Toward an Integration of Answer Set and Constraint Solving

S. Baselice, P.A. Bonatti, M. Gelfond

Submitted

The Problem

Current answer set solvers start with grounding of a logic program, P.

If P has a variable which ranges over a finite domian then $ground(P)$ may be too large for the solvers.

We need to avoid complete grounding of P!

The Basic Idea

- **Expend the language of ASP to allow:**

1. regular relations;

2. relations over predefined finite domains;

3. mixed relations;

- **A program P of the new language consists of a regular program R and a collection C of constraints of the form**

$\leftarrow body.$

where *body* is a set of arbitrary literals.

Use constraint satisfaction methods to solve C.

Example

Consider a logic program R describing possible agent's trajectories in some dynamic domain.

Each trajectory is of the form:

$\langle \sigma_0, a_0, \sigma_1, \ldots, a_{n-1}, \sigma_n \rangle$

To reason about actual time of various events R can be used in conjunction with constraints:

$\leftarrow time(I_1, T_1), time(I_2, T_2), I_1 < I_2, T_1 \geq T_2$

$\leftarrow time(I, T_1), time(I, T_2), T_1 \neq T_2$

$\leftarrow time(I, T_1), time(I+1, T_2), o(A, I), |T_2 - T_1| < 3$

What is done?

- **Define syntax and semantics of the new language, ASP^c.**

- **Designed and proved correctness of an algorithm which computes answer sets of ASP^C. The algorithm combines the "classical" ASP algorithm with the use of CSP solvers.**

- **Started an implementation of the algorithm. Even the simplest implementation allows us to reason with time ranging over integers from 0 to 600000 which is absolutely impossible with ASP solvers.**