# Abduction, ASP and Open Logic Programs

**P. A. Bonatti**
Dipartimento di Tecnologie dell'Informazione
Università di Milano
bonatti@dti.unimi.it

## Abstract

Open logic programs and open entailment have been recently proposed as an abstract framework for the verification of incomplete specifications based upon normal logic programs and the stable model semantics. There are obvious analogies between open predicates and abducible predicates. Their extension is not specified in the program. However, despite superficial similarities, there are features of open programs that have no immediate counterpart in the framework of abduction and viceversa. Similarly, open programs cannot be immediately simulated with answer set programming (ASP).

In this paper we start a thorough investigation of the relationships between open inference, abduction and ASP. We shall prove that open programs generalize the other two frameworks. Similarities and differences between the three frameworks will be analyzed formally. The generalized framework suggests interesting extensions of abduction under the generalized stable model semantics. In some cases, we will be able to reduce open inference to abduction and ASP, thereby characterizing the computational complexity of credulous and skeptical open inference for finite, function-free open programs. At the same time, the aforementioned reduction opens the way to new applications of abduction and ASP.

## 1 Introduction

Open logic programs and open entailment [Bonatti 2001] have been recently proposed as an abstract framework for the verification of incomplete specifications based upon normal logic programs and the stable model semantics.

An important example of incomplete specifications is given by compound security policies [Bonatti et al. 2000], where details such as the set of users and the formulation of certain subpolicies are tipically unknown at verification time. In this setting, it is interesting to verify whether the policy will necessarily satisfy privacy laws, all parties' requirements, etc.

Logic-based agents are a second important example. In IMPACT [Subrahmanian et al. 2000], agent programs must satisfy a property called *conflict freedom*. This property depends on a variable state that extends the agent program. The state is unknown at verification time, and can be regarded as a runtime extension of the agent program. Conflict freedom should hold for all possible such extensions.

Incomplete specifications are modelled by identifying a set of predicates, called *open predicates*, that are not completely defined in the program. Different forms of entailment capture what could possibly be true and what must necessarily be true, across a space of possible complete specifications of the open predicates.

There are obvious analogies between open predicates and abducible predicates. Their extension is not specified in the program. In the framework of abduction, a suitable definition of abducible predicates must be found as part of the reasoning task, while in the framework of open programs one may also quantify over all possible definitions of the open predicates (skeptical open inference). Moreover, abduction looks for simple definitions of the abducible predicates, consisting of ground facts, while the framework of open programs admits arbitrary definitions. It should be pointed out that arbitrary rules can be abduced by introducing new abducibles, but the point is whether the result-

ing abduction framework must necessarily be infinite when the set of abducible rules is.

Thus, despite superficial similarities, there are features of open programs that cannot be immediately reproduced in the framework of abduction and viceversa. Similarly, answer set programming (ASP) cannot trivially simulate open programs, because of the complex structure of some forms of open inference (called *mixed* inference), and because of the simultaneous treatment of diverse Herbrand domains, featured by open program semantics.

In this paper we start a thorough investigation of the relationships between open inference, abduction and ASP. We shall consider the abduction frameworks originated by [Eshghi and Kowalski 1989], refined in [Kakas and Mancarella 1990], and further studied – from a procedural standpoint – in [Satoh and Iwayama 1992]. The semantics is based upon *generalized stable models*, due to [Kakas and Mancarella 1990]. The main contributions of the paper are the following.

- We shall prove that open programs generalize both ASP and the generalized stable model semantics. A related benefit is that the applicability range of abduction and ASP is potentially extended to the intended applications of open programs (under some restrictions). As a second benefit, the relationships between the three frameworks open the way to a cross-fertilization of the proof procedures for abduction, ASP and open inference. Moreover, the generalized framework suggests some interesting extensions:
  - The generalized stable model semantics does not allow to abduce the existence of new individuals. Open logic programs provide such extension, previously supported only under 3-valued completion semantics [Denecker and De Schreye 1998] and stratified negation [Shanahan 1989]. Furthermore, open logic programs can express upper bounds on domain cardinality. None of the existing approaches handles such constraints.
  - The existing abductive procedures for the generalized stable model semantics do not support nonground inference. Open inference is a first step toward nonground abduction under the generalized stable model semantics. Moreover, open inference suggests new computation strategies, e.g., based on lemma generation.
  - The *mixed* inferences introduced in the open framework suggest a new form of abduction, that stands to standard abduction as *diagnosis as entailment* stands to *diagnosis as satisfaction*.
- In some cases, we will be able to reduce open inference to ASP. As a result, we characterize the complexity of two forms of open inference for finite and function-free open programs. Moreover, we show how ASP can be used to compute open inference in this special case. A complete embedding will be left as an open issue.
- The (partial) embedding of open programs into ASP makes it possible to use ASP engines such as Smodels [Niemelä and Simons 1997] for abductive reasoning under the open version of the generalized stable model semantics.

The basic definitions concerning abduction frameworks and open programs will be recalled in Section 2. Then we will study the relationships between open programs and abduction frameworks in Section 3. In Section 4, credulous and skeptical open inferences are embedded into standard credulous and skeptical inferences in ASP. The main results are summarized and discussed in Section 5.

## 2 Preliminaries

We assume the reader to be familiar with normal logic programs and the stable model semantics [Gelfond and Lifschitz 1988]. We say that a normal logic program is *consistent* if it has at least one stable model.

An *abduction framework* is a pair $\langle T, A \rangle$, where $T$ is a normal logic program and $A$ is a set of *abducible predicates*. Let $\mathsf{Abducibles}(T, A)$ be the set of all ground atoms $p(t_1, \ldots, t_n)$ such that $p \in A$ and $t_i$ belongs to the Herbrand domain of $T$ $(1 \leq i \leq n)$.

**Definition 2.1** A *generalized stable model* of an abduction framework $\langle T, A \rangle$ is a stable model of $T \cup E$, for some $E \subseteq \mathsf{Abducibles}(T, A)$. ∎

An *open program* is a triple $\langle P, F, O \rangle$ where $P$ is a normal logic program, $F$ is a set of function and constant symbols not occurring in $P$, and $O$ is a set of predicate symbols. A *completion*[1] of $\langle P, F, O \rangle$ is a normal logic program $P'$ such that

[1] Note that this notion of completion has nothing to do with the standard notion of completion derived from Clark's work. We have not been able to identify an alternative term to express correctly the idea of complete predicate specification.

1. $P' \supseteq P$,
2. the constant and function symbols of $P'$ occur in $P$ or $F$,
3. if $r \in P' \setminus P$, then $head(r) \in O$.

The set of all the completions of $\langle P, F, O \rangle$ is denoted by $\mathsf{Comp}(P, F, O)$. There exist four kinds of *open inference*:

1. (Credulous open inference) $\langle P, F, O \rangle \models^c \Psi$ iff for some $P' \in \mathsf{Comp}(P, F, O)$, $P'$ credulously entails $\Psi$.
2. (Skeptical open inference) $\langle P, F, O \rangle \models^s \Psi$ iff for all $P' \in \mathsf{Comp}(P, F, O)$, $P'$ skeptically entails $\Psi$.
3. (Mixed open inference I) $\langle P, F, O \rangle \models^{cs} \Psi$ iff for some consistent $P' \in \mathsf{Comp}(P, F, O)$, $P'$ skeptically entails $\Psi$.
4. (Mixed open inference II) $\langle P, F, O \rangle \models^{sc} \Psi$ iff for each consistent $P' \in \mathsf{Comp}(P, F, O)$, $P'$ credulously entails $\Psi$.

The four open entailments are pairwise dual and form a diamond-shaped lattice [Bonatti 2001].

**Proposition 2.2** *(Duality) For all open programs $\Omega$ and all sentences $\Psi$,*

1. $\Omega \models^c \Psi$ *iff* $\Omega \not\models^s \neg\Psi$;
2. $\Omega \models^{sc} \Psi$ *iff* $\Omega \not\models^{cs} \neg\Psi$.

**Proposition 2.3** *(Entailment lattice) Suppose there exists a consistent $P' \in \mathsf{Comp}(\Omega)$. Then, for all sentences $\Psi$,*

1. $\Omega \models^s \Psi$ *implies* $\Omega \models^{cs} \Psi$ *and* $\Omega \models^{sc} \Psi$;
2. $\Omega \models^{cs} \Psi$ *implies* $\Omega \models^c \Psi$;
3. $\Omega \models^{sc} \Psi$ *implies* $\Omega \models^c \Psi$.

## 3 Open programs and abduction

Abducible predicates are similar to open predicates, as their extension is not specified in the program. However, open predicates can be given arbitrarily complex definitions (including all sorts of cycles, possibly involving non-open predicates), while abducible predicates are always defined by sets of ground facts (cf. the clause $E \subseteq \mathsf{Abducibles}(T, A)$ in Definition 2.1).

Therefore, in order to relate the two approaches, we need a representation lemma showing that a similar restriction can be posed on the completions of open predicates, without loss of generality.

**Lemma 3.1** *If $M$ is a stable model of $P' \in \mathsf{Comp}(P, F, O)$, then $M$ is a stable model of some $P'' \in \mathsf{Comp}(P, F, O)$ such that $P'' \setminus P$ is a set of ground facts.*

**Proof.**(Sketch) Let $P'' = P \cup \{p(\vec{t}) \in M \mid p \in O\}$. It can be verified that $M$ is also a stable model of $P''$. Moreover, $P'' \setminus P$ is a set of ground facts by definition. ∎

A rule $r = H \leftarrow B$ could also be abduced by introducing a new abducible $n_r$ (that plays the role of $r$'s name), and inserting $H \leftarrow B, n_r$ in the program (cf. [Poole 1988]). Clearly, if arbitrary rules are to be abduced (as it happens in open programs), then this method leads necessarily to an infinite abduction framework (more precisely, both the program and the set of abducibles are infinite). Lemma 3.1 improves rule abduction by showing that in fact one needs not add any new symbols and rules, provided that no constraint is posed on the space of abducible rules.

Now the basic correspondence between abduction and open programs follows easily.

**Theorem 3.2** *$M$ is a generalized stable model of an abductive framework $\langle T, A \rangle$ iff $M$ is a stable model of some $P \in \mathsf{Comp}(T, \emptyset, A)$.*

The relationships between abductive frameworks and open programs suggest two natural extensions of abductive reasoning under the generalized stable model semantics. First note that the set $F$ of possible function symbols for the completions is empty in Theorem 3.2. By allowing a nonempty $F$, the generalized stable model semantics can be given the ability of assuming the existence of new individuals and functions (as in [Denecker and De Schreye 1998] and [Shanahan 1989]). Typically, in the framework of reasoning about action and change, this feature is applied to explain a sequence of events with unknown actions. Abductive frameworks can be extended accordingly, by a straightforward generalization of $\mathsf{Abducibles}(\cdot, \cdot)$.

**Definition 3.3** For each abduction framework $\langle T, A \rangle$, let $\mathsf{Abducibles}^o(T, A)$ be a set of ground atoms $p(t_1, \ldots, t_n)$ such that $p \in A$ and $t_i$ is a term built from the function and constant symbols of $T$, plus a denumerable set $Sk$ of (skolem) constants not occurring in $T$. $\mathsf{Abducibles}^o(T, A)$ will be called a set

of *open abducibles* of the abduction framework $\langle T, A \rangle$ (w.r.t. the set of skolem constants $Sk$). ∎

Clearly, the choice of $Sk$ is irrelevant, as long as it does not intersect the vocabulary of $T$.

**Definition 3.4** An *open generalized stable model* of an abduction framework $\langle T, A \rangle$ is a stable model of $T \cup E$, for some $E \subseteq \mathsf{Abducibles}^o(T, A)$. ∎

By analogy with the previous theorem we have:

**Proposition 3.5** *$M$ is an open generalized stable model of an abductive framework $\langle T, A \rangle$ iff $M$ is a stable model of some $P \in \mathsf{Comp}(T, Sk, A)$, where $Sk$ is the set of skolem constants adopted in $\mathsf{Abducibles}^o(T, A)$.*

The second extension concerns inference modalities. Mixed inference of type I suggests a new form of abduction that can be understood by analogy with the two main forms of diagnosis, based on satisfaction and entailment, respectively. More precisely, diagnostic reasoning can be carried out either by looking for a model of the domain theory $T$ where the observations $Q$ are *satisfied*, or by finding a set of explanations $E$ such that $T \cup E$ *entails* $Q$. The two kinds of diagnosis can be regarded as the analogues of standard generalized stable model computation (which is equivalent to credulous open reasoning, by Theorem 3.2) and mixed open inference of type I, respectively. The latter can be reformulated for abductive frameworks as follows.

**Definition 3.6** A sentence $Q$ is a *generalized skeptical consequence* of an abductive framework $\langle T, A \rangle$ if there exists $E \subseteq \mathsf{Abducibles}(T, A)$ such that $Q$ holds in *all* the stable models of $T \cup E$.

Similarly, $Q$ is an *open generalized skeptical consequence* of an abductive framework $\langle T, A \rangle$ if there exists $E \subseteq \mathsf{Abducibles}^o(T, A)$ such that $Q$ holds in *all* the stable models of $T \cup E$. ∎

Open programs improve standard abduction in one more respect: open predicates can be *partially specified* (cf. [Bonatti 2001], Example 14). The same effect can be obtained by adding one more abducible predicate for each partially defined predicate (cf. [Kakas et al. 1992]). A direct approach (such as the open framework) that introduces no auxiliary symbols may be considered more elegant.

## 4 Open programs and ASP

Open programs obviously generalize answer set programming. If $F = O = \emptyset$, then $\mathsf{Comp}(P, F, O) = \{P\}$, and the four entailment relations collapse to the standard credulous and skeptical entailment of the stable model semantics.

**Proposition 4.1** *Let $P$ be a normal logic program. A sentence $Q$ is a credulous (resp. skeptical) consequence of $P$ iff $\langle P, \emptyset, \emptyset \rangle \models^c Q$ (resp. $\langle P, \emptyset, \emptyset \rangle \models^s Q$). Moreover, if $P$ is consistent, then $\langle P, \emptyset, \emptyset \rangle \models^c Q$ is equivalent to $\langle P, \emptyset, \emptyset \rangle \models^{sc} Q$, and $\langle P, \emptyset, \emptyset \rangle \models^s Q$ is equivalent to $\langle P, \emptyset, \emptyset \rangle \models^{cs} Q$.*

In the rest of this section, we focus on the opposite embedding.

Given Lemma 3.1, embedding open programs into ASP may seem trivial. Apparently, the possible extensions of open predicates could all be generated by suitable cyclic definitions, by analogy with the embedding of generalized stable model semantics into (standard) stable model semantics [Satoh and Iwayama 1991]. However, handling the different Herbrand domains of the completions is not so trivial.

**Example 4.2** Let $P = \{p(a),\ q \leftarrow \neg p(X)\}$ and $F = \{b\}$. Suppose that $p \notin O$ and $q \notin O$. The stable models of the completions $P' \in \mathsf{Comp}(P, F, O)$ satisfy $q$ iff $b$ occurs in $P'$. As a consequence, neither $q$ nor $\neg q$ are skeptical open consequences of $\langle P, F, O \rangle$. Now consider two naive attempts at embedding $\langle P, F, O \rangle$ into ASP:

$$\begin{aligned} P_1 &= P \cup \{p(a) \leftarrow \neg \bar{p}(a)\} \cup \{\bar{p}(a) \leftarrow \neg p(a)\} \\ P_2 &= P_1 \cup \{p(b) \leftarrow \neg \bar{p}(b)\} \cup \{\bar{p}(b) \leftarrow \neg p(b)\} \end{aligned}$$

(where $\bar{p}$ is a new predicate symbol). Note that $\neg q$ is a skeptical consequence of $P_1$ while $q$ is a skeptical consequence of $P_2$, so none of the two programs is sound with respect to open inference. ∎

A faithful (partial) embedding of open programs into ASP can be obtained by modeling the Herbrand domains explicitly (a device used also for modeling quantification in Kripke structures where different worlds have different domains).

**Definition 4.3** Let $\langle P, F, O \rangle$ be an open program. Define the corresponding normal logic program $\Pi(P, F, O)$ as follows. For each predicate symbol $p \in O$, introduce a new distinct predicate symbol $\bar{p}$. Moreover, let $U$, $S$ and $\bar{S}$ be three new predicate symbols distinct from the symbols $\bar{p}$. $\Pi(P, F, O)$ consists of the following rules, for all rules $H \leftarrow Body$ in $P$, for all $n$-ary function symbols $f$ occurring in $P$ or $F$, and

for all $p \in O$:

$$H \leftarrow Body, U(x_1), \ldots, U(x_n) \qquad \text{where the } x_i\text{s are the variables of } H \leftarrow Body$$
$$U(f(x_1, \ldots, x_n)) \leftarrow U(x_1), \ldots, U(x_n), S(f)$$

$$S(f) \qquad \text{if } f \notin F$$
$$S(f) \leftarrow \neg \bar{S}(f) \qquad \text{if } f \in F$$
$$\bar{S}(f) \leftarrow \neg S(f) \qquad \text{if } f \in F$$

$$p(x_1, \ldots, x_n) \leftarrow \neg \bar{p}(x_1, \ldots, x_n), U(x_1), \ldots, U(x_n)$$
$$\bar{p}(x_1, \ldots, x_n) \leftarrow \neg p(x_1, \ldots, x_n), U(x_1), \ldots, U(x_n)$$

∎

Intuitively, $S$ and $\bar{S}$ select a vocabulary from $P$ and $F$, $U$ captures the corresponding ground terms (i.e., the Herbrand domain of some completion), and the last rules generate an arbitrary set of ground open atoms.

**Remark 4.4** Strictly speaking, in the above definition the atoms $S(f)$ and $\bar{S}(f)$ should be replaced by $S(\hat{f})$ and $\bar{S}(\hat{f})$, where $\hat{f}$ is a *name* for $f$. For simplicity, here we use a Prolog-like, relaxed syntax.

**Example 4.5** Let $P$ and $F$ be as in Example 4.2, let $O = \{r\}$. Then $\Pi(P, F, O)$ consists of the following rules:

$$p(a)$$
$$q \leftarrow \neg p(X), U(X)$$
$$U(a) \leftarrow S(a)$$
$$U(b) \leftarrow S(b)$$
$$S(a)$$
$$S(b) \leftarrow \neg \bar{S}(b)$$
$$\bar{S}(b) \leftarrow \neg S(b)$$
$$r(X) \leftarrow \neg \bar{r}(X), U(X)$$
$$\bar{r}(X) \leftarrow \neg r(X), U(X)\,.$$

∎

The following theorem proves that the above embedding is correct. Let $M|_{P,F,O} = \{\, p(t_1, \ldots, t_n) \in M \mid p \text{ occurs in } \langle P, F, O \rangle \text{ and } U(t_i) \in M \ (1 \leq i \leq n) \,\}$.

**Theorem 4.6** *$M$ is a stable model of $\Pi(P, F, O)$ iff there exist $P' \in \mathsf{Comp}(P, F, O)$ and a stable model $M'$ of $P'$ such that $M|_{P,F,O} = M'$.*

**Corollary 4.7** *For all sentences $Q$ with no occurrences of the new predicates $\bar{p}$ ($p \in O$), $U$, $S$ and $\bar{S}$,*

1. *$\langle P, F, O \rangle \models^c Q$ iff $Q$ is a credulous consequence of $\Pi(P, F, O)$.*
2. *$\langle P, F, O \rangle \models^s Q$ iff $Q$ is a skeptical consequence of $\Pi(P, F, O)$.*

Note that $\Pi(P, F, O)$ may contain function symbols (then its consequences may be undecidable). If either $F$ or $O$ are infinite then $\Pi(P, F, O)$ is infinite, otherwise $\Pi(P, F, O)$ is finite and it can be computed in polynomial time (w.r.t. $|P \cup F \cup O|$). The following corollary immediately follows.

**Corollary 4.8** *Suppose that $P$, $F$ and $O$ are finite and contain only 0-ary function symbols.*

1. *If $P$ is ground, then credulous and skeptical open inferences are NP-complete and coNP-complete, respectively.*
2. *If $P$ is not ground, then credulous and skeptical open inferences are NEXPTIME-complete and coNEXPTIME-complete, respectively.*

Corollary 4.7 and Corollary 4.8 cannot be easily extended to mixed open inference. The above translation technique (that basically derives from Lemma 3.1) captures faithfully the credulous consequences of the completions, but apparently it does not scale to their skeptical consequences. Currently, we do not know whether mixed inference can be polynomially embedded into ASP.

The results of the previous section can be combined with the above translation to obtain an embedding of abduction into ASP.

**Corollary 4.9** *For all interpretations $M$,*

1. *$M$ is a generalized stable model of an abduction framework $\langle T, A \rangle$ iff $M = M'|_{T,\emptyset,A}$, for some stable model $M'$ of $\Pi(T, \emptyset, A)$.*
2. *$M$ is an open generalized stable model of an abduction framework $\langle T, A \rangle$ iff $M = M'|_{T,Sk,A}$, for some stable model $M'$ of $\Pi(T, Sk, A)$ (where $Sk$ is the set of skolem constants adopted in $\mathsf{Abducibles}^o(T, A)$).*

The translation $\Pi(T, \emptyset, A)$ is slightly more redundant than the translation introduced in [Satoh and Iwayama 1991], but a trivial unfolding process yields an equivalent embedding.

**Example 4.10** Consider the abduction framework $\langle P, O \rangle$, where $P$ and $O$ are specified as in Example 4.5. $\Pi(P, \emptyset, O)$ is

$$p(a)$$
$$q \leftarrow \neg p(X), U(X)$$
$$U(a) \leftarrow S(a)$$
$$S(a)$$
$$r(X) \leftarrow \neg \bar{r}(X), U(X)$$
$$\bar{r}(X) \leftarrow \neg r(X), U(X)\,.$$

Unfolding yields the (expected) simplified program

$$\begin{array}{l} p(a) \\ q \leftarrow \neg p(a) \\ r(a) \leftarrow \neg \bar{r}(a) \\ \bar{r}(a) \leftarrow \neg r(a)\,. \end{array}$$

∎

The translation $\Pi(T, Sk, A)$ extends the approach by [Satoh and Iwayama 1991] with the ability of abducing new individuals. Unfortunately, $\Pi(T, Sk, A)$ is infinite if $Sk$ is infinite. Therefore, in the absence of any upper bound on the domain's cardinality, the existing credulous engines for ASP can only be applied to a finite approximation $\Pi(T, Sk', A)$ of the original problem, where $Sk'$ is a finite subset of $Sk$. Clearly, this approximation yields all and only the abductions that postulate the existence of at most $|Sk'|$ individuals, besides those explicitly mentioned in the domain knowledge.

**Example 4.11** Let

$$\begin{array}{rcl} T & = & \{\ p(a),\ q \leftarrow r(X), \neg p(X)\ \} \\ A & = & \{\, r\,\}\,. \end{array}$$

No generalized stable model of $\langle\, T, A\,\rangle$ satisfies $q$, whereas if $Sk = \{s_0, s_1, \ldots, s_i, \ldots\}$, then $\langle\, T, A\,\rangle$ has an open generalized stable model

$$\{p(a), q, \} \cup \{r(s_i) \mid i \in I\}$$

for each set of natural numbers $I$. Only one individual needs to be abduced in order to explain $q$. Accordingly, the program $\Pi(T, \{s_0\}, A)$ consisting of the rules

$$\begin{array}{l} p(a) \\ q \leftarrow r(X), \neg p(X), U(X) \\ U(a) \leftarrow S(a) \\ U(s_0) \leftarrow S(s_0) \\ S(a) \\ S(s_0) \leftarrow \neg \bar{S}(s_0) \\ \bar{S}(s_0) \leftarrow \neg S(s_0) \\ r(X) \leftarrow \neg \bar{r}(X), U(X) \\ \bar{r}(X) \leftarrow \neg r(X), U(X) \end{array}$$

has a stable model $\{p(a), q, r(s_0)\}$, that "explains" $q$ with $\{r(s_0)\}$. In this case, the bound on the open domain does not cause any significant loss of information, in the sense that all the alternative minimal explanations are isomorphic to $\{r(s_0)\}$. ∎

## 5 Summary and Conclusions

Open programs constitute a simple unifying framework that generalizes both abduction frameworks (under the generalized stable model semantics) and answer set programming, as shown by Theorem 3.2 and Proposition 4.1. In particular, any abduction framework $\langle\, T, A\,\rangle$ corresponds to the open program $\langle\, T, \emptyset, A\,\rangle$, while each normal logic program $P$ is captured by the degenerate open program $\langle\, P, \emptyset, \emptyset\,\rangle$.

The reduction of the generalized stable model semantics to open inference highlights similarities and differences between the two frameworks. In particular, it shows that the only real difference lies in the treatment of the domains, that are fixed in abduction frameworks, and variable in the open framework (cf. the discussion after Theorem 3.2). On the contrary, the different kinds of predicate specifications considered by the two frameworks (ground facts vs. arbitrary definitions) have no influence, as they select the same set of models (Lemma 3.1). This result implies also that in the absence of constraints on the space of abducible rules, no new abducible predicates and rules need to be introduced in order to abduce complex sentences (as opposed to the standard formulation [Poole 1988]).

We have applied the open framework to give the generalized stable model semantics the ability of abducing new individuals (Definition 3.4). Then we extended the embedding of abduction into ASP accordingly, using the more general result for open programs (Corollary 4.9). As a result, ASP engines such as Smodels can be applied to abductive reasoning over *bounded* open domains (cf. the discussion after Corollary 4.9). Because of the bound on the domain size, ASP can only approximate abduction with unbounded open domains. We are currently generalizing *finitary programs* [Bonatti 2001b] to achieve semidecidable, *exact* abduction with unbounded open domains.

Note, however, that the bound on domain size is not always a restriction. In some cases, the bound may be part of the domain knowledge. The existing calculi that support open domains do not express nor reason about such restrictions [Denecker and De Schreye 1998, Shanahan 1989].

The open framework suggests a new form of abduction analogous to *diagnosis-as-entailment*, called *generalized skeptical consequences* (Definition 3.6).

Our results improve the understanding of open inference. From the embedding into ASP we derived the complexity of skeptical and credulous open inference in the finite case, and in the absence of proper functions (Corollary 4.8). In the other cases, the translation may be infinite or contain function symbols, and the only known automated deduction method is the skeptical open resolution calculus introduced in [Bonatti 2001].

This calculus derives sentences that hold no matter how the open program is completed. It may be in-

teresting to investigate the use of such skeptical inferences to pre-compute lemmata, in order to speed-up the computation of explanations.

Moreover, the skeptical open calculus is nonground, and may return nonground answers. This may be a starting point for nonground abduction under the open generalized stable model semantics.

On the other hand, Theorem 3.2 and Corollary 4.7 open the way to the application of abduction procedures and ASP engines to open inference problems, at least under some cardinality and arity restrictions. In particular, abduction and ASP might be applied to static verification problems for various kinds of incomplete specifications.

Currently it is not clear whether the mixed forms of open inference can be embedded into ASP and abduction frameworks. We conjecture that no polynomial reduction based on normal logic programs exists. It should be verified whether the calculus for mixed inference illustrated in [Bonatti 2001] can be immediately applied to the generalized skeptical consequences introduced in Definition 3.6.

**Acknowledgements**

The author is grateful to the anonymous referees for their comments, that led to significant improvements in the paper.

The work reported in this paper was partially supported by the European Community within the Fifth (EC) Framework Programme under contract IST-1999-11791 – FASTER project.

## References

[Bonatti 2001] P. A. Bonatti. Reasoning with open programs. *Proc. of LPNMR'01*, pp. 147-159, LNAI 2173, Springer Verlag, 2001.

[Bonatti 2001b] P. A. Bonatti. Reasoning with infinite stable models. *Proc. of IJCAI*, Morgan Kaufmann, 2001.

[Bonatti et al. 2000] P.A. Bonatti, S. De Capitani Di Vimercate, P. Samarati. An algebra for composing access control policies. *ACM Transactions on Information and System Security, 5(1), 2002*, to appear. Preliminary version in Proc. of the 7th ACM Conference on Computer and Communication Security, CCS'2000, Atene, 2000.

[Denecker and De Schreye 1998] D. Denecker, D. De Schreye. SLDNFA: An abductive procedure for abductive logic programs. *The Journal of Logic Programming*, 34(2):111-167, 1998.

[Eshghi and Kowalski 1989] K. Eshghi, R.A. Kowalski. Abduction compared with negation as failure. In *Proc. of the 6th Int.l Conf. on Logic Programming*, MIT Press, 1989.

[Gelfond and Lifschitz 1988] M. Gelfond, V. Lifschitz. The stable model semantics for logic programming. In *Proc. of the 5th ICLP*, pp.1070-1080, MIT Press, 1988.

[Kakas et al. 1992] A.C. Kakas, R.A. Kowalski, F. Toni. Abductive Logic Programming. *Journal of Logic and Computation*, 2(6):719-770, 1992.

[Kakas and Mancarella 1990] A.C. Kakas, P. Mancarella. Generalized Stable Models: a semantics for abduction. *Proc. of ECAI'90*, pp.385-391, 1990.

[Niemelä and Simons 1997] I. Niemelä, P. Simons. Smodels - an implementation of the stable model and well-founded semantics for normal LP. In J. Dix, U. Furbach, A. Nerode (eds.), *Logic Programming and Nonmonotonic Reasoning: 4th international conference, LPNMR'97*, LNAI 1265, Springer Verlag, Berlin, 1997.

[Poole 1988] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27-47, 1988.

[Satoh and Iwayama 1992] K. Satoh, N. Iwayama. A query evaluation method for abductive logic programming. In *Proc. of the Joint Int.l Conf. and Symposium on Logic Programming* (JICSLP'92), 671-685, MIT Press, 1992.

[Satoh and Iwayama 1991] K. Satoh, N. Iwayama. Computing abduction by using the TMS. In *Proc. of the Int.l Conf. on Logic Programming* (JICSLP'92), 504-ff, MIT Press, 1991.

[Shanahan 1989] M. Shanahan. Prediction is deduction but explanation is abduction. In *Proc. of IJCAI'89*, 1055-ff, 1989.

[Subrahmanian et al. 2000] V.S. Subrahmanian, P.A. Bonatti, J. Dix, T. Eiter, S. Kraus, F. Ozcan, R. Ross. *Heterogeneous Active Agents*. MIT Press, 2000.