A Logic of Knowledge and Justified Assumptions[1]

Fangzhen Lin and Yoav Shoham

Department of Computer Science

Stanford University

Stanford, CA 94305

[1]This work was supported in part by a grant from the Air Force Office of Scientific Research.

Abstract

In this paper we define the logic GK of knowledge and justified assumptions. GK is best understood as a formalization of autoepistemic reasoning processes that are more general than those in Moore's autoepistemic logic, and is formally defined via a modification of Shoham's preference semantics. We show that GK includes not only Moore's autoepistemic logic, but also Reiter's default logic. To our knowledge GK is the first complete semantic unification of the two logics. Similarly to circumscription, GK is based on the notion of logical minimization, and thus provides a bridge between circumscription and fixed-point nonmonotonic logics, an outstanding problem in nonmonotonic logics. As an application of this bridge, we propose a formalization of logic programs with negation-as-failure in circumscription.

1 Introduction

Imagine an ideal rational agent trying to decide which set of propositions to believe.[1] Suppose the agent has a set of prior beliefs, and a set of rules to derive new beliefs. Then the natural answer would be that the agent should take the "logical closure" of the prior belief set under the set of rules as her belief set. However, it is not always clear how to define the notion of "logical closure." If the prior beliefs are modeled as atomic sentences in a first-order language, and the rules as definite Horn clauses as in logic programs, then there would be no problem in defining "logical closure." However, in many cases, both prior beliefs and rules can take more complicated forms. In particular, the applicability of a rule may depend on what the agent does and does not believe. For example, consider the following famous example due to Moore's [21]:

> If I do not know that I have a brother, then I do not have one.

Thus in order to assure herself that she does not have a brother, the agent must first make sure that she does not "know" that she has a brother, a somewhat circular argument. A common way of capturing circular arguments is by using fixed-points. In this case, the agent first assumes a belief set, and refers all statements about beliefs in the rules to this belief set. This will eliminate all references to beliefs in the rules. If, from the prior belief set and the resulting rules, the agent deduces a set of beliefs that coincides with the one the agent assumed earlier, then this assumed belief set will be "justified." This "justified" belief set will then be a "logical closure" of the set of rules applied to the prior belief set, and a candidate for the agent to believe in.

It is clear that this kind of "autoepistemic" reasoning processes is the one used in Moore's autoepistemic logic [21]. However, Moore's logic has serious limitations. Largely through Konolige's work [6], it is well-known by now that Reiter's default rules [25] are closely related to these "autoepistemic rules." From the same work, it is also clear that

[1] This is a revised version of Chapter 3 in [14]. An extended abstract of section 2 to section 4 appeared as [13].

Moore's logic has difficulty in accounting precisely for the mechanisms used in Reiter's default logic. If we study carefully the autoepistemic reasoning process outlined above, we see that there are two kinds of beliefs involved. The first one is the beliefs in the fixed belief set that the agent assumed. The other one is the new beliefs derived by applying the rules. In Moore's autoepistemic logic, all beliefs in the statement of a rule are assumed. This means that in Moore's logic, derived beliefs are meta-logical, and can not be referred to in the language. As we shall see later, the formalization of default rules in an epistemic logic requires both kinds of beliefs to be explicit in the language. Of course, this does not mean that default logic is more general than Moore's autoepistemic logic. In fact, as we shall see, default rules are also very limited as they restrict the forms of assumed and derived beliefs that can be expressed.

In this paper we propose a logic that formalizes the above reasoning procedure in a more general way than either Moore's autoepistemic logic and Reiter's default logic. We do so by directly introducing two modal operators, one for knowledge (K), which corresponds to derived beliefs, and the other for assumptions (A), which correspond to assumed beliefs. By doing so we provide a clear and intuitive epistemic semantics for default logic, until recently an open problem in nonmonotonic logics. Furthermore, our logic is based on Shoham's preferential semantics, and thus provides a bridge between minimal-model nonmonotonic logics such as circumscription and fixed-point nonmonotonic logics such as default logic, another significant problem in nonmonotonic logics.

This paper is organized as follows. We first define our basic logical language, which is simply a propositional one augmented by the two epistemic modalities. Then we define the logic GK[2] of knowledge and justified assumptions, which is the result of imposing an augmented version of preference semantics on our basic language. We then offer translations of both default logic and autoepistemic logic into GK, translations that are similar but subtly different. We prove the correctness of the translations, and show how the common

[2]The name GK is historical. Initially, we thought of the logic as accounting for Grounded Knowledge; we now consider it more appropriate to view it as a logic of knowledge and justified assumption.

framework explicates the relationship between the two logics. Finally, as an application of the bridge GK provides between minimal-model and fixed-point nonmonotonic logics, we show a formalization of logic programs in circumscription.

2 The basic logical language

Our language is a propositional one, augmented with two modalities K and A, which are at this point mutually unconstrained. Well-formed formulas are defined as usual. Intuitively $K\varphi$ means that φ is known or believed (the distinctions between the two are not important in this paper), while $A\varphi$ means that φ is assumed.

A *Kripke structure* is a tuple (W, π, R_K, R_A), where W is a nonempty set, $\pi(w)$ a truth assignment to the primitive propositions for each $w \in W$, and R_K, R_A are binary relations over W (the accessibility relations for K and A, respectively). A *Kripke interpretation* M is a pair $((W, \pi, R_K, R_A), w)$, where (W, π, R_K, R_A) is a Kripke structure, and $w \in W$. We call w the *actual world* of M.

Notice that no restrictions on the two accessibility relations have been placed. This means that we are assuming the weakest normal modal system for both K and A. However, as we shall see later, our key results still hold if other modal systems such as $S5$ are assumed for K and/or A.

An interpretation M satisfies a formula φ if φ is true in the actual world of M. Formally, the satisfaction relation "$\models$" between Kripke interpretations and formulas is defined as follows:

1. $((W, \pi, R_K, R_A), w) \models p$ iff $\pi(w)(p) = 1$, where p is a primitive proposition.

2. $M \models \varphi_1 \vee \varphi_2$ iff $M \models \varphi_1$ or $M \models \varphi_2$.

3. $M \models \neg\varphi$ iff it is not the case that $M \models \varphi$.

4. $((W, \pi, R_K, R_A), w) \models K\varphi$ iff $((W, \pi, R_K, R_A), w') \models \varphi$ for any $w' \in W$ such that $(w, w') \in R_K$.

5. $((W, \pi, R_K, R_A), w) \models A\varphi$ iff $((W, \pi, R_K, R_A), w) \models \varphi$ for any $w' \in W$ such that $(w, w') \in R_A$.

We say that a Kripke interpretation M is a *model* of a set of formulas S if M satisfies every member of S. We emphasis here that according to our definition, a Kripke interpretation M is a model of a set of formulas S if every member of S is true in the *actual world* of M, rather than in *every world* of M.

We conclude this section with two definitions that will be used throughout the remainder of the paper: For each Kripke interpretation M, we define

$$K(M) = \{\varphi \mid M \models K\varphi, \varphi \text{ is a base formula}\},$$

and

$$A(M) = \{\varphi \mid M \models A\varphi, \varphi \text{ is a base formula}\},$$

where a *base formula* is one that does not contain modal operators.

3 Preference semantics and the logic GK

In this section we modify the semantics of our basic language, defining the logic GK. We will augment the semantical framework for nonmonotonic logics developed by Shoham [28, 29]. Shoham's basic idea is that a nonmonotonic logic (called a *preference logic* by Shoham) can be viewed semantically as the result of imposing a partial order (called a *preference relation*) "$\sqsubset$" on interpretations of a monotonic logic. The syntax of the resulting logic is unchanged, but the definition of semantic entailment changes: Φ now entails φ if φ is true in all models of Φ that are minimal according to that partial order (but not necessarily in all models of Φ). This new notion of entailment is called *preferential entailment*, and is denoted by $\models_\sqsubset$. We start by endowing our basic language with such preference semantics.

To do so we define a preference relation on the Kripke interpretations defined in the previous section. The following relation "$\sqsubset$" has the effect of minimizing knowledge with assumptions fixed.

Definition 3.1 *Let M_i, $i = 1, 2$, be two Kripke interpretations. We say that M_1 is preferred over M_2, written $M_1 \sqsubset M_2$, if:*

1. $A(M_1) = A(M_2)$.

2. $K(M_1) \subset K(M_2)$.

M is a *minimal* model of S if M is a model of S and there is no other model M' of S such that $M' \sqsubset M$. Referring back to the informal discussions in the introduction, we can say that if M is a minimal model of S, then $K(M)$ will be a "logical closure" of the "rules" in S under the assumed belief set $A(M)$. As was said in the introduction, the assumptions made by the agent should be justified. We capture this justification by an added requirement of the minimal models – that the assumptions coincide with the knowledge. This naturally leads to the following definition:

Definition 3.2 *Let S be a set of formulas, and M a Kripke interpretation. We say that M is a* preferred model *of S if:*

1. *M is a minimal model of S.*

2. $K(M) = A(M)$.

If M is a preferred model of S, then $K(M)$, or $A(M)$ since they are the same, would be a candidate belief set for the agent to hold.

It is clear that our definition of preferred models depends only on sets of base formulas that are known or assumed in a model. This is part of the reason that our logic GK defined as follows is insensitive to the modal systems used for K and A. The other part of the reason is that Theorem 1, our key result about GK used in embedding both default logic and autoepistemic logic, concerns only with formulas that have no nested modalities, and contain only primitive propositions that are under the range of a modal operator.

Definition 3.3 *The logic GK is defined as follows:*

Syntax: The syntax of our basic language.

Semantic entailment: $\Phi \models_{GK} \varphi$ *iff* φ *holds in all preferred models of* Φ.

Example 3.1 In the following, p, q, r are primitive propositions.

$S_1 = \{Kp\}$ has a unique preferred model in the sense that M is a preferred model of $\{Kp\}$ iff $K(M)$ is the tautological closure of p, and $A(M) = K(M)$. This means that if M_1 and M_2 are both preferred models of S_1, then $K(M_1) = K(M_2)$, that is, M_1 and M_2 agree on formulas of the form $K\varphi$, where φ is a base formula. It is possible that M_1 and M_2 may disgree on other formulas. In fact, it is easy to see that there are two preferred models M_1, and M_2 such that $M_1 \models p$, and $M_2 \models \neg p$. Of course, Kp is true in both models.

$S_2 = \{\neg Ap \supset Kp\}$ has no preferred models. For let M be a preferred model of S_2. If Ap is true in M, then Kp must be false because of the minimality of M. But then $K(M) \neq A(M)$ and M can not be a preferred model of S_2. Now if Ap is false in M, then Kp must be true in M because it is a model of S_2, and this again will make $K(M) \neq A(M)$.

As we will see in the next section, S_1 corresponds to the default theory $(\{p\}, \emptyset)$, and S_2 to $(\emptyset, \{: \neg p/p\})$.

$S_3 = \{Kp \vee Kq, Kp \wedge \neg A\neg r \supset Kr, Kq \wedge \neg A\neg r \supset Kr\}$. We can distinguish two kinds of preferred models of S_3: One in which $Kp \wedge \neg Kq$ and Kr are true; and the other in which $Kq \wedge \neg Kp$ and Kr are true. Notice that Kr is true in all preferred models.

There is no default theory that corresponds to S_3. In fact, S_3 is an instance of disjunctive default theories [4]. For instance, we can interpret p as "Tweety is a penguin", q as "Tweety has a broken wing", and r as "Tweety does not fly". ■

We conclude this section by proving a generalization of Reiter's Theorem 2.1 in [25]. This is our key technical result about GK, and is used in embedding default logic and autoepistemic logic in GK.

Theorem 1 *Let S be a set of formulas of the form:*

$$Kp \wedge Aq \wedge \neg Ar_1 \wedge \ldots \wedge \neg Ar_n \supset Ks, \tag{1}$$

where p, q, r_1, ..., r_n, s are base formulas, $n \geq 0$, and both Kp and Aq may be absent. If both Kp and Aq are absent, and $n = 0$, then (1) will be Ks. Then a Kripke interpretation M is a preferred model of S iff:

1. *$K(M) = A(M)$.*
2. *$K(M) = E_1 \cup E_2 \cup ...$, where E_i, $i = 1, 2, ...$, are defined inductively as follows:*
   (a) *$E_1 = \{s \mid Ks \in S\}$.*
   (b) *$E_{i+1} = Th(E_i) \cup \{s \mid Kp \wedge Aq \wedge \neg Ar_1 \wedge \ldots \wedge \neg Ar_n \supset Ks \in S$, where $p \in E_i$, $q \in A(M)$, and $r_1, r_2, ..., r_n \notin A(M)\}$, where $i > 0$ and $Th(E_i)$ is the tautological closure of E_i.*

Proof: In the following, let $E = E_1 \cup E_2 \cup ....$

($\Leftarrow$): Let M be a Kripke interpretation that satisfies condition (1) and (2). We show that M is a preferred model of S. First, we show that M is a model of S. Let $\varphi \in S$. If $\varphi = Ks \in S$, then by condition 2(a), $s \in E_1 \subset E = K(M)$. Thus $M \models \varphi$. Let $\varphi \in S$ has the form

$$Kp \wedge Aq \wedge \neg Ar_1 \wedge \ldots \wedge \neg Ar_n \supset Ks.$$

Suppose

$$M \models Kp \wedge Aq \wedge \neg Ar_1 \wedge \ldots \wedge \neg Ar_n.$$

Since $K(M) = E$, there exists a number $n \geq 1$ such that $p \in E_n$. By the construction of E_{n+1}, we have $s \in E_{n+1}$. Thus $s \in K(M)$, and $M \models Ks$. Therefore $M \models \varphi$. This proves that M is a model of S. M must be minimal for if M' is a model of S such that $A(M) = A(M')$, then it is easy to prove by induction on i that $E_i \subseteq K(M')$ for $i > 0$; thus $K(M) = E \subseteq K(M')$. Thus M is a preferred model of S.

($\Rightarrow$): Let M be a preferred model of S. We show that $K(M) = E$. Since M is a model of S, as we have proved in the ($\Leftarrow$) part, we have $E \subseteq K(M)$. Thus E is a tautologically consistent set of sentences. We claim that there is a Kripke interpretation M' such that

$A(M') = A(M)$ and $K(M') = E$. If this is the case, then as we have proved above, M' must be a model of S. Thus by the minimality of M, we must have $K(M) = K(M') = E$. We now show the existence of such M'. Since E is consistent, and closed under tautological consequence, by a well-known result in [5], there is a Kripke interpretation M'' such that $K(M'') = E$. Let $M = ((W, \pi, R_K, R_A), w)$, and $M'' = ((W'', \pi'', R''_K, R''_A), w'')$. Without the loss of generality, we can assume that $W \cap W'' = \emptyset$ (otherwise, we can always rename elements in W). Define $M' = ((W', \pi', R'_K, R'_A), w')$ as follows:

1. $w' = w$;
2. $W' = W \cup W''$;
3. For any $w_1 \in W'$, if $w_1 \in W$, then $\pi'(w_1) = \pi(w_1)$, otherwise, $\pi'(w_1) = \pi''(w_1)$;
4. For any $w_1, w_2 \in W''$, $(w_1, w_2) \in R'_K$ iff either $(w_1, w_2) \in R''_K$ or $w_1 = w$, and $(w'', w_2) \in R''_K$; and
5. For any $w_1, w_2 \in W''$, $(w_1, w_2) \in R''_A$ iff $(w_1, w_2) \in R_A$.

It is easy to see that $K(M') = K(M'') = E$, and $A(M') = A(M)$. ■

4 Fixed-point nonmonotonic logics

The two major fixed-point nonmonotonic logics in the AI literature are Reiter's default logic [25] and Moore's autoepistemic logic [21].

4.1 Default logic

Default logic [25] was originally defined with respect to a first-order language. For the sake of consistency with autoepistemic logic, the default logic used here is with respect to a propositional language. In this subsection, we define sentences as base formulas.

A default theory, adapted to a propositional language, is a pair (W, D), where W is a set of sentences and D a set of defaults, which are expressions of the form

$$p : q_1, \ldots, q_n/r,$$

where $p, q_1, \ldots, q_n, r$ are sentences. According to Reiter [25], a set of sentence E is a *default extension* of a default theory $\Delta = (W, D)$ iff $E = \Gamma(E)$, where Γ is the following operator: For any set of sentences S, $\Gamma(S)$ is the smallest set satisfying the following three properties:

1. $W \subseteq \Gamma(S)$.

2. $\Gamma(S)$ is closed under tautological consequence.

3. If $(p : q_1, ..., q_n/r) \in D$, $p \in \Gamma(S)$, and $\neg q_1, ..., \neg q_n \notin S$, then $r \in \Gamma(S)$.

The following proposition, proved as Theorem 2.1 in [25], seems central for default logic:

Proposition 4.1 *A set of sentences E is a default extension of the closed default theory $\Delta = (W, D)$ iff*

$$E = \bigcup_{i=0}^{\infty} E_i,$$

where the E_i's are defined as follows: $E_0 = W$, and for any $i \geq 0$,[3]

$$E_{i+1} = Th(E_i) \cup \{r \mid (p : q_1, ..., q_n/r) \in D, p \in E_i, \neg q_1, ..., \neg q_n \notin E\},$$

where for any set S, $Th(S)$ is the logical closure of S under tautological consequence.

A default theory $\Delta = (W, D)$ is translated into the following set of formulas Δ_{GK} in GK:

1. If $p \in W$, then $Kp \in \Delta_{GK}$.

2. If $(p : q_1, \ldots, q_n/r) \in D$, then

$$Kp \wedge \neg A \neg q_1 \wedge \ldots \wedge \neg A \neg q_n \supset Kr \in \Delta_{GK}.$$

[3]Notice the appearance of E in the definition of E_{i+1}.

Theorem 2 *A consistent set of sentences E is a default extension of Δ iff there is a preferred model M of Δ_{GK} such that $E = K(M)$.*

Proof: This follows straightforwardly from Theorem 1 and Proposition 4.1, and the fact that if E is consistent and closed under tautological consequence, then there is a Kripke interpretation M such that $K(M) = A(M) = E$. The last fact was proved in the proof of Theorem 1. ∎

4.2 Autoepistemic logic

Autoepistemic logic was proposed by Moore [21] as a reconstruction of McDermott and Doyle's nonmonotonic logic [17]. The language Moore used is a propositional one augmented by a modal operator L for belief. In the following, by a L-sentence we mean one that may contain the modal operator L but not K nor A.

Let S be a set of L-sentences. According to Moore, a set of L-sentences E is a *stable expansion (or AE extension)* of S if

$$E = Th(S \cup \{L\varphi \mid \varphi \in E\} \cup \{\neg L\varphi \mid \varphi \notin E\}) \tag{2}$$

where Th is the tautological closure operator. Konolige [6] proves that for any set of L-sentences S, there is a set of L-sentences S' such that (a) a set E is a stable expansion of S iff it is a stable expansion of S', and (b) every member of S' is in *normal* form, that is, of the form:

$$Lp \wedge \neg Lq_1 \wedge \ldots \wedge \neg Lq_n \supset r$$

where $p, q_1, ..., q_n, r$ are base formulas, $n \geq 0$, and Lp may be absent. Thus without the lose of generality, in the following we assume that S is always a set of L-sentences of the normal form.[4]

[4]Theoretically, it is not necessary to use the *normal form*. Under the assumption that A satisfies the weak S5 system, and K and A satisfy some properties such as $K(Ap \vee q) \equiv Ap \vee Kq$, an arbitrary L-sentence

Similar to Proposition 4.1 for default logic, we have the following proposition for autoepistemic logic:

Proposition 4.2 *A stable set E is a stable expansion of S iff $Base(E) = E_2$ where $E_2 = Th(E_1 \cup \{r \mid Lp \wedge \neg Lq_1 \wedge \ldots \wedge \neg Lq_n \supset r \in S$, where $p \in Base(E)$ and $q_1, ..., q_n \notin Base(E)\})$, and $E_1 = \{p \mid p \in S$ is a base formula$\}$.*

This proposition was stated in [12]. It was also given independently in [18].

For any such S, we define S_{GK} by the following equation:

$$S_{GK} = \{Ap \wedge \neg Aq_1 \wedge \ldots \wedge \neg Aq_n \supset Kr \mid Lp \wedge \neg Lq_1 \wedge \ldots \wedge \neg Lq_n \supset r \in S\}.$$

Parallel to Theorem 2, we have the following result:

Theorem 3 *A consistent stable set of L-sentences E is a stable expansion of a set of normal L-sentences S iff there is a preferred model M of S_{GK} such that $K(M) = Base(E)$, where $Base(E)$ is the set of base formulas in E.*

Proof: Again, this follows straightforwardly from Theorem 1 and Proposition 4.2, and the fact that if E is consistent and closed under tautological consequence, then there is a Kripke interpretation M such that $K(M) = A(M) = E$. The last fact was proved in the proof of Theorem 1. ■

Our semantics for autoepistemic logic is closely related to Moore's possible world semantics [22].[5] As we said earlier in the introduction, our logic is to be understood in epistemic terms the same way that Moore's autoepistemic logic is. In fact, our requirement of knowledge and assumption being *identical* corresponds to Moore's requirement that the actual world must be *one of* the worlds that are compatible with what the agent believes.

can be translated into a sentence in our language by inserting K in front of the L-sentence and replacing every L by A. The following Theorem 3 will also be true for this transformation.

[5]We are thankful to one of the anonymous referees for pointing this out.

In summary, by Theorem 2, we have the following correspondence between GK and default logic:

$$p : q_1, \ldots, q_n/r \Leftrightarrow Kp \wedge \neg A\neg q_1 \wedge \ldots \wedge \neg A\neg q_n \supset Kr,$$

and by Theorem 3, we have the following correspondence between GK and autoepistemic logic:

$$Lp \wedge \neg L\neg q_1 \wedge \ldots \wedge \neg L\neg q_n \supset r \Leftrightarrow Ap \wedge \neg A\neg q_1 \wedge \ldots \wedge \neg A\neg q_n \supset Kr.$$

Thus the difference between default and autoepistemic logics lies in their different interpretations of the premise p. While default logic treats the premise p and the conclusion r in the same way (as knowledge, or derived beliefs, in our terminology), autoepistemic logic treats the premise p and the consistency assumptions q's in the same way (as assumptions, or assumed beliefs, in our terminology). The effect of this difference is that the notion of default extensions is stronger than that of stable expansions, that is, under Konolige's transformation [6]

$$p : q_1, \ldots, q_n/r \Leftrightarrow Lp \wedge \neg L\neg q_1 \wedge \ldots \wedge \neg L\neg q_n \supset r,$$

if E is a default extension of a default theory and T is a stable set such that $Base(T) = E$, then T is also a stable expansions of the corresponding autoepistemic theory, but the converse is not true in general [6]. There is, however, a special case in which the converse is also true. If Δ is a default theory without premises, that is, if every default in Δ has the form

$$: q_1, \ldots, q_n/r,$$

then Konolige's transformation is exact, that is, a set of base sentence E is a default extension of Δ iff there is a stable set T such that $Base(T) = E$ and T is a stable expansion of the corresponding autoepistemic theory. This result is also proved in [12], and now follows trivially from Theorem 2 and Theorem 3.

We mentioned in the introduction that both default logic and Moore's autoepistemic logic are formalizations of certain autoepistemic rules, and GK is intended as a generalization of both. For instance, as we have seen in Example 3.1, formulas of the form

$$Kp \wedge \neg Aq_1 \wedge \ldots \wedge \neg Aq_n \supset Kr_1 \vee \ldots \vee Kr_m$$

formalize so-called disjunctive defaults [4], which are not expressible in default logic. Similarly, formulas of the form

$$Ap \wedge \neg Aq_1 \wedge \ldots \wedge \neg Aq_n \supset Kr_1 \vee \ldots \vee Kr_m$$

can be thought of as expressing "disjunctive autoepistemic rules," and formulas of the form

$$Kp_1 \wedge Ap_2 \wedge \neg Aq_1 \wedge \ldots \wedge \neg Aq_n \supset Kr_1 \vee \ldots \vee Kr_m$$

generalize both disjunctive default and disjunctive autoepistemic rules.

In recent years, we have seen some related work. Siegel [30] also explicitly introduced the notions of knowledge and assumptions. However, Siegel defined assumptions in terms of knowledge, and used a different minimization strategy. As a result, Siegel was only able to account for default logic. Recently, Marek, Schwarz, and Truszczynski [19, 20, 31, 32] showed that default logic and autoepistemic logic, among others, can be captured by varying the underlying monotonic logics in McDermott and Doyle's fixed-point construction [17]. Furthermore, preference semantics were proposed for a large class of such logics by Schwarz [26]. Using the preference semantics, Schwarz and Truszczynski [27] show that for a class of formulas including the ones in Theorem 1, GK can be embedded into S4F modal nonmonotonic logic[6] by replacing the assumption operator A with $\neg K \neg K$.

5 Negation-as-failure as circumscription

As with circumscription, our logic GK is based on minimization, and thus provides a bridge between fixed-point nonmonotonic logics and minimal model nonmonotonic logic. As an

[6]S4F modal nonmonotonic logic is McDermott and Doyle's logic with the modal system S4F as the underlying monotonic logic

application of this bridge, we propose a formalization of logic programs in circumscription in this section.

Recall that the basic strategy in GK is to minimize knowledge with a fixed set of assumptions, and then compare the resulting minimal knowledge and the set of assumptions. If we reify propositions, then this strategy can be easily captured in circumscription.

Consider a first-order language with a distinguished *formula* sort for base formulas in GK. Instead of modal operators, we have two unary predicates, $Known(x)$ for knowledge, and $Assumed(x)$ for assumption, where x is of sort *formula*. Thus a modal formula in GK such as

$$Kp \wedge \neg Aq \supset Kr$$

can be translated into the first-order language as:

$$Known('p') \wedge \neg Assumed('q') \supset Known('r').$$

Of course, there are some axioms relating, for example, $Known('p \wedge q')$ and $Known('p') \wedge Known('q')$, and possibly p and $Known('p')$. Suppose $W0$ is the set of such axioms. Given any GK formula φ without nested modalities, let $Reify(\varphi)$ be the translation of φ in the two-sorted first order language. It can be shown that preferred models of φ correspond to models of the following second-order sentence:

$$Circum(W0 \wedge Reify(\varphi); Known) \wedge \forall x(Known(x) \equiv Assumed(x)), \qquad (3)$$

where $Circum(W0 \wedge Reify(\varphi); Known)$ is the circumscription of $Known$ in $W0 \wedge Reify(\varphi)$ (see [8]).

In the general case, the price of reification seems too high[7]. However, in logic programs, where the sentences in a rule are literals, reification is easy. In this case, the above connection between GK and circumscription yields an interesting relationship between logic

[7]John McCarthy pointed out that for propositional logic, we only need about five axioms (in order to reify propositional logic). But the effects of circumscribing predicates in these axioms seem hard to capture, witness the mechanisms needed in [11]. In any event, the advantages that circumscription can bring us in the general case by reification are less clear than in the special case of logic programs.

programs and circumscription. In this section, we shall explore this relationship. Our work can be considered an extension of [9] and [23], which provide semantics for stratified logic programs in circumscription.

Consider a first-order language. A logic program consists of rules of the following form:

$$p_0 \leftarrow p_1, ..., p_m, \neg p_{m+1}, ..., \neg p_n, \tag{4}$$

where p_i, $0 \leq i \leq n$, is an atomic formula, and $0 \leq m \leq n$. If (4) contains variables, then it is an *open* rule, otherwise, it is a *closed* rule. Usually, an open rule is considered to be the set of closed rules obtained by replacing the variables with the closed terms.

Let $\mathcal{P}$ be a logic program with only closed rules. Let us call an atomic sentence an *atom*. A set of atoms E is an *answer set* of $\mathcal{P}$ iff $\Gamma(E) = E$, where for any set S, $\Gamma(S)$ is the smallest set of atoms satisfying the following property:

> If $p_0 \leftarrow p_1, \ldots, p_m, \neg p_{m+1}, \ldots, \neg p_n \in \mathcal{P}$, $p_1, ..., p_m \in \Gamma(S)$, and $p_{m+1}, ..., p_n \notin S$, then $p_0 \in \Gamma(S)$.

The rule (4) corresponds to the following default[8]

$$(p_1 \wedge \ldots \wedge p_m : \neg p_{m+1}, \ldots, \neg p_n)/p_0,$$

which can be translated into the following formula in GK (see Section 4.1):[9]

$$(Kp_1 \wedge ... \wedge Kp_m \wedge \neg Ap_{m+1} \wedge ... \wedge \neg Ap_n) \supset Kp_0,$$

which is translated into the following first-order sentence:

$$Known('p_1') \wedge ... \wedge Known('p_m') \wedge \neg Assumed('p_{m+1}') \wedge ... \wedge \neg Assumed('p_n') \supset Known('p_0')$$

[8] See, e.g., [3, 1, 12].

[9] Since every p_i is an atomic sentence, thus can be considered a primitive proposition, the difference between first-order language used in logic programs and propositional language used in GK is not important here.

Now since p_0,...,p_n are atomic sentences, and the only function of $Known('p')$ is labeling p as known, rather than assumed. Thus the above sentence can be replaced by the following one:

$$p_1 \wedge ... \wedge p_m \wedge \neg p'_{m+1} \wedge ... \wedge \neg p'_n \supset p_0, \tag{5}$$

where p'_i, $m+1 \leq i \leq n$, is the result of replacing the predicate P in p_i by a corresponding new predicate P' of the same arity. Now the second-order sentence (3) corresponds to that we first minimize all predicates in the original language with the new predicates fixed, and then for every predicate P in the original langauge, we equate it with the corresponding P'. Let us see a simple example.

Example 5.1 Let $\mathcal{P}$ be a logic program with the following rules:

$$Q(b),$$
$$Q(a) \leftarrow Q(b), \neg Q(c).$$

$E = \{Q(b), Q(a)\}$ is the unique answer set of $\mathcal{P}$.

Let F be the conjunction of the following sentences:

$$a \neq b \neq c,$$
$$Q(b) \wedge (Q(b) \wedge \neg Q'(c) \supset Q(a)).$$

Then F is the reification of $\mathcal{P}$. It can be shown that $Circum(F;Q)$ is equivalent to

$$F \wedge [Q'(c) \supset \forall x(Q(x) \equiv x = b)] \wedge [\neg Q'(c) \supset \forall x(Q(x) \equiv x = b \vee x = a)].$$

Thus the sentence $\forall x(Q(x) \equiv Q'(x)) \wedge Circum(F;Q)$ is equivalent to the following one:

$$F \wedge \forall x(Q(x) \equiv Q'(x)) \wedge \forall x(Q(x) \equiv x = b \vee x = a),$$

which implies the set of ground atoms given by E. ■

In general, for any given finite logic program $\mathcal{P}$, let $C(\mathcal{P})$ be the conjunction of the sentences obtained by translating every rule (4) in $\mathcal{P}$ into the universal closure of (5). If $\mathcal{P}$

contains only closed rules, then let $EQ(\mathcal{P})$ be the unique names assumption for the closed terms in $\mathcal{P}$. For any formula W, and a set of predicates $\vec{P}$, let $Circum(W;\vec{P})$ be the (parallel) circumscription of the predicates in $\vec{P}$ in the formula W [16, 8].

Theorem 4 *Let $\mathcal{P}$ be a finite logic program with only closed rules. For any set E of ground atoms, E is an answer set of $\mathcal{P}$ iff there is a model M of the following sentence*

$$\bigwedge_{P\in\mathcal{P}} \forall x(P(x) \equiv P'(x)) \wedge Circum(EQ(\mathcal{P}) \wedge C(\mathcal{P}); \vec{P}) \tag{6}$$

such that E is the set of the ground atoms in $\mathcal{P}$ satisfied in M, where $\vec{P}$ is the set of predicates appeared in $\mathcal{P}$.

Proof: This theorem is a direct consequence of the relationships between logic prorgam and default logic, between default logic and GK, and between GK and circumscription as we have outlined above. It is also interesting to see a direct proof of the theorem. Suppose E is an answer set of $\mathcal{P}$. Then E will contain only ground atoms that appear in $\mathcal{P}$. Construct an Herbrand model M satisfying the following property

> For any predicate $P \in \mathcal{P}$, any tuple of closed terms t, $M \models P(t)$ iff $M \models P'(t)$ iff $P(t) \in E$.

We show that M satisfies (6). Since M is an Herbrand model, thus it satisfies $EQ(\mathcal{P})$. $M \models C(\mathcal{P})$ follows easily from the construction of M, and the assumption that E is an answer set. We now show that M is minimal. If not, suppose there is a M' such that $M' \models C(\mathcal{P})$, and $M' \sqsubset_{\bigwedge_{P\in\mathcal{P}} P(x)} M$. Let E' be the set of ground atoms that are true in M' and appear in $\mathcal{P}$. It can be seen that E' satisfies the property for $\Gamma(E)$. Thus $\Gamma(E) \subseteq E' \subset E = \Gamma(E)$, a contradiction. Therefore, M must be minimal, and satisfy (6).

Suppose M is a model of (6). Let E be the set of the ground atoms that are true in M and appear in $\mathcal{P}$. We show that E is an answer set of $\mathcal{P}$, i.e. $E = \Gamma(E)$. It is easy to see that E satisfies the property for $\Gamma(E)$. Thus $\Gamma(E) \subseteq E$. Now let M' be just like M except that for any ground atoms p in $\mathcal{P}$, $M' \models p$ iff $p \in \Gamma(E)$. This is a sound definition since M

is a model of $EQ(\mathcal{P})$. It is easy to check that M' is a model of $C(\mathcal{P})$, and $M' \sqsubseteq_{\bigwedge_{P\in\mathcal{P}} P(x)} M$. Thus $M = M'$, and $E = \Gamma(E)$. ■

Corollary 4.1 *Under the assumptions of the theorem, a ground atom in $\mathcal{P}$ is in every answer set of $\mathcal{P}$ iff it is a logical consequence of (6).*

The proof of Theorem 4 also gives the following result:

Theorem 5 *Let $\mathcal{P}$ be a finite logic program. A set E is an answer set of $\mathcal{P}$ iff there is an Herbrand model M such that (1) for any ground atom p in the language of $\mathcal{P}$, $p \in E$ iff $M \models p$, and (2) M is a model of the following sentence:*

$$\bigwedge_{P\in\mathcal{P}} \forall x(P(x) \equiv P'(x)) \wedge Circum(C(\mathcal{P}); \vec{P}).$$

For finite logic programs with free variables, we may still have the corresponding unique names assumption for them. In this case, Theorem 4 will be applicable.

Example 5.2 Consider the following program $\mathcal{P}$ [2]:

$$\begin{aligned} &P(1,2), \\ &Q(x) \quad \leftarrow \quad P(x,y), \neg Q(y). \end{aligned}$$

The unique names assumption for $\mathcal{P}$ is $1 \neq 2$. Let A be the following sentence

$$1 \neq 2 \wedge P(1,2) \wedge \forall xy(P(x,y) \wedge \neg Q'(y) \supset Q(x)).$$

Then $Circum(A; \{P, Q\})$ is equivalent to

$$A \wedge \forall xy[P(x,y) \equiv (x = 1 \wedge y = 2)] \wedge \forall x[Q(x) \equiv (x = 1 \wedge \neg Q'(2))]$$

Thus $\mathcal{P}$ is captured by the following sentence:

$$\forall xy(P(x,y) \equiv P'(x,y)) \wedge \forall x(Q(x) \equiv Q'(x)) \wedge Circum(A; \{P, Q\}),$$

which is equivalent to the following sentence as far as P and Q are concerned:

$$\forall xy(P(x,y) \equiv x = 1 \wedge y = 2) \wedge \forall x(Q(x) \equiv x = 1) \wedge 1 \neq 2,$$

which concisely summarize the unique answer set of $\mathcal{P}$: $\{P(1,2), Q(1)\}$. ■

It is particularly interesting to note that if a logic program has more than one answer set, then the circumscription of the program gives us the intersection of the answer sets.

Example 5.3 Consider the following logic program $\mathcal{P}$:

$$P(a) \leftarrow \neg P(b),$$
$$P(b) \leftarrow \neg P(a).$$

The circumscription of P in

$$(\neg P'(a) \supset P(b)) \wedge (\neg P'(b) \supset P(a)) \wedge a \neq b$$

is

$$\forall x(x \neq a \wedge x \neq b \supset \neg P(x)) \wedge (\neg P'(a) \equiv P(b)) \wedge (\neg P'(b) \equiv P(a)) \wedge a \neq b.$$

Together with $\forall x(P(x) \equiv P'(x))$, it gives

$$P(a) \equiv \neg P(b) \wedge \forall x(x \neq a \wedge x \neq b \supset \neg P(x)).$$

■

Our formalization of logic programs in circumscription can be generalized to logic programs with disjunctive heads (see, for example, [3, 24]).

Formally, a *disjunctive data base* is a set of rules of the following form:

$$p_0|p_1|...|p_l \leftarrow p_{l+1}, ..., p_m, \neg p_{m+1}, ..., \neg p_n, \tag{7}$$

where $p_0, ..., p_n$ are atoms, and $0 \leq l \leq m \leq n$. Intuitively, $p_0|p_1|...|p_l$ is $p_0 \vee p_1 \vee ... \vee p_l$, i.e., one of p_i, $0 \leq i \leq l$, must be true.

The rule (7) can be traslated into the universal closure of the following sentence:

$$p_{l+1} \wedge \dots \wedge p_m \wedge \neg p'_{m+1} \wedge \dots \wedge \neg p'_n \supset p_0 \vee \dots \vee p_l, \tag{8}$$

and Theorem 4, Corollary 4.1, and Theorem 5 can be extended appropriately. We shall not go into detail here. Rather, we shall see a simple example.

Example 5.4 Consider $\mathcal{P}_1 = \{P(a)|P(b) \leftarrow\}$. There are two answer sets, $E_1 = \{P(a)\}$, and $E_2 = \{P(b)\}$.

$C(\mathcal{P}_1)$ is $P(a) \vee P(b)$, and $EQ(\mathcal{P}_1)$ is $a \neq b$. The circumscription of P in $C \wedge EQ$ is

$$a \neq b \wedge [\forall x(P(x) \equiv x = a) \vee \forall x(P(x) \equiv x = b)],$$

which summarizes E_1 and E_2.

Let $\mathcal{P}_2 = \{P(a), P(x)|Q(x) \leftarrow\}$. $\{P(a)\}$ is the unique answer set of $\mathcal{P}_2$. $C(\mathcal{P}_2)$ is

$$P(a) \wedge \forall x(P(x) \vee Q(x)).$$

The circumscription of P and Q in $C(\mathcal{P}_2)$ is

$$P(a) \wedge \forall x(P(x) \equiv \neg Q(x)).$$

■

6 Summary

We have defined the logic GK, of knowledge and justified assumptions, and provided two very similar transformations from default logic and autoepistemic logic into GK. The transformations provide for the first time uniform epistemic semantics for the two logics, and thus a common semantic background against which they can be clearly compared.

Using a relationship between GK and circumscription, we have shown that logic programs with negation as failure can be easily captured n circumscription by introducing, for each predicate, a new one with the same arity.

Recently, Lifschitz [10] has extended a version of GK to first-order case, and shown that it can be used to formalize disjunctive data bases with classical negation, and to define epistemic queries to logic programs. We expect other similar applications of GK in the future.

Acknowledgements

We would like to thank the anonymous referees for helpful comments on an earlier draft of this paper.

References

[1] N. Bidoit and C. Froidevaux, Negation by default and nonstratifiable logic programs, Technical Report 437, Universite Paris XI (1988).

[2] M. Gelfond and V. Lifschitz, The stable model semantics for logic programming, in: R. Kowalski and K. Bowen, editors, *Logic Programming: Proc. of the Fifth International Conference*, (1988) 1070–1080.

[3] M. Gelfond and V. Lifschitz, Classical negation in logic programs and disjunctive databases, *New Generation Computing*, 1991. To appear.

[4] M. Gelfond, V. Lifschitz, H. Przymusinska, and M. Truszczynski, Disjunctive defaults, in: *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1991) 230–237.

[5] J. Y. Halpern and Y. O. Moses, Towards a theory of knowledge and ignorance: preliminary report, IBM Technical Report RJ 4448 (48136), 1984.

[6] K. Konolige, On the relation between default logic and autoepistemic logic, *Artificial Intelligence*, 35 (1988) 343–382.

[7] H. Levesque, All I know: a study in autoepistemic logic, in: *Conference Proceedings of AAAI–87*, Seatle, WA (1987) 426–431.

[8] V. Lifschitz, Pointwise circumscription, in: M. Ginsberg, ed., *Readings in Nonmonotonic Reasoning*, (Morgan Kaufmann, San Mateo, CA, 1987) 179–193.

[9] V. Lifschitz, On the declarative semantics of logic programs with negation, in: M. Ginsberg, ed., *Readings in Nonmonotonic Reasoning*, (Morgan Kaufmann, San Mateo, CA, 1987) 337–350.

[10] V. Lifschitz, Nonmonotonic databases and epistemic queries, in: *Conference Proceedings of IJCAI–91*, Sydney, Australia (1991) 387–396.

[11] F. Lin, Circumscription in a modal logic, in: *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, Asilomar, CA (1988) 113–128.

[12] F. Lin and Y. Shoham, Argument systems: a uniform basis for nonmonotonic reasoning, in: *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Canada (1989) 245–255.

[13] F. Lin and Y. Shoham, Epistemic semantics for fixed-point nonmonotonic logics, in: *Proceedings of the Third Conference on Theoretical Aspects of Reasoning about Knowledge*, Asilomar, CA (1990) 111–120.

[14] F. Lin, A Study of Nonmonotonic Reasoning, Ph.D. Thesis, Department of Computer Science, Stanford University, CA (1991).

[15] J. McCarthy, Circumscription — A form of non-monotonic reasoning, *Artificial Intelligence* 13(1980) 27–39.

[16] J. McCarthy, Applications of circumscription to formalizing commonsense knowledge, *Artificial Intelligence* 28 (1986) 89–118.

[17] D. McDermott and J. Doyle, Nonmonotonic logic I, *Artificial Intelligence* 13(1980) 41–72.

[18] W. Marek and M. Truszczynski, Relating autoepistemic and default logics, in: *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Canada (1989) 276–288.

[19] W. Marek and M. Truszczynski, Modal logic for default reasoning, *Annals of Mathematics and Artificial Intelligence,* 1 (1990) 275–302.

[20] W. Marek, G. F. Shvarts, and M. Truszczynski, Modal nonmonotonic logics: ranges, characterization, computation, in: *Proceedings of the Second Conference on the Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1991) 395–404.

[21] R. Moore, Semantical considerations on nonmonotonic logic, in: *Conference Proceedings of IJCAI–83*. Also *Articifial Intelligence*, 25 (1985) 75–94.

[22] R. Moore, Possible-world semantics for autoepistemic logic, in: *Proceedings of 1984 Workshop on Nonmonotonic Reasoning*, New Paltz, NY, 1984. Reprinted in: M. Ginsberg, ed., *Readings in Nonmonotonic Reasoning*, (Morgan Kaufmann, San Mateo, CA, 1987) 137–142.

[23] T. Przymusinski, On the declarative and procedural semantics of logic programs, *Journal of Automated Reasoning*, 5 (1989) 167–205.

[24] T. Przymusinski, Extended stable semantics for normal and disjunctive programs, in: D. Warren and P. Szeredi, eds., *Logic Programming: Proc. of the Seventh International Conference* (1990) 459–477.

[25] R. Reiter, A logic for default reasoning, *Artificial Intelligence* 13 (1980) 81–132.

[26] G. Schwarz, Minimal model semantics for nonmonotonic modal logics, Technical Report 184-91, Department of Computer Science, University of Kentucky, Kentucky (1991).

[27] G. Schwarz and M. Truszczynski, Modal logic S4F and the minimal knowledge paradigm, to appear in: *Proceedings of the Fourth Conference on Theoretical Aspects of Reasoning about Knowledge*, Asilomar, CA (1992).

[28] Y. Shoham, Nonmonotonic logics: meaning and utility, in: *Proceedings of IJCAI–1987*, Milan, Italy (1987) 388–393.

[29] Y. Shoham, *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*, (MIT Press, Cambridge, MA, 1988).

[30] P. Siegel, A modal language for nonmonotonic logic, in: *Proceedings of Workshop DRUMS/CEE,* Marseille, France (1990) 1–8.

[31] M. Truszczynski, Modal interpretations of default logic, in: *Proceedings of IJCAI–91*, Sydney, Australia (1991) 393–398.

[32] M. Truszczynski, Embedding default logic into modal nonmonotonic logics, in: *Proceedings of the Workshop on Logic Programming and Non-monotonic Reasoning*, Washington (1991) 151–165.