AN ARGUMENTATION SEMANTICS FOR LOGIC PROGRAMMING WITH EXPLICIT NEGATION

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

Abstract

The paper presents two novel results:

A theory of argumentation which can handle different kinds of conflicts between arguments

Based on the newly developed theory of argumentation we define a simple framework for semantics of extended logic programming with explicit negation generalizing in a natural way the major approaches to semantics of normal logic programming.

1. Introduction

In general, a logic program can be viewed as a schema for forming arguments which may possibly support conflicting conclusions. The semantics of a logic program is determined by the way the conflicts between its arguments are resolved. In normal logic programming, there is only one kind of conflict caused by the negation as failure assumptions. The nature of this kind of conflict has been analysed deeply in tne literature [ABW,C1,D1,D2,DL1,EK,GRS,GL1,I1,I2, KKT,KM1, KM2,L,P1], and is well-understood today.

Logic programming has been extended by Gelfond and Lifschitz [GL2], Kowalski and Sadri [KS] to allow reasoning with explicit negation. The introduction of a new kind of negation brings with it a new kind of conflict between arguments. Hence, to understand the nature of the semantics of extended logic programming, it is necessary to understand the "interaction" between the two kinds of conflicts between arguments. Though much work have been done lately to study the semantics of extended logic programming [ADP,AP,DR,GL2,GS,KS,I,PAA,P2], no satisfactory analysis on the "interaction" between two kinds of conflicts in extended logic programming has been given. As the result, the nature of semantics of extended logic programming is still not well-understood today. Hence, the goal of this paper is to clarify this problem.

To have an intuitive understanding of the nature of the conflicts in extended logic programming, let us take a look at a concrete example.

Example 1 (Why Tweety does not fly) Imagine the following argumentation between two persons P1 and P2 about Tweety.

P1: Tweety can not fly since Tweety is a penguin and penguins can not fly

P2 does not agree with this conclusion. He tries to attack the argument of P1 via Reductio Ad Absurdum, i.e. by pointing out that accepting the argument of P1 can lead to a contradiction. P2 introduces his argument skillfully in the following dialog:

P2: Well, I agree with you that Tweety can not fly. But this doesn't change the fact that Tweety is a bird. Doesn't it ?

P1: You are right. Tweety is still a bird though it can't fly.

P2: And you also agree with me that birds can fly. Don't you ?

P1: Of course.

P2: Well, so Tweety can fly too. Can't it ?

It seems that P2 has been successful in his attempt to reject the conclusion of P1 that Tweety can not fly. But wait ! P1 continues as follows:

P1: You "forget" that though most birds do fly, some of them may not. And penguins are among the exceptions. We all know it. Don't we ?

According to the commonsense principle that specific information overrides more general one, P1 has defeated P2's attack. So P1 has successfully defended his argument. And so, he does convince us all that Tweety indeed can not fly.

P1 and P2's knowledge about Tweety together with their arguments can be "encoded" in extended logic programming as follows:

$$
\begin{aligned}
&fly(X) \leftarrow bird(X), not\text{-}ab_B(X)\\
&\neg fly(X) \leftarrow penguin(X), not\text{-}ab_P(X)\\
&bird(X) \leftarrow penguin(X)\\
&ab_B(X) \leftarrow penguin(X), not\text{-}ab_P(X)\\
&penguin(Tweety)
\end{aligned}
$$

P2's philosophy can be explained as follows: If it can be assumed that Tweety is a normal penguin just because most penguins are normal then it can also be assumed that Tweety is a normal bird since most birds are normal, too. But since assuming both leads to a contradiction none should be assumed. And so one should not conclude that Tweety can not fly.

P1's argument is based on the assumption that Tweety is a normal penguin. So it can be characterized by the set {not-ab_P(Tweety)}. P2's argument is based on the asumption not-ab_B(Tweety). So it is characterized by the set {not-ab_B(Tweety)}. Let A1 = {not-ab_P(Tweety)}, and A2 = {not-ab_B(Tweety)}. So A2 represents an Reductio Ad Absurdum attack against A1. Since A1 implies ab_B(Tweety), A1 represents an attack against the basis of A2. We say that A1 represents a (ground) attack against A2.

The principle of specific information overriding more general one dictates that the ground attack from A1 against A2 is stronger than the Reductio Ad Absurdum attack from A2 against A1. So the intended semantics of the above extended logic program should conclude that $\neg$ fly(Tweety). ■

The conclusion from the above example is that there are two different ways to attack an argument in extended logic programming and an attack on a negation-as-failure assumption of an argument is stronger than an attack via Reductio Ad Absurdum.

The above example suggests that a natural and intuitive way to study the semantics of extended logic programming is within the framework of argumentation. In fact, this is not totally new as it has been showed in [D1,KKT,D2,KM1,KM2] that argumentation provides a simple and natural framework for studying the semantics of normal logic programming.

Roughly, the idea of argumentational reasoning is that a statement is believable if it can be argued successfully against attacking arguments. In other words, whether or not a rational agent believes in a statement depends on whether or not the argument supporting this statement can be successfully defended against the counterarguments. The goal of this paper is to show how this simple idea can be applied to provide a simple framework for the semantics of extended logic programming. We do so by developing an argumentation theory which can handle more than one kind of conflicts between arguments and then applying this theory to define the semantics of extended logic programming.

The paper is organized as follows: In chapter 2, we analyze the nature of conflicts between arguments and give a formal definitions of Reductio Ad Absurdum attacks as well as of ground attacks. Then in chapter three, we define the semantics of argumentation frameworks which allow two diffewrent kinds of attacks between arguments. We show that applying the newly developed theory of argumenation provides a simple and intuitive semantics for extended logic programming generalizing many major approaches to semantics of normal logic programming.

2. Extended Logic Programming as Argumentation Frameworks

An *objective literal* is either an atom A or the explicit negation ¬A of an atom A. A *subjective literal* is an expression of the form not-L where L is an objective literal. An extended program clause C is of the form

$$L_0 \leftarrow L_1,..,L_m,\text{not-}L_{m+1},..,\text{not-}L_{m+n}$$

where L_i's are objective literals. L_0 is said to be the head of C and denoted by head(C).

An extended logic program is a finite set of extended program clauses.

Let K be an objective literal. A *(defeasible) proof of* K can be viewed as a sequence $C_0,C_1,...,C_n$ of ground instances of clauses in P with head(C_n) = K such that for each i, $0 \leq i \leq n$, for each objective literal L in the body of C_i, there is a $j<i$ such that L = head(C_j).

The subjective literals appearing in a proof are the assumptions on which the proof is based. It is clear that the "acceptance" of a proof depends on the "acceptance" of the assumptions on which it is based. Hence arguing for an conclusion K means arguing for the assumptions on which some proof of K is based. So it is intuitive to view the set of assumptions on which a proof of K is based as an argument for K.

Definition 1 1) Let P be an extended logic program. An *argument* wrt P is defined as a set of ground subjective literals.

2) An objective literal L is *supported by* an argument A if there exists a proof of L whose assumptions are contained in A. ■

Definition 2 1) We say that an argument A is *selfdefeating* if there is an objective literal L such that A supports both L and ¬L.

2) An argument is said to be *sound* if it is not selfdefeating. ■

We give now a precise definition of attacks via Reductio Ad Absurdum.

Definition 3 Let A,A' be two sound arguments. Then we say that A' represents a *Reductio Ad Absurdum - attack* (or *RAA-attack* for short) against A if A ∪ A' is selfdefeating. ■

Another way to attack an argument is to attack its basis.

Definition 4 Let A,A' be sound arguments. Then A' represents a *ground attack* against A if there is an assumption not-L in A such that L is supported by A'. ■

To illustrate the interactions between the two kinds of conflicts between arguments, let us continue to consider the well-known example about Tweety.

Example 2 Let P be the program introduced in example 1. The semantics of P is determined by the conflicts between the arguments A0 = ∅, A1 = {not-ab_P(Tweety)}, A2 = {not-ab_B(Tweety)}. A2 is clearly an RAA-attack against A1. Since A1 supports ab_B(Tweety), A1 is an groundattack against A2. ■

Given an extended logic program P, the set of all sound arguments wrt P is denoted by $\mathbf{AR_P}$.

Definition 5 *Let P be an extended logic program. The semantics of P is defined by the semantics of the following argumentation framework*

AF(P) = <AR_P, attacks, g-attacks>

where following conditions are satisfied:

attacks = RAA-attacks ∪ g-attacks

RAA-attacks(A',A) iff A' represents a RAA-attack against A

g-attacks(A',A) iff A' represents a ground attack against A

■

The semantics of argumentation frameworks is given in the following chapter.

3. Argummentation Frameworks

An argumentation framework is defined as a triple of a set of arguments, and two binary relations representing the attack-relationship between arguments.

Definition 6 An *argumentation framework* is a triple

AF = <AR, attacks, g-attacks>

where AR is a set of arguments, and attacks, g-attacks are binary relations on AR such that g-attacks ⊆ attack. ■

An argumentation framework is *normal* if attacks = g-attacks.

For two arguments A,B, the meaning of attacks(A,B) is that A represents an attack against B. g-attacks(A,B) means that A represents a ground attack against B. The purpose of distinguishing between (normal) attacks and ground attacks is to classify different kinds of attacks among which the ground attacks are the most "lethal" ones.

A theory of normal argumentation and its relations to normal logic programming is given in [D2]. In the following, we generalize this theory for general argumentation frameworks with more than one kind of conflicts between arguments.

Remark From now on, if not explicitely mentioned otherwise, we always refer to an arbitrary but fixed argumentation framework AF = <AR,attacks,g-attacks>. Further, we say that A attacks (resp. g-attacks) B if attacks(A,B) (resp. g-attacks(A,B)) holds. Similarly, we say that a set S of arguments attacks (resp. g-attacks) B if B is attacked (resp. g-attacked) by an argument in S.

Definition 7 A set S of arguments is said to be *conflict-free* if no two elements of it attack each other, i.e. there are no two arguments A,B in S such that (A,B) $\in$ attacks or (B,A) $\in$ attacks. ■

For a rational agent G, an argument A is acceptable if G can defend A (from within his world) against all attacks on A. Further, it is reasonable to assume that a rational agent accepts an argument only if it is acceptable. That means that the set of all arguments accepted by a rational agent is a set of arguments which can defend itself against all attacks on it. This leads to the following definition of an admissible (for a rational agent) set of arguments.

Definition 8 *(1) An argument A is acceptable wrt a set S of arguments if for each argument B: if B attacks A then B is g-attacked by some A' $\in$ S.*

(2) A conflict-free set of arguments S is admissible if each argument in S is acceptable wrt S. ■

The (credoulous) semantics of an argumentation framework is defined by the notion of preferred extension.

Definition 9 A *preferred extension* of an argumentation framework AF is a maximal (wrt set inclusion) admissible set of arguments. ■

Example 3 (Continuation of examples 1,2) Let P be the program given in example 1. Let A0 = $\varnothing$, A1 = {not-ab_P(Tweety)} and A2 = {not-ab_B(Tweety)}. Then AR_P = {A0,A1,A2}, and RAA-attacks = {(A1,A2),(A2,A1}, and g-attacks = {(A1,A2)}. It is not difficult to see that $\varnothing$, {A0}, {A1}, and {A0,A1} are the only admissible sets of arguments in this example. Thus {A0,A1} is the unique preferred extension. ■

Example 4 P:

$$a \leftarrow \text{not-}p$$
$$\neg a \leftarrow \text{not-}q$$
$$b \leftarrow \text{not-}r$$

Let AF(P) = <AR_P,attacks,g-attacks>. It is not difficult to see that g-attacks = $\varnothing$. From that, we can conclude that preferred extensions of AF(P) contain only those arguments which are not attacked by any other arguments. There are only two such arguments in AF(P): A0 = ∅ and A1 = {not-r}. It follows that AF(P) has exactly one preferred extension which is {A0,A1}. Thus we can conclude b from P but neither a nor ¬a. ■

Lemma 1 (Fundamental lemma) Let S be an admissible set of arguments and A,A' be arguments which are acceptable wrt S. Then

1) S' = S ∪ {A} is admissible, and
2) A' is acceptable wrt S'. ■

The following theorem shows that preferred extension semantics is always defined for argumentation frameworks.

Theorem 2 (1) The set of all admissible sets of arguments form a complete partial order wrt set inclusion.

(2) For each admissible set S of arguments, there exists an preferred extension E such that S ⊆ E. ■

Since there can be no attacks against the empty set ∅ of arguments, it is clear that the empty set ∅ of arguments is always admissible. Thus, from the above theorem it follows immediately

Corollary 3 Every argumentation framework possesses at least one preferred extension. ■

It is not difficult to see that the semantics defined by the preferred extension of AF(P) for a normal logic program P coincides with the preferred extension semantics defined directly for P in [D1].

Stable Semantics for Argumentation

The following notion of stable extension represents the counterpart of the answer set semantics [GL2] of extended logic programs.

Definition 10 A conflict-free set of arguments S is a *stable extension* if S g-attacks each argument which does not belong to S. ■

The relations between stable extension and preferred extension are clarified in the following lemma.

Lemma 4 Every stable extension is a preferred extension, but not vice versa. ■

That the stable extensions capture the answer set semantics of an extended logic program P is showed in the following theorem.

Theorem 5 Let P be an extended logic program. Then S is an answer set of P iff there is a stable extension E of AF(P) such that

$$S = \{ L \mid L \text{ is supported by an argument from } E \}$$ ■

Fixpoint Semantics and Grounded (Skeptical) Semantics

We show in this chapter that argumentation can be characterized by a fixpoint theory which also provides an elegant way to introduce grounded (skeptical) semantics.

Definition 11 The *characteristic function* of an argumentation framework AF, denoted by F_{AF}, is defined as follows:

$$F_{AF}: 2^{AR} \rightarrow 2^{AR}$$

$$F_{AF}(S) = \{ A \mid A \text{ is acceptable wrt } S \}$$

■

Remark As we always refer to an arbitrary but fixed argumentation framework AF, *we often write F shortly for F_{AF}.*

Lemma 6 A conflict-free set S of arguments is admissible iff $S \subseteq F(S)$.

Proof The lemma follows immediately from the property "If S is conflict-free then F(S) is also conflict-free". So we need only to prove this property. Assume that there are A,A' in F(S) such that A attacks A'. Thus, there exists B in S such that B g-attacks A. Hence there is B' in S such that B' g-attacks B. Contradiction !! So F(S) is conflict free. ■

It is easy to see that if an argument A is acceptable wrt S then A is also acceptable wrt any superset of S. Thus, it follows immediately that

Lemma 7 F_{AF} is monotonic (wrt set inclusion). ■

The skeptical semantic of argumentation frameworks is defined by the notion of grounded extension introduced in the following.

Definition 12 The *grounded extension* of an argumentation framework AF, denoted by GE_{AF}, is the least fixed point of F_{AF}. ■

Example 5 (Continuation of the Tweety example) First let P be the program given in example 1 (AF(P) is given in example 3). Then $F(\emptyset) = \{A0\}$, and $F^2(\emptyset) = F(\emptyset) = \{A0\}$. Note that the grounded extension of AF(P) is different to its preferred extension. It is interesting to compare the grounded extension of AF(P) with the grounded extension of AF(P') where P' is obtained from P by replacing the clause $ab_B(X) \leftarrow penguin(X), not\text{-}ab_P(X)$ by the clause $ab_B(X) \leftarrow penguin(X)$ That means P' is the following program:

P':

$$\begin{aligned}
&fly(X) \leftarrow bird(X), not\text{-}ab_B(X)\\
&\neg fly(X) \leftarrow penguin(X), not\text{-}ab_P(X)\\
&bird(X) \leftarrow penguin(X)\\
&ab_B(X) \leftarrow penguin(X)\\
&penguin(Tweety)
\end{aligned}$$

Let F' denote the characteristic function of AF(P'). Then we have $F_{AF}(\emptyset) = \{A0\}$, and $F_{AF}{}^2(\emptyset) = F_{AF}{}^3(\emptyset) = \{A0, A1\}$. Hence the grounded extension of AF(P') is also a preferred one. ■

Example 6 (Continuation of exampl 4) P be the program given in example 4. From the fact that g-attacks = $\emptyset$, it is not difficult to see that $F(\emptyset) = \{A0, A1\}$, and $F(\emptyset) = F^2(\emptyset)$ where $A0 = \emptyset$ and $A1 = \{not\text{-}r\}$. Thus the grounded extension of AF(P) is also a preferred one, but not a stable one. ■

The following theorem shows that for a normal logic program P, the well-founded semantics [GRS] of P "coincides" with the grounded extension of AF(P).

Theorem 8 Let P be a normal logic program. Let P* be the program obtained from P by adding to P clauses of the form $\neg p(x_1,..,x_n) \leftarrow not\text{-}p(x_1,..,x_n)$ for each predicate symbol p. Let GE be the grounded extension of AF(P*). Then a consistent set of objective literals W is the well-founded model of P iff

$$W = \{ L \mid L \text{ is supported by an argument from GE } \}$$

■

The following notion of complete extension provides the link between preferred extensions (credulous semantics), and grounded extension (skeptical semantics).

Definition 13 An admissible set of arguments S is called a *complete extension* iff each argument which is acceptable wrt S, belongs to S. ■

Intuitively, the notion of complete extensions captures the kind of rational agents who believes in every thing he can defend.

Lemma 9 A conflict-free set of arguments E is a complete extension iff $E = F_{AF}(E)$. ■

The relations between preferred extensions, grounded extensions and complete extensions is given in the following theorem.

Theorem 10 (1) Each preferred extension is a complete extension, but not vice versa.

(2) The grounded extension is the least (wrt set inclusion) complete extension.

(3) The complete extensions form a complete semilattice[1] wrt set inclusion.

Proof (1) It is obvious from the fixpoint definition of complete extensions that every preferred extension is a complete extension. (2) Obvious
(3) Let SE be a nonempty set of complete extensions. Let LB = $\{ S \mid S$ is admissible and $S \subseteq E$ for each E in SE $\}$. It is clear that GE $\in$ LB. So LB is not empty. Let $S' = \cup\{ S \mid S \in LB\}$. It is clear that S' is admissible, i.e. $S' \subseteq F(S')$. Let $E = lub(F^i(S'))$ for ordinal i. Then it is clear that E is a complete extension and $E \in LB$. Thus $E = S'$. So E is the glb of SE. ■

In general, F_{AF} is not continuous, but if the argumentation framework is finitary then it is.

Definition 14 An argumentation framework AF = <AR,attacks,g-attacks> is finitary iff for each argument A, A is attacked by only finitely many number of arguments in AR. ■

Lemma 11 If AF is finitary then F_{AF} is ω-continuous.

Proof Let $S_0 \subseteq ... \subseteq S_n \subseteq ..$ be an increasing sequence of sets of arguments, and let $S = S_0 \cup ... \cup S_n \cup ...$ Let $A \in F_{AF}(S)$. Since there are only finitely many arguments which attack A, there exists a number m s.t. $A \in F_{AF}(S_m)$. Therefore, $F_{AF}(S) = F_{AF}(S_0) \cup ... \cup F_{AF}(S_n) \cup ....$ ■

We want to conclude this chapter with one more example. Let consider the knowledge base:

> Most US presidents serve out their full terms, except those who are impeached by the congress.
>
> Most quakers are pacifists.
> Most republicans are not pacifists.
>
> Nixon is an impeached US president who is also both a quaker and a republican.

This can be encoded by the following program:

---

[1] A partial order $(S,\leq)$ is a complete semilattice iff each nonempty subset of S has a glb and each increasing sequence of S has a lub.

P: serve-full-term(X) $\leftarrow$ US-president(X), not-Ab_{pres}(X)
$\neg$serve-full-term(X) $\leftarrow$ impeached-US-president(X)
Ab_{pres}(X) $\leftarrow$ impeached-US-president(X)
US-president(X) $\leftarrow$ impeached-US-president(X)

pacifist(X) $\leftarrow$ quaker(X), not-Ab_q(X)
$\neg$pacifist(X) $\leftarrow$ republican(X), not-Ab_r(X)

impeached-US-president(Nixon) $\leftarrow$
republican(Nixon) $\leftarrow$
quaker(Nixon) $\leftarrow$

For there are no ground attacks against the arguments {not-Ab_q) and {not-Ab_r}, and these two arguments attack each other, none of them could belong to any admissible scenario. Hence, it is easy to see that the only preferred extension of AF(P) is {∅} which is also the grounded extension. So we can conclude that Nixon does not serve out his term. But nothing about the pacifistness of Nixon can be concluded.

Coherent Argumentation Frameworks

Now, we want to give a condition for the coincidence between stable extensions and preferred extensions. In general, the existence of a preferred extension which is not stable indicates the existence of some anomalies in the corresponding argumentation framework, e.g. the argumentation framework corresponding to the logic program p $\leftarrow$ not p has an empty preferred extension which is not stable. Sc it is interesting to find sufficient conditions to avoid such anomalies.

Definition 15 An argumentation framework AF is *coherent* iff each preferred extension of AF is stable. ■

It follows directly from the definition that

Property There exists at least one stable extension in a coherent argumentation framework. ■

Now we want to give a sufficient condition for an argumentation framework to be coherent.

We say that an argument B *potentially attacks* A if there exists a finite sequence $A_0,..,A_n$ such that 1) $A = A_0$ and $B = A_n$, and 2) n is an odd number and, 3) for each i, A_{2i+1} attacks A_{2i}, and A_{2i} g-attacks A_{2i-1}.

An argument B *indirectly defends* A if there exists a finite sequence $A_0,..,A_n$ such that 1) $A = A_0$ and $B = A_n$, and 2) n is an even number and, 3) for each i, A_{2i+1} attacks A_{2i}, and A_{2i} g-attacks A_{2i-1}.

An argument B is said to be *controversial wrt A* if B potentially attacks A and indirectly defends A, at the same time. An argument is *controversial* if it is controversial wrt some argument A.

Definition 16 An argumentation framework is *uncontroversial* if there exists no controversial argument. ■

The following theorem is proved in [D2]

Theorem 12 Normal uncontroversial argumentation frameworks are coherent. ■

Now we want to introduce the class of seminormal argumentation frameworks which can be "reduced" to a normal one.

Definition 17 An argumentation framework AF is *seminormal* if for each $(A_0,A_1) \in$ attacks, either $(A_0,A_1) \in$ g-attacks or $(A_1,A_0) \in$ g-attacks holds. ■

It is clear that each normal argumentation framework is seminormal but not vice versa.

The intuition behind the seminormality is that though there are two kinds of attacks in a seminormal argumentation framework, the semantics of it is determined by the ground attacks. This is demonstrated in the following.

Each seminormal argumentation framework AF = (AR,attacks,g-attacks) can be transformed into a normal one N(AF) = (AR',attacks',g-attacks') with AR' = AR, and attacks' = g-attacks' = g-attacks.

Lemma 13 Let AF be a seminormal argumentation framework. Then a set of arguments S is admissible in AF if and only if it is admissible in N(AF).

Proof "=>" Obvious
"<=" Let S be an admissible set of arguments in N(AF). Let A be an attack on S. Let $B \in S$ s.t. $(A,B) \in$ attacks. If $(A,B) \in$ g-attacks then from the admissibility of S in N(AF), it is clear that S g-attacks A. If $(A,B) \notin$ g-attacks then from the seminormality of AF, it follows that $(B,A) \in$ g-attacks. Hence S g-attacks A. q.e.d. ■

It follows immediately

Theorem 14 A seminormal AF is coherent iff N(AF) is coherent. ■

An extended logic program P is called seminormal (resp. uncontroversial) if its argumentation framework AF(P) is seminormal (resp. uncontroversial).

It follows immediately that for seminormal and uncontroversial logic programs, preferred semantics and stable semantics coincide.

Corollary 15 Let P be an arbitrary seminormal and uncontroversial logic program. Then AF(P) is coherent. ■

For each extended logic program P, let P* denote the normal logic program obtained by replacing each objective literal $\neg p(t_1,...,t_n)$ by the positive literal $p^*(t_1,...,t_n)$ where p* is a new predicate symbol introdued into the language for each predicate symbol p in P. It follows then immediately from [K2,D4]

Corollary 16 A seminormal extended logic program P is coherent iff P* is call-consistent. ■

Conclusions

We have pointed out that there are two kinds of conflicts between arguments in an extended logic program. Then we have generalized the theory of normal argumentation from [D2] to handle this case. Our theory provides a simple and intuitive semantics for extended logic programs generalizing many existing approaches. Recently, also inspired by [D2], Bondarenko, Toni and Kowalski [BTK] have developed a very interesting assumption-based framework for nonmonotonic reasoning and logic programming where a deep analysis of the different kinds of attacks is also given. Though different, our works and that of Bondarenko et all complement each other, and together they provide the first deep insights into the nature of argumentation and its applications in logic programming and nonmonotonic reasoning.

Lately, based on the theory of admissible scenaria [D1], Alfere et all [ADP] have introduced another framework for semantics of explicit negation wrt the so-called coherence principle where it is required that if an negative objective literal ¬L is supported then not-L must be accepted. As we don't "enforce" this principle in this paper, it is not clear to us how these approaches are related. We plan to explore the relationship between them in the future.

Acknowledgement

I am grateful to some anonymous referees for the constructive criticisms.

This paper is partially supported by the Abduction Group at the Imperial College under a grant from Fujitsu.

References

[ABW] Apt K., Blair H., Walker A. 'Towards a theory of declarative knowledge' in Minker J (ed.) 'Foundation of dedcutive databsese and logic programming, 1988, Morgan Kaufman

[ADP] Alferes J.J., Dung P.M., Pereira L.M. 'Scenario semantics for extended logic programming', Proc. of the Second Inter. Workshop on Logic Programming and Nonmonotonic Reasoning, Lisbon, 1993, MIT Press

[AP] Alferes J.J., Pereira L.M. 'On logic programs with two kinds of negation' in Proc. of Int. Joint Conf. on LP, MIT Press, 1992.

[BTK] Bondarenko A., Toni F., Kowalski R.A. 'An assumption-based framework for nonmonotonic reasoning', in Proc. of the Second Inter. Workshop on Logic Programming and Nonmonotonic Reasoning, Lisbon, 1993, MIT Press

[C1] Clark,K.L. 'Negation as Failure' in Logic and Database, Gallaire H., Minker J. (eds), Plenum, New York,1978

[D1] Dung P.M. 'Negations as Hypotheses: an Abductive Foundation for Logic Programming' In Proc. of ICLP'91, MIT Press

[D2] Dung P.M. 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming', in Proc. of IJCAI'93

[D3] Dung P.M. 'Logic programming as dialog games', Technical report, December, 1992, AIT

[D4] Dung P.M. 'On the relations between stable and well-founded semantics of logic programs', Theoretical Computer science 105, 1992, 7-25

[DR] Dung P.M., Ruamviboonsuk P. 'Well-founded reasoning with classical negation' in Proc. of Int. Workshop on Logic Programming and Nonmonotonic reasoning, 1991, MIT press.

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

[GRS] Van Gelder A., Ross K., Schlipf J.S. 'Unfounded sets and Well-Founded Semantics for General Logic Programs' in PODS 1988

[GL1] Gelfond M., Lifschitz V. 'The stable model semantics for logic programming' in Proc. 5th ICLP, 1988, MIT press.

[GL2] Gelfond M.,Lifschitz V. 'Logic programs with classical negation' in Proc. of 7th ICLP, 1990, MIT press

[GS] Grant J., Subrahmanian V.S. 'Reasoning in inconsistent knowledge bases', To appear in IEEE Transaction on Knowledge & data Engineering

[I1] Inoue K. 'Extended logic programs with default assumptions' in Proc. of 8th ICLP, 1991, MIT Press.

[I2] Iwayama N. 'Acceptable Hypotheses: Semantics for negation by default', Draft, private communication, March 1993.

[KKT] Kakas T., Kowalski R., Tony F. 'Abductive Logic Programming', To appear in J. of Logic and Computations

[KM1] Kakas T., Mancarella P. 'Stable Theories for Logic Programs', Proc. ISLP'91, MIT Press

[KM2] Kakas T., Mancarella P. 'Preferred extensions are partial stable models', J of LP, 1992, 14, 341-348

[K1] Kowalski R.A. 'Logic for problem solving', Elsvier North Holland, New York, 1979

[K2] Kunen K. 'Signed data dependencies in logic programming' J. of LP, 7, 1989, 231-245

[KS] Kowalski R A., Sadri F. 'Logic Programs with Exceptions', ICLP 1990, MIT Press

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

[P1] Przymusinski T.C., 'On the Declarative Semantics of Deductive Databases and Logic Programs' in Foundations of Deductive Databases & Logic Programming, J. Minker (ed.) 1988

[P2] Przymusinski T.C. 'The well-founded semantics coincides with the three-valued stable semantics' Fundamenta Informaticae 13, 4, 1990

[PAA] Pereira L.M., Aparicio J.N., Alferes J.J. 'Nonmonotonic reasoning with well-founded semantics' in Proc. of 8th ICLP, 1991, MIT press.