ASPAL. Proof of soundness and completeness

Domenico Corapi and Alessandra Russo

Department of Computing
Imperial College London
180 Queen's Gate, SW7 2AZ
London, UK
E-mail: {d.corapi,a.russo}@ic.ac.uk

Abstract. We provide here a brief introduction and proof of soundness and completeness of the ILP system ASPAL. This document is in support of our ICLP 2011 submission, for the reviewers' benefits.

Inductive Logic Programming (ILP) [4] is a machine learning technique concerned with the induction of logic theories, called target theories, that generalise (positive and negative) examples with respect to a prior background knowledge. For example, from the observations $fly(a), fly(b), \neg fly(c)$ and a background knowledge containing the two facts $bird(a)$ and $bird(b)$, we can generalise the concept $fly(X) \leftarrow bird(X)$. In non-trivial problems it is crucial to define the space of possible solutions accurately. Target theories are within a space defined by a *language bias*, which can be expressed using mode declarations [4].

We refer to [3] for notations and preliminary definitions on logic programming. For a given theory T we denote with B_T the Herbrand base of T.

Definition 1. *A* mode declaration *is either a* head declaration*, written* $modeh(s)$*, or a* body declaration*, written* $modeb(s)$*, where* s *is a* schema*. A schema is a ground literal containing special terms called* placemarkers*. A* placemarker *is either '+type', '−type' or '#type' where* type *denotes the type of the placemarker and the three symbols '+', '−' and '#' indicate that the placemarker is an input, an output, a constant placemarker, respectively.*

In the previous example a possible language bias would be given by the three mode declarations $modeh(fly(+animal))$, $modeb(bird(+animal))$ and $modeb(fish(+animal))$.

A rule $h \leftarrow b_1, ..., b_n$ is *compatible* with a set M of mode declarations iff (a) h is the schema of a head declaration in M and b_i are the schemas of body declarations in M where every input and output placemarkers are replaced by variables and constant placemarkers are replaced by constants; (b) every input variable in any atom b_i is either an input variable in h or an output variable in some $b_j, j < i$; and (c) all variables and constants are of the corresponding type. From a user perspective, mode declarations establish how rules in the final hypotheses are structured, defining literals that can be used in the head and in the body of a well-formed hypothesis. $s(M)$ is the set of all the rules compatible with M. For a set M of mode declarations we denote with M_b the subset of body declaration and with M_h the subset of head declaration.

Definition 2. *An* ILP task *is a tuple* $\langle P, B, M\rangle$ *where* P *is a set of conjunctions of literals, called* properties*,* B *is a normal program, called* background theory*, and* M *is a set of mode declarations. A theory* H*, called* hypothesis*, is an inductive solution for the task* $\langle P, B, M\rangle$*, if (a)* $H \subseteq s(M)$*, and (b)* P *is true in all the stable models of* $B \cup H$*.*

The above notion is an instance of the *learning from entailment* setting [2], where positive and negative examples are our properties. In the following we omit the additional control on the type of the arguments in the mode declarations for simplicity and readability. The extension of the definitions and proofs with type check is straightforward, they can be added to the bodies of the hypotheses and of the rules of the additional theories introduced using a standard pre-processing procedure used in ASP. Furthermore, our definitions and proposition will be given for ground programs. Rules with variables will be used as a shorthand for the set of their ground instances. We define a strong completeness requirement, compared to alternative formulations where completeness only requires that a solution is found if one exists.

Given a mode declaration $id : modeh(s)$ (or $modeb(s)$), id is the (unique) *identifier* for the mode declaration and $sc(id)$ denotes the schema s of the mode declaration.

Definition 3. $\mathbf{s}$ *is the literal obtained from* s *by replacing all placemarkers with different variables* $var(\mathbf{s}, s) = (X_1, ..., X_n)$*, called a* variable list *and it is called the variabilisation of* s*.* $con(\mathbf{s}, s) = (C_1, ..., C_c)$ *is the* constant list *of variables in* $\mathbf{s}$ *that replace only constant placemarkers in* s*.* $inp(\mathbf{s}, s) = (I_1, ..., I_i)$ *and* $out(\mathbf{s}, s) = (O_1, ..., O_o)$ *are defined similarly for input and output placemarkers. A variable is in the variable list if and only if is one of the constant, input or output list. When* s *is clear from the context we omit the second argument from* $var(\mathbf{s}, s)$*,* $con(\mathbf{s}, s)$*,* $inp(\mathbf{s}, s)$ *and* $out(\mathbf{s}, s)$*.*

Given a set of mode declarations M, a *top theory* $\top = t(M)$ is constructed as follows:

– For each head declaration $modeh(s)$, with unique identifier id, the following rule is in $\top$

$$\begin{aligned} \mathbf{s} \leftarrow\ & rule(RId, (id, con(\mathbf{s}), ())), \\ & rule\_id(RId), \\ & body(RId, 1, inp(\mathbf{s})) \end{aligned} \tag{1}$$

– For each body declaration $modeb(s)$, with unique identifier id the following clause is in $\top$

$$\begin{aligned} body(RId, L, I) \leftarrow\ & rule(RId, L, (id, con(\mathbf{s}), Links)), \\ & link(inp(\mathbf{s}), I, Link), \\ & \mathbf{s}, \\ & append(I, out(\mathbf{s}), O), \\ & body(RId, L+1, O) \end{aligned} \tag{2}$$

- The following rules are in $\top$ together with a standard definition for the append predicate.

$$body(RId, L, \_) \leftarrow rule(RId, L, last) \quad (3)$$

$$\begin{array}{l} link((X),(X),1). \\ link((X),(X,\_),2). \\ link((X),(\_,X),2). \\ link((X,Y),(X,Y),(1,2)). \\ link((X,Y),(Y,X),(2,1)). \\ link((X),(X,\_,\_),1). \\ link((X),(\_,X,\_),2). \\ ... \end{array} \quad (4)$$

$$\begin{array}{l} rule\_id(1). \\ ... \\ rule\_id(rn). \end{array} \quad (5)$$

$rule\_id(rid)$ is true whenever $1 \leq rid \leq rn$ where rn is the maximum number of new rules allowed. $link((a_1, ..., a_m), (b_1, ..., b_n), (o_1, ..., o_m))$ are true if for each element in the first list a_i, there exists an element in the second list b_j such that a_i unifies with b_j and $o_i = j$. Given the top theory, we seek a set of *rule* atoms Δ, such that P is true in all models of $B \cup \top \cup \Delta$.

Δ has a one-to-one mapping to a set of rules $H = u(\Delta, M)$. Intuitively, each abduced atom represents a literal of the rule identified by the first argument. The second argument collects the constant used in the literal and the third defines the variable linking.

Δ_{rid} denotes the subset of Δ of all the *rule* abducibles with rid as a first argument. Each Δ_{rid} corresponds to a rule r in $u(\Delta, M)$, $r = u(\Delta_{rid}, M)$. For a given

$$\begin{aligned} \Delta_{rid} =\{&rule(rid, (id_h, con_h), \\ &rule(rid, 1, (id_1, con_1, links_1)), \\ &..., \\ &rule(rid, 1, (id_n, con_n, links_n))\} \end{aligned}$$

r is a rule $h \leftarrow b_1, ..., b_n$ such that, given a list of variables defined as follows:

- $avar_0$ is $inp(\mathbf{sc(id_h)})$
- $avar_i$ is such that $append(avar_{i-1}, out(\mathbf{sc(id_i)}), avar_i)$ is true for each $i = 1, ..., n$ for the standard definition of append

each atom is constructed as follows:

- $h = \mathbf{sc(id_h)}$ and $con(\mathbf{sc(id_h)})$ is unified with the list of constant con_h of the same length

– $b_i = \mathbf{sc}(\mathbf{id_i})$, $con(\mathbf{sc}(\mathbf{id_i}))$ is unified with the list of constant con_i of the same length and $link(inp(\mathbf{sc}(\mathbf{id_i})), avar_{i-1}, links_i)$ is true, according to the definition of link provided in the top theory for each $i = 1, ..., n$

Proposition 1. *Let H be a theory and M a set of mode declarations. Then $H \in s(M)$ if and only if there exists a Δ such that $H = u(\Delta, M)$.*

Proof. The proof is immediate as the transformation itself can be used to define $s(M)$. In fact the transformation requires that the head corresponds to the variabilisation of a schema of a head declaration where all the constants are replaced by arbitrary constants. Each other condition i in the body can only bind input to variables in $avar_{i-1}$ by definition and the constants are also replaced by a list of arbitrary constants. □

In the proof we will use the unfolding transformation or generalised principle of partial evaluation (GPPE)[1].

Definition 4. *Let P be a theory, $h \leftarrow b_1, ..., b_{i-1}, b_i, b_{i+1}, ..., b_n \in P$ a rule r and b_i a positive literal. We say we apply a* GPPE transformation *to r on b_i if P' is the theory obtained from P by replacing rule r with the p rules*

$$h \leftarrow b_1, ..., b_{i-1}, b_{i+1}, ..., A_j$$

where A_j is the conjunction of literals in $b_i \leftarrow A_j \in P$, $j = 1..p$.

Definition 5. *Let P and Q be theories. We say P is equivalent to Q if, for all interpretation I, I is a stable model for P if and only if I is a stable model for Q.*

Proposition 2. *[1] Let P be a theory. If P' is the theory obtained applying GPPE to any rule in P then P' and P are equivalent.*

Proposition 3. *Let B be a normal program, H be a hypothesis and M a set of mode declarations such that $H = u(\Delta, M)$ and $\top = t(M)$. Let α denote $B \cup H$ and β denote $B \cup \top \cup \Delta$. Let $K = I \cup J$ be an interpretation for β such that $I \subseteq B_\alpha$, $J \subseteq (B_\beta \setminus B_\alpha)$. Then K is a stable model for β if and only if I is a stable model for α.*

Proof. Consider rules (1) of the top theory. Applying GPPE to them on $body(RId, 1, inp(\mathbf{s}))$ we obtain a theory $\top_1$ in which rules (1) are replaced by the following:

$$\begin{aligned} \mathbf{sc}(\mathbf{id_h}) \leftarrow \quad & rule(RId, (id_h, con(\mathbf{sc}(\mathbf{id_h})), ()), \\ & rule\_id(RId), \\ & rule(RId, 1, (id_1, con(\mathbf{sc}(\mathbf{id_1})), Links_1)), \\ & link(inp(\mathbf{sc}(\mathbf{id_1})), inp(\mathbf{sc}(\mathbf{id_h})), Links_1), \\ & \mathbf{sc}(\mathbf{id_1}), \\ & append(inp(\mathbf{sc}(\mathbf{id_h})), out(\mathbf{sc}(\mathbf{id_1})), O_1), \\ & body(RId, 2, O_1) \end{aligned} \tag{6}$$

for all id_h such that $id_h \in M_h$ and for all $id_1 \in M_b$. We used the rules (2) where the second argument is 1. Using (3) in the transformation the following rules are included in $\top_1$

$$\begin{aligned}\mathbf{sc(id_h)} \leftarrow \quad & rule(RId, (id_h, con(\mathbf{sc(id_h)}), ()), \\ & rule\_id(RId), \\ & rule(RId, 1, last)\end{aligned} \tag{7}$$

We can now apply the transformation again on $body(RId, 2, O_1))$ and rules (2) and (3), and again on $body(RId, i+1, O_i)$ for each $i \leq nc$. nc the maximum number of conditions allowed is enforced by an additional condition on the second argument of the *body* predicate, omitted for ease of exposition. So we obtain a theory $\top'$ that contains the following rules for each $m = 1..nc$.

$$\begin{aligned}\mathbf{sc(id_h)} \leftarrow \quad & rule(RId, (id_h, con(\mathbf{sc(id_h)}), ()), \\ & rule\_id(RId), \\ & rule(RId, 1, (id_1, con(\mathbf{sc(id_1)}), Links_1)), \\ & link(inp(\mathbf{sc(id_1)}), inp(\mathbf{sc(id_h)}), Links_1), \\ & \mathbf{sc(id_1)}, \\ & append(inp(\mathbf{sc(id_h)}), out(\mathbf{sc(id_1)}), O_1), \\ & ..., \\ & rule(RId, n, (id_m, con(\mathbf{sc(id_m)}), Links_m)), \\ & link(O_{m-1}, inp(\mathbf{sc(id_m)}), Links_m), \\ & \mathbf{sc(id_m)}, \\ & append(O_{m-1}, out(\mathbf{sc(id_m)}), O_m), \\ & rule(RId, m+1, last),\end{aligned} \tag{8}$$

and for all id such that $id \in M_h$ and $id_i \in M_b, i = 1, .., m$.

Of these rules the only rules whose body can be true are those that contain exactly the atoms in Δ_{rid} for some rid. Consider $h \leftarrow b_1, ..., b_n = u(\Delta_{\overline{rid}}, M)$

$$\begin{aligned}\Delta_{\overline{rid}} = \{ & rule(\overline{rid}, (\overline{id_h}, \overline{con_h}, ())), \\ & rule(\overline{rid}, 1, (\overline{id_1}, \overline{con_1}, \overline{links_1})), \\ & ..., \\ & rule(\overline{rid}, 1, (\overline{id_n}, \overline{con_n}, \overline{links_n}))\}\end{aligned}$$

we can consider all the rules that contain the atoms in such $\Delta_{\overline{rid}}$ and apply GPPE on conditions *rule_id* and *append*. O_i as result of the append will correspond to $avar_i$ as defined previously:

$$
\begin{aligned}
\mathbf{sc}(\mathbf{id_h}) \leftarrow \quad & rule(\overline{rid}, (\overline{id_h}, \overline{con_h}, ())) \\
& rule(\overline{rid}, 1, (\overline{id_1}, \overline{con_1}, \overline{links_1})), \\
& link(inp((\mathbf{s})), \overline{avar_0}, \overline{links_1}), \\
& \mathbf{sc}(\overline{\mathbf{id_1}}), \\
& ..., \\
& rule(\overline{rid}, n, (\overline{id_m}, \overline{con_m}, \overline{links_n})), \\
& link(inp(\mathbf{sc}(\mathbf{id_n})), \overline{avar_{n-1}}, \overline{link_n}), \\
& \mathbf{sc}(\overline{\mathbf{id_n}}), \\
& rule(\overline{rid}, m+1, last)
\end{aligned} \tag{9}
$$

Using the truth values defined in J we can see that rules (9) correspond to rules $h \leftarrow b_1, ..., b_n$. We can then say $B \cup \top \cup \Delta$ is equivalent to $B \cup H \cup \top^- \cup \Delta$ where $\top^-$ is the original $\top$ without rules (1) that have been transformed into H. Now $\top^- \cup \Delta$ define a disjoint set of predicates from $B \cup H$. The set J only defines those predicates that appear in $\top$ and Δ but not in B, thus it must define the predicates *link*, *rule_id*, *append*, *rule* and *body* appearing in the head of rules in $\top^- \cup \Delta$. The first four predicates are defined extensionally by facts. The definition of *body* predicates is given by a stratified theory (as a result of the increasing value of the integer used as a second variable of *body*) and completes the definition of J. I is thus a stable model for the new theory if and only if it is a stable model for $B \cup H$ that proves the proposition. □

Proposition 4. *Given an ILP task $\langle P, B, M \rangle$, H is an inductive solution if and only if there is a Δ such that $H = u(\Delta, M)$, $\top = t(M)$ and P is true in all stable models of $B \cup \top \cup \Delta$.*

Proof. The proposition is a consequence of the Propositions 3 and 1. H is a solution if and only if (a) $H \subseteq s(M)$, and (b) P is true in all the stable models of $B \cup H$. $H \subseteq s(M)$ if and only if there exists a Δ such that $H = u(\Delta, M)$ according to Proposition 1. P is true in all answer sets of $B \cup H$. But all such answer sets have a corresponding answer set for β, according to Proposition 3 that "agrees" on P, thus proving the proposition. □

ASPAL is an ILP system that is based on the transformation introduced and on an ASP solver used abductively to derive Δ. The soundness (if a Δ is found than $u(\Delta, M)$ is an inductive solution) and completeness (if H is an inductive solution than a Δ is found such that $H = u(\Delta, M)$) of ASPAL follow from Proposition 1 and the soundness and completeness properties of the underlying *ASP* solver that ensures that for any given theory T and conjunction of literals P, any Δ such that P is true in a stable model of $T \cup \Delta$ is an output of the system.

References

1. Stefan Brass and Jürgen Dix. Characterizations of the stable semantics by partial evaluation. In *Logic Programming and Non-Monotonic Reasoning, Proceedings of the Third International Conference, LNCS 928*, pages 85–98. Springer, 1995.

2. Luc De Raedt. Logical settings for concept-learning. *Artif. Intell.*, 95(1):187–201, 1997.
3. J. W. Lloyd. *Foundations of logic programming.* Springer-Verlag New York, Inc., New York, NY, USA, 1984.
4. S. Muggleton. Inverse entailment and progol. *New Gen. Comp.*, 13(3&4):245–286, 1995.