ICOT Technical Report: TR-0842

TR-0842

Investigating Assumption-Semantics Through Open Positive Programs

by

N. Iwayama

April, 1993

ICOT

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

(03)3456-3191～5

Institute for New Generation Computer Technology

Investigating Assumption-based Semantics through Open Positive Programs

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

Brogi et al. provide us with the following framework (we call this framework, the model-based framework) to characterize the meaning of negation-as-failure: they regard normal logic programs as open positive logic programs and force conditions on their Herbrand models so that they are recognized as proper models capturing the meaning of negation-as-failure.

We contribute to the study of the semantics of normal logic programming in the model-based framework by modifying Brogi's way of forcing conditions on Herbrand models. We provide two new conditions on Herbrand models in terms of the model-based framework. Each of the two conditions coincides with each of two semantics, stable theory and acceptability semantics by Kakas and Mancarella. One of the conditions which coincides with acceptability semantics is considerably understandable because our definition is non-recursive whereas Kakas and Mancarella defined it recursively.

Keywords: Semantic analysis, Normal logic programs, Open logic programs, Argumentation.

1 Introduction

Bondarenko et al. state that the semantics proposed by Dung [Dung91] and Kakas et al. [Kakas91] is based on the following framework, called the *assumption-based framework*, which

- Regards negation-as-failure as "assumption," and
- Forces argumentation theoretic conditions on the set of assumptions so it is recognized as "proper" negation-as-failure.

Here, we state the details of the argument theoretic conditions from [Bondarenko93]. Argumentation theoretic conditions are formulated in terms of their ability to successfully "counterattack" any "attacking" set of assumptions. A set of assumptions is said to "attack" another if, together with the theory, it implies a consequence which is inconsistent with some assumption contained in the other set. For example, let us consider the program $\{p \leftarrow notq\}$. An assumption $\{notq\}$ attacks the assumption $\{notp\}$, because the program with assumption $\{notq\}$ implies p. Since this framework gives a unified view for various semantic proposals for logic programming, Bondarenko et al. argue that the framework is important.

Added to assumption-based framework, a framework provided by Brogi et al. [Brogi92] should be mentioned. We call this framework the model-based framework in this paper, and summarize the actions of the framework as follows:

- Regards normal logic programs as open "positive" logic programs and
- Forces conditions on the set of their Herbrand models so they are recognized as "proper" models.

To discuss the meaning of normal logic programs, they [Brogi92] provided a specific condition, the admissibility condition. The admissibility condition is based on the "coherency" of Herbrand models. A Herbrand model is "coherent" if the model does not contain both a positive atom (p) and a negation-as-failure atom (not_p) at the same time (the complement notion is "incoherency", i.e., there are both p and not_p in a Herbrand model). As Brogi at al. have pointed out, the point which we should emphasize is that this framework is based on purely model-theoretic arguments (Herbrand models) rather than in terms of the syntax of the program[1].

Although Brogi et al. did not argue that their condition is "argumentation-theoretic," the model-based framework seems to be similar to the assumption-based framework. You can find that there is some intuitive correspondence between the "attacking" of an assumption set and the "incoherency" of a Herbrand model. In fact, for the case of preferential semantics by Dung [Dung91], the admissibility condition in terms of the model-based framework coincides with the appropriate argumentation theoretic condition in terms of the assumption-based framework [Brogi92]. We

[1] For example, stable models [Gelfond88] are defined based on syntactic program transformation.

state that such a coincidence between two frameworks is very important, because we can provide a model-theoretic explanation for the assumption-based framework that is based on the derivability of inconsistent consequences with a set of assumptions.

In this paper, we contribute to the study of the semantics of normal logic programming in the model-based framework. We achieve the following by modifying Brogi's way of forcing conditions on Herbrand models:

1. We provide new two conditions for Herbrand models in terms of the model-based framework, and

2. We show that each of the two conditions coincides with each of the two semantics in terms of the assumption-based framework, i.e. stable theory and acceptability semantics by Kakas and Mancarella [Kakas91].

As a result, we represent stable theory and acceptability semantics in the model-based framework more simply than in the assumption-based framework; especially, our new non-recursive formulation of acceptability semantics is considerably understandable because in [Kakas91] the acceptability semantics is defined recursively. Moreover, our result provides a considerable evidence to show that both frameworks support each other, because we show that formulations from two different perspectives (the assumption-based and model-based frameworks) provide us with a coincidence with respect to many phenomena in the semantics of negation-as-failure.

This paper is organized as follows. We show the preliminary definitions of logic programming in Section 2, in which negation-as-failure is introduced as an abductive assumption. In Sections 3 and 4, we show the conditions of Herbrand models in terms of the model-based framework by modifying Brogi's way of forcing conditions. In Section 5, we summarize the definitions of assumption-based semantics in [Bondarenko93] and show that the condition in Section 3 coincides with stable theory semantics and that the condition in Section 4 coincides with acceptability semantics.

2 Preliminaries

We refer to the terminologies of logic programming. A normal logic program P is a set of clauses of the form

$$A_0 \leftarrow A_1, ..., A_m, not A_{m+1}, ..., not A_n.$$

where A_i are atoms and $n \geq m \geq 0$. Negation-as-failure $notA$ is clearly distinguished classical negation "$\neg A$." We only consider (possibly infinite) propositional programs[2].

Let P be a program and HB be the Herbrand base associated with P. negation-as-failure $notA$ is dealt with as positive atoms not_A, where not_A is a new propositional symbol. In other words, P is transformed into its positive version P' by

[2] This way of restricting programs is well known in the literature (for example, [Przymusinski91]).

replacing each negation-as-failure $notA$ in P's clause bodies with the corresponding positive literal not_A. The positive literal not_A is not only a propositional symbol but an abductive assumption also. We denote the set of abductive assumptions as not_HB $(= \{not\_A | A \in HB\})$.

A (Herbrand) interpretation I of P is defined as usual, i.e. $I \subseteq HB \cup not\_HB$. Given an interpretation I, I^+ stands for $I \cap HB$ and I^- stands for $I \cap not\_HB$. The following definition is a key definition in the model-based framework.

Definition 2.1 *Coherency and incoherency*
An interpretation I is coherent *iff for no atom $A \in HB$ both $A \in I^+$ and $not\_A \in I^-$. I is* incoherent *otherwise.*

We define the following *supportedness*, where we use the immediate consequence operator T_P for a positive program P.

Definition 2.2 *Supported interpretations*
Let P be a program. An interpretation I is a supported interpretation of P iff $H \subseteq I$ such that $I = T_{P \cup H} \uparrow \omega$.

In this paper, we only pay attention to the following class of supported interpretations.

Definition 2.3 *[Brogi92] Negatively supported interpretation*
Let P be a program. A supported interpretation M of P is a negatively supported interpretation *of P iff M is supported by its negative part M^-, or $M = T_{P \cup M^-} \uparrow \omega$. The set of negatively supported interpretations of P is denoted by $NSI(P)$.*

We may express a negatively supported interpretation M simply as $M(\Delta)$ $(\Delta = M^-)$.
As Brogi et al. have explained, if a supported interpretation M is a negatively supported interpretation of a program P, then the set of positive atoms true in M, M^+, contains exactly those which are derivable from P by assuming the negative atoms in M^-.

Example 2.1 *Let P be a program $\{a \leftarrow not\_b\}$. Then*

$$NSI(P) = \{\emptyset, \{not\_a\}, \{a, not\_b\}, \{a, not\_a, not\_b\}\}.$$

In Sections 3 and 4, we will discuss the conditions which we force on negatively supported interpretations to capture the declarative meaning of negation-as-failure.

3 Weak admissibility based on the model-based framework

In this section, we consider the conditions forced on Herbrand interpretations to capture the meaning of negation-as-failure. First, we state definitions in [Brogi92], then we show new definitions.

Brogi et al. use the notion of conservative extensions of negatively supported interpretations to discuss the meaning of negation-as-failure.

Definition 3.1 *[Brogi92] Conservative Extension*
Let M and N be negatively supported interpretations of a program P. N is a conservative extension *of M iff $N^- \supseteq M^-$ and $M^+ \cup N^-$ is coherent. The set of all conservative extensions of M is denoted by $CE(M)$.*

Example 3.1 *Let P be a program $\{a \leftarrow not\_b\}$ as in example 2.1. Then*

$$CE(\emptyset) = \{\emptyset, \{not\_a\}, \{a, not\_b\}, \{a, not\_a, not\_b\}\},$$
$$CE(\{not\_a\}) = \{\{not\_a\}, \{a, not\_a, not\_b\}\},$$
$$CE(\{a, not\_b\}) = \{\{a, not\_b\}\}, \text{ and}$$
$$CE(\{a, not\_a, not\_b\}) = \{\}.$$

To consider the semantics of a program P, we select the interpretation that satisfies the following condition for its conservative extensions.

Definition 3.2 *[Brogi92] Admissible supported interpretation and model*
Let M be a negatively supported interpretation of a program P. M is an admissible supported interpretation *of P iff*

$$\forall N \in CE(M) : M^- \cup N^+ \text{ is coherent.}$$

M is an admissible supported model if M is coherent. $ASI(P)$ ($ASM(P)$) denotes the set of all admissible supported interpretations (models) for P.

Example 3.2 *Let P be the program in example 2.1. Then*

$$ASI(P) = \{\emptyset, \{a, not\_b\}, \{a, not\_a, not\_b\}\}, \text{ and}$$
$$ASM(P) = \{\emptyset, \{a, not\_b\}\}.$$

Though Brogi et al. considered the conservative extensions in order to discuss the meaning of negation-as-failure, in this paper we consider other "non-conservative" extensions. In the definition of conservative extensions, N does not imply the truth of not_A whenever A is true in M, i.e., N is extended "so as to be conservative." Here, we change our point of view in extending M. In extending M, rather than paying attention to keeping coherency with M, we pay attention to expanding M coherently. We formalize this notion of extension as self-coherent extension. Intuitively, N self-coherently expands M unless N newly assumes the truth of both A and not_A.

Definition 3.3 *Self-coherent extension*
Let M and N be negatively supported interpretations of a program P. N is a self-coherent extension *of M iff $N^- \supseteq M^-$ and $(N^- - M^-) \cup N^+$ is coherent. The set of all self-coherent extensions of M is denoted by $SCE(M)$.*

Example 3.3 *Let P be the program in example 2.1. Then*

$$SCE(\emptyset) = \{\emptyset, \{not\_a\}, \{a, not\_b\}\},$$
$$SCE(\{not\_a\}) = \{\{not\_a\}, \{a, not\_a, not\_b\}\},$$
$$SCE(\{a, not\_b\}) = \{\{a, not\_b\}\}, \text{ and}$$
$$SCE(\{a, not\_a, not\_b\}) = \{\{a, not\_a, not\_b\}\}.$$

Note that $\{a, not\_a, not\_b\} \notin SCE(\emptyset)$ but $\{a, not\_a, not\_b\} \in CE(\emptyset)$, and $\{a, not\_a, not\_b\} \in SCE(\{a, not\_a, not\_b\})$ but $\{a, not\_a, not\_b\} \notin CE(\{a, not\_a, not\_b\})$.

As admissible interpretations, we define some kind of admissibility based on the self-coherent extension.

Definition 3.4 *Weakly admissible supported interpretation and model*
Let M be a negatively supported interpretation of a program P. M is a weakly admissible supported interpretation *of P iff*

$$\forall N \in SCE(M) : M^- \cup N^+ \text{ is coherent.}$$

M is a weakly admissible supported model if M is coherent. $WASI(P)$ ($WSAM(P)$) denotes the set of all weakly admissible supported interpretation (models) for P.

Proposition 3.1 *Let P be a program. Then*

$$WASI(P) = WASM(P).$$

Proof: Because $M \in SCE(M)$, we say that M is coherent if M is a weakly admissible interpretation.

Example 3.4 *Let P be the program in example 2.1. Then*

$$WASM(P) = \{\emptyset, \{a, not\_b\}\}.$$

Although, for the program in example 2.1, $ASM(P) = WASM(P)$, the following example shows the difference between admissible and weakly admissible models.

Example 3.5 *Let P be a program $\{a \leftarrow not\_b,\ b \leftarrow not\_b\}$, Then $ASM(P) = \{\emptyset\}$, but $WASM(P) = \{\emptyset, \{not\_a\}\}$ because*

$$SCE(\emptyset) = \{\emptyset, \{not\_a\}\},$$
$$SCE(\{not\_a\}) = \{\{not\_a\}\},$$
$$SCE(\{a, b, not\_b\}) = \{\{a, b, not\_b\}\}, \text{ and}$$
$$SCE(\{a, b, not\_a, not\_b\}) = \{\{a, b, not\_a, not\_b\}\}.$$

The program in the above example is the same as that used by Kakas and Mancarella in their paper [Kakas91] to argue why the weakly stable hypotheses set is important in the *assumption-based* framework. In Section 5, we show that this definition of weakly admissible interpretations based on the model-based framework coincides with the definition of weakly stable theories, that is based on the assumption-based framework. As a result, we will show the relation between weakly and non-weakly admissible supported interpretations.

4 Yet another admissibility based on the model-based framework

In this section, we consider another "non-conservative" extension to discuss the meaning of negation-as-failure, and apply the new notion of non-conservative extension to the definition of admissible interpretation, as in the previous section.

In the previous section, we considered how to expand M when defining the nature of extensions. Here, we force a stronger condition on the extensions of M. Namely, the extension of M is the maximal one among self-coherent extensions of M, otherwise the extension of M should be further expanded to a self-coherent extension of M.

Definition 4.1 *Strongly self-coherent extension*
Let M, N, and NN be negatively supported interpretations of a program P. N is a strongly self-coherent extension *of M iff $\forall NN : NN \supseteq N$, if $(NN^- - N^-) \cup NN^+$ is coherent, then $(N^- - M^-) \cup NN^+$ is coherent. The set of all strongly self-coherent extensions of M is denoted by $SSCE(M)$.*

In this definition, suppose that $(NN^- - N^-) \cup NN^+$ is coherent and that $(N^- - M^-) \cup NN^+$ is coherent, then it follows that $(NN^- - M^-) \cup NN^+$ is coherent. This means that NN is a self-coherent extension of M.

Example 4.1 *Let P be the program in example 2.1. Then*

$SSCE(\emptyset) = \{\emptyset, \{a, not\_b\}\}$,
$SSCE(\{not\_a\}) = \{\{not\_a\}, \{a, not\_a, not\_b\}\}$,
$SSCE(\{a, not\_b\}) = \{\{a, not\_b\}\}$, *and*
$SSCE(\{a, not\_a, not\_b\}) = \{\{a, not\_a, not\_b\}\}$.

Note that $\{not\_a\} \in SCE(\emptyset)$, but $\{not\_a\} \notin SSCE(\emptyset)$.

We point out the relation between strong and non-strong self-coherent extensions.

Proposition 4.1 *Let M be a negatively supported interpretation of a program P.*

$$SSCE(M) \subseteq SCE(M).$$

As weakly admissible interpretations, we define some kind of admissibility based on the strongly self-coherent extension.

Definition 4.2 *Acceptable supported interpretation and model*
Let M be a negatively supported interpretation of a program P. M is an acceptable supported interpretation *of P iff*

$$\forall N \in SSCE(M) : M^- \cup N^+ \textit{ is coherent.}$$

M is an acceptable supported model if M is coherent. $ACCSI(P)$ ($ACCSM(P)$) denotes the set of all acceptable supported interpretations (models) for P.

Proposition 4.2 *Let P be a program. Then*

$$ACCSI(P) = ACCSM(P).$$

Proof: Because $M \in SSCE(M)$, we say that M is coherent if M is an acceptable supported interpretation.

Example 4.2 *Let P be the program in example 2.1. Then*

$$ACCSM(P) = \{\emptyset, \{a, not\_b\}\}.$$

The next corollary follows from proposition 4.1.

Corollary 4.1 *Let P be a program. Then*

$$WASM(P) \subseteq ACCSM(P).$$

Although, for the program in example 2.1, $WASM(P) = ACCSM(P)$, the following example shows the difference between weakly admissible and acceptable models.

Example 4.3 *Let P be a program* $\{a \leftarrow not\_b,\ b \leftarrow not\_b, not\_c\}$, *Then*

$$WASM(P) = \{\emptyset, \{not\_c\}, \{not\_a, not\_c\}\}, \textit{ but}$$

$$ACCSM(P) = \{\emptyset, \{not\_a\}, \{not\_c\}, \{not\_a, not\_c\}\},$$

because of $SCE(P)$ and $SSCE(P)$ in Appendix A.

In [Kakas91], Kakas and Mancarella proposed the notion of stable theories so as to approximate their intuitive meaning of negation-as-failure, and they mentioned how to formalize their accurate intuition itself. Kakas et al. [Kakas92] and Bondarenko et al. [Bondarenko93] discussed the formalization (called *acceptability semantics*) in the assumption-based framework. In the next section, we show that the notion of acceptable supported interpretations coincides with acceptability semantics.

5 The class of assumption-based semantics

In this section, we show that the class of conditions on negatively supported interpretations in the previous sections is equivalent to the class of assumption-based semantics. The class of assumption-based semantics is the class of semantics defined by Dung [Dung91], Kakas and Mancarella [Kakas91], and Kakas, Kowalski and Toni [Kakas92].

Bondarenko et al. [Bondarenko93] summarize and generalize the class of semantics in terms of argumentation theoretic criterion as the "assumption-based" framework. In the assumption-based framework for logic programming, negation-as-failure is regarded as an abductive "assumption," and the argumentation theoretic conditions are forced on the set of assumptions so that the set is recognized as "proper" negation-as-failure. The argumentation theoretic conditions are formulated in terms of their ability to successfully "counterattack" any "attacking" set of assumptions. A set of assumptions is said to "attack" another if, together with the theory, it implies a consequence which is inconsistent with some assumption contained in the other set. The variety of assumption-based semantics is defined by determining the different notions of "counterattacking."

In the sequel, we recall the definitions in [Bondarenko93] in terms of the argumentation theory. Given a set Δ ($\subseteq$ not_HB), a set $\mathcal{A}$ ($\subseteq$ not_HB) *attacks* Δ if $P \cup \mathcal{A} \vdash A$ for some $not\_A \in \Delta$. In [Bondarenko93], the notion of admissibility of an assumption set for an appropriate definition of counterattack is defined as follows:

$$\forall \mathcal{A} \text{ attacks } \Delta, \Delta \text{ counterattacks } \mathcal{A}.$$

Dung [Dung91] defined the *preferential semantics*. In Dung's work, the *admissibility* of a set of assumption Δ is:

$$\forall \mathcal{A}\text{: } \mathcal{A} \text{ attacks } \Delta,\ \Delta \text{ attacks } \mathcal{A}.$$

Namely, Dung used the following notion of counterattacking in the framework of Bondarenko: given a program P and the sets of assumptions Δ and $\mathcal{A}$, Δ $counterattacks_1$ $\mathcal{A}$ iff Δ attacks $\mathcal{A}$.

Then, to negate disadvantages (including those in example 3.5) in the preferential semantics, Kakas and Mancarella [Kakas91] consider one other semantics, *stable theory*. The *weakly stability* of an set of assumptions Δ is :

$$\forall \mathcal{A}\text{: } \mathcal{A} \text{ attacks } \Delta,\ \mathcal{A} \cup \Delta \text{ attacks } \mathcal{A} - \Delta.$$

Namely, they used the following notion of counterattacking in the framework of Bondarenko: given a program P and the sets of assumptions Δ and $\mathcal{A}$, Δ $counterattacks_2$ $\mathcal{A}$ iff $\Delta \cup \mathcal{A}$ attacks $\mathcal{A} - \Delta$.

The following proposition by Kakas and Mancarella shows the relation between the preferential semantics and stable theories.

Proposition 5.1 *[Kakas91] Let Δ be an assumption set of a program P. If Δ is admissible (in Dung's definition), then Δ is weakly stable.*

In the next theorem, we restate the result in [Brogi92], which bridges the gap between assumption-based and model-based frameworks.

Theorem 5.1 *[Brogi92] admissible supported model = admissible assumption set*

1. *a coherent assumption set Δ is admissible $\Rightarrow$ a negatively supported interpretation $M(\Delta)$ is an admissible supported model.*

2. *M is an admissible supported model $\Rightarrow$ M^- is admissible.*

In addition to the above theorem, we state the next theorem, which gives the equivalence between weakly admissible supported models proposed in Section 3 and weakly stable assumption sets by Kakas and Mancarella.

Theorem 5.2 *weakly admissible supported model = weakly stable assumption set*

1. *a coherent assumption set Δ is weakly stable $\Rightarrow$ a negatively supported interpretation $M(\Delta)$ is a weakly admissible supported model.*

2. *M is a weakly admissible supported model $\Rightarrow$ M^- is weakly stable.*

Proof: See Appendix B.

Here, we show the relation between admissible and weakly admissible supported models.

Corollary 5.1 *Let P be a program. Then*

$$ASM(P) \subseteq WASM(P).$$

Proof: From theorem 5.1, theorem 5.2, and proposition 5.1.

In [Kakas91], to capture the basic intuition of negation-as-failure more closely than stable theories, they provide an alternative condition of acceptability in their *acceptability semantics* [Kakas92]. The *acceptability* of assumption set Δ is defined as follows: for some initial assumption set Δ_0,

> Δ is acceptable to Δ_0 iff $\forall \mathcal{A}$: $\mathcal{A}$ attacks $\Delta - \Delta_0$, $\mathcal{A}$ is not acceptable to $\Delta \cup \Delta_0$.

In the sequel, we only consider whether Δ is acceptable to $\emptyset$. Bondarenko et al. [Bondarenko93] defined the equivalent notion of the acceptability in a complicated way in terms of attacking and counterattacking (we do not show the definition here). We can unfold the recursion using the above definition of acceptability once.

> Δ is acceptable to $\emptyset$
> iff $\forall \mathcal{A}$: $\mathcal{A}$ attacks Δ, $\exists \mathcal{D}$: $\mathcal{D}$ attacks $\mathcal{A} - \Delta$ s.t. $\mathcal{D}$ is acceptable to $\mathcal{A} \cup \Delta$.

We will show that this definition is equivalent to the acceptable supported models proposed in the previous section after we discuss the non-recursive version of the above definition; the non-recursive version is used in the proof of the equivalence.

Definition 5.1 *A set of assumptions Δ is* acceptable' *to Δ_0 iff*

$$\forall \mathcal{A}: \mathcal{A} \supseteq \Delta \cup \Delta_0,\ \mathcal{A} \text{ attacks } \Delta - \Delta_0,\ \exists \mathcal{D}: \mathcal{D} \supseteq \mathcal{A} \cup \Delta \cup \Delta_0,\ \mathcal{D} \text{ attacks } \mathcal{A} - (\Delta \cup \Delta_0) \text{ and } \mathcal{D} \text{ does not attack } \mathcal{D} - (\mathcal{A} \cup \Delta \cup \Delta_0).$$

Theorem 5.3 *Δ is acceptable to Δ_0 iff Δ is acceptable' to Δ_0.*

Proof: See Appendix B.

From this theorem, we state the next theorem which gives the equivalence between acceptable supported models and acceptable assumption sets.

Theorem 5.4 *acceptable supported model = acceptable assumption set*

1. *Δ is acceptable to $\emptyset \Rightarrow$ a negatively supported interpretation $M(\Delta)$ is an acceptable supported model.*

2. *M is an acceptable supported model $\Rightarrow M^-$ is acceptable to $\emptyset$.*

Proof: It follows that definition 5.1 is equivalent to definition 4.2 by substituting $\emptyset$ for Δ_0 in definition 5.1.

6 Concluding Remarks

We provide two conditions on Herbrand models in terms of the model-based framework proposed by Brogi et al. We show that each of the conditions coincides with each of two assumption-based semantics by Kakas and Mancarella, as Brogi et al. showed that their definition of admissibility coincides with preferential semantics by Dung. The table shows these coincidences in each condition:

Model-based	Assumption-based
Acceptable	Acceptable
Weakly Admissible	Weakly Stable
Admissible	Admissible

Each row shows all correspondences between the two frameworks. A condition in the upper row in the table generalizes the condition in the lower row.

Now, we have two directions in which to proceed with future work. The first direction is concerned with the reformulation of our formalism in autoepistemic logic. In this paper, we discuss ordinary Herbrand models by regarding *not_A* as a positive hypothesis. If we may treat *not_A* as $\neg LA$ in autoepistemic logic (L is a modal operator, and LA means that "A is believed"), we may provide the conditions in autoepistemic logic terminologies. We have already shown one way of that in [Iwayama93] although it is necessary to investigate what we formalize in [Iwayama93] more deeply.

The second direction is related to extensions of languages. We would like to specify the semantics of extended logic programs and abductive logic programs which have positive and negative hypotheses (with or without integrity constraints) in the same way as in this work. Bondarenko et al. [Bondarenko93] do that in the assumption-based framework.

References

[Bondarenko93] Bondarenko, A., Toni, F., Kowalski, R. A., An Assumption-based Framework for Non-monotonic Reasoning, to appear in *2nd International Logic Programming and Non-Monotonic Reasoning Workshop*, (1993).

[Brogi92] 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.

[Iwayama93] Iwayama, N., Rules representing two types of epistemic statements, *Proc. of 4th International Workshop on Extensions of Logic Programming*, (1993).

[Kakas91] Kakas, A. C., Mancarella, P., Stable Theories for Logic Programs, *Proc. of ILPS'91*, pp. 85 – 100 (1991).

[Kakas92] Kakas, A. C., Kowalski, R. A., Toni, F., Abductive Logic Programming, to appear in *Journal of Logic and Computation*, (1993).

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

Appendix A

Self-coherent extensions and strongly self-coherent extensions of a program $\{a \leftarrow not\_b,\ b \leftarrow not\_b, not\_c\}$ are shown in the following:

$$\begin{aligned} SCE(\emptyset) &= \{\emptyset, \{not\_a\}, \{not\_c\}, \{not\_a, not\_c\}\}. \\ SCE(\{not\_a\}) &= \{\{not\_a\}, \{a, not\_a, not\_b\}, \{not\_a, not\_c\}\}. \end{aligned}$$

$$
\begin{aligned}
SCE(\{a, not\_b\}) &= \{\{a, not\_b\}, \{a, b, not\_b, not\_c\}\}. \\
SCE(\{not\_c\}) &= \{\{not\_c\}, \{not\_a, not\_c\}\}. \\
SCE(\{a, not\_a, not\_b\}) &= \{\{a, not\_a, not\_b\}, \{\{a, b, not\_a, not\_b, not\_c\}\}. \\
SCE(\{a, b, not\_a, not\_b\}) &= \{\{a, b, not\_a, not\_b\}\}. \\
SCE(\{not\_a, not\_c\}) &= \{\{not\_a, not\_c\}\}. \\
SCE(\{a, b, not\_a, not\_b, not\_c\}) &= \{\{\{a, b, not\_a, not\_b, not\_c\}\}.
\end{aligned}
$$

$$
\begin{aligned}
SSCE(\emptyset) &= SCE(\emptyset) - \{\{not\_a\}\}. \\
SSCE(\{not\_a\}) &= SCE(\{not\_a\}) - \{\{a, not\_a, not\_b\}\}. \\
SSCE(\{a, not\_b\}) &= SCE(\{a, not\_b\}). \\
SSCE(\{not\_c\}) &= SCE(\{not\_c\}). \\
SSCE(\{a, not\_a, not\_b\}) &= SCE(\{a, not\_a, not\_b\}). \\
SSCE(\{a, b, not\_a, not\_b\}) &= SCE(\{a, b, not\_a, not\_b\}). \\
SSCE(\{not\_a, not\_c\}) &= SCE(\{not\_a, not\_c\}). \\
SSCE(\{a, b, not\_a, not\_b, not\_c\}) &= SCE(\{a, b, not\_a, not\_b, not\_c\}).
\end{aligned}
$$

Appendix B

Proof of theorem 5.2
1) Assume $M(\Delta)$ is not a weakly admissible supported interpretation. Then $\exists N : N^- \supseteq M$ s.t. $M^- \cup N^+$ is incoherent and $N^- - M^- \cup N^+$ is coherent, that is $\forall not\_A \in N^- - M^-$ s.t. $P \cup N^- \not\vdash A$. This contradicts the weakly stability of Δ because we can assume that $N^- = \mathcal{A} \cup \Delta$.
2) It is enough to consider the case where $N^- = \mathcal{A}$ s.t. $\mathcal{A} \supseteq \Delta$. □
Proof of theorem 5.3
$\Rightarrow$) Assume that Δ is not acceptable' to Δ_0, i.e., $\exists \mathcal{A}$: $\mathcal{A} \supseteq \Delta \cup \Delta_0$, $\mathcal{A}$ attacks $\Delta \cup \Delta_0$, $\forall \mathcal{D}$: $\mathcal{D} \supseteq \mathcal{A} \cup \Delta\Delta_0$, $\mathcal{D}$ does not attack $\mathcal{A} - (\Delta \cup \Delta_0)$, or $\mathcal{D}$ attacks $\mathcal{D} - (\mathcal{A} \cup \Delta \cup \Delta_0)$. We obtain $\exists \mathcal{D}'$: $\mathcal{D}' \supseteq \mathcal{A} \cup \Delta \cup \Delta_0$, $\mathcal{D}'$ attacks $\mathcal{A} - \Delta$, and $\mathcal{D}'$ is acceptable to $\mathcal{A} \cup \Delta \cup \Delta_0$ because Δ is acceptable to $\emptyset$. (If $\mathcal{D}' \not\supseteq \mathcal{A} \cup \Delta \cup \Delta_0$, we can make a new $\mathcal{D}'$ where $\mathcal{D}' \cup \mathcal{A} \cup \Delta \cup \Delta_0$.) It follows that $\mathcal{D}'$ attacks $\mathcal{D}' - \mathcal{A} \cup \Delta \cup \Delta_0$. Since $\mathcal{D}'$ is acceptable to $\mathcal{D}' \cup \mathcal{A} \cup \Delta \cup \Delta_0 = \mathcal{D}'$, $\mathcal{D}'$ is *not* acceptable to $\mathcal{A} \cup \Delta \cup \Delta_0$. This is a contradiction.
$\Leftarrow$) Assume that Δ is not acceptable to Δ_0. Instead of original recursive definition of *not-acceptable*, we show the constructive definition of notion of *not-acceptable*. Then, by using constructive definition, we will show the contradiction.

First, we provide us with the constructive definition of the notion of not-acceptable.

Definition B.1 *Let S be a set of pairs of sets of assumptions.*

$$NotAcc(S) = \{\langle \Delta, \Delta_0 \rangle \mid \exists \mathcal{A} \text{ attacks } \Delta - \Delta_0, \forall \mathcal{D} \text{ attacks } \mathcal{A} - (\Delta \cup \Delta_0), \langle \mathcal{D}, \mathcal{A} \cup \Delta \cup \Delta_0 \rangle \in S\},$$

$$NotAcc^0(\emptyset) = NotAcc(\emptyset),$$
$$NotAcc^i(\emptyset) = NotAcc(NotAcc^{i-1}(\emptyset)).$$

The intuitive meaning of S in $NotAcc(S)$ is that S includes assumption set pairs which have already been computed as not-acceptable.

Next lemma shows the constructive definition of notion of not-acceptable.

Lemma B.1 *Δ is not acceptable to Δ_0 iff $\langle\Delta, \Delta_0\rangle \in \bigcup_{i=0}^{\omega} NotAcc^i(\emptyset)$.*

In the following, we will prove the case where $\langle\Delta, \Delta_0\rangle \in NotAcc^1(\emptyset)$ contradicts the fact that Δ is acceptable' to Δ_0. Although we do not show the proof for the cases where $\langle\Delta, \Delta_0\rangle \in NotAcc^i(\emptyset)$ $(i > 1)$, we can construct the proof for the cases same as the case for $i = 1$. (We can prove obviously the case where $\langle\Delta, \Delta_0\rangle \in NotAcc^0(\emptyset)$.)

Since $\langle\Delta, \Delta_0\rangle \in NotAcc^1(\emptyset)$, $\exists\mathcal{A}_1$ such that $\mathcal{A}_1$ attacks $\Delta - \Delta_0$, $\forall\mathcal{D}_i$: $\mathcal{D}_i$ attacks $\mathcal{A}_1 - (\Delta \cup \Delta_0)$, $\exists\mathcal{A}_i$: $\mathcal{A}_i$ attacks $\mathcal{D}_i - (\mathcal{A}_1 \cup \Delta \cup \Delta_0)$ and there is no assumption set $\mathcal{D}$ such that $\mathcal{D}$ attacks $\mathcal{A}_i - (\mathcal{D}_i \cup \mathcal{A}_1 \cup \Delta \cup \Delta_0)$.

We can consider $\mathcal{A}_1 \supseteq \Delta \cup \Delta_0$, because it is enough to consider same $\mathcal{D}_i$ as before. Let us consider the following assumption set:

$$H = \bigcup_i (\mathcal{A}_i - (\mathcal{D}_i \cup \mathcal{A}_1 \cup \Delta \cup \Delta_0)).$$

So, we can consider $\mathcal{A}_1 \supseteq \Delta \cup \Delta_0 \cup H$, because it is enough to consider same $\mathcal{D}_i$ as before (since there is no assumption set $\mathcal{D}$ such that $\mathcal{D}$ attacks $\mathcal{A}_i - (\mathcal{D}_i \cup \mathcal{A}_1 \cup \Delta \cup \Delta_0)$).

Then, let us consider all the assumption sets $\mathcal{D}_j$ such that $\mathcal{D}_j \supseteq \mathcal{A}_1$ and $\mathcal{D}_j$ attacks $\mathcal{A}_1 - (\Delta \cup \Delta_0)$ (there may exist $\mathcal{D}$ such that $\mathcal{D} \not\supseteq \mathcal{A}_1$ and $\mathcal{D}$ attacks $\mathcal{A}_1 - (\Delta \cup \Delta_0)$). For such a $\mathcal{D}_j$, $\mathcal{D}_j$ attacks $\mathcal{D}_j - (\mathcal{A}_1 \cup \Delta \cup \Delta_0)$. The reason is because there exists $\mathcal{A}_j$ which attacks $\mathcal{D}_j - (\mathcal{A}_1 \cup \Delta \cup \Delta_0)$ and because $\mathcal{D}_j \supseteq \mathcal{A}_j$ since $\mathcal{D}_j \supseteq H$. However, this contradicts the fact that Δ is acceptable' to Δ_0. □