Answer Set Programming in Intuitionistic Logic

Aleksy Schubert and Paweł Urzyczyn[1]

University of Warsaw
[alx,urzy]@mimuw.edu.pl

Abstract

We propose the first interpretation of propositional answer set programming (ASP) in terms of intuitionistic proof theory, in particular in terms of simply typed lambda calculus. While connections between ASP and intuitionistic logic are well-known, they usually take the form of characterizations of stable models with the help of some intuitionistic theories represented by specific classes of Kripke models. As such the known results are model-theoretic rather than proof-theoretic. In contrast, we propose an interpretation of ASP in terms of constructive proofs.

Answer Set Programming (ASP) is a programming paradigm originated from logic programming with negation understood as "fixpoint" [1, 2, 3]. The paradigm proved useful by reducing search problems to so caled stable models of declarative programs.

It has been observed long ago [5, 6, 7] that ASP can be interpreted using certain intuitionistic theories or intermediate logics, of which *equilibrium logic* of Pearce [7] is the most fundamental example. This is commonly summarized as applying intuitionistic logic to ASP. But what is actually done is representing answer sets as specific Kripke models (often two-state models).

We propose another way to interpret ASP in intuitionistic logic, namely we want to represent ASP inference (entailment in stable models) by intuitionistic provability. We use the ordinary intuitionistic logic (without any additional axioms) for this purpose. Our proof-theoretical account brings new insights to the operational semantics of answer set programming. This seems to suggest that intuitionistic provers can serve as natural ASP-solvers. Yet differently, perhaps intuitionistic logic (together with its highly intuitive and natural lambda-notation) should itself be seen as a programming paradigm, competitive to ASP?

In the present note we only consider propositional ASP and we reduce it to the implicational fragment (in particular no negation is needed) of propositional intuitionistic logic (IPC). We believe this approach should turn out very flexible by easily accomodating various extensions of ASP. In particular, a similar approach should be applicable to first-order ASP.

Propositional ASP: A *literal* is a propositional atom or its negation. A *clause* is an expression of the form $X :- X_1, \ldots, X_n$, where X is a propositional atom and $X_1, \ldots, X_n$ are literals. A *program* is a finite set of clauses. A *model* is a set M of atoms, identified with a valuation v_M such that $v_M(X) = \mathit{true}$ if and only if $X \in M$. Given a program P and a model M, we obtain P^M from P as follows. First, for all $X \notin M$, we delete $\neg X$ from the rhs of all clauses of P. Then, for all $X \in M$, delete all clauses of P with $\neg X$ at the rhs. Now the *interpretation* of P under M, denoted $I(P, M)$, is defined as the least fixed point of the operator:

$$F(\mathcal{S}) = \mathcal{S} \cup \{X \mid \text{there is a clause } X :- X_1, \ldots, X_n \text{ in } P^M \text{ such that all } X_i \text{ are in } \mathcal{S}\}.$$

A model M is a *stable* model of P (or it is an *answer set* for P) iff $M = I(P, M)$. We also say that P *entails* an atom X *under SMS*, written $P \models_{SMS} X$, iff every stable model of P satisfies X. It is known [2, 4] that the existence of a stable model and the entailment under SMS are, respectively, NP and *co*-NP complete problems.

Intuitionistic logic: We consider formulas of minimal propositional logic with $\to$ as the only propositional connective. That is, our formulas are just simple types, and intuitionistic proofs can be identified with simply-typed lambda-terms.

Given a program P we define an implicational formula φ such that P entails X under stable model semantics if and only if φ is intuitionistically provable. Let $X_1, \ldots, X_n$ be all propositional atoms occurring in P, including X. Without loss of generality we assume that X_i and $\neg X_i$ never occur together in the rhs of a clause.

Assume that P consists of m clauses, numbered from 1 to m. The vocabulary of φ consists of atoms $0, 1, \ldots, n, X_1, \ldots, X_n, \overline{X}_1, \ldots, \overline{X}_n, X_1!, \ldots, X_n!, X_1?, \ldots, X_n?, A, B, K_1, \ldots, K_m$.

The formula φ has the form $\psi_1 \to \cdots \to \psi_d \to 0$. The assumption formulas $\psi_1, \ldots, \psi_d$ are defined below. The first n assumptions are as follows:

$$\psi_1 = (X_1 \to 1) \to (\overline{X}_1 \to 1) \to 0, \qquad \ldots \qquad \psi_n = (X_n \to n) \to (\overline{X}_n \to n) \to n - 1.$$

The next m assumptions are the clauses of P, where every atom X is replaced by $X!$ and every $\neg X$ is replaced by $\overline{X}$. Then we have three formulas: $X \to n$, $A \to n$, $B \to n$, with target n. For every $i = 1, \ldots, n$, there are assumptions $\overline{X}_i \to X_i! \to A$ and $X_i \to X_i? \to B$. For $i = 1, \ldots, n$, there is an assumption $(X_i? \to K_{s_1}) \to \cdots \to (X_i? \to K_{s_{r_i}}) \to X_i?$, where $s_1, \ldots, s_{r_i}$ are (numbers of) all the clauses of P with target X_i.

If the atom X_i occurs at the rhs of the sth clause of P then there is an assumption $X_i? \to K_s$. And if $\neg X_i$ occurs at the rhs of the sth clause of P then there is an assumption $X_i \to K_s$.

The formulas naturally reflect the semantical character of the definition of ASP. Their intended meaning is as follows: Formulas $\psi_1, \ldots, \psi_n$ choose a binary valuation of atoms in a certain model. In order to prove 0 one proves n under assumptions representing every model. Now there are three ways in which the entailment $P \models X$ holds in a stable model: either X holds, or the model is unstable because P is unsound (proves too much) or the model is unstable because P is incomplete (does not prove what is needed). The assumption $X \to n$ can be used to complete the proof when X holds in the model. The two other possibilities are represented by A and B, respectively. One proves A when P is unsound for our model: it forces some X_i to hold (that is, $X_i \in I(P, M)$), but $\overline{X}_i$ is chosen. Proving B means that P^M is unable to derive an atom X_i which is present in the model. The propositional atoms $X?$ and K_s are understood as "X has no proof" and "the target of the sth clause has no proof", respectively. The formula with target $X_i?$ should be understood as follows: no clause with target X_i has a proof unless such a proof recursively refers to the proof goal X_i.

References

[1] Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Commun. ACM*, 54(12):92–103, December 2011.

[2] Phokion G. Kolaitis and Christos H. Papadimitriou. Why not negation by fixpoint? In *PODS '88*, pages 231–239, ACM, 1988.

[3] Vladimir Lifschitz. What is answer set programming? In *AAAI'08*, pages 1594–1597. AAAI Press, 2008.

[4] Wiktor Marek and Mirosław Truszczyński. Autoepistemic logic. *J. ACM*, 38(3):587–618, July 1991.

[5] Mauricio Osorio, Juan A. Navarro, and José Arrazola. Applications of intuitionistic logic in answer set programming. *Theory Pract. Log. Program.*, 4(3):325–354, May 2004.

[6] David Pearce. Stable inference as intuitionistic validity. *The Journal of Logic Programming*, 38(1):79–91, 1999.

[7] David Pearce. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence*, 47(1-2):3–41, 2006.