# A semantics for the Kakas-Mancarella procedure for abductive logic programming

**Francesca Toni**
*Department of Computing*
*Imperial College*
*180 Queen's Gate, London SW7 2BZ, UK*
`ft@doc.ic.ac.uk` Tel: +44 171 594 8228 Fax: +44 171 589 1552

**Abstract**

The paper presents a soundness result for the Kakas-Mancarella proof procedure for abductive logic programming with respect to an argumentation-theoretic semantics. Furthermore, it discusses the relationship of the Kakas-Mancarella procedure and its semantics with other proof procedures and semantics for abductive logic programming.

**Keywords**: Abduction, Argumentation.

## 1 Introduction

Abductive logic programming (ALP) is the extension of normal logic programming (NLP) to incorporate abducibles and integrity constraints. Abducibles are atoms that can be added to programs, provided their addition does not violate the integrity constraints.

Various forms of ALP have been presented in the literature (see [6] for a survey). Among those, the form of ALP presented by Kakas and Mancarella [8, 7] has played an important role in the development of the field. Kakas and Mancarella propose both a semantics for ALP [8], by generalising the stable model semantics for NLP, and a proof procedure [7], by generalising the abductive proof procedure for NLP by Eshghi and Kowalski [5]. However, the Kakas-Mancarella (KM) procedure is not sound with respect to the generalised stable model semantics, in the same way as the Eshghi-Kowalski (EK) procedure is not sound with respect to the stable model semantics. Kakas and Mancarella, though, prove soundness for a special class of abductive logic programs, namely for abductive logic programs that admit at least a generalised stable model [1] and whose normal logic program is "well-behaved" [7].

[1] This condition is not always explicitly stated in the works of Kakas and Mancarella.

This paper presents an alternative semantics for ALP and shows that the KM procedure is sound (but not complete) with respect to this semantics, for any abductive logic program. This semantics is based upon argumentation-theoretic notions presented in [9], that are variations of those presented in [6, 4, 2, 10, 1]. The suggested semantics for ALP can be seen as an extension of the argumentation-theoretic formulation of the NLP semantics proposed by Dung [3] for the EK procedure. Therefore, the soundness result presented in this paper for the KM procedure can be seen as an extension of the soundness result given by Dung [3] for the EK procedure.

Toni and Kowalski [10] propose a transformation to reduce abductive logic programs into normal logic programs. Then, proof procedures for ALP can be obtained by applying existing procedures for NLP to the result of the transformation. In this paper, we compare the behaviour of the KM procedure applied to the original abductive logic programs with the behaviour of the EK procedure applied to the transformed normal logic programs, and we argue that the behaviour obtained in the second case is preferable.

The paper is organised as follows. In section 2 we revise the KM procedure and the generalised stable model semantics. In section 3 we present the argumentation-theoretic semantics that is the basis for providing the ALP semantics and the soundness result for the KM procedure, given in section 4 and proved in the appendix. Section 5 compares the behaviour of the KM procedure with the behaviour of the EK procedure applied to the result of reducing abductive logic programs to normal logic programs as in [10].

# 2 Kakas and Mancarella ALP

An abductive logic program [8, 7] is a triple $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, where

- $\mathcal{P}$ is a **normal logic program**,
- $\mathcal{AB}_{KM}$ is a set of predicate symbols, called **abducible predicates**,
- $\mathcal{I}_{KM}$ is a set of closed first-order formulas, called **integrity constraints**.

Without loss of generality, abducible predicates in $\mathcal{AB}_{KM}$ are assumed to be undefined in $\mathcal{P}$, namely, for every clause $H \leftarrow L_1, \ldots, L_n$ in $\mathcal{P}$, $H \neq a(t)$ where $a \in \mathcal{AB}_{KM}$ and $t$ is a tuple of terms. In the sequel, with an abuse of notation, we will use $\mathcal{AB}_{KM}$ to indicate also the set of all the variable-free instances of predicates in $\mathcal{AB}_{KM}$.

The semantics of an abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$ is given by its **generalised stable models** [8]. A set $M$ of atoms in the Herbrand base of $\mathcal{P}$ [2] is a generalised stable model of $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$ iff $M$ is a stable model of $\mathcal{P} \cup \Delta$, for

[2] We assume that the Herbrand base of $\mathcal{P}$ includes all predicates, function and constant symbols occurring in $\mathcal{AB}_{KM}$ and $\mathcal{I}_{KM}$.

some $\Delta \subseteq \mathcal{AB}_{KM}$ (abducibles in $\Delta$ are interpreted as unitary clauses), and $M \models \phi$, for every integrity constraint $\phi \in \mathcal{I}_{KM}$.

In [7], Kakas and Mancarella define a proof procedure for the computation of the generalised stable model semantics. While defining the procedure they make the further assumptions that the integrity constraints are expressed in the form of denials of literals, and that all literals in integrity constraints are either abducible atoms or the negation of abducible atoms. In this paper, until section 5, we relax the second condition by assuming that every (denial) integrity constraint contains at least an abducible atom or the negation of an abducible atom.

Following [5], the procedure treats negative literals as additional abducibles, and, for any query $\mathcal{Q}$, generates a set $\Delta$ of abducibles and negative literals as answer. We will refer to literals that are either abducibles or negative literals as **assumptions**. We will assume a safe selection rule $\Re$, namely one that selects an assumption only if it is variable-free. Moreover, for any literal $L$, let $\overline{L}$ be defined as follows: if $L = p(t)$ then $\overline{L} = not\ p(t)$, and if $L = not\ p(t)$ then $\overline{L} = p(t)$.

An **abductive derivation** from $(G_1, \Delta_1)$ to $(G_n, \Delta_n)$ with respect to $\Re$ is a sequence $(G_1, \Delta_1), \ldots, (G_n, \Delta_n)$ such that for all $i = 1, \ldots, n-1$, $G_i$ is a goal of the form $\leftarrow L, \mathcal{Q}$, where $\Re$ selects $L$ in $G_i$ and $\mathcal{Q}$ is a possibly empty conjunction of literals; for all $i = 1, \ldots, n$, $\Delta_i$ is a set of assumptions; and $(G_{i+1}, \Delta_{i+1})$ is obtained according to one of the following rules:

**(A1)** if $L$ is not an assumption, then $G_{i+1} = C$ and $\Delta_{i+1} = \Delta_i$,
where $C$ is the resolvent of some clause in $\mathcal{P}$ with $G_i$ on $L$;

**(A2)** if $L$ is an assumption and $L \in \Delta_i$, then $G_{i+1} = \leftarrow \mathcal{Q}$ and $\Delta_{i+1} = \Delta_i$;

**(A3)** if $L$ is an assumption, $L \notin \Delta_i$, $\overline{L} \notin \Delta_i$,
and there exists a *successful consistency derivation* $(L, \Delta_i \cup \{L\}), \ldots, (\emptyset, \Delta')$
then $G_{i+1} = \leftarrow \mathcal{Q}$ and $\Delta_{i+1} = \Delta'$.

A **successful abductive derivation** is an abductive derivation of the kind $(G_1, \Delta_1)$, ..., $(\Box, \Delta_n)$, with $n \geq 1$.

A **consistency derivation** from $(L, \Delta_1)$ to $(S_n, \Delta_n)$ is a sequence $(L, \Delta_1)$, $(S_1, \Delta_1), \ldots, (S_n, \Delta_n)$ such that $L$ is an assumption; $S_1$ is the set of all goals of the form $\leftarrow \phi$ obtained by resolving the assumption $L$ with the integrity constraints in $\mathcal{I}_{KM} \cup \{\neg[Q \wedge not\ Q] \mid Q \text{ is an atom in } \mathcal{P}\}$, with $\Box \notin S_1$; for all $i = 1, \ldots, n-1$, $S_i$ is a set of goals of the form $\{\leftarrow L, \mathcal{Q}\} \cup S'$; for all $i = 1, \ldots, n$, $\Delta_i$ is a set of assumptions; and $(S_{i+1}, \Delta_{i+1})$ is obtained according to one of the following rules:

**(C1)** if $L$ is not an assumption, then $S_{i+1} = C' \cup S'$ and $\Delta_{i+1} = \Delta_i$,
where $C'$ is the set of all resolvents of rules in $\mathcal{P}$ with $\leftarrow L, \mathcal{Q}$ on $L$ and $\Box \notin C'$;

**(C2)** if $L$ is an assumption, $L \in \Delta_i$ and $\mathcal{Q} \neq \Box$, then $S_{i+1} = \{\leftarrow \mathcal{Q}\} \cup S'$ and $\Delta_{i+1} = \Delta_i$;

**(C3)** if $L$ is an assumption and $\overline{L} \in \Delta_i$, then $S_{i+1} = S'$ and $\Delta_{i+1} = \Delta_i$;

**(C4)** if $L$ is an assumption, $L \notin \Delta_i$ and $\overline{L} \notin \Delta_i$, then
(i) if there exists a *successful abductive derivation* $(\leftarrow \overline{L}, \Delta_i), \ldots, (\Box, \Delta')$

then $S_{i+1} = S'$ and $\Delta_{i+1} = \Delta'$,

(ii) otherwise, if $\mathcal{Q} \neq \Box$, then $S_{i+1} = \{\leftarrow \mathcal{Q}\} \cup S'$ and $\Delta_{i+1} = \Delta_i$.

A **successful consistency derivation** is a consistency derivation of the kind $(L, \Delta_1), \ldots, (\emptyset, \Delta_n)$, for $n \geq 1$.

In general, this procedure is not sound with respect to the generalised stable model semantics, namely, given an abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, a query $\mathcal{Q}$ and a successful abductive derivation from $(\leftarrow \mathcal{Q}, \emptyset)$ to $(\Box, \Delta)$, there might exist no generalised stable model $M$ of $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$ such that $M \models \mathcal{Q}$ and $M \cap \mathcal{AB}_{KM} = \Delta \cap \mathcal{AB}_{KM}$. We illustrate this unsoundness problem by means of two examples. The first example is adapted from the one given in [5] to show that the EK procedure is not sound with respect to the stable model semantics for NLP.

**Example 2.1** Consider the abductive logic program $\langle \mathcal{P}, \emptyset, \emptyset, \rangle$ with $\mathcal{P}$

$$\begin{array}{lcl} r & \leftarrow & not\, r \\ r & \leftarrow & q \\ p & \leftarrow & not\, q \\ q & \leftarrow & not\, p \end{array}$$

There is a successful abductive derivation from $(\leftarrow p, \emptyset)$ to $(\Box, \{not\ q\})$. However, the only generalised stable model of $\langle \mathcal{P}, \emptyset, \emptyset, \rangle$ is $\{q, r\}$, and $\{q, r\} \not\models p$.

**Example 2.2** Consider the abductive program $\langle \{p \leftarrow b\}, \{a, b\}, \{\neg a, \ \neg not\ a\} \rangle$. There is a successful abductive derivation from $(\leftarrow p, \emptyset)$ to $(\Box, \{b\})$. However, there is no generalised stable model of the given abductive program.

The soundness result has been proved to hold for abductive logic programs that admit at least a generalised stable model and whose normal logic program component is "well-behaved" [7]. (The logic program in the first example above is not "well-behaved".) To obtain a more general soundness result one might modify the procedure or adopt a new semantics. In this paper we follow the second alternative. The new semantics, defined in argumentation-theoretic terms, is based upon notions presented in the next section.

# 3 Argumentation semantics

The semantics underlying the KM procedure is expressed in the context of the instance for ALP of an abstract assumption-based framework [9] which is a variant of those proposed in [4, 2, 1] as a semantics for non-monotonic reasoning in general. An **assumption-based framework** is a tuple $\langle \mathcal{T}, \vdash, \mathcal{AB}, \mathcal{IC} \rangle$ where

- $\mathcal{T}$ is a **theory** in some formal language,
- $\vdash$ is a notion of **monotonic derivability** for the given language,

- $\mathcal{AB}$ is a set of **assumptions**, which are sentences of the language, and
- $\mathcal{IC}$ is a set of **integrity constraints**, which are denials of sentences of the language.

In such an (abstract) framework a sentence is a non-monotonic consequence if it follows monotonically from the theory extended by means of an "acceptable" set of assumptions. Various notions of "acceptability" can be defined, based upon a single notion of "attack" between sets of assumptions. Intuitively, one set of assumptions "attacks" another if the two sets together with the theory violate an integrity constraint (i.e. the two sets together with the theory derive, via $\vdash$, all sentences in the denial integrity constraint), and the second set is deemed responsible for the violation. We assume that some of the sentences in the integrity constraints in $\mathcal{IC}$ are explicitly indicated as **retractibles**, with the intended meaning that if a violation takes place, then integrity satisfaction should be restored by "retracting" the sentences explicitly indicated as retractible. Then, the derivation of some of the retractibles sanctions the responsibility in the integrity violation.

**Definition 3.1** Given an argumentation framework $\langle \mathcal{T}, \vdash, \mathcal{AB}, \mathcal{IC} \rangle$, where $\mathcal{IC}$ is a set of denial integrity constraints, each one with at least one retractible:

- a set of assumptions $\mathcal{A} \subseteq \mathcal{AB}$ **attacks** another set $\Delta \subseteq \mathcal{AB}$ iff for some integrity constraint $\neg[L_1 \wedge \ldots \wedge L_i \wedge \ldots \wedge L_n] \in \mathcal{IC}$ with $L_i$ retractible,
  $\mathcal{T} \cup \mathcal{A} \vdash \ L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n$, and
  $\mathcal{T} \cup \Delta \vdash L_i$.

Various notions of "acceptability" can be defined in terms of the same notion of attack. Here we mention only some of the notions presented in [4, 2, 1, 9]: A set of assumptions which does not attack itself is called **stable**, iff it attacks all assumptions it does not contain ($\mathcal{A} \subseteq \mathcal{AB}$ attacks $\delta \in \mathcal{AB}$ iff $\mathcal{A}$ attacks $\{\delta\}$); **admissible**, iff it attacks all sets of assumptions that attack it; **preferred**, iff it is maximally (with respect to set inclusion) admissible.

ALP can be given a semantics by appropriately instantiating any of these abstract semantics. Given an abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, the corresponding assumption-based framework is $\langle \mathcal{T}, \vdash, \mathcal{AB}, \mathcal{IC} \rangle$ where

- $\mathcal{T}$ is the set of all variable-free instances of clauses in $\mathcal{P}$;
- $\vdash$ is modus ponens for the clause implication symbol $\leftarrow$;
- $\mathcal{AB}$ is $\mathcal{AB}_{KM}$ together with the set of all variable-free negative literals;
- $\mathcal{IC}$ is the set consisting of ($A$ is a variable-free atom)
  (1) all denials of the form $\neg[A \wedge \mathit{not}\ A]$ with $\mathit{not}\ A$ retractible,
  (2) all denials of the form $\neg[A \wedge \mathit{not}\ A]$ with $A$ abducible and retractible, and
  (3) all variable-free instances of the integrity constraints in $\mathcal{I}_{KM}$, with all assumptions (i.e. abducibles or negative literals) retractible.

Note that, in the domain specific integrity constraints, literals that are not assumptions are not considered as retractible. This choice is made to conform to the intended meaning of retractibles in integrity constraints, as sentences that can be retracted to restore integrity satisfaction. In fact, it might not be possible to retract literals that are not assumptions, if these literals can be derived from the program without the addition of any assumption. In general, however, it might be convenient to allow the user to choose directly retractibles in the integrity constraints when defining the domain-specific abductive logic program (e.g. see [10]). In this case, the user should take care of choosing retractibles so that the choice conforms to their intended meaning.

Note also that, in the instance of the assumption-based framework for ALP, negative literals are treated as syntactic object, whose derivation solely depends on their assumption. Therefore, a theory $\mathcal{P}\cup\Delta$, for some set of assumptions $\Delta$ in the assumption-based framework for $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM}\rangle$, can be seen as a Horn program. This feature is made explicit in [5] and [8, 7] by explicitly renaming negative literals as positive atoms.

The instance for ALP of the notion of attack between sets of assumptions is:

- a set of assumptions $\mathcal{A}$ attacks another set $\Delta$ iff
  - (i) there exists an atom $Q$ such that $\mathcal{P} \cup \mathcal{A} \vdash Q$ and *not* $Q \in \Delta$, or
  - (ii) there exists an abducible atom $Q$ such that *not* $Q \in \mathcal{A}$ and $Q \in \Delta$, or
  - (iii) there exists a variable-free instance of an integrity constraint $\neg[L_1 \wedge \ldots \wedge L_i \wedge \ldots \wedge L_n] \in \mathcal{I}_{KM}$ with $L_i$ retractible such that $\mathcal{P} \cup \mathcal{A} \vdash L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n$ and $L_i \in \Delta$ [3].

In this concrete assumption-based framework, stable sets of assumptions correspond to generalised stable models (see [9]). Note that, since NLP is a special case of ALP, the semantics of NLP is a special instance of the semantics of ALP given by the assumption-based framework above. In the assumption-based framework for NLP, stable sets of assumptions correspond to stable models and preferred sets of assumptions correspond to partial stable models/preferred extensions [4, 2, 1].

# 4 Soundness result

**Theorem 4.1** Given an abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM}\rangle$ and a conjunction of literals $\mathcal{Q}$, if there is an abductive derivation from $(\leftarrow \mathcal{Q}, \emptyset)$ to $(\Box, \Delta)$, then, in the assumption-based framework for $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM}\rangle$,

1. $\mathcal{P}\cup\Delta \vdash \mathcal{Q}$, and

[3]This condition replaces the condition $\mathcal{P} \cup \Delta \vdash L_i$ which is equivalent to it due to the fact that all retractibles in $\mathcal{I}_{KM}$ are assumptions and assumptions are not defined in $\mathcal{P}$.

2. for each set of assumptions $\mathcal{A}$,
   if $\mathcal{A}$ attacks $\Delta$ (by cases (i) or (ii) or (iii)),
   then $\Delta$ attacks $\mathcal{A} - \Delta$ by case (i) or by case (ii).

The proof of this theorem can be found in the appendix.

The procedure is not complete with respect to the semantics expressed in theorem 4.1, due to the possible non-termination of the procedure for some queries.

Directly from the soundness result, the KM procedure allows domain-specific integrity constraints to be used to attack $\Delta$ but not to counter attack the attacks against $\Delta$. The following example illustrates the disadvantages of this asymmetric use of domain-specific integrity constraints.

**Example 4.1** Consider the propositional abductive logic program

$$\begin{array}{lcl} \mathcal{P} & = & \{q \leftarrow b, \;\; p \leftarrow a\} \\ \mathcal{AB}_{KM} & = & \{a, \, b\} \\ \mathcal{I}_{KM} & = & \{\neg[p \wedge b], \;\; \neg a, \;\; \neg \mathit{not}\ a\} \end{array}$$

There is no successful abductive derivation from $(\leftarrow q, \emptyset)$. In fact, in the corresponding assumption-based framework, the set $\{b\}$ (needed to prove $q$) is attacked by the set $\{a\}$, but $\{b\}$ does not counter attack the attack $\{a\}$ via case (i) or (ii). Moreover, the set $\{b, \mathit{not}\ a\}$, which counter attacks the attack $\{a\}$ via case (i) or (ii), is attacked by $\emptyset$, which can not be counter attacked via case (i) or (ii). However, $q$ should intuitively hold, since $p$ can not possibly hold. By allowing a symmetric use of domain-specific integrity constraints to counter attack, $\{b\}$ can be shown to be "acceptable", since $\emptyset$, and therefore $\{b\}$, counter attacks the attack $\{a\}$ against it via case (iii), and thus $q$ can be shown to hold.

One way to obtain the behaviour suggested in the example is to modify the KM procedure to allow a symmetric use of domain-specific integrity constraints. The following section illustrates an alternative way of obtaining the same behaviour, by transforming the given abductive logic program into a normal logic program first [10] and then applying an existing proof procedure for NLP, the EK procedure.

# 5 Comparison with the EK procedure

New proof procedures for ALP can be obtained by applying existing proof procedures for NLP to the normal logic programs $\mathcal{P}' = \mathcal{P}^{NAF} \cup \mathcal{P}^{I_{KM}} \cup \mathcal{P}^{AB_{KM}}$ obtained by applying the transformation of [10] to abductive logic programs $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, where, if for every abducible predicate $a$ in $\mathcal{AB}_{KM}$, $a'$ stands for the "complement" of $a$:

- $\mathcal{P}^{NAF}$ is $\mathcal{P}$ with all abducible atoms $a(t)$ replaced by a negative literal $\mathit{not}\ a'(t)$;

- for every $\neg[L_1 \wedge \ldots \wedge L_i \wedge \ldots \wedge L_n] \in \mathcal{I}_{KM}$ and every retractible $L_i$ in it, $\mathcal{P}^{I_{KM}}$ contains a clause
$$\alpha \leftarrow L'_1, \ldots, L'_{i-1}, L'_{i+1}, \ldots, L'_n$$
where for each $j = 1, \ldots, n$, $j \neq i$, if $L_j = a(t)$ with $a \in \mathcal{AB}_{KM}$ then $L'_j = not\ a'(t)$, otherwise $L'_j = L_j$, and
if $L_i = not\ a(t)$ then $\alpha = a(t)$, otherwise, if $L_i = a(t)$, then $\alpha = a'(t)$;

- for every abducible predicate $a$ in $\mathcal{AB}_{KM}$, $\mathcal{P}^{AB_{KM}}$ contains a pair of clauses
$$a(X) \leftarrow not\ a'(X)$$
$$a'(X) \leftarrow not\ a(X)$$

If we adopt Kakas and Mancarella's restriction, that all literals in domain-specific integrity constraints are either abducible atoms or the negation of abducible atoms, then it can be shown that the original abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$ and the transformed normal logic program $\mathcal{P}'$ are equivalent with respect to all semantics defined in the underlying assumption-based frameworks. This result is a corollary of the basic result that attacks before and after the transformation are preserved (see [9, 10] for more details). As a consequence, any proof procedure for NLP, sound with respect to an argumentation-theoretic semantics expressed in the assumption-based framework for NLP, provides a proof procedure for ALP, sound with respect to the same semantics but expressed in the assumption-based framework for ALP. In particular, since the EK procedure is sound with respect to the preferred extension semantics [3], by applying the EK procedure to the normal logic program $\mathcal{P}'$ corresponding to an abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, we obtain a procedure for ALP computing preferred sets of assumptions, which therefore treats symmetrically integrity constraints to attack and counter attack. This is illustrated by the following example.

**Example 5.1** The abductive logic program in example 4.1 is reduced to the normal logic program $\mathcal{P}'$

| | | |
|---|---|---|
| $q \leftarrow not\ b'$ | $b' \leftarrow p$ | $a \leftarrow not\ a'$ |
| $p \leftarrow not\ a'$ | $a'$ | $a' \leftarrow not\ a$ |
| | $a$ | $b \leftarrow not\ b'$ |
| | | $b' \leftarrow not\ b$ |

where $a'$ and $b'$ are new atoms standing for the "complement" of the abducibles $a$ and $b$, respectively. By applying the EK procedure to the program $\mathcal{P}'$, the query $\leftarrow q$ succeeds to be proved, with underlying negative assumption $not\ b'$, corresponding to the positive abducible $b$ with respect to $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$. In the proof of $\leftarrow q$, the attacks $\{not\ a'\}$ and $\{not\ b\}$ against $\{not\ b'\}$, corresponding to the attacks $\{a\}$ and $\{not\ b\}$ against $\{b\}$ with respect to $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, are counter attacked by $\emptyset$ and $\{not\ b'\}$, respectively. The attack $\emptyset$ against $\{not\ a'\}$ corresponds to the attack $\emptyset$ against $\{a\}$ via case (iii) with respect to $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$.

## 6 Conclusions

We have presented a soundness result for the Kakas-Mancarella proof procedure for ALP with respect to an argumentation-theoretic semantics. We have shown other semantics for ALP and pointed to proof procedures for their computation. These procedures are based upon transforming abductive logic programs into normal logic programs first, and then applying existing semantics for NLP to the result of the transformation. We have argued that the behaviour of the Eshghi-Kowalski procedure applied the transformed normal programs is preferable to the behaviour of the Kakas-Mancarella proof procedure applied to the original logic programs.

## Acknowledgements

This research was supported by the Fujitsu Research Laboratories. The author is grateful to Robert A. Kowalski for helpful suggestions and to Tony Kakas and Paolo Mancarella for helpful discussions.

## References

[1] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic framework for default reasoning. In preparation, 1995.

[2] A. Bondarenko, F. Toni, and R. A. Kowalski. An assumption-based framework for non-monotonic reasoning. In L.M. Pereira and A. Nerode, editors, *Proc. International Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 171–189, Lisbon, 1993. MIT press.

[3] P.M. Dung. Negation as hypothesis: an abductive foundation for logic programming. In K. Furukawa, editor, *Proc. ICLP*, pages 3–17, Paris, 1991. MIT Press.

[4] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming. In R. Bajcsy, editor, *Proc. IJCAI*, pages 852–857, Chambery, France, 1993. Morgan Kaufmann.

[5] K. Eshghi and R.A. Kowalski. Abduction compared with negation by failure. In G. Levi and M. Martelli, editors, *Proc. ICLP*, pages 234–255, Lisbon, 1989. MIT Press.

[6] A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.

[7] A.C. Kakas and P. Mancarella. Abductive logic programming. In W. Marek, A. Nerode, D. Pedreschi, and V.S. Subrahmanian, editors, *Proc. NACLP Workshop on Non-Monotonic Reasoning and Logic Programming*, Austin, Texas, 1990.

[8] A.C. Kakas and P. Mancarella. Generalised stable models: a semantics for abduction. In L. Carlucci Aiello, editor, *Proc. ECAI*, pages 385–391, Stockolm, 1990. Pitman.

[9] F. Toni. *Abductive logic programming.* PhD Thesis, Imperial College, London, 1995.

[10] F. Toni and R.A. Kowalski. Reduction of abductive logic programs to normal logic programs. To appear in *Proc. ICLP*, 1995.

# Appendix

First, we define partial orders $>$ and $\gg$ between (abductive and consistency) derivations, by appropriately modifying the definitions in [3]. Given an abductive derivation $\alpha$: $(G_1, \Delta_1^\alpha), \ldots, (\Box, \Delta_n^\alpha)$, a consistency derivation $\kappa$: $(L, \Delta_1^\kappa), \ldots, (\emptyset, \Delta_m^\kappa)$, where $n, m \geq 1$, and an assumption $A$,

$\kappa > \alpha$ via $A$ at step $i$, where $1 \leq i < m$, if $A$ is selected in the chosen goal in $S_i$ and a step of type (C4)(i) takes place, invoking $\alpha$,
i.e. $S_i = \{\leftarrow L', \mathcal{Q}\} \cup S'$, $\Re$ selects $L'$ in $\leftarrow L', \mathcal{Q}$, $L' = A$, $L' \notin \Delta_i^\kappa$, $G_1 = \leftarrow \overline{A}$, $\Delta_1^\alpha = \Delta_i^\kappa$, $S_{i+1} = S'$ and $\Delta_{i+1}^\kappa = \Delta_n^\alpha$;

$\alpha > \kappa$ via $A$ at step $i$, where $1 \leq i < n$, if $A$ is selected in $G_i$ and a step of type (A3) takes place, invoking $\kappa$,
i.e. $G_i = \leftarrow L', \mathcal{Q}$, $\Re$ selects $L'$ in $G_i$, $L' = A$, $L' \notin \Delta_i^\alpha$, $L = A$, $\Delta_1^\kappa = \Delta_i^\alpha \cup \{A\}$, $G_{i+1} = \leftarrow \mathcal{Q}$ and $\Delta_{i+1}^\alpha = \Delta_m^\kappa$;

$\gg$ is the transitive closure of $>$.

Let $\alpha_0$ denote the successful abductive derivation $(\leftarrow \mathcal{Q}, \emptyset)$ to $(\Box, \Delta)$ as in theorem 4.1. By definition of successful abductive derivation, $\alpha_0$ is finite and there exists a finite number of successful and finite derivations $\delta$ such that $\alpha_0 \gg \delta$.

**Lemma 6.1** For each derivation $\delta$: $(U_1, \Delta_1), \ldots, (U_n, \Delta_n)$ with $\alpha_0 \gg \delta$:
$\Delta_i \subseteq \Delta_j$, for all $i, j = 1, \ldots, n$ with $i < j$, and
$\Delta_i \subseteq \Delta$, for all $i = 1, \ldots, n$.

This lemma can be easily proved by induction on the number of derivations $\delta'$ such that $\delta \gg \delta'$ (see [9]).

Let $\mathcal{HB}_{not}$=$\{not\ A | A$ is an atom in the Herbrand base of $\mathcal{P}\}$. The **assumption set of an abductive derivation** $\alpha$: $(G_1, \Delta_1), \ldots, (G_n, \Delta_n)$ is the set $ass(\alpha) = \{A \mid \exists G_i$ such that $A \in G_i \cap (\mathcal{AB}_{KM} \cup \mathcal{HB}_{not})\}$. (We interpret goals, i.e. conjunctions of literals, as sets of literals.)

**Lemma 6.2** Given an abductive derivation $\alpha$:$(\leftarrow \mathcal{Q}', \Delta_1), \ldots, (\Box, \Delta_n)$ with $\alpha_0 \gg \alpha$,

1. $ass(\alpha) \subseteq \Delta$, and

2. $\mathcal{P} \cup ass(\alpha) \vdash \mathcal{Q}'$.

**Proof:**

1. By definition, each $A \in ass(\alpha)$ occurs as a subgoal in some $G_i$ in $\alpha$. By definition of abductive derivation, for each such $A$ there exists $\Delta_i$ such that $A \in \Delta_i$. Therefore, directly from Lemma 6.1, each such $A$ is in $\Delta$. As a consequence, $ass(\alpha) \subseteq \Delta$.

2. Directly from the definition of abductive derivation, $\mathcal{P} \cup ass(\alpha) \vdash_{SLD} \mathcal{Q}'$. By soundness of SLD resolution, $\mathcal{P} \cup ass(\alpha) \vdash \mathcal{Q}'$.

**Corollary 6.1** Given an abductive derivation $\alpha$: $(\leftarrow L, \Delta_1), \ldots, (\Box, \Delta_n)$ with $\alpha_0 \gg \alpha$ and $L$ an atom or the negation of an abducible atom, $\Delta$ attacks the assumption $\overline{L}$ via case (i) or case (ii).

Let the **assumption set of a consistency derivation** $\kappa$: $(L, \Delta_1), \ldots, (S_n, \Delta_n)$ be the set $ass(\kappa) = \{ass(\mathcal{B}) \,|\mathcal{B}$ is a branch of the tree $\mathcal{T}(\kappa)$ corresponding to $\kappa\}$, where the **assumption set of a branch** $\mathcal{B}$ in $\mathcal{T}(\kappa)$ is the set $ass(\mathcal{B}) = \{A \,| \exists G_i$ in $\mathcal{B}$ such that $A \in G_i \cap (\mathcal{AB}_{KM} \cup \mathcal{HB}_{not})\}$, and the **tree** $\mathcal{T}(\kappa)$ **corresponding to** $\kappa$ is $\mathcal{T}_n$, such that, assuming that for each $i = 1, \ldots, n$, $S_i = \{\leftarrow L_i, \mathcal{Q}_i\} \cup S_i'$, and for each $i = 1, \ldots, n-1$, $S_{i+1}$ is obtained by choosing the goal $\leftarrow L_i, \mathcal{Q}_i$ and by selecting the literal $L_i$ in it:

$\mathcal{T}_0$ consists only of the root, which is the empty goal;
for each $i \geq 1$, given $\mathcal{T}_i$ and $S_{i+1}$, then $\mathcal{T}_{i+1}$ is obtained as follows:

1. if $L_i$ is an atom and case (C1) applies, then $\mathcal{T}_{i+1}$ is $\mathcal{T}_i$ where all resolvents of $\leftarrow L_i, \mathcal{Q}_i$ on $L_i$ in $\mathcal{P}$ are children of $\leftarrow L_i, \mathcal{Q}_i$; if there is no such resolvent, failure is the only child of $\leftarrow L_i, \mathcal{Q}_i$;

2. if $L_i$ is an assumption and $L_i \in \Delta_i$, i.e. case (C2) applies, then $\mathcal{T}_{i+1}$ is $\mathcal{T}_i$ where $\leftarrow \mathcal{Q}_i$ is the only child of $\leftarrow L_i, \mathcal{Q}_i$;

3. if $L_i$ is an assumption and $\overline{L_i} \in \Delta_i$, i.e. case (C3) applies, then $\mathcal{T}_{i+1}$ is $\mathcal{T}_i$ where failure is the only child of $\leftarrow L_i, \mathcal{Q}_i$;

4. if $L_i$ is an assumption, $L_i \notin \Delta_i$ and $\overline{L_i} \notin \Delta_i$, i.e. case (C4) applies, then,
   if there exists an abductive derivation from $(\leftarrow \overline{L_i}, \Delta_i)$ to $(\Box, \Delta')$ (case (i)), then $\mathcal{T}_{i+1}$ is $\mathcal{T}_i$ where failure is the only child of $\leftarrow L_i, \mathcal{Q}_i$;
   otherwise (case (ii)), $\mathcal{T}_{i+1}$ is $\mathcal{T}_i$ where $\leftarrow \mathcal{Q}_i$ is the only child of $\leftarrow L_i, \mathcal{Q}_i$.

**Lemma 6.3** Given a consistency derivation $\kappa : (L, \Delta_1), \ldots, (\emptyset, \Delta_n)$ with $\alpha_0 \gg \kappa$, $ass(\kappa) \supseteq \{\Delta' | \Delta'$ is a minimal (with respect to set inclusion) attack against $L\}$.
**Proof:** By definition of consistency derivation, since all resolvents are considered at any step of kind (C1), $ass(\kappa)$ contains all sets of assumptions $\Delta'$ such that
$\mathcal{P} \cup \Delta' \vdash_{SLD} \overline{L}$, and
$\mathcal{P} \cup \Delta' \vdash_{SLD} L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_m$, for some $\neg[L_1 \wedge \ldots \wedge L_m]$ in $\mathcal{I}_{KM}$
with $L_i = L$ for some $i = 1, \ldots, m$.
But note that, for any (definite) program $\mathcal{P}'$ and query $\mathcal{Q}'$, if there exists a minimal

subset $\mathcal{P}''$ of $\mathcal{P}'$ such that $\mathcal{P}'' \vdash \mathcal{Q}'$ then $\mathcal{P}' \vdash_{SLD} \mathcal{Q}'$. As a consequence, directly from the definition of (minimal) attack, each minimal attack $\Delta'$ against $L$ belongs to $ass(\kappa)$.

**Lemma 6.4** For each consistency derivation $\kappa$ such that $\alpha_0 \gg \kappa$:

1. there exists an abductive derivation $\alpha$ such that $\alpha > \kappa$;
2. for any abductive derivation $\alpha$ such that $\alpha > \kappa$, via $L$ at step $i$, for each branch $\mathcal{B} \in \mathcal{T}(\kappa)$, if $ass(\mathcal{B})$ is a minimal attack against $L$ then for some $L' \in ass(\mathcal{B})$, (a) $\Delta$ attacks $L'$ via case (i) or case (ii), and (b) $L' \notin \Delta$.

**Proof:** Part 1. is obvious. Let us prove part 2. To prove (a), note that each $\mathcal{B} \in \mathcal{T}(\kappa)$ terminates with failure. If $ass(\mathcal{B})$ (minimally) attacks $L$, then, by completeness of SLD resolution, failure cannot be generated via case (C1). Therefore, failure must be generated via case (C3) or (C4)(i), due to the selection of an assumption $L'$ at some step $j$. (Note that necessarily $L' \in ass(\mathcal{B})$.) In the first case $\Delta$ attacks $L'$ via case (i) or via case (ii) trivially holds. In the second case, there must exist an abductive derivation $\alpha'$ such that $\kappa > \alpha'$ via $L'$. By definition, $\alpha'$ is a derivation from $(\leftarrow \overline{L'}, \Delta_j)$, where $\Delta_j$ is the set of assumptions accumulated at step $j$ in $\kappa$. By corollary 6.1, $\Delta$ attacks $L'$ via case (i) or via case (ii).

To prove (b), suppose by contradiction that $L' \in \Delta$. Necessarily, $L' \notin \Delta_j$, otherwise a step of type (C2) would have taken place. Therefore, there must exist an abductive derivation $\alpha''$ such that $L' \in ass(\alpha'')$ and $\alpha' \gg \alpha''$, and a consistency derivation $\kappa'$ such that $\alpha'' > \kappa'$ via $L'$. Since $\Delta$ attacks $L'$ (see (a)), by completeness of SLD resolution there must exist a branch $\mathcal{B} \in \mathcal{T}(\kappa')$ such that $ass(\mathcal{B}) = \Delta'$ for some $\Delta' \subseteq \Delta$, and $\mathcal{B}$ terminates with $\square$. As a consequence, $\alpha_0$ cannot terminate successfully. This gives a contradiction.

**Lemma 6.5** For all sets of assumptions $\mathcal{A}$ and $\Delta$ in the assumption-based framework corresponding to a KM abductive logic program $\langle \mathcal{P}, \mathcal{AB}_{KM}, \mathcal{I}_{KM} \rangle$, the following statements are equivalent:

1. if $\mathcal{A}$ attacks $\Delta$, then $\Delta$ attacks $\mathcal{A}- \Delta$;
2. if $\mathcal{A}$ is a minimal attack against $\Delta$, then $\Delta$ attacks $\mathcal{A}- \Delta$.

**Proof:** 1. $\Rightarrow$ 2. trivially holds. To prove 2. $\Rightarrow$ 1., note that if $\mathcal{A}$ is a non-minimal attack against $\Delta$, then there exists $\mathcal{A}' \subset \mathcal{A}$ such that $\mathcal{A}'$ is a minimal attack against $\Delta$. Then, by 2., $\Delta$ attacks $\mathcal{A}' - \Delta$. Therefore, $\Delta$ attacks $\mathcal{A}- \Delta$.

**Proof of theorem 4.1**: Part 1. holds by lemma 6.2. Part 2. can be proved as follows: by lemma 6.5 we only need to consider minimal attacks against $\Delta$. By definition of attack, for each (minimal) attack $\mathcal{A}$ against $\Delta$, there must exist $L \in \Delta$ such that $\mathcal{A}$ attacks $L$. Moreover, by construction of $\Delta$, for each such $L$, there must exist an abductive derivation $\alpha$ such that $\alpha_0 \gg \alpha$ and $L \in ass(\alpha)$, and a consistency derivation $\kappa$ such that $\alpha > \kappa$ via $L$. By lemma 6.3, $\mathcal{A} \in ass(\kappa)$. By lemma 6.4 $\Delta$ attacks $\mathcal{A} - \Delta$ via case (i) or via case (ii). This concludes the proof.