NEGATIONS AS HYPOTHESES: AN ABDUCTIVE FOUNDATION FOR LOGIC PROGRAMMING.

Phan Minh Dung
Division of Computer Science,
Asian Institute of Technology
GPO Box 2754, Bangkok 10501, THAILAND.
E-mail: dung@ait.th

Abstract

A logic program is considered as an abductive framework with negative literals as hypotheses. We define a simple declarative semantics for abduction and show that the new semantics captures, generalizes and unifies different semantic concepts (e.g. well-founded models, stable models) in logic programming. We study the operational semantics of abduction, and prove the soundness of the Eshghi and Kowalski's abductive procedure with respect to our new declarative semantics.

1. Introduction

For a successful application of logic programming as a paradigm for knowledge representation, it is necessary to clarify the semantic problems of negation in logic programming and its relations to nonmonotonic logics. This paper presents a contribution to the study of this problem. Our goal is to reveal the inherent relations between abduction and logic programming.

Two major approaches to semantics of negation in logic programming are the stable semantics and the well-founded semantics.

The stable semantics of a program is defined by the set of its stable models. This semantics has its root in nonmonotonic logics where a logic program is considered as an autoepistemic theory whose stable extensions correspond to the stable models of the logic program [GL]. The problem of stable semantics is that it is not defined for every logic program, e.g. the program consisting of the only clause $p \leftarrow \neg p$, has no stable models. To illustrate the seriousness of this problem, let us consider one more example.

Example 1 (The Barber's paradox) [GCH]

"Beardland is a small city where the barber Noel shaves every citizen who does not shave himself.
Does Noel shaves the city mayor Casanova ?
Does Noel shave himself ? "

The problem can be represented by a logic program consisting of the clauses shave(Noel,t)<— ┐shave(t,t)
mayor(Casanova)<-

Despite the confusion about who shaves Noel, we expect that Noel shaves the city mayor Casanova. But this program has no stable model, i.e. we could not conclude any things with respect to the stable semantics. ##

The idea of well-founded semantics is negation as (possibly infinite) failure, i.e. the failure (possibly in infinitary) to prove a fact (a ground atom) to be true leads to the acceptance of this fact being false. Formally, the well-founded semantics is defined by the well-founded model which is defined as the least fixed point of a monotonic operator [GRS]. In contrast to the stable semantics, the well-founded semantics is defined for every logic program. Its major shortcomming is its inability to handle conclusions which can be reached only by "proof by cases". The following example illustrates this problem.

Example 2 Let P be
$a \leftarrow \neg b$
$b \leftarrow \neg a$
$c \leftarrow a$
$c \leftarrow b$

It is reasonable to expect that c holds. But wrt the well-founded semantics, all a,b,c are unknown. Note that in this case the stable semantics provides the expected conclusions. ##

The diversity of different approaches in semantics of negation suggests that there is probably not an unique intended semantics for logic programs. Which semantics should be used depends on concrete applications. To be able to choose the "right" semantics among different ones, it is of great importance to understand the inherent relations between them.

Relations between logic programming and nonmonotonic logics are subjects of intensive study in the literature. While the relations between stable semantics and various different nonmonotonic logics have been largely clarified [MT,GL,EK,KM, YY,D1,W], the same can not be said for well-founded semantics.[1]

One of the well-known and simple approach in nonmonotonic reasoning is abduction. In the simplest case, it has the form:

From A and $A \leftarrow B$
infer B as a possible "explanation" of A.

Though abduction has been the focus of intensive research [K,EK,KM,PO,CP], many questions concerning both declarative and operational semantics of abduction remain unclear. Eshghi and Kowalski [EK] have given an abductive procedure as the

---

1. Since well-founded semantics is defined in the framework of a three-valued logic [GRS],it is implicitely accepted in the literature that the logical nature of the semantics is three-valued. So the few works [PT2,PAA] devoted to clarify the relations between well-founded semantics and nonmonotonic logics are all based on three-valued logics. In this paper, we will show that this point of view is not undisputed, by giving a simple two-valued abductive characterization of well-founded semantics. This constitutes an interesting result of this paper.

operational semantics of abduction and have also pointed out that stable semantics does not provide the expected semantics for abduction. But it is left open the question of the soundness of the abductive procedure as well as the question of what is the expected semantics of abduction.

In this paper, we attack all these problems, namely

* We define a simple and natural declarative semantics, called preferential semantics, for abduction.

* We clarify the relationship between abduction and logic programming by showing that the preferential semantics captures, generalizes and unifies different semantic concepts (e.g. well-founded models, stable models) in logic programming.

* We study the operational semantics of abduction. We show that the Eshghi and Kowalski's abductive procedure is sound wrt the preferential semantics.[2]

2. Declarative Semantics

2.1. PREFERENTIAL SEMANTICS

Abduction can be considered as a special form of hypothetical reasoning and the idea of hypothetical reasoning following in this paper is that to predict the expected observations from a incomplete knowledge base represented as a set of facts known to be true the user supplies a set of hypotheses they are prepared to accept as a part of an explanation to the expected observations which is consistent with the facts. This explanation is considered as a logical theory based on a restricted set of possible hypotheses. An explanation can also be viewed as a scenario in which some goal is true. **The user provides which hypotheses are acceptable in such scenarios** [PO].

In general, the theory of abductive reasoning is based on the notion of **abduction frameworks** [PO,EK] defined as triples (KB,H,IC) where KB is a first order theory representing the user supplied rules and facts, H is a set of first order formulae representing the possible hypotheses, and IC is a set of integrity constraints used to determine the admissible explanations.

Given an abduction framework (KB,H,IC), a set of hypotheses E is an abductive solution for a query Q iff

$KB \cup E \vdash Q$ and

$KB \cup E$ satisfies IC

Thus, any theory of abductive reasoning has to provide answers to the following two questions:

---

[2]. In the full version of this paper, we point out an error in the original version of the abductive algorithm of Eshghi and Kowalski. The algorithm given in this paper is a modified version of the original one.

What does "KB U E satisfies IC" mean ?
(declarative semantics)

How to compute the abductive solutions E's for a query Q ?
(operational semantics)

Since our goal in this paper is to study the relations between abduction and logic programming, we restrict ourself on a special class of abduction frameworks corresponding to logic programs.

We assume the existence of a fixed finite alphabet L, big enough to contain all constants, function symbols, and predicate symbols occurring in any program considered in this paper. The **Herbrand base** (resp. **Herbrand universe**) of L is denoted by **HB** (resp. **HU**). A **logic program** is a set of clauses of the form $A \leftarrow L_1 \& .. \& L_n$ where A is an atom and L_i's are literals. To define the class of abduction frameworks corresponding to logic programs, we introduce for each predicate symbol p contained in L, a new predicate symbol p^- of the same arity. The new predicates are called **abducible predicates**. Atoms of the abducible predicates are called **abducible atoms**. Ground abducible atoms are called **hypotheses**. The set of all hypotheses is denoted by **HY**. Atoms in HB are called **ordinary atoms**. For every ordinary atom $A=p(t_1,..,t_n)$, A^- denotes the corresponding abducible atom $p^-(t_1,..,t_n)$. From now on, if we speak of an atom then we always mean an ordinary one.

An **abductive program** over the language L is an abduction framework (KB,H,IC) such that

KB is a definite Horn theory over L U $\{p^- \mid p \in L\}$ with no abducible predicates appearring in the heads of its clauses.
IC = $\{ \leftarrow p(x) \& p^-(x) \mid p$ is a predicate symbol in L $\}$[3]
H = HY

Since the set of hypotheses and integrity constraints are fixed for all abductive programs over the fixed language L, we often write shortly KB for an abductive program <KB,IC,HY>.

A logic program P is transformed into an abductive program **P*** by replacing every negative literal $\neg p(t_1,..,t_n)$ in each clause body by $p^-(t_1,..,t_n)$. For example, let P be $p \leftarrow \neg q$. Then P* is $p \leftarrow q^-$. Dually, the logic program corresponding to an abductive program KB is denoted by **KB^**.

The semantics of abductive programs is based on the notions of scenario and extension [PO] recalled in the following definition.

Definition A **scenario** of an abductive program KB is a first order theory KB U H where H is a subset of HY such that KB U H U IC is consistent.

An **extension** of a abductive program KB is a maximal (with respect to set inclusion) scenario of KB. ##

---

[3]. All variables occurring in any clause of KB U IC are universally quantified at the front of this clause.

For any scenario S, **hyp(S)** denotes the set of hypotheses in S, i.e. if S=KB U H then hyp(S)=H. Further, let Der(H,KB) = {A $\in$ HB | KB U H $\vdash$ A }.

As indicated elsewhere [PO], not every extension specifies an expected semantics of an abductive program. For example, let KB = { p<—q^- }. KB has two extensions C_1 = KB U {q^-}, C_2 = KB U {p^-}. But it is clear that only C_1 captures the expected semantics of KB.

The problem we are facing here is to determine those extensions, called preferred extensions, which specify the intended semantics of an abductive program.

Note that in practice, to explain some expected observation, **the user provides the hypotheses** and **a hypothesis is acceptable only if there is no evidence to the contrary.** Let us take a closer look at this plausible rule.

It is clear that the contrary of a hypothesis A^- is the atom A. Hence, an evidence to the contrary of A^- can be considered as an evidence of A which itself can be seen as a set of hypotheses E such that KB U E $\vdash$ A. The existence of no evidence to the contrary for a hypothesis A^- wrt a scenario S could mean that each evidence E of A is inconsistent with S U IC. But unluckily, this can not go well as the following example shows.

Example 3 Let KB: p<— p^-. Let S be the scenario KB U {}. The only evidence of p is {p^-}. It is obvious that S U {p^-} U IC is inconsistent. Thus the hypothesis p^- has no evidence to the contrary. So p^- should be acceptable. But it is clear that p^- could not be accepted since S U {p^-} U IC is inconsistent. ##

How could we interprete the condition "No Evidence to the Contrary" in the plausible rule ?

Note that the hypotheses are provided by the user. An user observes and characterizes a system by its input-output-behavior. Thus in a scenario S = KB U H, a user "sees" only the "inputs" (the set H of hypotheses) and the corresponding "outputs" (the set T of atoms which are logically implied by S,i.e. T={ A$\in$HB | S$\vdash$ A }). To decide whether a hypothesis is acceptable or not, the user has to rely on what he gets from the systems, i.e. on the sets T,H. Thus the following definitions of the acceptability of hypotheses. Let denote **inout(S)** = H U T.

Definition Let KB be an abductive program. A set of hypotheses E is an **evidence** of an atom A $\in$ HB wrt KB if KB U E $\vdash$ A.[4]

A hypothesis A^- is **acceptable** wrt a scenario S if for every evidence E of A, E U inout(S) U IC is inconsistent. ##

It is clear that we are only interested in scenarios whose hypotheses are acceptable. Hence the following definition.

---

[4]. Note that KB U E U IC can be inconsistent.

Definition A scenario S is **admissible** if every hypothesis accepted in S is also acceptable wrt S, i.e. if $A^- \in S$ then A^- is acceptable wrt S.

A **preferred extension** of an abductive program KB is a maximal (wrt set inclusion) admissible scenario of KB
##

Let call the semantics defined by the preferred extensions **preferential semantics.**

The following lemma shows the correctness of this definition.

Lemma (Fundamental Lemma)

Let S be an admissible scenario and let A^-, B^- be acceptable with respect to S. Then

1) $S' = S \cup \{A^-\}$ is admissible
2) B^- is acceptable with respect to S'. ##

Example 3 (Continued)
Let KB be $p \leftarrow p^-$. Let S = KB U {}. Then inout(S) = {}. Since $\{p^-\}$ is an evidence of p and $\{p^-\}$ U inout(S) U IC is consistent, p^- is not acceptable with respect to S. Thus the only admissible scenario of KB is KB U {} which is also its unique preferred extension. ##

The following examples demonstrate that preferential semantics captures and generalizes stable semantics.

Example 4 Let KB = { $p \leftarrow q^-$ }. C_1 = KB U $\{q^-\}$, C_2 = KB U $\{p^-\}$ are two extensions of KB. Since q has no evidence, q^- is acceptable wrt C_1. Since $\{q^-\}$ is an evidence of p and $\{q^-\}$ U inout(C_2) U IC = $\{p^-, q^-\}$ U IC is consistent, p^- is not acceptable wrt C_2. Thus only C_1 is admissible. Hence C_1 is the unique preferred extension of KB corresponding to the unique stable model of KB^: $p \leftarrow \neg q$). ##

Example 5 (Continuation of example 2)
Let consider again the abductive program

shave(Noel,t)<— shave⁻(t,t)
mayor(Casanova)<-

It is not difficult to see that C = KB U {shave⁻(c,c) | c≠Noel} is the only preferred extension. Thus $C \vdash$ shave(Noel,c) for each c≠Noel. That means that our new semantics implies that Noel shaves every person who is not Noel himself. Thus Noel shaves the mayor Casanova. ##

Let denote the set of all admissible scenarios of KB by $\mathbf{AS}_{KB}$. The existence of at least one preferred extension for every program KB is guaranted by the following theorem.

Theorem 1

1) $(AS_{KB}, \subseteq)$ is a complete partial order, i.e. every directed subset of AS_{KB} has a least upper bound.
2) For every admissible scenario S, there is at least one preferred extension K such that $S \subseteq K$. ##

The stable semantics of an abductive program is defined by stable extensions which are defined as folows: A **stable extension** is a scenario S such that for every ordinary atom $A \in HB$, either $S \vdash A$ or $A^- \in S$ holds.

Lemma [EK] Let P be a logic program and let M be a Herbrand model of P. Then M is a stable model of P iff there is a stable extension S of P* such that $M = \{A \mid S \vdash A\}$. ##

The following theorem gives a formal account of the relations between stable semantics and preferential semantics.

Theorem 2

1) Every stable extension is a preferred extension but not vice versa.
2) If P is a locally stratified logic program then the unique preferred extension S of P* is stable and $M = \{A \mid S \vdash A\}$ is the unique stable model (also called perfect model) of P. ##

2.2. WELL-FOUNDED SEMANTICS

We will show in this chapter how the idea of well-founded semantics of logic programming can be captured in our framework.

Let denote the least admissible scenario KB U {} of an abductive program KB by S_{KB}.

An abductive program may have different semantics which can be defined as certain sets of admissible scenarios. The interesting question is whether or not there is a general characterization of these semantics. Since every semantics of an abductive program represents in some sense a possible world of the program, we expect that this world is complete in the sense that every acceptable hypothesis must be accepted. Thus, the class of scenarios in which all acceptable hypotheses are accepted is of special interest to us.

Definition An admissible scenario S is **complete** if every acceptable hypothesis wrt S is accepted in S, i.e. for any hypothesis A^-, if A^- is acceptable wrt S then $A^- \in S$.
##

From the definition of preferred extensions as maximal admissible scenarios, it follows immediately:

Lemma Preferred extensions are complete scenarios, but not vice versa. ##

An example for the existence of a complete scenario which is not a preferred extension is the least scenario $ of the following program KB = { $p \leftarrow q^-$, $q \leftarrow p^-$ }.

Let CS_{KB} be the class of all complete scenarios of KB.

Each admissible scenario S has a **complete closure** which is the least (wrt set inclusion) complete scenario containing S. We give now a construction for computing the complete closure of a scenario S.

Let $S \in AS_{KB}$. Define $\mathbf{V_{KB}: AS_{KB} \rightarrow AS_{KB}}$ by

$$V_{KB}(S) = S \cup AC_{KB}(S)$$

where $AC_{KB}(S)$ is the set of all acceptable hypotheses wrt S (the correctness of the definition follows directly from the fundamental lemma).

The complete closure of an admissible scenario S can be constructed as the limit of the following sequence $(S_i)_i$ of scenarios.

$S_0 = S$

$S_i = \bigcup_{j<i} S_j$ for limit ordinal i

$S_{i+1} = V_{KB}(S_i)$

It is not difficult to see that $(S_i)_i$ is an increasing sequence (wrt set inclusion). Hence, it has a limit **S*** at some countable ordinal.

Lemma S* is the complete closure of S, i.e. S* is a complete scenario and for every complete scenario R, if $S \subseteq R$ then $S^* \subseteq R$. ##

Definition The complete closure of the least admissible scenario $ is called the **well-founded scenario** denoted by $\mathbf{WFS_{KB}}$ (i.e. WFS_{KB} = $*). ##

Theorem 3

1) $(CS_{KB}, \subseteq)$ is a complete semilattice[5].
2) WFS_{KB} is the least element of $(CS_{KB}, \subseteq)$. ##

If a program may have different semantics (represented by some complete scenarios) representing the different views peoples may draw from the program, then it is meaningful to ask whether all of these different views may have some things in common. From theorem 3, it follows immediately that the well-founded semantics defined by well-founded scenario, represents the common ground for different semantics of a program. In other words, well-founded semantics is some kind of a skeptical semantics.

That the well-founded semantics of a abductive program P* corresponds to the well-founded semantics of its corresponding logic program P is demonstrated by the following theorem.

---

[5]. A partial order $(R, \leq)$ is a complete semilattice if every nonempty subset of R has an inf and every nonempty directed subset of R has a sup.

Theorem 4

Let P be a logic program and WFM be the well-founded model[6] of P. Then

$$WFM = \{ A \mid A \in HB \text{ and } WFS_{P^*} \vdash A \} \cup \{ \neg A \mid A \in HB \text{ and } A^- \in WFS_{P^*} \}$$
##

Summary

We can say that the set of complete scenarios, CS_{KB}, represents the universe of all possible semantics (the set of all possible worlds) of an abductive program in which well-founded semantics corresponds to the "minimalism" semantics where only things which hold in all possible worlds are "believed" while the preferential semantics corresponds to the "maximalism" semantics where each preferred extension represents a "belief world" of an agent who tries to conclude as many knowledges as possible from an abductive program considered as an incomplete knowledge base.[7]

3. Operational Semantics

As we have seen in the previous part, there are two different semantic intuitions, the minimalism corresponding to well-founded semantics, and the maximalism corresponding to preferential semantics, for abductive programs.

Questions related to the proof theoretic aspects of well-founded semantics have been studied in the literature [R,Ck,L]. It has been shown that the SLS-resolution is sound and complete wrt well-founded semantics [R]. SLDNF-resolution based on negation as (finite) failure can be considered as a computable approximation of the SLS-resolution.

In contrast, not much work has been done to study the proof theoretic aspects of abduction with respect to the preferential semantics. In [EK], an abduction procedure is given as a proof theory for abduction. But it is left open the question about the declarative semantics of abduction. In this chapter, we show the soundness of this abductive procedure wrt preferential semantics.

---

[6]. See [GRS] for a formal definition of well-founded model

[7]. It seems that minimalism and maximalism are two main semantic intuitions for knowledge representation schemes. In the theory of nonmonotonic inheritance, these two intuitions are known as skepticism and credulism respectively. A skeptical reasoner refuses to daw conclusions in ambiguous situations where a credulous (belief-hungry) reasoner tries to conclude as much as possible [THT]. The correspondence between the minimalism (resp. maximalism) in logic programming and skepticism (resp. credulism) in nonmonotonic inheritance theory is studied throughoutly in [D2].

The Eshghi and Kowalski's Abductive Procedure

The abductive procedure can be viewed as an extension of the SLDNF-resolution consisting of two interleaving activities: (a) reasoning backward for a refutation and collecting the required hypotheses, as shown inside an ordinary box in example 6, and (b) checking that the collected hypotheses are consistent, as shown in the bold boxes in example 6.

Let KB be a abductive program. Let R be a safe computation rule (one that selects an abducible atom only if it is ground).

An **abductive derivation** from (G_1,H_1) to (G_n,H_n) is a sequence

$$(G_1,H_1),(G_2,H_2),\ldots,(G_n,H_n)$$

such that, for each i, $0<i<n$, G_i has the form $\leftarrow l,l'$ where (without loss of generality) R selects l, and l' is a (possibly empty) collection of atoms, H_i is a set of hypotheses, and

abd1) If l is not abducible
then $G_{i+1} = C$ and $H_{i+1} = H_i$

where C is the resolvent of some clause in KB with the clause G_i on the selected literal l.

abd2) If l abducible and $l \in H_i$
then $G_{i+1} = \leftarrow l'$ and $H_{i+1} = H_i$

abd3) If l is abducible ($l=k^-$) and $l \notin H_i$ and there is a consistency derivation from $(\{\leftarrow k\}, H_i \cup \{l\})$ to $(\{\}, H')$

then $G_{i+1} = \leftarrow l'$ and $H_{i+1} = H'$

An **abductive refutation** is an abductive derivation to a pair $([\,],H')$.

A **consistency derivation** from (F_1,H_1) to (F_n,H_n) is a sequence

$$(F_1,H_1),\ (F_2,H_2),\ldots,\ (F_n,H_n)$$

such that, for each i, $0<i<n$, F_i has the form $\{\leftarrow l,l'\} \cup F_i'$, where (without loss of generality) the clause $\leftarrow l,l'$ has been selected (to continue the search), R selects l and

con1) If l is not abducible
then $F_{i+1} = C' \cup F_i'$ and $H_{i+1} = H_i$

where C' is the set of all resolvents of clauses in KB with the selected clause on the selected literal, and $[\,] \notin C'$.

con2) If l is abducible, $l \in H_i$ and l' is not empty
then $F_{i+1} = \{\leftarrow l'\} \cup F_i'$ and $H_{i+1} = H_i$

con3) If l is abducible ($l=k^-$), $l \notin H_i$

then if there is an abductive derivation from
($\leftarrow$k, H_i) to ([], H')

then F_{i+1} = F_i' and H_{i+1} = H'

else if l' is not empty

then F_{i+1} = {$\leftarrow$l'} U F_i' and H_{i+1} = H_i [8]

##

The correctness of the abduction procedure for a abductive program KB means that whenever there exists an abductive derivation from ($\leftarrow$A,{}) to ([],H) for A$\in$HB then there exists a preferred extension K of KB such that K $\vdash$ A and H $\subseteq$ K. We will prove the soundness of the abductive procedure by proving the following stronger result.

Theorem 5 (Soundness of the abductive procedure)

Let ($\leftarrow$A,{}),...,([],H) be an abductive refutation. Then KB U H is an admissible scenario and KB U H $\vdash$ A.

Proof See appendix. ##

Another characterization of the abduction procedure is given in the following.

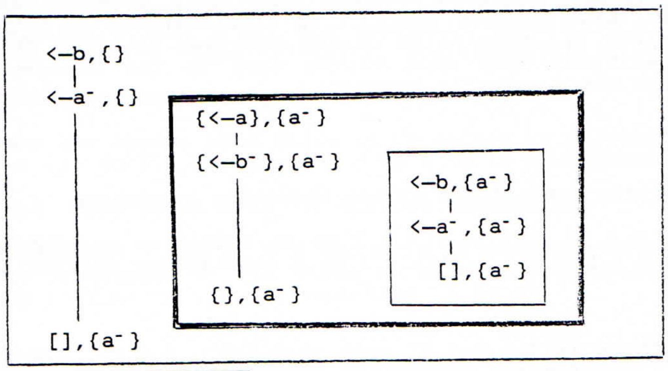
Theorem 6 Let KB be an abductive program and (G,{}),... ([],K) be an abductive refutation. Then

$$\text{comp(KB) U R U IC} \vdash \neg b \qquad \text{for any } b^{-} \notin K,$$

where R = Der(K,KB) and comp(KB) is the Clark's predicate completion of KB without the completed definitions of abducible predicates. ##

Example 6 Let KB: $a \leftarrow b^{-}$
$b \leftarrow a^{-}$

We obtain the following search space for the goal (<-b,{})



[8]. In the original definition of this procedure, H_{i+1} = H_i U {l} which is an error. See [D3] for more details.

Conclusion

We have shown that abductive frameworks provide a simple basis for declarative and operational semantics of logic programs. We have introduced the notions of preferred extensions and complete scenarios and demonstrated how these new notions provide an unified framework which captures and generalizes different semantic concepts (e.g. well-founded models, stable models, negation as failure) in logic programming. The key step is the way we interprete the plausible rule that a hypothesis is acceptable if there is no evidence to the contrary.

We argue that, in general, there are two main semantic intuitions for knowledge representation schemes: the minimalism (formalized by well-founded semantics) and the maximalism (formalized by preferential semantics).

In [D4], we apply the theory developed in this paper to give a logical foundation for nonmonotonic truth maintenance systems.

Following interesting special classes of programs can be identified and a further study is anticipated:

Stable programs: Programs whose preferred extensions are stable. This class contains the locally stratified programs [PT] as well as the sufficiently stratified programs [DK].

Well-founded programs: Programs where the well-founded scenario is the unique complete scenario.

Constructive programs: Programs which are both stable and well-founded.

Acknowledgements

I am grateful to Robert Kowalski for his insightful discussions as well as his many valuable comments to an earlier version of this paper. Many thanks to Tony Kakas, Mark Wallace and Wolfgang Wechler for their comments and suggestions. Last but not least, I am also thankful to Huynh Ngoc Phien for his many helps during the time in which this paper is written. Very much thanks also to the anonymous referees for their valuable comments .

Appendix

We show now the correctness of the abductive procedure.

Let β: $(G_1, H_1), (G_2, H_2), \ldots, ([\,], H_n)$ be an abductive refutation
μ: $(F_1, K_1), (F_2, K_2), \ldots, (\{\}, K_n)$ be a consistency derivation

Define

$\beta > \mu$ if there is $G_i = \leftarrow l, l'$ such that $l = k^-$, $l \notin H_i$, and μ is a consistency derivation from $(\{\leftarrow k\}, H_i \cup \{l\})$ to $(\{\}, H')$ such that $G_{i+1} = \leftarrow l'$ and $H_{i+1} = H'$

$\mu > \beta$ if there is $F_i = \{\leftarrow l, l'\} \cup F_i'$ such that $l = k^-$, $l \notin K_i$ and β is an abductive derivation from $(\leftarrow k, K_i)$ to $([\,], K')$ such that $F_{i+1} = F_i'$ and $K_{i+1} = K'$

If $G_1 = \leftarrow A$ then the set of all hypotheses appearing in goals G_i's in β is called the **evidence of A generated by β.**

Further let $>>$ be the transitive closure of $>$.

It is obvious that the following lemmas A1,A2 hold.

Lemma A1

1) Let β be an abductive refutation and μ: $(\{\leftarrow B\},K),\ldots,(\{\},K')$ be a consistency derivation such that $\beta >> \mu$. Then $B^- \in K$.
2) Let β be an abductive derivation and β': $(\leftarrow B,M),\ldots,([\,],M')$ be an abductive derivation such that $\beta >> \beta'$. Then $B^- \notin M$.
3) Let μ be a consistency derivation from $(\{\leftarrow A\},K)$ to $(\{\},K')$. Let E be an evidence of A. Then for some $B^- \in E$, there exists an abductive refutation β from $(\leftarrow B,H)$ to $([\,],H')$ such that $\mu > \beta$. ##

Lemma A2 If there exists an abductive refutation from $(\leftarrow A,H)$ to $([\,],H')$ then there exists no consistency derivation from $(\{\leftarrow A\},K)$ to $(\{\},K')$ for any K with $H' \subseteq K$. ##

Lemma A3 Let β be an abductive refutation from $(\leftarrow A,H)$ to $([\,],H')$ such that $A^- \notin H$. Then $A^- \notin H'$.

Proof Assume the contrary. Then there exists a consistency derivation μ: $(\{\leftarrow A\},K),\ldots,(\{\},K')$ for some K such that $\beta >> \mu$. Let E be the evidence of A generated by β. Thus there exists an abductive refutation β': $(\leftarrow B,M),\ldots,([\,],M')$ for some $B^- \in E$ such that $\mu > \beta'$ (lemma A1.3). It is clear that $B^- \notin M$ (lemma A1.2). Since $B^- \in E$ and E is generated by β, there is a consistency derivation μ' from $(\{\leftarrow B\},R)$ to $(\{\},R')$ such that $\beta > \mu'$. It is clear that $B^- \in R$. Since $B^- \notin M$ and $B^- \in R$, it follows that μ' (as a process) starts after β' in the process of β. Then either $\beta' >> \mu'$ or β',μ' are disjoint. Since $\beta > \mu'$ and $\beta >> \mu > \beta'$, it follows immediately that β',μ' are disjoint. Thus $M' \subseteq R$. Lemma A2 implies that μ' does not exist. Contradiction !! ##

It follows immediately

Lemma A4 Assume that there is an abductive refutation β from $(\leftarrow A,\{\})$ to $([\,],H)$. Then $H \cup KB \cup IC$ is consistent.

Proof From lemma A1.3, it follows immediately

Proposition Let $B^- \in H$ and E is an evidence of B. Then for some hypothesis $X^- \in E$ there exist a abductive refutation β' from $(\leftarrow X,R)$ to $([\,],R')$ such that $\beta >> \beta'$.
#

Assume that $KB \cup H \cup IC$ is inconsistent. Then there is an atom X such that $X^- \in H$ and $X \in Der(H,KB)$. From $X^- \in H$ and H is an evidence of X, it follows from the above proposition that there is $B^- \in H$ with an abductive refutation β' from $(\leftarrow B,K)$

to ([],K') such that ß >> ß'. It is clear that $B^- \notin K$ (lemma A1.2). But since $B^- \notin H$, there is a consistency derivation μ from ({<–B},R) to ({},R') such that ß>>μ and $B^- \notin R$. There are two cases:

Case 1 ß'>>μ. That means that $B^- \notin K'$. Contradiction to lemma A3.

Case 2 μ and ß' are disjoint. Thus μ' (as a process) starts after ß' (as a process) terminates in ß (as a process). Then it is clear that K' $\subseteq$ R. Lemma A2 implies that μ does not exist. Contradiction !! ##

It follows immediately

Theorem 5 (Soundness of the abduction procedure)
Let (<–A,{}),...,([],H) be an abductive refutation. Then KB U H is an admissible scenario such that KB U H $\vdash$ A.
##

References

[Ck] Clark,K.L.
'Negation as Failure' in Logic and Database, Gallaire H., Minker J. (eds), Plenum, New York,1978

[CP] Cox P.T., Pietrzykowski T.
'Causes for events: their computation and application' Proc. of CADE-86,

[D1] Dung P.M.
'On the strong completion of logic programs' In Proc. of Second International Conference on Algeraic and Logic Programming. October 1-3, 1990, France. Lecture Notes in Computer Science 463

[D2] Dung P.M.
'Logic programming as a nonmonotonic inheritance reasoner' Technical report, March 1990, Division of Computer Science, Asian Institute of Technology

[D3] Dung P.M.
'Negations as hypotheses: an abductive foundation for logic programming' Technical report, October 1990, Division of Computer Science, Asian Institute of Technology

[D4] Dung P.M.
'An abductive foundation for NTMS' Technical report, January 1991, Division of Computer Science, Asian Institute of Technology

[DK] Dung P.M., Kanchanasut K.,
'A fixpoint approach to declarative semantics of logic programs' in Proc. of NACLP-89, Cleveland Ohio 1989, MIT Press

[EK] Eshghi K., Kowalski R.A.
'Abduction Compared with Negation by Failure' In Proc. of 6th ICLP, 1989

[GCH] Guerreiro R.A.de T., Casanova M.A., Hemerly A.S.
'Contributions to a proof theory for generic defaults' ECAI-90, Stockholm, Sweden

[GRS] Van Gelder A., Ross K., Schlipf J.S.
'Unfounded sets and well-founded semantics for general logic programs' in PODS 1988

[GL] Gelfond M.,Lifschitz V.
'The stable model semantics for logic programs' in Proc. of the 5th Int Conf/Sym on Logic Programming MIT Press, 1988

[K] Kowalski R.A.
'Logic for problem solving'
Elsvier North Holland, New York, 1979

[KM] Kakas A.C., Mancarella P.
'Generalized stable models: A semantics for abduction'
In Proc. ECAI-90

[L] Lloyd J.W.
'Foundations of Logic Programming',
second edition, Springer Verlag, 1987

[MT] Marek W., Truszczynski M.
'Stable semantics for logic programs and default theories' in Proc. of NACLP-89, USA

[PT1] Przymusinski T.C. ,
'On the Declarative Semantics of Deductive Databases and Logic Programs' in Foundations of Deductive Databases & Logic Programming, J. Minker (ed.) 1988

[PT2] Przymusinski T.C.
'Three-valued nonmonotonic formalism and logic programming' In Proc. of first Inter. Conf. on Principles of KR and reasoning

[PO] Poole D.
'A Logical Framework for Default Reasoning'
Artificial Intelligence 36 (1988), 27-47

[PAA] Pereira L.M., Aparicio J.N., Alferes J.J.
'Nonmonotonic reasoning with well-founded semantics'
Preprint, Oct. 21, AI center, Uninova, Portugal

[R] Ross, K.A.
"A procedural semantics for well-founded negation in logic programs", Technical report, Stanford, Department of Computer Science, 1988.

[THT] Touretzky D.S., Horty J. F., Thomason R.H.
'A clash of intuitions: the current state of nonmonotonic multiple inheritance systems'
IJCAI 1987

[YY] Yuan L.Y., You J.H.
'A new circumscription method for stable model semantics' Submitted

[W] Wallace, M.
'Unrestricted logic programs or if stratification is the cure, what is the malady?' In Proc. of ECAI-90