F(x, a, s)$ is a formula that is uniform in s [Reiter, 2001] and whose free variables are among x, a, s;
- $\mathcal{D}_{ap}$: a set of action precondition axioms of the form

  $$Poss(A(x), s) \leftrightarrow \Pi_A(x, s),$$

  where $\Pi_A(x, s)$ is a formula that is uniform in s and whose free variables are among x, s;
- $\mathcal{D}_{una}$: the set of unique name axioms for fluents and actions;
- $\mathcal{D}_{S_0}$: a set of first-order sentences that are uniform in S_0.

4.2 The Frame Problem, the Ramification Problem, and the Qualification Problem

McCarthy and Hayes identified the Frame Problem as the problem of expressing a dynamical domain without explicitly specifying which conditions are not affected by an action [McCarthy and Hayes, 1969]. McCarthy [1986] initially proposed to solve the Frame Problem by the following *generic frame axiom*:

$$Holds(p, s) \wedge \neg abnormal(p, a, s) \rightarrow Holds(p, do(a, s))$$

with the abnormality predicate abnormal circumscribed. However, Hanks and McDermott showed that McCarthy's approach does not work [Hanks and McDermott, 1987]. Reiter proposed a simple syntactic manipulation that turns a set of effect axioms into a set of successor state axioms that completely captures the true value of each fluent in any successor situation [Reiter, 1991].

$$F(x, do(a, s)) \equiv \gamma^+(a, x, s) \vee (F(x, s) \wedge \neg\gamma^-(a, x, s))$$

The Ramification Problem, first discussed by Finger [1986], is about how to encode constraints like this in an action domain, and how these constraints can be used to derive the effects of the actions in the domain. Lin [1995] represents this constraint as a causal constraint, axiomatizing this by introduced a ternary predicate $Caused(p, v, s)$, meaning that fluent p is caused to have truth value v in situation s.

- $\mathcal{D}_{caused}$ is a set of axioms of the form

  $$Poss(A(x), s) \rightarrow \\ (\phi(s) \rightarrow Caused(F(y), V, do(A(x), s))$$

  (direct effects) and

  $$\phi(s) \wedge Caused(F_1(x_1), V_1, s) \wedge \ldots \\ \wedge Caused(F_n(x_n), V_n, s) \rightarrow Caused(F(x), V, s)$$

  (indirect effects).

The Qualification Problem is concerned with the impossibility of listing all the preconditions required for a real-world action to have its intended effect [McCarthy, 1977]. One possible solution to this problem is to assume that an action is always executable unless explicitly ruled out by the theory. This can be achieved by maximizing the predicate $Poss$, or in terms of circumscription, circumscribing $Poss$. The problem becomes more complex when some domain constraints-like axioms can influence $Poss$. Lin and Reiter [1994] called those constraints that yield indirect effects of actions ramification constraints, and those that yield additional qualifications of actions qualification constraints. They are both represented as sentences of the form $\forall sC(s)$, and it is up to the user to classify which category they belong to. Under this framework, only constraints represented as causal rules using $Caused$ can derive new effects of actions, and ordinary situation calculus sentences of the form $\forall sC(s)$ can only derive new qualifications on actions.

4.3 Representing the Frame and the Qualification Problem with Point-Sensitive Circumscription

Amir [1997] adjusted the discrete situation calculus [Lin and Reiter, 1994] to fit his set theoretic language and proposed his solution to the Frame and the Qualification problems. Amir gave a theoretic solution with point-sensitive circumscription and we achieve it practically in this section.

Suppose $Ab(l, a, s)$, which is an predicate constant on fluent l, action a, and situation s, means an abnormality. Normally, fluent l remains after performing action a in situation s. In other word, $Ab(l, a, s)$ denotes l changes in situation $do(a, s)$, which is represented by the following axioms:

$$\neg Ab(l, a, s) \rightarrow (Holds(l, s) \leftrightarrow Holds(l, do(a, s)))$$

To find all fluents keeping persistence in the next situation, we minimize Ab on one situation at a time. Let $V_1(x, y)$ be the formula $\lambda x, y\ \exists l_x, a_x, s_x\ \exists l_y, a_x, s_x\ x =< l_x, a_x, s_x > \wedge\ y =< l_y, a_y, s_y > \wedge\ s_x = s_y$, and $V_2(x, y)$ be the formula $\lambda x, y\ \exists l_x, a_x, s_x\ \exists l_y, a_x, s_x\ x =< l_x, a_x, s_x > \wedge\ y =< l_y, a_y, s_y > \wedge\ do(a_x, s_x) = s_y$. Intuitively, V_1 considers that situation s_x while V_2 considers the next situation. According to the meaning of V in pointwise view, predicates can change arbitrarily inside V with remaining unchanged outside V. In point-sensitive circumscription, Ab/V_1 means that the abnormality in different situations and Ab is allowed to vary in the same situation, while $Holds/V_2$ means that fluents can be changed in the next situation.

$$EQ_{V_1x}(Ab^*, Ab) \equiv s_x \neq s_y \rightarrow (Ab^*(y) \leftrightarrow Ab(y))$$

$$EQ_{V_2x}(Holds^*, Holds) \equiv \\ do(a_x, s_x) \neq s_y \rightarrow (Holds^*(y) \leftrightarrow Holds(y))$$

With this circumscription policy, for each situation, there is only one situation, the next situation, being considered rather than all situations. To minimize Ab in all fluents and actions, we let R be $True$. The point-sensitive circumscription can solve the Frame Problem one situation at a time:

$$\mathrm{C_{PS}}[\varphi; Ab/R; Ab/V_1, Holds/V_2] \quad (17)$$

The models of this point-sensitive circumscription coincide with minimal models of discrete situation calculus, no matter whether nondeterministic actions are allowed to be done.

As far as the Qualification Problem is concerned, the point-sensitive circumscription can solve it. Similarly, we let Abq describe the abnormality of allowance to preform actions. Action a is executable in situation s normally and $Abq(a, s)$ denote a can not be preformed in s.

$$\neg Abq(a, s) \rightarrow Poss(a, s)$$

In situation calculus, there is a class of formulas called constraints, which consist of ramification constraints RC and quantification constraints QC. We use a predicate constant $AllowedS(s)$ to denote situation s satisfying all quantification constraints. The following axiom can guarantee every situation not in $AllowedS$, its next situation is also not in it.

$$\neg AllowedS(s) \rightarrow \neg AllowedS(do(a, s))$$

Next, an action is said to be applicable if and only if its preconditions are met and it does not lead to the violation of QC, which is denoted as follows.

$$App(a, s) \leftrightarrow Poss(a, s) \wedge AllowedS(do(a, s))$$

To consider all situations, we need consider as many as possible action. Thus, we must maximize the predicates $Poss$. Furthermore, predicate Abq should be circumscribed. The point-sensitive circumscription can solve the Quantification Problem with the following policy.

$$\mathrm{C_{PS}}[\varphi; Abq/R; Abq/V_1, Ab/V_1, Poss/V_1, \\ Holds/V_2, AllowedS/True, App/True] \quad (18)$$

As a result, action quantifications is expressed explicitly by a conjunction of simple formulas of the following form:

$$Poss(a, s) \leftrightarrow \theta_1(a, s) \wedge ... \wedge \theta_n(a, s)$$

5 Implementation and Example

This section shows how to implement a point-sensitive circumscription solver psc2lp[3]. One example: a variant of the Yale Shooting Scenario is presented how psc2lp can be used in solving the Qualification Problem in situation calculus.

5.1 Implementation

A point-sensitive circumscription solver psc2lp is developed based on our approach. psc2lp firstly accepts a point-sensitive circumscriptive theory, then translates it into a logic program, and finally invokes an ASP solver with a corresponding finite extensional database[4].

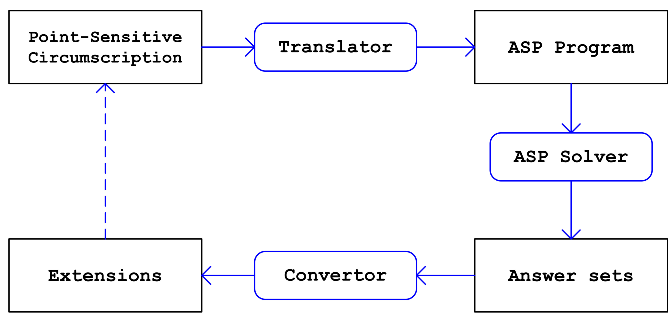

Figure 1: Outline of psc2lp

Figure 1 illustrates how psc2lp works. An input point-sensitive circumscriptive theory is firstly translated to an answer set program by the *translator* in psc2lp. Then, an *ASP solver* is called to compute the answer sets of the program. Finally, the answer sets will be interpreted back to all solutions of the original point-sensitive circumscriptive theory by an *convertor*. For the *ASP solver* module in psc2lp, we just use claspD [5] [Gebser *et al.*, 2007]. The *convertor* in psc2lp is trivial. Hence, the main issue in psc2lp is the *translator*.

Indeed, after the translation Tr^{ps}, the fixed and varied predicates in circumscription are treated as the extensional predicates under stable model semantics which need to be eliminated when translating into a logic program by introducing a sentence of the form $\forall \bar{x}(Q(\bar{x}) \vee \neg Q(\bar{x}))$ for each extensional predicate Q.

[3] http://ss.sysu.edu.cn/~wh/psc2lp.html

[4] An extensional database is a structure consisting of extensional predicate and function constants under stable model semantics.

[5] http://www.cs.uni-potsdam.de/clasp/

More precisely, we can compute point-sensitive circumscription by 4 steps:

1. Turn the input into the sentence in both prenex normal form and negation normal form;
2. Apply the translation Tr^{ps} to obtain a first-order sentence under stable model semantics;
3. Use Zhang's first order stable model semantics solver T2LP repeatedly till a logic program can be obtained;
4. Add $\forall \bar{x}(Q(\bar{x}) \vee \neg Q(\bar{x}))$ for each fixed and varied predicate.

5.2 Example

Let us examine a variant of the Yale Shooting Scenario (YSS) [Hanks and McDermott, 1987]. Assume that there are two turkeys. As a result of the gun's being shot, exactly one turkey dies. There is only one bullet can be loaded. For this, we have initially, both turkeys are alive, and the gun is loaded. The knowledge representation is provided below with some formulas omitted:

$\mathcal{D}_{ap}$:

$$Holds(loaded, s) \rightarrow Poss(shoot(x), s) \tag{19}$$

$$Holds(loaded, s) \vee \neg Holds(loaded, s) \rightarrow Poss(load, s) \tag{20}$$

$\mathcal{D}_{ss}$:

$$Holds(loaded, do(a, s)) \leftrightarrow (Holds(loaded, s) \wedge a \neq load) \vee (\neg Holds(loaded, s) \wedge a = load) \tag{21}$$

$$Holds(alive(x), do(a, s)) \leftrightarrow Holds(alive(x), s) \wedge a \neq shoot(x) \tag{22}$$

Especially, in this kind of circumscription policy, R is $True$ and $LS_{Rx}(Ab^*, Ab)$ reduces to $Ab^* < Ab$. So Formulas (10)-(13) are not necessary in the translation Tr^{ps}. When the translation is applied on (18), according to (14) and (15) in Tr^{ps} we can get:

$$s_x \neq s_y \wedge \neg\neg Abq(y) \rightarrow Abq'(y)$$
$$s_x \neq s_y \wedge \neg Abq(y) \rightarrow \neg Abq'(y)$$

where $x =< a_x, s_x >$, $y =< a_y, s_y >$ and Abq' is an auxiliary predicate corresponding to Abq, introduced like the introduction of Q. Additionally, according to (14) and (15) we can obtain:

$$do(a_x, s_x) \neq s_y \wedge \neg\neg Holds(y) \rightarrow Holds'(y)$$
$$do(a_x, s_x) \neq s_y \wedge \neg Holds(y) \rightarrow \neg Holds'(y)$$

where $Holds'$ is an auxiliary predicate corresponding to $Holds$. Other similar formulas are omitted because of the limited space.

Besides, (7) in Tr^{ps} applied to (19) is denoted below:

$$\varphi^{\neg\neg} : \quad \neg Holds(loaded, s) \vee Poss(shoot(x), s)$$
$$\tilde{\varphi} : \quad (Holds'(loaded, s) \rightarrow \gamma) \vee Poss'(shoot(x), s)$$

where $Holds'$ and $Poss'$ are auxiliary predicates corresponding to $Holds$ and $Poss$ respectively.

After applying the translation Tr^{ps}, we can get a first-order theory under stable model semantics. Next, via T2LP we can reduce it into an universal theory, which is equivalent to an answer set program. When we obtain a logic program, we can invoke an existing ASP solver to find all solutions.

6 Conclusion

The relationship among McCarthy's circumscription, Lifschitz's pointwise circumscription and Amir's point-sensitive circumscription was clarified in this paper. Furthermore, we proposed and proved a translation Tr^{ps} from point-sensitive circumscription to stable model semantics over finite structures. We can compute point-sensitive circumscription over finite structures by reducing stable model semantics to ASP. Our approach is not only theoretically interesting but also of practical relevance with an example in situation calculus. With point-sensitive circumscription, the Frame problem and the Quantification problem can be solved and we can find all solutions via our solver.

Now we summarize the contributions of this paper. First, we propose a translation from point-sensitive circumscription to stable model semantics over finite structures. Secondly, with psc2lp, we can compute practical problems represented by point-sensitive circumscription effectively, such as the Frame problem and the Quantification problem. Thirdly, compared with propositional case, we can represent problems in a flexible and natural way with allowing existential quantifiers in point-sensitive circumscription.

Acknowledgments

We would like to thank Heng Zhang for his valuable suggestions. Yongmei Liu and her group commented on this paper. Our deep thanks to all of them. We also would like to thank anonymous reviewers for their valuable comments. This research has been partially supported by the Doctoral Program Foundation of Institutions of Higher Education of China under Grant 20110171120041, the Natural Science Foundation of Guangdong Province under Grant S2012010009836, and the Guangzhou Science and Technology Project under Grant 13200059.

References

[Amir, 1997] Eyal Amir. Formalizing action using pointwise circumscription and set theory. In *NRAC'97*, pages 1–17, 1997.

[Amir, 1998] Eyal Amir. Pointwise circumscription revisited. In *KR'98*, pages 202–211, 1998.

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

[Cabalar *et al.*, 2005] Pedro Cabalar, David Pearce, and Agustłn Valverde. Reducing propositional theories in equi-

References

[Baral and Gonzalez, 2011] Chitta Baral and Marcos Alvarez Gonzalez. Using Inverse Lambda and Generalization to Translate English to Formal Languages. In *Proceedings of the Ninth International Conference on Computational Semantics*, number 2000, 2011.

[Chen *et al.*, 2010] Xiaoping Chen, Jianmin Ji, Jiehui Jiang, Guoqiang Jin, Feng Wang, and Jiongkun Xie. Developing High-level Cognitive Functions for Service Robots. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-10)*, pages 989–996, Toronto, Canada, 2010. IFAAMAS.

[Chen *et al.*, 2012] Xiaoping Chen, Jiongkun Xie, Jianmin Ji, and Zhiqiang Sui. Toward Open Knowledge Enabling for Human-Robot Interaction. *Journal of Human-Robot Interaction (JHRI)*, 1(2):100–117, 2012.

[Gelfond and Lifschitz, 1988] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference on Logic programming (ICLP-88)*, pages 1070–1080, 1988.

[Gupta and Kochenderfer, 2004] Rakesh Gupta and Mykel J. Kochenderfer. Common Sense Data Acquisition for Indoor Mobile Robots. In *Proceedings of the 19th National Conference on Artificial Intelligence and the 16th Conference on Innovative Applications of Artificial Intelligence (AAAI-04)*, volume 300, pages 605–610, San Jose, California, USA, 2004. AAAI.

[Kunze *et al.*, 2010] Lars Kunze, Moritz Tenorth, and Michael Beetz. Putting People's Common Sense into Knowledge Bases of Household Robots. In *Proceedings of KI 2010: Advances in Artificial Intelligence, 33rd Annual German Conference on AI*, pages 151–159, Karlsruhe, Germany, 2010. Springer.

[Kwiatkowski *et al.*, 2010] Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing Probabilistic CCG Grammars from Logical Form with Higher-Order Unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*, pages 1223–1233, MIT Stata Center, Massachusetts, USA, 2010. ACL.

[Kwiatkowski *et al.*, 2011] Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical Generalization in CCG Grammar Induction for Semantic Parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, pages 1512–1523, John McIntyre Conference Centre, Edinburgh, UK, 2011. ACL.

[Lu *et al.*, 2008] Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. A Generative Model for Parsing Natural Language to Meaning Representations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pages 783–792, Honolulu, Hawaii, USA, 2008. ACL.

[Matuszek *et al.*, 2012] Cynthia Matuszek, Evan Herbst, Luke S. Zettlemoyer, and Dieter Fox. Learning to Parse Natural Language Commands to a Robot Control System. In *Proceedings of the 13th International Symposium on Experimental Robotics (ISER-12)*, 2012.

[Steedman, 1996] Mark Steedman. *Surface Structure and Interpretation*. MIT Press, 1996.

[Steedman, 2000] Mark Steedman. *The Syntactic Process*. MIT Press, 2000.

[Tenorth *et al.*, 2010] Moritz Tenorth, Daniel Nyga, and Michael Beetz. Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA-10)*, pages 1486–1491, Anchorage, Alaska, USA, 2010. IEEE.

[Wong and Mooney, 2007] Yuk Wah Wong and Raymond J. Mooney. Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, number June, pages 960–967, Prague, Czech Republic, 2007. ACL.

[Zettlemoyer and Collins, 2005] Luke S. Zettlemoyer and Michael Collins. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI-05)*, number x, pages 658–666, Edinburgh, Scotland, 2005. AUAI Press.

[Zettlemoyer and Collins, 2007] Luke S. Zettlemoyer and Michael Collins. Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL-07)*, pages 678–687, Prague, Czech Republic, 2007. ACL.

[Zettlemoyer and Collins, 2009] Luke S. Zettlemoyer and Michael Collins. Learning Context-Dependent Mappings from Sentences to Logical Form. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL/AFNLP-09)*, pages 976–984, Singapore, 2009. ACL.