A semantics for positive abductive logic programs with implicative integrity constraints

Paolo Mancarella and **Francesca Toni**
Università di Pisa, Italy Imperial College London, UK
paolo@di.unipi.it ft@imperial.ac.uk

Abstract

We propose a novel semantics for positive abductive logic programs with *implicative integrity constraints*, namely integrity constraints in the form of implications. We show that this semantics is better suited to deal with several applications of abductive logic programming. Moreover, we prove that, in the propositional case, the existing abductive proof procedure IFF is sound and "strongly" complete w.r.t. the proposed semantics. Thus, we improve upon the existing "weak" completeness results for IFF.

Introduction

Abduction is a powerful mechanism for hypothetical reasoning with incomplete knowledge, that has found broad applications in artificial intelligence (Kakas, Kowalski, and Toni 1998; Denecker and Kakas 2002). This form of reasoning is handled by labeling some pieces of information as abducibles, i.e. as possible hypotheses, that can be assumed to hold provided that they are compatible with the available knowledge.

Abductive Logic Programming (ALP) combines abduction with standard logic programming, by assuming that the available knowledge is modelled as a logic program and abducibles are atoms not defined by the logic program. A number of *abductive proof procedures* have been proposed in the literature, e.g. (Kakas and Mancarella 1990; Console, Dupre, and Torasso 1991; Denecker and Schreye 1998; Fung and Kowalski 1997; Mancarella et al. 2009), to compute hypotheses/abducibles to explain observations seen as standard logic programming queries. These proof procedures allow the use of *integrity constraints* to restrict the range of possible hypotheses. Abductive proof procedures compute *abductive answers* to queries Q, meant to provide explanations for these Q: answers specify which abducibles have to be assumed to hold for Q to hold as well, while also validating the integrity constraints.

Integrity constraints can in principle be any logical formulas, but are more conventionally assumed to be in the form of "denials" and/or implications. ALP with implicative integrity constraints has been advocated as a useful knowledge representation mechanism to support several applications, including agents (Kowalski and Sadri 1999; Sadri and Toni 1999; Mancarella and Terreni 2003; Kakas et al. 2008), active databases (Sadri and Toni 1999) and automated repairing of web sites (Mancarella, Terreni, and Toni 2009). However, the current notion of abductive answer is not suitable to model implicative integrity constraints, when these are used for these applications. Indeed, this current notion allows to validate implicative integrity constraints by arbitrarily enforcing their premises (and, as a consequence, their conclusion) even when these premises have no reason to be enforced. For example, the integrity constraint $alarm \rightarrow run$, modeling the reactive behaviour of an agent, with $alarm$ and run both abducible[1], can be validated by arbitrarily abducing $alarm$, and as a consequence run. The resulting abductive answer is counter-intuitive (in the absence of other information) and gives unwanted behaviour. Interestingly, existing abductive proof procedures refrain from computing these counter-intuitive abductive answers. For instance, in the earlier example, IFF (Fung and Kowalski 1997) would compute the empty abductive answer. Indeed, IFF is shown to be "weakly complete" w.r.t. the current notion of abductive answer: IFF is only guaranteed to compute *a subset* of every such answer. Thus, the existing notion of abductive answers can be deemed to be "weak".

In this paper we give a novel notion of abductive answer, overcoming the limitations of the existing notion for implicative integrity constraints, and prove a "strong" completeness result for IFF. Namely we prove that IFF is guaranteed to compute every abductive answer in our novel sense. Moreover, we prove that IFF is still sound w.r.t. our new notion of abductive answer (as it was w.r.t. the old notion).

Our new notion of abductive answer is given in terms of a notion of *computation*, inspired by a corresponding notion recently proposed in (Liu et al. 2010) to understand answer set programming.

[1]When ALP is used to model agents, both observations and actions by agents are modelled as abducibles, see (Kowalski and Sadri 1999).

The paper is organised as follows. First, we give background on ALP and its existing semantics. Then, we discuss some examples, motivating the inadequacy of the existing semantics for ALP for a class of applications. In the main part of the paper, we propose our novel semantics for ALP, illustrate it for the motivating examples and prove some properties of this semantics, including a comparison with the existing semantics. We then prove that the IFF proof procedure for ALP is sound and complete w.r.t. our proposed semantics. Finally we conclude.

Background

An *abductive logic program* (ALP) (Kakas, Kowalski, and Toni 1998) is a tuple $\langle P, A, IC\rangle$ where:

- P is a *normal logic program*, namely a set of *clauses* of the form: $p \leftarrow l_1 \wedge \ldots \wedge l_n \quad (n \geq 0)$
  where p is an atom and each l_i is a literal, i.e. an atom a or the negation (as failure) $\neg a$ of an atom a. All variables in $p, l_1, \ldots, l_n$ are implicitly universally quantified over $p \leftarrow l_1 \wedge \ldots \wedge l_n$. We refer to p as the *head* and to $l_1 \wedge \ldots \wedge l_n$ as the *body* of the clause.
- A is a set of (ground) atoms, referred to as *abducibles*. The predicates of abducibles do not occur in the head of any clause of P (without loss of generality, see (Kakas, Kowalski, and Toni 1998)).
- IC is a set of *implicative integrity constraints* of the form: [2] $l_1 \wedge \ldots \wedge l_n \rightarrow p \quad (n \geq 0)$
  where p is an atom and each l_i is a literal. All variables in $p, l_1, \ldots, l_n$ are implicitly universally quantified over the implication. We refer to $l_1 \wedge \ldots \wedge l_n$ as the *body* and to p as the *head* of the integrity constraint.

We refer to the set of all predicates occurring in $\langle P, A, IC\rangle$ as the *signature* of $\langle P, A, IC\rangle$ and to all literals that can be built from predicates in the signature of $\langle P, A, IC\rangle$ as the *Herbrand base* of $\langle P, A, IC\rangle$, denoted $HB_{\langle P, A, IC\rangle}$. Clauses with an empty body ($n = 0$) will be represented as $p \leftarrow true$, with $true$ not already in $HB_{\langle P, A, IC\rangle}$. Integrity constraints with an empty body ($n = 0$) will be represented as $true \rightarrow p$.

A *query* Q to an ALP $\langle P, A, IC\rangle$ is a (possibly empty) conjunction of literals whose predicates belong to the signature of $\langle P, A, IC\rangle$. The variables in Q are implicitly existentially quantified, with scope the query. The emtpy query is represented as *true*.

Informally, given an ALP $\langle P, A, IC\rangle$ and a query Q, an "abductive answer" for a query Q is a set of (ground) abducibles Δ that, together with P, "entails" both Q and IC, w.r.t. some notion of "entailment". The notion of "entailment" depends on the semantics associated with the logic program P (there are many different possible choices for such semantics (Kakas, Kowalski, and Toni 1998)). Formally, an *abductive answer* to a query Q w.r.t. an ALP $\langle P, A, IC\rangle$ is a finite set Δ of abducibles such that, for some ground substitution σ for the variables of Q:

- $P \cup \Delta \models_{LP} Q\sigma$ and
- $P \cup \Delta \models_{LP} IC$

where $\models_{LP}$ stands for entailment w.r.t. the chosen semantics for logic programming.

In the remainder we will focus on *positive* ALPs and queries, where no negative literals occur. Note that in this case $\models_{LP}$ is necessarily entailment under the least Herbrand model, referred to below as $\models_{lhm}$. Moreover, as conventional in logic programming when defining semantics, we will assume that any ALP $\langle P, A, IC\rangle$ stands for its ground instantiation (w.r.t. $HB_{\langle P, A, IC\rangle}$), or, equivalently, that $\langle P, A, IC\rangle$ is propositional.

[2] Note that in some approaches to ALP, e.g. (Kakas, Kowalski, and Toni 1998), integrity constraints can also be *denials*, namely p can be $false$. Also, in some approaches to ALP, e.g. (Fung and Kowalski 1997), the head of integrity constraints can be a disjunction of atoms. For simplicity, we do not consider these other forms of integrity constraints.

Motivation

As mentioned in the introduction, ALP with implicative constraints has been advocated as a useful knowledge representation mechanism to support several applications. In this section, we show that the current notion of abductive answer is not suitable to model implicative integrity constraints, when these are used for the aforementioned applications.

Example 1 Let $\langle P, A, IC\rangle$ be

$$P = \{\}; \qquad A = \{a, b\}; \qquad IC = \{a \rightarrow b\}$$

In line with (Kowalski and Sadri 1999; Sadri and Toni 1999; Kakas et al. 2008), this could be used to determine the reactive behaviour of a hardware agent (robot) that, when a fire alarm goes off (a) should immediately evacuate the building in which it is situated (b). In addition, in line with (Sadri and Toni 1999), it could be used to represent an active rule over a database sanctioning that every employee (a) should have a social security number (b). Finally, in line with (Mancarella, Terreni, and Toni 2009), it could be used to represent a rule over a web site about books, that each book documented on the site (a) should have an author (b).

Consider three possible queries $Q_1 = true$, $Q_2 = b$, $Q_3 = a$. Then, given the earlier notion of abductive answer, $\{a, b\}$ is the only possible answer to Q_3, whereas $\{a, b\}$ and $\{b\}$ are alternative answers to Q_2 and $\{a, b\}$ and $\{\}$ are alternative answers to Q_1. However, for the applications mentioned earlier, $\{a, b\}$ is not an appropriate answer to $Q_1 = true$ and $Q_2 = b$. Indeed, this answer unnecessarily and arbitrarily contains a.

Example 2 Let $\langle P, A, IC\rangle$ be

$$P = \{p \leftarrow b\}; \qquad A = \{a, b\}; \qquad IC = \{a \rightarrow p\}$$

This simple $\langle P, A, IC\rangle$ could be used for example to represent the reactive behaviour of a software agent that should increase the amount held by a bank account (p) when this amount goes below some threashold (a). One way to do so may be to transfer some money from another account (b).

Consider again queries $Q_1 = true$, $Q_2 = p$, $Q_3 = a$. Intuitively, the abductive answers should be for Q_1: $\{\}$; for Q_2: $\{b\}$; for Q_3: $\{a, b\}$. However, $\{a, b\}$ is an additional abductive answer for Q_1 and Q_2 according to the earlier definition. This is counter-intuitive for the intended application.

In the next section we give a novel notion of abductive answer overcoming the limitations of the existing notion when used with implicative integrity constraints.

Revised abductive answers

Throughout this section we take as given a (propositional) positive ALP $\langle P,\ A,\ IC\rangle$ and a (propositional) positive query Q. We first give some preliminary definitions and notations, then define the notion of *r-abductive answer* in terms of *computations*, illustrate this notion, and give some properties for it.

Preliminary notions

We first define the notion of implicative integrity constraints "fired" by a set of abducibles. This notion is given in terms of the following notation:

Notation 1 For any $\Delta \subseteq A$,

$$M(\Delta) = \left\{x \in HB_{\langle P,\ A,\ IC\rangle} \mid P \cup \Delta \models_{lhm} x\right\}$$

Definition 1 Given $\Delta \subseteq A$ and a set of (implicative) integrity constraints S, the *integrity constraints in S fired by Δ* are given by

$$fired_S(\Delta) = \{\alpha \rightarrow \beta \in S | \alpha \subseteq M(\Delta) \cup \{true\}\}$$

As an illustration, given $IC = \{a \rightarrow p\}$ as in example 2 and $S = IC \cup \{true \rightarrow a\}$, $fired_S(\{a\}) = S$ and $fired_S(\{\}) = \{true \rightarrow a\}$. Also, given $S = \{a \wedge b \rightarrow p, c \rightarrow p, d \rightarrow e\}$, $fired_S(\{a, c, d\}) = \{c \rightarrow p, d \rightarrow e\}$ and $fired_S(\{a, b, c, d\}) = S$.

We then define the notion of relevant explanation of a conjunction of atoms, used in the definition of *r-abductive answer* both for given queries and heads of fired implicative integrity constraints. This definition is inspired by the notion of argument in (Dung, Kowalski, and Toni 2009).

Definition 2 Given $\langle P,\ A,\ IC\rangle$ and a conjunction of atoms X, $\mathcal{E} \subseteq A$ is a *relevant explanation* for X w.r.t. $\langle P,\ A,\ IC\rangle$ if and only if

- if $X = true$ then $\mathcal{E} = \{\}$
- if X is an atom, let T_X be a tree with nodes labelled by literals in $HB_{\langle P,\ A,\ IC\rangle}$ or by the symbol τ (not already occurring in $HB_{\langle P,\ A,\ IC\rangle}$), such that the root of T_X is labelled by X and for every node N
  - if N is a leaf then N is labelled either by an abducible or by τ;
  - if N is not a leaf and l_N is the label of N, then there is a clause $l_N \leftarrow b_1, \ldots, b_m \in P$ and either $m = 0$ and the child of N is τ or $m > 0$ and N has m children, labelled by $b_1, \ldots, b_m$ (respectively);

  then $\mathcal{E}$ is the set of all abducibles labelling the leaves of T_X;
- if X is a (non-empty) conjunction $l_1 \wedge \ldots \wedge l_n$ ($n > 0$) and $\mathcal{E}_{l_i}$ is a relevant explanation for l_i, then $\mathcal{E} = \mathcal{E}_{l_1} \cup \ldots \cup \mathcal{E}_{l_n}$.

Note that integrity constraints play no role in the definition of relevant explanation.

As an illustration, consider $\langle P,\ A,\ IC\rangle$ of example 2: here, $\{b\}$ is a relevant explanation of p, whereas $\{\}$ and $\{a, b\}$ are not.

Example 3 Consider $\langle P,\ A,\ IC\rangle$ with $P = \{p \leftarrow a \wedge q, q \leftarrow b \wedge r, r \leftarrow, q \leftarrow b \wedge c\}$ and $A = \{a, b, c\}$. Both $\{a, b\}$ and $\{a, b, c\}$ are relevant explanations of p.

Thus, relevant explanations may be non-minimal.

It is easy to see that relevant explanations correspond to SLD derivations:

Lemma 1 If $\mathcal{E} \subseteq A$ is a relevant explanation for a conjunction of atoms X then there exists a SLD derivation for X from $P \cup \mathcal{E} \cup \{true\}$.

Thus, by soundness of SLD resolution (and since *true* is assumed to hold):

Lemma 2 If $\mathcal{E}$ is a relevant explanation of a conjunction of atoms X then $P \cup \mathcal{E} \models_{lhm} X$.

Note that the converse of this lemma does not hold, e.g., in example 3, $P \cup \{a, b\} \models_{lhm} q$ but $\{a, b\}$ is not a relevant explanation of q. However, the following result holds:

Lemma 3 If $P \cup \Delta \models_{lhm} Q$ then there exists $\mathcal{E} \subseteq \Delta$ such that $\mathcal{E}$ is a relevant explanation of Q.

The following notation will be used to define the notion of explanation of (heads of) implicative integrity constraints (definition 3 below).

Notation 2 Given any $x \in HB_{\langle P,\ A,\ IC\rangle}$,

$$\mathcal{E}_P(x) = \{\mathcal{E} | \mathcal{E} \subseteq A \text{ is a relevant explanation of } x\}$$

Note that, if x admits no relevant explanation, then $\mathcal{E}_P(x)$ is empty, and if x admits $\{\}$ as a relevant explanation, then $\{\}$ belongs to $\mathcal{E}_P(x)$. Moreover, if $a \in A$, then $\mathcal{E}_P(a) = \{\{a\}\}$. As an illustrative example, given $\langle P,\ A,\ IC\rangle$ with $P = \{p \leftarrow a, p \leftarrow b, q \leftarrow c\}$ and $A = \{a, b, c\}$, then $\mathcal{E}_P(p) = \{\{a\}, \{b\}\}$, $\mathcal{E}_P(q) = \{\{c\}\}$, and $\mathcal{E}_P(r) = \{\}$.

Definition 3 Let $\alpha \rightarrow \beta$ be an implicative integrity constraint and S a set of implicative integrity constraints.

- $expl_P(\alpha \rightarrow \beta)$ (*explanation of $\alpha \rightarrow \beta$ w.r.t. P*) is defined as:

$$expl_P(\alpha \rightarrow \beta) = \begin{cases} \mathcal{E} \in \mathcal{E}_P(\beta) & \text{if } \mathcal{E}_P(\beta) \neq \{\} \\ undefined & \text{otherwise} \end{cases}$$

- $expl_P(S)$ (*explanation of S w.r.t. P*) is defined as:

$$expl_P(S) = \begin{cases} \bigcup_{x \in S} expl_P(x) & \text{if, } \forall x \in S, expl_P(x) \subseteq A \\ undefined & \text{otherwise} \end{cases}$$

Note that, if $expl_P(x) = undefined$ for some $x \in S$, then $expl_P(S) = undefined$. Note also that $expl_P$ returns *one* single relevant explanation, if one exists, for the head of each integrity constraint it receives in input. Thus, there is a non-deterministic choice underlying the definition of $expl_P$. As an illustration, in example 3, assuming $IC = \{true \rightarrow p\}$, both $expl_P(IC) = \{a, b\}$ and $expl_P(IC) = \{a, b, c\}$ are acceptable.

Computations and *r-abductive answer*s

Let $IC_Q = IC \cup \{true \rightarrow q | q \text{ is a conjunct in } Q\}$. Trivially, the following statements are equivalent (for the existing notion of abductive answer given in the Background section)

1. Δ is an abductive answer to Q w.r.t. $\langle P,\ A,\ IC\rangle$
2. Δ is an abductive answer to *true* w.r.t. $\langle P,\ A,\ IC_Q\rangle$

We will define the notion of *r-abductive answer* (see definition 5) in the context of $\langle P,\ A,\ IC_Q\rangle$.

Notation 3 Given a sequence $\Delta_0, \ldots, \Delta_i, \ldots$ of sets of abducibles ($\Delta_i \subseteq A$, for $i \geq 0$), we denote $\Delta_\infty = \bigcup_{i \geq 0} \Delta_i$.

Definition 4 A *computation* (for $\langle P,\ A,\ IC_Q\rangle$) is a sequence $\Delta_0, \ldots, \Delta_i, \ldots$ such that $\Delta_i \subseteq A$, for $i \geq 0$, $\Delta_0 = \{\}$, and the following properties are fulfilled:

- Monotonicity:
  $\Delta_{i-1} \subseteq \Delta_i$ for each $i > 0$
- Groundedness:
  $\Delta_i = expl_P(fired_{IC_Q}(\Delta_{i-1}))$ for each $i > 0$
- Convergence:
  $\Delta_\infty = expl_P(fired_{IC_Q}(\Delta_\infty))$

Definition 5 A finite $\Delta \subseteq A$ is a *revised abductive answer* (*r-abductive answer* in short) of a positive Q given $\langle P,\ A,\ IC\rangle$ if and only if $\Delta = \Delta_\infty$ for some computation $\Delta_0, \ldots, \Delta_i, \ldots$ for $\langle P,\ A,\ IC_Q\rangle$.

Groundedness of the computation ensures that the head of each integrity constraint that is fired "so far" can be derived from the *r-abductive answer*, specifically from a subset of this that is a relevant explanation for the head (by definition of $expl_P$). Convergence guarantees that all heads of integrity constraints that are fired can be derived from the *r-abductive answer*. Monotonicity of the computation guarantees that relevant explanations for (the heads of) integrity constraints already fired "so far" can only be enlarged during the computation. This is illustrated by the following example.

Example 4 Consider $P = \{p \leftarrow a, p \leftarrow a \wedge b, p \leftarrow d\}$, $A = \{a, b, c, d\}$, $IC = \{c \rightarrow p\}$ and $Q = c$ or p. Then,

$\{\}, \{a\}, \{a\}, \ldots$
$\{\}, \{a, b\}, \{a, b\}, \ldots$
$\{\}, \{a\}, \{a, b\}, \{a, b\}, \ldots$

are all computations, whereas

$\{\}, \{a\}, \{d\}, \{d\}, \ldots$
$\{\}, \{a\}, \{a, b\}, \{a\}, \{a\}, \ldots$

corresponding to changing relevant explanation for p from $\{a\}$ to $\{d\}$ and from $\{a, b\}$ to $\{a\}$, respectively, are not, since they do not fulfil the property of monotonicity. Moreover, $\{\}, \{a\}, \{a, d\}, \{a, d\}, \ldots$ is not a computation, as it does not fulfil the property of groundedness (since $\{a, d\}$ is not a relevant explanation for p).

Illustration

Let us illustrate the notion of *r-abductive answer* for the motivating examples given earlier in the paper.

Example 1 (revisited) $Q_1 = true$ and $Q_2 = b$ admit *r-abductive answer*s $\{\}$ and $\{b\}$ respectively, with computations (respectively):

$\{\}, \{\}, \ldots$
$\{\}, \{b\}, \{b\}, \ldots$

To see why $\{a, b\}$ is not a *r-abductive answer* for Q_2, observe that, in any computation for Q_2, $\Delta_1 = \{b\}$ necessarily (since this is the only possible relevant explanation of b). Since $fired_{IC_Q}(\{b\}) = \{\}$, then $\Delta_i = \Delta_1$ for all $i > 1$, Thus, $\Delta_\infty = \{b\}$ and $\{a, b\}$ cannot possibly be a *r-abductive answer*.

Finally, $\{a, b\}$ is a *r-abductive answer* for $Q_3 = a$ since $\{\}, \{a\}, \{a, b\}, \{a, b\}, \ldots$ is a computation.

Example 2 (revisited) $\{a, b\}$ is a *r-abductive answer* for $Q_3 = a$ since $\{\}, \{a\}, \{a, b\}, \{a, b\}, \ldots$ is a computation. Instead, $\{a, b\}$ is not a *r-abductive answer* for $Q_2 = b$ since the only possible computation in this case is $\{\}, \{b\}, \{b\}, \ldots$

If we extend P in example 2 to also include $p \leftarrow c$ with c added to A, then $Q_3 = a$ admits two *r-abductive answer*: $\{a, b\}$ and $\{a, c\}$. However, $\Delta = \{a, b, c\}$ is not a *r-abductive answer* for Q_3, since the only possible computations in this case are

$\{\}, \{a\}, \{a, b\}, \{a, b\}, \ldots$
$\{\}, \{a\}, \{a, c\}, \{a, c\}, \ldots$

Properties of *r-abductive answer*s

Every *r-abductive answer* is guaranteed to be an abductive answer in the old sense. Formally:

Theorem 1 Let Δ be a *r-abductive answer* for a positive query Q given a positive $\langle P,\ A,\ IC\rangle$. Then Δ is an abductive answer for Q given $\langle P,\ A,\ IC\rangle$ (w.r.t. $\models_{lhm}$).

Proof. By definition of *r-abductive answer*, there exists a computation $\Delta_0 = \{\}, \Delta_1, \ldots,$ with $\Delta = \Delta_\infty$. Then there exists $\Delta_Q \subseteq \Delta_1$ that is a relevant explanation for Q (since integrity constraints with a *true* body are all fired by $\{\}$), and, by lemma 2, $P \cup \Delta_Q \models_{lhm} Q$. Thus, by monotonicity of $\models_{lhm}$, $P \cup \Delta \models_{lhm} Q$. To prove that $P \cup \Delta \models_{lhm} IC$ we need to check that $P \cup \Delta \models_{lhm} h$ for each h such that $B \rightarrow h \in IC$ and $P \cup \Delta \models_{lhm} B$. But if $P \cup \Delta \models_{lhm} B$ then $B \rightarrow h \in fired_{IC_Q}(\Delta_i)$ for some $i > 0$ and some $\Delta_{B \rightarrow h} \subseteq \Delta_{i+1}$ is a relevant explanation for h. As a consequence, by lemma 2, $P \cup \Delta_{B \rightarrow h} \models_{lhm} h$ and, by monotonicity of $\models_{lhm}$, $P \cup \Delta \models_{lhm} h$. *qed*

Notice that an abductive answer may not be a *r-abductive answer*. For instance, in example 1, $\{a, b\}$

is an abductive answer but not a *r-abductive answer* for Q_1. However, if an abductive answer exists, a *r-abductive answer* is guaranteed to exist too. Formally:

Theorem 2 If there exists an abductive answer, w.r.t. $\models_{lhm}$, for a positive query Q given a positive $\langle P, A, IC\rangle$, then there exists a *r-abductive answer* for Q given $\langle P,\ A,\ IC\rangle$.

We have seen, in example 4, that relevant explanations for heads of fired integrity constraints can "grow" in computations. We now define a notion of "persistent" computation where such explanations cannot "grow" over computations. Naturally, these kinds of computations lend themselves better to be constructed by proof procedures for ALP, and indeed we will see that IFF constructs such computations.

Definition 6 A *persistent computation* (for $\langle P,\ A,\ IC_Q\rangle$) is a computation (for $\langle P,\ A,\ IC_Q\rangle$) fulfilling the following property

- Persistence of explanations:
  for each $x \in fired_{IC_Q}(\Delta_\infty)$, there exists one $\mathcal{E}_x \in expl_P(x)$ such that $\mathcal{E}_x \subseteq \Delta_i$ for all $i > k$ where k is the least integer such that $x \in fired_{IC_Q}(\Delta_k)$.

  For example 4, given either c or p as query:

  $\{\}, \{a, b\}, \{a, b\}, \{a, b\}, \ldots$

  is a persistent computation whereas

  $\{\}, \{a\}, \{a, b\}, \{a, b\}, \ldots$

  is a non-persistent computation.

Note that there could be multiple $\mathcal{E}_x$ fulfilling definition 6, as illustrated by the following example.

Example 5 Given $P = \{p \leftarrow a, p \leftarrow b, q \leftarrow a\}$, $A = \{a, b\}$, $IC = \{\}$ and $Q = p \wedge q$, the computation (w.r.t. $\langle P,\ A,\ IC_Q\rangle$) $\{\}, \{a, b\}, \{a, b\}, \ldots$ is persistent. Here, there are two relevant explanations ($\{a\}$, $\{b\}$) for (the head p of) $true \rightarrow p$ fulfilling definition 6.

The notion of persistent computation is sufficiently expressive so that we can restrict *r-abductive answer*s to be obtained from persistent computations. Indeed, for every non-persistent computation, there exists a persistent computation from which the same *r-abductive answer* can be obtained (and vice versa, trivially, since persistent computations are computations). Formally:

Lemma 4 Let $\Delta_0, \ldots, \Delta_i, \ldots$ be a non-persistent computation. Then, there exists a persistent computation $\Delta'_0, \ldots, \Delta'_i, \ldots$ such that $\Delta_\infty = \Delta'_\infty$.

Proof (Sketch). If $\Delta_0, \ldots, \Delta_i, \ldots$ is non-persistent then there exist at least one $x \in fired_{IC_Q}(\Delta_\infty)$ with at least two different relevant explanations $\mathcal{E}^1_x \neq \mathcal{E}^2_x$, both in $expl_P(x)$, such that $\mathcal{E}^1_x \subseteq \Delta_{k_1}$ and $\mathcal{E}^2_x \subseteq \Delta_{k_2}$ with $\Delta_{k_1} \subseteq \Delta_{k_2}$ in the computation. Assume that there is exactly one such x and exactly two such explanations $\mathcal{E}^1_x, \mathcal{E}^2_x$. (The case with $m > 1$ such xs and k_i explanations for each x ($k_i \geq 2$) is similar.) By monotonicity of computations, $\mathcal{E}^1_x \subset \mathcal{E}^2_x$. We can then obtain a persistent computation $\Delta'_0, \ldots, \Delta'_i, \ldots$ by replacing $\mathcal{E}^1_x$ in Δ_{k_1} with $\mathcal{E}^2_x$. Trivially, $\Delta_\infty = \Delta'_\infty$. *qed*

Correctness of IFF

In this section we show that our newly defined notion of *r-abductive answer* is a perfect fit for the existing IFF proof procedure for ALP, in the sense that IFF is sound and complete, in a "strong" sense, w.r.t. this notion. We first we describe the procedure, and then prove our soundness and completeness results.

The IFF proof procedure

We give here a simplified version, for *ground* and *positive* ALPs and queries, of the fully-fledged IFF proof procedure of (Fung and Kowalski 1997; Fung 1996). This procedure uses the *selective completion* of the logic program P w.r.t. the abducibles A, denoted $comp_A(P)$ and defined as the union of the completions of all the atoms in $HB_{\langle P,\ A,\ IC\rangle} \setminus A$. As conventional, the completion of an atom p such that $p \leftarrow D_1, \ldots, p \leftarrow D_k$ are all the clauses in P with head p ($k \geq 1$) is the *iff-definition* $p \leftrightarrow D_1 \vee \ldots \vee D_k$, and the completion of an atom p for which no clause in P has p as its head is $p \leftrightarrow false$ (where $false$ does not belong to $HB_{\langle P,\ A,\ IC\rangle}$).

Given $\langle P,\ A,\ IC\rangle$, an *IFF derivation* for a query Q is defined as a sequence of "goals", $G_1, \ldots, G_k$, such that $G_1 = Q \wedge IC$. These goals are disjunctions of *disjuncts*, which are conjunctions of the form [3]

$A_1 \wedge \ldots \wedge A_n \wedge I_1 \wedge \ldots \wedge I_m$

where $n, m \geq 0$, $n + m > 0$, the A_i are atoms, and the I_i are *implications*, with the same syntax as implicative integrity constraints. Each G_{i+1} ($1 \leq i < k$) is obtained from G_i by application of one of the inference rules defined below, using the notation $G\left[{}^{\varphi}/{\psi}\right]$ to denote the goal obtained from goal G by replacing a conjunct ψ in it with φ.

Unfolding an atomic conjunct: given $p \leftrightarrow D_1 \vee \ldots \vee D_m$ in $comp_A(P)$ and an atom p which is a conjunct of a disjunct G in G_i, then G_{i+1} is G_i with G replaced by $\bigvee_{j=1}^{m} G\left[{}^{D_j}/{p}\right]$

Unfolding an atom in the body of an implication: given $p \leftrightarrow D_1 \vee \ldots \vee D_m$ in $comp_A(P)$ and an implication $[l_1 \wedge \ldots \wedge l_j \wedge \ldots \wedge l_k \rightarrow q]$ which is a conjunct of a disjunct G of G_i with $l_j = p$, then G_{i+1} is G_i with the implication in G replaced by the conjunction $\bigwedge_{s=1}^{m} [l_1 \wedge \ldots \wedge D_s \wedge \ldots \wedge l_k \rightarrow q]$

Propagation: given an atom p and an implication $[imp = l_1 \wedge \ldots \wedge l_j \wedge \ldots \wedge l_k \rightarrow q]$ with $l_j = p$, both conjuncts of the same disjunct G in G_i, if

$$imp' = \begin{cases} l_1 \wedge \ldots l_{j-1} \wedge l_{j+1} \wedge \ldots \wedge l_k \rightarrow q & \text{if } k > 1 \\ q & \text{if } k = j = 1 \end{cases}$$

then $G_i\left[{}^{imp}/{imp'}\right]$

[3] These disjuncts are simplified versions of the *simple disjuncts* of the original IFF, that may also include disjunctions as additional conjuncts. By merging splitting into other rules, discussed below, we do not need general simple goals.

Logical simplification replaces, within disjuncts:

$B \wedge true$ or $true \wedge B$ or $true \rightarrow B$ by B

$B \wedge false$ or $false \wedge B$ by $false$

$false \rightarrow B$ by $true$

In this variant of IFF we do not explicitely use the splitting rule, which distributes disjunctions over conjunctions. In the original IFF (Fung and Kowalski 1997) splitting was introduced as a separate inference rule, but, at the same time, its systematic use as a rule with higher priority was suggested, in order to simplify the overall procedure. In our variant, splitting is directly incorporated into the *unfolding* rule which is the only rule that can potentially introduce disjunctions within disjuncts in the case of ground positive ALPs.

Note that we have not included the simplification rules for disjunction, as disjunction never occurs in disjuncts, given that splitting is implicitly applied within unfolding. Note also that we have not included the simplification rules involving negation, nor the negation elimination rule as we are considering positive ALPs and queries. Further, we do not include inference rules such as factoring and case analysis, since they have to do with non-propositional ALPs and queries.

Finally, notice also that Fung and Kowalski define the propagation rule so that G_{i+1} is obtained by conjoining imp' to G_i (rather than replacing imp in G_i with imp' as we have done), and associate a *propagation history* with atoms in the body of implications in disjuncts, in order to avoid applying the same propagation step to the same implication and atom (see page 67 of (Fung 1996)). Our propagation rule renders this propagation history unnecessary. Moreover, it prevents the same integrity constraint to be propagated with several times unnecessarily, as in the following example.

Example 6 Consider $\langle P, A, IC\rangle$ with $P = \{p \leftarrow c, p \leftarrow d\}$, $IC = \{a \wedge b \rightarrow p\}$ and $A = \{a, b, c, d\}$. Consider $Q = a \wedge b$. Our variant of IFF computes

$G_1 = Q \wedge IC$

$G_2 = a \wedge b \wedge [a \rightarrow p]$ (by propagation)

$G_3 = a \wedge b \wedge p$ (by propagation)

$G_4 = [a \wedge b \wedge c] \vee [a \wedge b \wedge d]$ (by unfolding).

Instead, the original formulation of IFF may compute

$G'_1 = Q \wedge IC$

$G'_2 = a \wedge b \wedge IC \wedge [a \rightarrow p]$ (by propagation)

$G'_3 = a \wedge b \wedge IC \wedge [a \rightarrow p] \wedge [b \rightarrow p]$ (by propagation)

$G'_4 = [a \wedge b \wedge IC \wedge [a \rightarrow p] \wedge [b \rightarrow p] \wedge p$ (by propagation with $a \rightarrow p$)

$G'_5 = [a \wedge b \wedge IC \wedge [a \rightarrow p] \wedge [b \rightarrow p] \wedge p \wedge p$ (by propagation with $b \rightarrow p$)

$G'_6 = [\ldots c \wedge p] \vee [\ldots d \wedge p]$ (by unfolding the first occurrence of p)

$G'_7 = [\ldots c \wedge c] \vee [\ldots c \wedge d] \vee [\ldots d \wedge p]$ (by unfolding p in the first disjunct)

Given an IFF derivation $G_1, \ldots, G_n$ for a query Q, let G be a disjunct of G_n. G is called

conclusive if no inference rule can be applied to G;

failed if $false$ is a conjunct in G;

successful if G is conclusive and not failed.

Then, an IFF derivation $G_1, \ldots, G_n$ is *successful* if and only if there exists a successful disjunct G in G_n. An *answer extracted from* a successful IFF-derivation $G_1, \ldots, G_n$ for a query Q is the set of all abducible atoms in a successful disjunct G in G_n.

In the propositional case, our variant of IFF (notably with the simplified propagation rule) is trivially equivalent to the original IFF, in the sense that every answer computed by our variant is also computed by the original IFF, and (some subset of) every answer computed by the original IFF is computed by ours.

Correctness results for IFF

Theorem 3 (Soundness of IFF)
Given $\langle P, A, IC\rangle$, let Δ be an answer extracted from a successful IFF-derivation for a query Q. Then Δ is a *r-abductive answer* for Q given $\langle P, A, IC\rangle$.

Proof (Sketch). We first define inductively a construction from an IFF derivation $G_1 = Q \wedge IC, \ldots, G_n$ to a sequence $S_1, \ldots, S_n$ where each S_i is a set of forests of trees, each forest corresponding to a disjunct in G_i. We then define an order $\leq$ over trees in the forest F corresponding to the node of G_n from which Δ is extracted. All trees in F are "complete", in that they have abducibles or *true* as their leaves. Basically, a tree is ordered before another if it has become "complete" before the other in the construction of F in the sequence $S_1, \ldots, S_n$. The resulting order has a top element T_k (since the IFF derivation is finite). Finally, we map F onto a computation $\Delta_0, \ldots, \Delta_i, \ldots$ such that $\Delta_0 = \{\}$, for $0 < i \leq k$, Δ_i is the union of all sets of abducibles at the leaves of trees with i-th position w.r.t. $\leq$, and for $j > k$, $\Delta_j = \Delta_k$. *qed*

We illustrate this result in the case of example 6, for the answer $\{a, b, c\}$ extracted from the first disjunct in G_4, given derivation $G_1, \ldots, G_4$. The corresponding computation is $\{\}, \{a, b\}, \{a, b, c\}, \{a, b, c\}, \ldots$, obtained from $S_1, \ldots, S_4$ where S_4 consists of two forests, one of which consists of three trees, T_a, T_b and T_p, with, respectively: root (and leaf) a, root (and leaf) b, and root p with child (and leaf) c. The order $\leq$ is such that $T_a{=}T_b < T_p$ (with T_p the top element). Note that the resulting computation is persistent.

We prove completeness for *persistent r-abductive answer*, namely *r-abductive answer* obtained from persistent computations. Then, by lemma 4, completeness holds for any computation.

Theorem 4 (Completeness of IFF)
Let Δ be a persistent *r-abductive answer* for a query Q, given $\langle P, A, IC\rangle$. Then, Δ is an answer extracted from a successful IFF-derivation for Q.

Proof (Sketch). If Δ is a persistent *r-abductive answer* for Q, then there exist a persistent computation $\Delta_0, \ldots, \Delta_i, \ldots$ such that $\Delta = \Delta_\infty$. It is easy to see that, if ic is fired by Δ_i, then there is an SLD derivation for its body, from $P \cup \Delta_i \cup \{true\}$. Moreover, if the head of

ic can be explained, then by lemma 1, there is an SLD derivation for this head, from $P \cup \Delta_{i+1} \cup \{true\}$. It is also easy to see that SLD derivations can be mapped onto IFF derivations. All these IFF derivations can be combined into a single successful IFF derivation (including suitable steps corresponding to "firing") from which Δ can be extracted. *qed*

Conclusions

We have defined a new notion of abductive answer for positive ALPs with implicative integrity constraints that is better suited to a class of applications of ALP and provides a "better fit" than the existing notion for the IFF abductive proof procedure. Our new notion is defined in terms of *relevant explanations*, adapted from the notion of argument in (Dung, Kowalski, and Toni 2009), and a notion of *computation*, adapted from a corresponding notion in answer set programming (Liu et al. 2010). In particular, our monotonicity is the same as the notion of "persistence of beliefs" in (Liu et al. 2010) and our groundedness corresponds to the notion of "revision" in (Liu et al. 2010), but, whereas revision there amounts to obtaining each element in the computation by applying the standard logic programming T_P operator to the previous element, in our case groundedness amounts to obtaining each element in the computation by adding relevant explanations for the head of newly fired integrity constraints. The notion of convergence is also present in (Liu et al. 2010), but again defined in terms of T_P rather than $expl_P(fired_{IC_Q})$ as in our case. Finally, our persistence of explanations corresponds to the "persistence of reasons" in (Liu et al. 2010), but there this notion amounts to making sure that the same rules guarantee the derivation of atoms over (their kind of) computations.

Inoue and Sakama (Inoue and Sakama 1996) also propose a fixpoint semantics for abductive logic programming, based upon their rewriting as disjunctive logic programs and the use of (a suitable) T_P operator. Their semantics agrees with ours in some example, e.g. example 2, but does not enforce relevance of explanations (in our sense) in general. The formal relationships between our approach and the approach of (Inoue and Sakama 1996) deserves further study.

The applications that have inspired our approach use implicative integrity constraints to *determine* behaviour (e.g. of agents, or database or web management systems, see examples 1 and 2). It would be interesting to study whether our approach would be suitable to *explain* behaviour.

We have restricted attention to positive ALPs and queries, and omitted (for lack of space) to consider denials. Future work includes considering negation in ALPs and queries and denials alongside implicative integrity constraints.

We have studied soundness and completeness of IFF in the propositional case and for positive ALPs and queries. Future work is needed to consider the non-propositional case and negation, in particular the NAF extension of IFF given in (Sadri and Toni 1999). Moreover, it would be interesting to consider other abductive proof procedures that use implicative integrity constraints, e.g. the variant (Mancarella and Terreni 2003) of the procedure of (Kakas and Mancarella 1990).

References

Console, L.; Dupre, D. T.; and Torasso, P. 1991. On the relationship between abduction and deduction. *Journal of Logic and Computation* 1(5):661–690.

Denecker, M., and Kakas, A. C. 2002. Abduction in logic programming. In *Comp. Log.: LP and Beyond*, volume 2407 of *LNCS*, 402–436. Springer.

Denecker, M., and Schreye, D. D. 1998. SLDNFA: an abductive procedure for abductive logic programs. *J. Log. Progr.* 34(2):111–167.

Dung, P.; Kowalski, R.; and Toni, F. 2009. Assumption-based argumentation. In *Argumentation in AI: The Book*. Springer. 199–218.

Fung, T. H., and Kowalski, R. A. 1997. The IFF proof procedure for abductive logic programming. *J. Log. Progr.* 33(2):151–165.

Fung, T. H. 1996. *Abduction by deduction.* Ph.D. Dissertation, Imperial College, University of London.

Inoue, K., and Sakama, C. 1996. A fixpoint characterization of abductive logic programs. *J. Log. Progr.* 27(2):107–136.

Kakas, A. C., and Mancarella, P. 1990. Abductive logic programming. In *Proc. LPNMR*, 49–61.

Kakas, A. C.; Mancarella, P.; Sadri, F.; Stathis, K.; and Toni, F. 2008. Computational logic foundations of KGP agents. *J. of Artificial Intelligence Research.*

Kakas, A.; Kowalski, R.; and Toni, F. 1998. The role of abduction in logic programming. In *Handbook of Logic in AI and LP*, volume 5. OUP. 235–324.

Kowalski, R. A., and Sadri, F. 1999. From logic programming towards multi-agent systems. *Annals of Mathematics and AI* 25(3/4):391–419.

Liu, L.; Pontelli, E.; Son, T. C.; and Truszczynski, M. 2010. Logic programs with abstract constraint atoms: The role of computations. *Artificial Intelligence.* Forthcoming.

Mancarella, P., and Terreni, G. 2003. An abductive proof procedure handling active rules. In *Proc. AI*IA*, volume 2829 of *LNCS*, 105–117. Springer.

Mancarella, P.; Terreni, G.; Sadri, F.; Toni, F.; and Endriss, U. 2009. The CIFF proof procedure for abductive logic programming with constraints: Theory, implementation and experiments. *TPLP* 9:691–750.

Mancarella, P.; Terreni, G.; and Toni, F. 2009. Web sites repairing through abduction. *Electr. Notes Theor. Comput. Sci.* 235:137–152.

Sadri, F., and Toni, F. 1999. Abduction with negation as failure for active and reactive rules. In *Proc. AI*IA*, volume 1792 of *LNCS*, 49–60. Springer.