Abductive Disjunctive Logic Programming

Ullrich Hustadt
Max-Planck-Institut für Informatik,
Im Stadtwald, W-6600 Saarbrücken, Germany
Phone: (+49 681) 302-5431, Fax: (+49 681) 302-5430,
E-mail: Ullrich.Hustadt@mpi-sb.mpg.de

1 Introduction

[EK89] introduced the notion of an abductive framework and proposed stable models as a semantics for abduction. They showed that abductive frameworks can be used to provide an alternative basis for negation-as-failure in logic programming. [KM90] introduced the notion of generalized stable models by suitably extending the definition of stable models. The semantics of generalized stable models clarifies the meaning of integrity constraints within an abductive framework. In ([SI92]) a goal-directed method for computing the generalized stable models of an abductive framework has been proposed. Their method is correct for any consistent abductive framework. Whereas abductive frameworks correspond to normal logic programs with integrity constraints, I propose an extension to disjunctive normal logic programs. Disjunctive normal logic programs extend normal logic programs to disjunctive logic programs and therefore, provide full first-order expressibility.

2 Abductive Frameworks

Definition 2.1 (Abductive framework)
An *atom* is an expression $P(t_1, \ldots, t_n)$, where P is a predicate symbol and $t_1,\ldots, t_n$ are terms. A *positive literal* is an atom, a *negative literal* is an expression $not(A_1)$, where A_1 is an atom. A *literal* is either a positive or a negative literal. Let L be a literal. Then L^c denotes the complement of L.

A *clause* is either of the form

$$A_1 \vee \ldots \vee A_m \leftarrow L_1 \wedge \ldots \wedge L_n,$$

where $A_1,\ldots, A_m$, $m \geq 1$ are atoms, and $L_1,\ldots, L_n$ are literals, or

$$\perp \leftarrow L_1 \wedge \ldots \wedge L_n,$$

where $L_1,\ldots, L_n$ are literals. The left hand side of a clause is the *head*, denoted by $head(C)$, the right hand side is the *body* of the clause, denoted by $body(C)$.

A *program* is a set of clauses. An *abductive framework* is a pair $\langle T, A\rangle$ where A is a set of predicate symbols, called *abducible predicates*, and T is a set of clauses such that no predicate symbols of head atoms are in A. A set of ground atoms for predicates in A is called *abducibles*. The set of all abducibles is denoted by $\mathcal{A}$.

Given an abductive framework $\langle T, A\rangle$, $pos(C)$ is the set of positive literals in the body of a clause C which are not abducibles, $neg(C)$ is the set of negative literals in the body of C, $abd(C)$ is the set of abducibles in the body of C. △

The *Herbrand base* of a program T is denoted by $HB(T)$, its *Herbrand universe* by $HU(T)$.

We impose the restriction that the clauses of a program must be *range-restricted*, i.e. any variable in a clause C must occur in $pos(C)$. Any clause can be transformed to a range-restricted clause by inserting for every variable violating the range-restrictedness condition a predicate *dom* describing the Herbrand universe.

Definition 2.2 (Minimal Model)
An *interpretation* I for a program T is a subset of $HB(T)$. An interpretation I satisfies a ground atom A_1 iff $A_1 \in I$. It satisfies a ground literal $not(A_1)$ iff $A_1 \notin I$. No interpretation satisfies $\bot$. An interpretation I satisfies a clause

$$A_1 \vee \ldots \vee A_m \leftarrow L_1 \wedge \ldots \wedge L_n,$$

iff for every ground substitution σ either one of $A_1\sigma, \ldots, A_m\sigma$ is satisfied by I or one of $L_1\sigma, \ldots, L_n\sigma$ is not satisfied by I.

An interpretation I is a *model* of T if I satisfies every clause in T. An model I of T is *minimal* if there is no interpretation $I' \subset I$ such that I' is a model of T. $\triangle$

Definition 2.3 (Gelfond-Lifschitz Transformation)
Let T be a program and I be an interpretation. The *Gelfond-Lifschitz Transformation* $GL(T, I)$ of T is defined by

$$\begin{aligned} GL(T,I) = \\ &\{(A_1 \vee \ldots \vee A_m \leftarrow B_1 \wedge \ldots \wedge B_n)\theta \mid \\ &\quad A_1 \vee \ldots \vee A_m \leftarrow B_1 \wedge \ldots \wedge B_n \wedge not(C_1) \wedge \ldots \wedge not(C_k) \in T, \\ &\quad \theta \text{ is a ground substitution, and} \\ &\quad C_1\theta, \ldots, C_k\theta \notin I\} \end{aligned}$$

$\triangle$

Definition 2.4 (Generalized Stable Model)
An interpretation I is a *stable model* of T iff I is a minimal model of $GL(T, I)$.

Let $\langle T, A\rangle$ be an abductive framework and Δ be a set of abducibles. A *generalized stable model* $M(\Delta)$ of $\langle T, A\rangle$ is a stable model of $T \cup \{H \leftarrow \mid H \in \Delta\}$. $\triangle$

An abductive framework $\langle T, A\rangle$ is *consistent* if there exists a generalized stable model $M(\Delta)$ of $\langle T, A\rangle$ for some set Δ. In the following, we restrict our intention to consistent abductive frameworks.

3 Proof Procedure for Abductive Frameworks

Definition 3.1 (Goal)
A *goal* is a disjunction of conjunctions of literals, written

$$(L^1_1 \wedge \ldots \wedge L^1_{n_1}) \vee \ldots \vee (L^m_1 \wedge \ldots \wedge L^m_{n_m}).$$

An interpretation I satisfies a goal if there exists a ground substitution σ such that for some i, $1 \leq i \leq m$, I satisfies $L^i_j\sigma$ for every $1 \leq j \leq n_i$. $\triangle$

Let D be a disjunction of atoms $A_1 \vee \ldots \vee A_m$. Then D^c denotes the conjunction of negative literals $A^c_1 \wedge \ldots \wedge A^c_m$.

Definition 3.2 (Abductive Explanation)
Let $\langle T, A\rangle$ be an abductive framework and G a goal. We call a set of abducibles Δ an *abductive explanation* for G if there exists a generalized stable model $M(\Delta)$ that satisfies G. $\triangle$

We can define an abductive proof procedure generating abductive explanations by combining the proof procedure given in ([SI92]) with a proof procedure for programs. Such a proof procedure is described in ([RLS91]). We need the following definition for the description of the abductive proof procedure.

Definition 3.3
Let $\langle T, A\rangle$ be an abductive framework and L be a ground literal. Then the set of resolvents with respect to L and T, $resolve(L, T)$, is defined by

$$
\begin{aligned}
&resolve(L,T) = \\
&\quad \{\, (H_1 \vee \ldots \vee H_{i-1} \vee H_{i+1} \vee \ldots \vee H_k \leftarrow L_1 \wedge \ldots \wedge L_m)\theta \mid \\
&\qquad\qquad \text{L is negative and} \\
&\qquad\qquad (H_1 \vee \ldots \vee H_k \leftarrow L_1 \wedge \ldots \wedge L_m) \in T \text{ and} \\
&\qquad\qquad L^c = H_i\theta \text{ by a ground substitution } \theta \,\} \cup \\
&\quad \{\, (H_1 \vee \ldots \vee H_k \leftarrow L_1 \wedge \ldots \wedge L_{i-1} \wedge L_{i+1} \wedge \ldots \wedge L_m)\theta \mid \\
&\qquad\qquad (H_1 \vee \ldots \vee H_k \leftarrow L_1 \wedge \ldots \wedge L_m) \in T \text{ and} \\
&\qquad\qquad L = L_i\theta \text{ by a ground substitution } \theta\}
\end{aligned}
$$

The set of deleted clauses with respect to L and T, $delete(L, T)$, is defined by

$$
\begin{aligned}
&delete(L,T) = \\
&\quad \{(H_1 \vee \ldots \vee H_k \leftarrow L_1 \wedge \ldots \wedge L_m)\theta \mid (H_1 \vee \ldots \vee H_k \leftarrow L_1 \wedge \ldots \wedge L_m) \in T \text{ and} \\
&\qquad\qquad L^c = L_i\theta \text{ by a ground substitution } \theta\}
\end{aligned}
$$

△

Definition 3.4 (Deduction rules)
Instead of using a kind of pseudo-code to describe the abductive proof procedure, we will provide inference rules for deriving judgements of the form

$$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_a \langle G, \sigma, \Delta_2\rangle,$$

where $\langle T, A\rangle$ is an abductive framework, Δ_1, Δ_2 are sets of abducibles, G is a goal, and σ is a substitution. Intuitively, the judgement above means that Δ_2 is an abductive explanation for $G\sigma$. To define the inference rules for $\vdash_a$, we need additional judgements of the form

$$
\begin{array}{ll}
\langle\langle T, A\rangle, \Delta_1\rangle \vdash_p \langle G, \sigma, \Delta_2\rangle, & \langle\langle T, A\rangle, \Delta_1\rangle \vdash_r \langle \mathcal{C}, \Delta_2\rangle, \\
\langle\langle T, A\rangle, \Delta_1\rangle \vdash_l \langle L_1, \Delta_2\rangle, & \langle\langle T, A\rangle, \Delta_1\rangle \vdash_d \langle \mathcal{C}, \Delta_2\rangle,
\end{array}
$$

where L_1 is a literal and $\mathcal{C}$ is a set of clause. We will provide inference rules for these judgements too.

Abductive Inference

$\langle\langle T, A\rangle, \Delta\rangle \vdash_a \langle K_1 \vee \ldots \vee K_n, \sigma, \Delta\rangle$
  if $\langle\langle T, A\rangle, \Delta \cup \{\bot\}\rangle \vdash_p \langle \bot, \sigma, \Delta \cup \{\bot\}\rangle$

$\langle\langle T, A\rangle, \Delta\rangle \vdash_a \langle K_1 \vee \ldots \vee K_n, \sigma, \Delta'\rangle$
  if $\langle\langle T \cup \{q(\bar{x}) \leftarrow K_1, \ldots, q(\bar{x}) \leftarrow K_n\}, A\rangle, \Delta\rangle \vdash_p \langle q(\bar{x}), \sigma, \Delta'\rangle$
    where $K_1, \ldots, K_n$ are conjunctions of literals, q is a fresh predicate symbol, and $\bar{x}$ are the free variables of $K_1, \ldots, K_n$.

Hypothesis Rule

$\langle\langle T, A\rangle, \Delta \cup \{A_1\}\rangle \vdash_p \langle B_1 \wedge B_2 \wedge \ldots \wedge B_k, \sigma\theta, \Delta_2\rangle$
if $\langle\langle T, A\rangle, \Delta \cup \{A_1\}\rangle \vdash_p \langle B_2\sigma \wedge \ldots \wedge B_k\sigma, \theta, \Delta_2\rangle$
where σ is the most general unifier of A_1 and B_1.

Resolution Rule

$\langle\langle T \cup \{A_1 \leftarrow L_1 \wedge \ldots \wedge L_n\}, A\rangle, \Delta_1\rangle \vdash_p \langle B_1 \wedge B_2 \wedge \ldots \wedge B_k, \sigma\theta, \Delta_3\rangle$
if $\langle\langle T \cup \{A_1 \leftarrow L_1 \wedge \ldots \wedge L_n\}, A\rangle, \Delta_1\rangle \vdash_p \langle L_1\sigma \wedge \ldots \wedge L_n\sigma \wedge B_2\sigma \wedge \ldots \wedge B_k\sigma, \theta, \Delta_2\rangle$ and
$\langle\langle T, A\rangle, \Delta_2\rangle \vdash_l \langle A_1\sigma\theta, \Delta_3\rangle$,
where σ is the most general unifier of A_1 and B_1.

Abduction Rule

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_p \langle A_1 \wedge L_2 \wedge \ldots \wedge L_m, \sigma, \Delta_3\rangle$
if $\langle\langle T, A\rangle, \Delta_1\rangle \vdash_l \langle A_1, \Delta_2\rangle$,
$\langle\langle T, A\rangle, \Delta_2\rangle \vdash_p \langle L_2 \wedge \ldots \wedge L_m, \sigma, \Delta_3\rangle$, and
A_1 is in $\mathcal{A}$.

Negation Rule

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_p \langle not(A_1) \wedge L_2 \ldots \wedge L_m, \sigma, \Delta_3\rangle$
if $\langle\langle T, A\rangle, \Delta_1\rangle \vdash_l \langle not(A_1), \Delta_2\rangle$ and
$\langle\langle T, A\rangle, \Delta_2\rangle \vdash_p \langle L_2 \wedge \ldots \wedge L_m, \sigma, \Delta_3\rangle$.

Splitting Rule

$\langle\langle T \cup \{A_1 \vee \ldots \vee A_m \leftarrow L_1 \wedge \ldots \wedge L_n\}, A\rangle, \Delta_1\rangle \vdash_p \langle G, \Delta_{m+1}\rangle$
if $\langle\langle T \cup \{A_i \leftarrow L_1 \wedge \ldots \wedge L_n\}, A\rangle, \Delta_1\rangle \vdash_p \langle G, \Delta_2\rangle$ for some $1 \leq i \leq m$,
$\langle\langle T \cup \{A_j\}, A\rangle, \Delta_{j+1}\rangle \vdash_a \langle G, \Delta_{j+2}\rangle$ for each $j = 1, \ldots, i-1$, and
$\langle\langle T \cup \{A_j\}, A\rangle, \Delta_j\rangle \vdash_a \langle G, \Delta_{j+1}\rangle$ for each $j = i+1, \ldots, m$.

Consistency of literals $\vdash_l$

$\langle\langle T, A\rangle, \Delta_1 \cup \{\bot\}\rangle \vdash_l \langle L, \Delta_1 \cup \{\bot\}\rangle$

$\langle\langle T, A\rangle, \Delta_1 \cup \{L\}\rangle \vdash_l \langle L, \Delta_1 \cup \{L\}\rangle$

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_l \langle L, \Delta_3\rangle$
if $\langle\langle T, A\rangle, \Delta_1 \cup \{L\}\rangle \vdash_r \langle resolve(L, T), \Delta_2\rangle$,
$\langle\langle T, A\rangle, \Delta_2\rangle \vdash_d \langle delete(L, T), \Delta_3\rangle$, and
L is not $\bot$.

Consistency of rule deletions $\vdash_d$

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_d \langle \{C\} \cup \mathcal{C}, \Delta_3\rangle$
if $\langle\langle T, A\rangle, \Delta_1\rangle \vdash_p \langle head(C), \epsilon^1, \Delta_2\rangle$ and
$\langle\langle T, A\rangle, \Delta_2\rangle \vdash_d \langle \mathcal{C}, \Delta_3\rangle$.

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_d \langle \{C\} \cup \mathcal{C}, \Delta_3\rangle$
if $\langle\langle T, A\rangle, \Delta_1\rangle \vdash_l \langle head(C)^c, \Delta_2\rangle$ and
$\langle\langle T, A\rangle, \Delta_2\rangle \vdash_d \langle \mathcal{C}, \Delta_3\rangle$.

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_d \langle \emptyset, \Delta_1\rangle$

[1] The identity substitution is denoted by ϵ

Consistency of rules $\vdash_r$

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_r \langle\{C\} \cup \mathcal{C}, \Delta_3\rangle$
  if $\langle\langle T, A\rangle, \Delta_1\rangle \vdash_p \langle L^c, \epsilon, \Delta_2\rangle$ for some literal L in the body of C and
    $\langle\langle T, A\rangle, \Delta_2\rangle \vdash_d \langle\mathcal{C}, \Delta_3\rangle$.

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_r \langle\{C\} \cup \mathcal{C}, \Delta_4\rangle$
  if $\langle\langle T, A\rangle, \Delta_1\rangle \vdash_p \langle body(C), \epsilon, \Delta_2\rangle$,
    $\langle\langle T, A\rangle, \Delta_2\rangle \vdash_l \langle H_1, \Delta_3\rangle$ for some atom H_1 in the head of C, and
    $\langle\langle T, A\rangle, \Delta_3\rangle \vdash_d \langle\mathcal{C}, \Delta_4\rangle$.

$\langle\langle T, A\rangle, \Delta_1\rangle \vdash_r \langle\emptyset, \Delta_1\rangle$ $\triangle$

Theorem 3.5 *Let $\langle T, A\rangle$ be an consistent abductive framework and G a goal. Then G has an abductive explanation Δ iff*

$$\langle\langle T, A\rangle, \emptyset\rangle \vdash_p \langle G, \sigma, \Delta'\rangle$$

can be derived for some substitution σ and a set of literals Δ', such that $\Delta' \cap \mathcal{A} \subseteq \Delta$.

4 Future Work

[SI92] introduced the notion of the relevant ground program Ω_T for a normal logic program T which is a subset of the set of ground rules obtainable from T. Using the relevant ground program it is possible to reduce the size of the sets $resolve(L, T)$ and $delete(L, T)$. Only if these two sets are finite, the proposed abductive proof procedure is applicable. Although there is a notion of the relevent ground program for a disjunctive normal logic program, it is not obvious that it can be used to reduce the size of $resolve(L, T)$ and $delete(L, T)$ without loosing correctness of the abductive proof procedure.

References

[EK89] K. Eshghi and R.A. Kowalski. Abduction compared with negation by failure. In Georgio Levi and Maurizio Martelli, editors, *Proceedings of the Sixth International Conference on Logic Programming*, pages 234–254, Lisabon, Portugal, June 19–23 1989. IEEE, MIT Press.

[KM90] A.C. Kakas and P. Mancarella. Generalized stable models: A semantics for abduction. In Luigia Carlucci Aiello, editor, *Proceeding of the 9th European Conference on Artifical Intelligence*, pages 385–391, Stockholm, Sweden, August, 6–10 1990. Pitman Publishing.

[RLS91] David W. Reed, Donald W. Loveland, and Bruce T. Smith. The near-horn approach to disjunctive logic programming. In L.-H. Eriksson, Hallnäs, and P. Schroeder-Heister, editors, *Proceedings of the Second International Workshop on Extension of Logic Programming (ELP '91)*, volume 596 of *LNCS*, pages 345–369, Stockholm, Sweden, January, 27–29 1991. Springer-Verlag.

[SI92] Ken Satoh and Noboru Iwayama. A query evaluation method for abductive logic programming. In Krzysztof R. Apt, editor, *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 671–685, Washington, D.C., USA, November, 9–13 1992. The MIT Press.