

The Meta-program Injection feature in ILASP

Mark Law

Alessandra Russo

Krysia Broda

June 15, 2018

Abstract

This document presents an experimental feature, introduced in ILASP3.2, that allows a learning task to be given along with a meta-program defining the conditions (over example coverage) that are sufficient for a hypothesis to be an inductive solution. This allows the user to weaken the standard conditions, for example, by specifying that “at least one (but not necessarily all) of the following examples must be covered”.

1 Background

We first recall the definitions of context-dependent partial interpretations (CDPIs), context-dependent ordering examples (CDOEs) and the $ILP_{LOAS}^{context}$ framework from [LRB18]. An ASP^{ch} program is a program consisting of only normal rules, choice rules and hard constraints. All other programs considered in this document are additionally allowed to contain weak constraints.

Definition 1.1. ([LRB18]) A *context-dependent partial interpretation* (CDPI) is a pair $\langle e, C \rangle$, where e is a partial interpretation and C is an ASP^{ch} program, called a *context*. A *context-dependent ordering example* (CDOE) o is a tuple $\langle \langle e_1, C_1 \rangle, \langle e_2, C_2 \rangle, op \rangle$, where the first two elements are CDPIs and op is a binary comparison operator ($<$, $>$, $=$, $\leq$, $\geq$ or $\neq$). P is said to *bravely respect* o if $\exists A_1 \in AS(P \cup C_1), \exists A_2 \in AS(P \cup C_2)$ such that A_1 extends e_1 , A_2 extends e_2 and $\langle A_1, A_2, op \rangle \in ord(P, AS(P \cup C_1) \cup AS(P \cup C_2))$. A program P is said to *cautiously respect* o if $\forall A_1 \in AS(P \cup C_1), \forall A_2 \in AS(P \cup C_2)$ such that A_1 extends e_1 and A_2 extends e_2 , $\langle A_1, A_2, op \rangle \in ord(P, AS(P \cup C_1) \cup AS(P \cup C_2))$.

Definition 1.2. ([LRB18]) A *Context-dependent Learning from Ordered Answer Sets* ($ILP_{LOAS}^{context}$) task is a tuple $T = \langle B, S_M, \langle E^+, E^-, O^b, O^c \rangle \rangle$ where B is an ASP program called the background knowledge, S_M is the set of rules allowed in the hypotheses (the hypothesis space), E^+ and E^- are finite sets of CDPIs called, respectively, positive and negative examples, and O^b and O^c are finite sets of CDOEs over E^+ called, respectively, brave and cautious context-dependent orderings. A hypothesis $H \subseteq S_M$ is an inductive solution of T (written $H \in ILP_{LOAS}^{context}(T)$) if and only if:

1. $\forall \langle e^+, C \rangle \in E^+, \exists A \in AS(B \cup C \cup H)$ st A extends e^+
2. $\forall \langle e^-, C \rangle \in E^-, \nexists A \in AS(B \cup C \cup H)$ st A extends e^-
3. $\forall o \in O^b, B \cup H$ bravely respects o
4. $\forall o \in O^c, B \cup H$ cautiously respects o

2 Meta-program injection

The idea of *meta-program injection* is that a task may be given along with an $\mathcal{ASP}^{ch}$ program whose answer sets define the subsets of the examples that, if covered by a hypothesis H , mean that this H is an acceptable solution. Note that H is not necessarily a solution of the whole task, as it may not cover all of the examples. We say that H is an inductive solution of T with respect to the injection. This notion is formalised by the following definition. The definition assumes that every example in the task has a unique identifier (which is a constant) – given an example e , e_{id} denotes the unique identifier of e .

Definition 2.1. Let $T = \langle B, S_M, \langle E^+, E^-, O^b, O^c \rangle \rangle$ be an $ILP_{LOAS}^{context}$ task and $\mathcal{Q}$ be an $\mathcal{ASP}^{ch}$ program. Given any interpretation I of $\mathcal{Q}$, T_I denotes the task $\langle B, S_M, \langle E_I^+, E_I^-, O_I^b, O_I^c \rangle \rangle$, where:

- $E_I^+ = \{e \in E^+ \mid \text{example_active}(e_{id}) \in I\}$
- $E_I^- = \{e \in E^- \mid \text{example_active}(e_{id}) \in I\}$
- $O_I^b = \{o \in O^b \mid \text{example_active}(o_{id}) \in I\}$
- $O_I^c = \{o \in O^c \mid \text{example_active}(o_{id}) \in I\}$

A hypothesis $H \subseteq S_M$ is said to be an *inductive solution* of T with respect to the injection of $\mathcal{Q}$ if and only if $\exists I \in AS(\mathcal{Q})$ such that $H \in ILP_{LOAS}^{context}(T_I)$.

One should note that in the implementation of ILASP, there are several predicates which are considered “reserved” and which should not be used in $\mathcal{Q}$. If these are used, then the behaviour of ILASP is undefined (as they may interfere with ILASP’s own meta-representation). To see a full list of these predicates, please run “ILASP --version=3 --reserved-predicates”.

References

- [LRB18] Mark Law, Alessandra Russo, and Krysia Broda. The complexity and generality of learning answer set programs. *Artificial Intelligence*, 259:110–146, 2018.