TR-0833

Rules Representing Two Types of Epistemic Statements

by

N. Iwayama

March, 1993

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

Rules representing two types of epistemic statements

Noboru IWAYAMA

Institute for New Generation Computer Technology
1-4-28 Mita, Minato-ku, Tokyo 108, Japan
phone: +81-3-3456-2514, fax: +81-3-3456-1618
email: iwayama@icot.or.jp

Abstract

We propose a new semantics for a variant of Moore's autoepistemic logic, GK, defined by Lin and Shoham. By using the new semantics, we can describe undefinedness as a certain state in an agent's belief set. We emphasize that our semantics does not use three-valued interpretations but two-valued ones. Our new semantics reduces a drawback of Lin and Shohams' semantics in translating Reiter's default logic into GK formulas. Logic programming is captured in GK through the new semantics. For normal logic programs, the semantics of GK coincides with Przymusinski's stationary semantics. Even though a logic program has a partial model, we can check whether integrity constraints are satisfied in the new semantics of GK.

1 Introduction

Moore's autoepistemic logic is a logic for modeling the beliefs of agents who reflect on their own beliefs, and formalizes the beliefs of rational agents in *stable* expansions by Stalnaker and Moore [Moore85]. They call a belief set S *stable* if it satisfies the following three properties:

1. S is closed under first-order consequence,

2. If $\phi \in S$, then $L\phi \in S$, and

3. If $\phi \notin S$, then $\neg L\phi \in S$,

where an expression $L\phi$ means that an agent believes ϕ. According to stable expansion, all propositions are either believed or not believed. Namely, if we adopt the view that the belief states of the rational agent are formalized in the stable expansions, the agent should decide whether she knows every matter in the world. However, we cannot hypothesize that an agent has such a strong ability, because she may answer, for some proposition, that she cannot decide whether she knows or not. Therefore, it is more appropriate to formalize the possibility that the agent reserves her belief state.

Przymusinski [Przymusinski89] introduced the *three-valued* autoepistemic logic to investigate the relationship among major non-monotonic formalisms. In the proposed autoepistemic logic, the "undefinedness" of propositions is represented in a meta-logical way, that is, each proposition may be assigned to one of three possibilities — true, false, or *undefined*. We can regard the truth value of "undefined" as reserevation by the agent.

Lin and Shoham [Lin92] argue that there are two kinds of beliefs involved in the autoepistemic reasoning process; the beliefs *assumed* by the agent and the new beliefs *derived* by applying the rules in a fixed belief set. Because, in Moore's autoepistemic logic, all beliefs in the statement of a rule are assumed, derived beliefs are meta-logical in the logic, and cannot be referred to in the language. Therefore, they proposed a new epistemic logic, called GK, in which both kinds of beliefs are explicit. They directly introduce two model operators in GK, one for knowledge (K), which corresponds to derived beliefs, and the other for assumptions (A), which corresponds to assumed beliefs. They define the semantics, called *preferred models*, based on a preference relation over Kripke interpretations. They do not mention the possibility of agent's reservations, namely, "undefined" propositions.

In this paper, we propose a new semantics for the epistemic logic, GK. By using the new semantics, we can describe undefinedness as a certain state in an agent's belief set. We emphasize that our semantics does not use three-valued interpretations, but two-valued ones, although Przymusinski did use three-valued interpretations.

A proposed semantics has many excellent properties related to Reiter's default logic and logic programming. The property of the semantics to be argued first is that the semantics avoids a drawback of Lin and Shohams' semantics. Lin and Shoham [Lin92] showed a translation of default theories into formulas in GK, and achieved a coincidence of default extensions for a default theory with the semantics of the translated formulas in GK. However, this means that the semantics has drawbacks in the default extensions. For example, any default theory containing the default $(: \neg p/p)$ and no other defaults in which the proposition p occurs has no extension. Its translation into GK, $\neg Ap \supset Kp$, has no preferred model. The new semantics proposed here (and the relevant translation of default logic) does not have such a drawback.

Our new semantics for the epistemic logic contributes to the study of semantics for logic programming. Bidoit and Froidevaux [Bidoit91] showed a correspondence between Reiter's default logic and logic programming. This correspondence provides us with a correspondence between the epistemic logic GK and logic programming. With this relation between GK and logic programming, for normal logic programs, our semantics for GK coincides with the stationary semantics proposed by Przymusinski [Przymusinski91].

Another contribution of this work to logic programming is involved with integrity constraints. There is much work that discusses the semantics of logic programming with integrity constraints. In previous work, the underlying semantics for the logic programs was a semantics which allows only *total* models, because it is not easy to consider the satisfaction of integrity constraints by a semantics which allows partial models, or "undefined" propositions. Since, in this paper, we translate integrity constraints into GK formulas as rules, we can deal with integrity satisfaction in terms of our new semantics of GK. Therefore, we can distinguish unsatisfaction of integrity from the existence of undefined propositions, and can see that our semantics will be a useful formalism for applications such as deductive databases and diagnosis.

This paper is organized as follows. We define the logic GK, which is first defined in [Lin92]. Then we define a new semantics for GK. In Section 4, we describe the translation of Reiter's default logic into GK. We discuss the contributions of GK to logic programming in Sections 5 and 6. We compare our results with Przymusinska and Przymusinskis' stationary expansions [Przymusinska91] for default logic.

2 Language Definition

In this section, we refer to the preliminary definitions from [Lin92].

The logic GK is a propositional one, augmented with two modalities, K and A. Well-formed formulas are defined as usual. Intuitively, $K\varphi$ means that φ is known or believed, while $A\varphi$ means that φ is assumed. The important point to note is that, although the distinctions between $K\varphi$ and φ are not important in [Lin92], we should distinguish the two to identify the undefinedness as an agent's belief state.

A *Kripke structure* is a tuple (W, π, R_K, R_A), where W is a nonempty set, $\pi(W)$ is a truth assignment to the primitive propositions for each $w \in W$, and R_K and 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.

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:

- $((W, \pi, R_K, R_A), w) \models \varphi$ iff $\pi(w)(p) = 1$, where p is a primitive proposition.
- $M \models \varphi_1 \wedge \varphi_2$ iff $M \models \varphi_1$ or $M \models \varphi_2$.
- $M \models \neg\varphi$ iff it is not the case that $M \models \varphi$.
- $((W, \pi, R_K, R_A), w) \models K\varphi$ iff $((W, \pi, R_K, R_A), w') \models \varphi$ for any $w' \in W$ such that $(w, w') \in R_K$.
- $((W, \pi, R_K, R_A), w) \models A\varphi$ 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 define the following for each Kripke interpretation M:

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

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

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

and,

$$\overline{K}(M) = \{\varphi | M \not\models K\varphi, \varphi \text{ is a base formula}\},$$
$$\overline{A}(M) = \{\varphi | M \not\models A\varphi, \varphi \text{ is a base formula}\},$$
$$\overline{B}(M) = \{\varphi | M \not\models \varphi, \varphi \text{ is a base formula}\},$$

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

3 Proposed Semantics

In this section, we provide the semantics for GK. Our semantics is not new, because we obtain the semantics by modifying the semantics proposed by Lin and Shoham. Thus, we first show the semantics by Lin and Shoham.

Definition 3.1 *Let M_1 and M_2 be two Kripke interpretations. We say that M_1 is* K-preferred over *M_2, written $M_1 \sqsubset_K M_2$, if:*

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

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

A model M of S is K-minimal *if there is no other model M' of S such that $M' \sqsubset_K M$.*

Lin and Shoham defined a semantics for GK as follows.

Definition 3.2 *[Lin92]*
Let S be a set of formulas, and M be a Kripke interpretation. We say that M is a preferred model *of S if:*

1. *M is a K-minimal model of S and*

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

The semantics proposed by Lin and Shoham is concerned only with beliefs, not propositions (or formulas that does not contain any modal operators) derived under the belief set. Namely, they do not pay attention to whether the propositions are derived under the belief set, if the assumed beliefs coincide with derived beliefs. As mentioned in Section 1, for Moore's autoepistemic logic, *stable* expansion has been considered as the semantic framework [Moore85]. In stable expansion, a belief set should be consistent with the derived propositions. We incorporate this point in our new semantics for GK.

Definition 3.3 *Let M_1 and M_2 be two K-minimal models. We say that M_1 is* B-preferred *over M_2, written $M_1 \sqsubset_B M_2$, if:*

1. $A(M_1) = A(M_2)$,
2. $K(M_1) = K(M_2)$, *and*
3. $B(M_1) \subset B(M_2)$.

A model M of S is B-minimal *if there is no other model M' of S such that $M' \sqsubset_B M$.*

Definition 3.4 *Let S be a set of formulas, and M be a Kripke interpretation. We say that M is an* acceptable model *of S if:*

1. *M is a B-minimal model of S,*
2. $A(M) \subseteq K(M)$, *and*
3. $B(M) = A(M)$.

In acceptable models, if the agent assumes some proposition, then the agent should know the proposition and the proposition should be derived. On the other hand, if the agent does not assume some proposition, then the proposition must not be derived. But, it is possible that the agent knows the proposition which the agent does not assume. For example, $\{\neg Ap \supset Kp\}$ has the acceptable model, $\{\neg Ap, \neg p, Kp\}$. (She knows the "unjustified" proposition or she cannot determine the belief with respect to the proposition.) The proposition p in this example is regarded as "undefined". In short, base formulas in an acceptable model M are divided into the following 3 types:

$$Pos(M) = A(M) \cap K(M),$$
$$Neg(M) = \overline{A}(M) \cap \overline{K}(M), \text{ and}$$
$$Undef(M) = \overline{A}(M) \cap K(M).$$

We note that $A(M) \cap \overline{K}(M) = \emptyset$ because $A(M) \subseteq K(M)$. We say that an acceptable model M is *partial* if $Undef(M) \neq \emptyset$, otherwise it is *total*.

To close this section, we show that preferred models are special cases of acceptable models.

Proposition 3.1 *Let S be a set of formulas, and M be a Kripke interpretation. M is an acceptable model of S if M is a preferred model of S.*

4 Default Logic Translated in GK

In this section, we describe the translation of Reiter's default logic into GK. First, we show the result by Lin and Shoham [Lin92] which provides the correspondence between preferred models and Reiter's default extensions. Then, we expand their result in terms of our new semantics. The default logic used here is restricted to propositional logic.

The following definitions show Reiter's default logic [Reiter80]. A default theory $\Delta = (\mathcal{D}, \mathcal{W})$ where $\mathcal{D}$ is a (possibly infinite) set of defaults and $\mathcal{W}$ is a first-order theory. The defaults are expressions of the form, $p : q_1, ..., q_n/r$, where $p, q_1, ..., q_n$ and r are first order sentences.

Definition 4.1 (Gamma operator) *[Reiter80]*
Let $\Delta = (\mathcal{D}, \mathcal{W})$ be a default theory and E be a first order theory. We denote the smallest first order theory as $\Gamma_\Delta(E)$ such that

- $\Gamma_\Delta(E) \supseteq W$,
- $\Gamma_\Delta(E)$ *is closed under tautological consequence, and*
- *If* $(p : q_1, ..., q_n/r) \in \mathcal{D}$, $p \in \Gamma_\Delta(E)$ *and* $\neg q_1, ..., \neg q_n \notin E$, *then* $r \in \Gamma_\Delta(E)$.

Definition 4.2 (Default extensions) *[Reiter80]*
Given a default theory Δ, E is a default extension *iff $E = \Gamma_\Delta(E)$.*

Lin and Shoham [Lin92] showed that the set of formulas in GK, into which a default theory is translated, has preferred models which are equivalent to the default extensions for the default theory. They translate a default theory $\Delta = (\mathcal{D}, \mathcal{W})$ into the following set of formulas Δ_{GK} in GK:

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

2. If $(p : q_1, ..., q_n/r) \in \mathcal{D}$, then $Kp \wedge \neg A\neg q_1 \wedge ... \wedge \neg A\neg q_n \supset Kr \in \Delta_{GK}$.

Theorem 4.1 *[Lin92]*
A consistent set of E is a default extension of Δ iff there is a preferred model M of Δ_{GK} such that $E = K(M)$.

We match the above translation with our semantics provided in the previous section. We translate a default theory $\Delta = (\mathcal{D}, \mathcal{W})$ into the following set of formulas Δ^{*}_{GK} in GK:

1. If $p \in \mathcal{W}$, then $\{p, Kp\} \subseteq \Delta^{*}_{GK}$ and

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

$$Kp \wedge \neg A\neg q_1 \wedge ... \wedge \neg A\neg q_n \supset Kr \in \Delta^{*}_{GK}, \text{ and}$$
$$p \wedge \neg K\neg q_1 \wedge ... \wedge \neg K\neg q_n \supset r \in \Delta^{*}_{GK}.$$

We characterize default extensions of a default theory Δ by the acceptable models of Δ^{*}_{GK}. Here, we show the example mentioned in Section1: a default thoery $\Delta = (\{: \neg p/p\}, \emptyset)$ has no extension. Lin and Shohams' translation $\Delta_{GK} = \{\neg Ap \supset Kp\}$ has no preferred model. On the other hand, our translation $\Delta^{*}_{GK} = \{\neg Ap \supset Kp, \neg p \supset Kp\}$ has an acceptable model $\{\neg Ap, \neg p, Kp\}$.

5 Correspondence to Stationary Semantics

In this section, we consider the translation of logic programming into GK formulas. Because we can see the natural correspondence between logic programming and default theory ([Bidoit91]), we map logic programming into default theories, and the default theories into GK formulas. As a result, acceptable models provide us with a semantic tool for considering the semantics of logic programming.

Here, we refer to the terminology used in logic programming. An *extended logic program* is a (possibly infinite) set which consists of the following rules of the form:

$$l_0 \leftarrow l_1, ..., l_m, not\, l_{m+1}, ..., not\, l_n,$$

where l_i are literals, which are either atoms p or "classically negated" atoms $\neg p$, and $n \geq m \geq 0$. In this paper, we consider only (possibly infinite) propositional programs[1].

Let P be an extended logic program and $\mathcal{D}$ be the corresponding set of defaults of the form for each rule in P:

$$l_1 \wedge ... \wedge l_m : \neg l_{m+1}, ..., \neg l_n / l_0.$$

The default theory $\Delta(P) = (\mathcal{D}, \emptyset)$ is called the translation of the extended program into default logic.

Based on the translation of default logic into GK provided in the previous section, we show the natural translation of the logic program into the set of formulas in GK. A normal logic program P is translated into $\Delta^*_{GK}(P) = \Delta^*_{KA}(P) \cup \Delta^*_{KB}(P)$, in which $\Delta^*_{KA}(P)$ contains the following formula in GK for each rule in P:

$$Kl_1 \wedge ... \wedge Kl_m \wedge \neg Al_{m+1} \wedge ... \wedge \neg Al_n \supset Kl_0,$$

and $\Delta^*_{KB}(P)$ contains the following formula in GK for each rule in P:

$$l_1 \wedge ... \wedge l_m \wedge \neg Kl_{m+1} \wedge ... \wedge \neg Kl_n \supset l_0.$$

In the following, we consider *normal logic programs* which have no classically negated atoms in thier programs, namely, sets consisting of rules of the form:

$$p_0 \leftarrow p_1, ..., p_m, not\, p_{m+1}, ..., not\, p_n.$$

Przymusinski proposed *stationary semantics* [Przymusinski91]. In [Brogi91], Brogi et al. showed that stationary semantics is equivalent to the complete scenario [Dung91] for normal logic programs. We see that acceptable models are useful for investiging logic programming, because the following theorem gives a one-to-one correspondence between stationary models and acceptable models for normal programs.

Theorem 5.1 *Stationary Semantics = Acceptable Models*
$\mathcal{M}$ is a stationary model of a logic program P iff M is an acceptable model of $\Delta^*_{GK}(P)$ such that $B(M) = \{A | \mathcal{M} \models A, A$ is an atom$\}$.

[1]This way of placing restrictions on programs is well known in the literature.

We can prove theorem 5.1 by using the following definition and two propositions. Note that the following definitions and proposition 5.1 are quoted from [Przymusinska91].

Let P be a program and let us use $\mathcal{H}$ to denote its *Herbrand base*, i.e., the set of all ground atoms. Any partial model $\mathcal{M}$ of P can be identified with a pair $< T; F >$ of disjoint subsets of $\mathcal{H}$ in which T contains those atoms which are *true* in $\mathcal{M}$ and F contains those atoms which are *false* in $\mathcal{M}$. Let us use N to denote the complement $\mathcal{H} - F$ of F, i.e., the set of all atoms which are *not* false. For our purposes it is more convenient to view a partial model as a pair $< T; N >$ of subsets of $\mathcal{H}$ so that $T \subseteq N$, T contains those atoms which are *true* in $\mathcal{M}$, and N contains those atoms which are *not false* in $\mathcal{M}$.

The following is a definition of Gelfond-Lifschitz' transformation [Gelfond88] for normal logic programs.

Definition 5.1 *Let P be a normal logic program and M be a set of ground atoms. By the* quotient of P modulo M *we mean the new positive program $\frac{P}{M}$ obtained from P by:*

- *Removing all rules in P which contain negative literal notp in their bodies with $p \in M$ and*
- *Deleting all negative literals from the remaining rules in P.*

Let us use $LEAST(P)$ to denote the least model of a positive program of P. Przymusinska and Przymusinski showed the following proposition in [Przymusinska91].

Proposition 5.1 *Let P be a normal logic program. A partial model $\mathcal{M} =< T; N >$ is a stationary model of P iff the following equalities hold:*

$$N = LEAST(\frac{P}{T}) \text{ and } T = LEAST(\frac{P}{N}).$$

Proposition 5.2 *Let P be a normal logic program and $\Delta^*_{GK}(P)$ be its translation.*

1. *For a model M of $\Delta^*_{GK}(P)$, M is a K-minimal model iff the following equality holds:*

$$K(M) = LEAST(\frac{P}{A(M)}).$$

2. *For a K-minimal model M, M is a B-minimal model M iff the following equality holds:*

$$B(M) = LEAST(\frac{P}{K(M)}).$$

Sketch of the proof:
1) Let M_P be a K-minimal model of $\Delta^*_{GK}(P)$ and $M_{P(KA)}$ be a K-minimal model of $\Delta^*_{KA}(P)$ such that $A(M_P) = A(M_{P(KA)})$.

$K(M_P) \supseteq K(M_{P(KA)})$ because $P \supseteq P(KA)$. If there is a formula $K\varphi \in K(M_P) - K(M_{P(KA)})$, then it contradicts the K-minimality of M_P because we can make $K(M_P)$ smaller by removing $K\varphi$ from $K(M_P)$. We conclude that $K(M_P) - K(M_{P(KA)})$. It follows that $M_{P(KA)}$ is a K-minimal model iff

$$K(M_{P(KA)}) = LEAST(\frac{P}{A(M_{P(KA)})})$$

from the definition of modulo and theorem 3.5 in [Lin92], in which the construction of a model corresponding to Γ operator in Reiter's default theory, gives K-minimal model.
2) Let M_P be a B-minimal model of $\Delta^*_{GK}(P)$ and $M_{P(KB)}$ be a B-minimal model of $\Delta^*_{KB}(P)$ such that $A(M_P) = A(M_{P(KB)})$ and $K(M_P) = K(M_{P(KB)})$. We conclude that $B(M_P) = B(M_{P(KB)})$ in the same way as in 1. It follows that $M_{P(KB)}$ is a B-minimal model iff

$$B(M_{P(KB)}) = LEAST(\frac{P}{K(M_{P(KB)})})$$

from the definition of modulo. □

In the following, we consider extended logic programs. Although the acceptable models of translated formulas for normal logic programs correspond to the stationary semantics, such a correspondence does not hold for extended programs. Let us consider the following program from [Przymusinski91]:

$$p \leftarrow q.$$

$$p \leftarrow \neg q.$$

The stationary model implies that p is true, because, in stationary semantics, programs without negation-as-failure literals are essentially viewed as classical theories. On the other hand, the acceptable model does not imply that p is true.

6 Translation of Logic Programs with Integrity Constraints into GK

In the previous section, we discussed the relationship between the stationary semantics of logic programs *without* integrity constraints and acceptable models of their translated formulas. In this section, we discuss the case where programs have integrity constraints.

Integrity constraints in logic programming represent conditions which should be satisfied in the model. Based on the translation provided in the previous section, we are able to distinguish the following two situations in acceptable models of translated formulas of programs:

- there is a proposition such that its truth value is *undefined* and
- there is a integrity constraint that is not satisfied.

As far as the author is aware, no previous work treats the above two situations separetely.

The formal definition of integrity constraints is as follows: a set of *integrity constraints* of the form:

$$\leftarrow l_1, ..., l_m, not\, l_{m+1}, ..., not\, l_n,$$

where l_i are literals, which are either atoms p or "classically negated" atoms $\neg p$, and $n \geq m \geq 0$.

The translation of integrity constraints is very straightforward. A set of integrity constraints in program P is translated into $IC_{GK}(P) = IC_{KA}(P) \cup IC_{KB}(P)$, in which $IC_{KA}(P)$ contains the following formula in GK for each integrity constraint:

$$Kl_1 \wedge ... \wedge Kl_m \wedge \neg Al_{m+1} \wedge ... \wedge \neg Al_n \supset false,$$

and $IC_{KB}(P)$ contains the following formula in GK for each integrity constraint:

$$l_1 \wedge ... \wedge l_m \wedge \neg Kl_{m+1} \wedge ... \wedge \neg Kl_n \supset false.$$

The translation of a logic program P with integrity constraints is defined as $\Delta^{*}_{GK}(P) = \Delta^{*}_{GK}(P - IC) \cup IC_{GK}(P)$, where $P - IC$ represents rules in program P.

Let us consider the following programs:

$$P_1 = \{p \leftarrow .\ a \leftarrow not\, a\} \text{ and}$$

$$P_2 = \{p \leftarrow .\ \leftarrow not\, a\}.$$

Translation $\Delta^*_{GK}(P_1)$ has an acceptable model $\{Ap, Kp, p, Ka\}$ and $\Delta^*_{GK}(P_2)$ has no acceptable model. Based on stable model semantics, there is no stable model for both programs. Therefore, we cannot distinguish the above two programs in terms of stable model semantics, although we can do that using acceptable semantics in GK.

7 Comparison with Stationary Extensions

Przymusinska and Przymusinski provided a new concept of extensions for Reiter's default theory. Because our definition of acceptable models and the definition of stationary extensions both seem to be based on the same idea, we should compare our approach with that of stationary extensions.

Definition 7.1 (Stationary default extensions) *[Przymusinska91] Given a default theory Δ, E is a* stationary default extension *iff:*

1. $E \subseteq \Gamma_\Delta(E)$ *and*
2. $E = \Gamma^2_\Delta(E)$.

Przymusinska and Przymusinski [Przymusinska91] showed that, for normal logic programs, stationary default extensions correspon to stationary models. Since our result in Section 5 shows that acceptable models of GK formulas for normal logic programs correspond to stationary models, we conclude that the acceptable models correspond to stationary extensions. You can find the following correspondence between stationary extensions and acceptable models.

Proposition 7.1 *For a normal logic program P, let $\Delta(P)$ be the corresponding default theory and its translation into GK be $\Delta^*_{GK}(P)$. We use E to denote a stationary extension of $\Delta(P)$ and M to denote an acceptable model of $\Delta^*_{GK}(P)$. Then:*

$$E = A(M)(= B(M)) \text{ and}$$

$$\Gamma_{\Delta(P)}(E) = K(M).$$

Our approach in GK seems to provide us with more intuitive answers than do stationary extensions. Let us consider the default theory $\Delta = (\mathcal{D}, \emptyset)$ from [Przymusinska91], where:

$$\mathcal{D} = \{: \neg p/\neg p, : p/p, : q/r\}.$$

The least stationary extension E of Δ is empty, because $\Gamma_\Delta(E)$ is the set of all first order sentences (this means contradiction). However, the acceptable model M of Δ^*_{GK} whose $A(M)$ is minimal is $\{Ar, Kr, r\}$.

8 Concluding Remarks

We have proposed acceptable model semantics for an autoepistemic logic, GK, defined by Lin and Shoham. Undefinedness is described as a certain state in an agent's belief set by using the semantics. Our new semantics provides us with new relationships between default logic and GK, and between logic programming and GK. We should study the features of acceptable models for general cases of default logic in the future.

In the Definition 3.4 of acceptable models, we can replace conditions 2 and 3 with the following condition:

$$\overline{A}(M) \cup \overline{K}(M) - \overline{B}(M),$$

and, thus, deal with abductive logic programming [Kakas90] in our framework. In this case, $Abd(M) = A(M) \cap \overline{K}(M) \neq \emptyset$ (for proposition $p \in Abd(M)$, $\neg Kp, \neg p$, and p are satisfied in the model M). This means that the agent assumes some proposition, but neither derives the belief nor the proposition, that is, the proposition might be called an **abducible**. We are currently investigating this new definition of acceptable models.

References

[Bidoit91] Bidoit, N., Froidevaux, C., General Logical Databases and Programs: Default Logic Semantics and Stratification, Journal of *Information and Control*, 91, pp. 15 – 54 (1991).

[Brogi91] Brogi, A., Mancarella, P., Lamma, P., Mello, P., Normal Logic Programs as Open Logic Programs, *Proc. of JISCLP'92*, pp. 783 – 797 (1992).

[Dung91] Dung, P. M., Negation as Hypotheses: An Abductive Foundation of Logic Programming, *Proc. of ICLP'91*, pp. 3 – 17 (1991).

[Gelfond88] Gelfond, M., Lifschitz, V., The Stable Model Semantics for Logic Programming, *Proc. of ICLP'88*, pp. 1070 – 1080, 1988.

[Kakas90] Kakas, A. C., Mancarella, P., Generalized Stable Models: A Semantics for Abduction, *Proc. of ECAI'90*, pp. 385 – 391 (1990).

[Lin92] Lin, F., Shoham, Y., A logic of knowledge and justified assumptions, *Artificial Intelligence*, 57, pp. 271 – 289 (1992).

[Moore85] Moore, R. C., Semantical considerations on nonmonotonic logic, *Artificial Intelligence*, 25, pp. 75 – 94 (1985).

[Przymusinski89] Przymusinski, C. T., Three-valued Non-Monotonic Formalisms and Semantics of Logic Programs, *Proc. of KR'89*, pp. 341 – 348 (1989).

[Przymusinska91] Przymusinska, H., Przymusinski, C. T., Stationary Default Extensions, Technical Report, California State Polytechnic and University of California at Riverside, (1991).

[Przymusinski91] Przymusinski, C. T., Semantics of Disjunctive Logic Programs and Deductive Databases, *Proc. of DOOD'91*, LNCS566, pp. 85 – 107 (1991).

[Reiter80] Reiter, R., A logic for default theory, *Artificial Intelligence*, 13, pp. 81 – 132 (1980).