KNOWLEDGE REPRESENTATION, REASONING AND DECLARATIVE PROBLEM SOLVING

Chitta Baral

Department of Computer Science and Engg.
Arizona State University
Tempe, AZ 85287
chitta@asu.edu

http://www.public.asu.edu/∼cbaral

Class 3: Answer set semantics of AnsProlog* and solvers

Answer set semantics of AnsProlog*

- Given a program, what can we conclude from it? What are the world models (also called stable models, answer sets) of the program?
  - A rule of AnsProlog*

    $$L_0 \ or \ \ldots \ or \ L_k \leftarrow \quad L_{k+1}, \ldots, L_m, \ \texttt{not} \ L_{m+1}, \ldots, \texttt{not} L_n.$$

    means that if L_{k+1}, ..., L_m are true and if L_{m+1}, ..., $L_n$can assumed to be false, then at least one of L_0, ..., L_k must be true.
  - If $k < 0$, the head of the rule is empty, we replace the head as $\perp$. $\perp$ means inconsistent or FALSE. For example the statement '$\perp \leftarrow b$' means that if b is true, then that is a wrong (inconsistent) world view.

- There are different subsets of $AnsProlog^*$ (or $AnsProlog^{\neg, or, \perp}$). Some of them are listed in the table below.

Language	Specification	Complexity
$AnsProlog^{or}$	L_is are atoms, and $k \geq 0$	Π_2P-complete
$AnsProlog$	L_is are atoms, and $k = 0$	NP-complete
$AnsProlog^{-not}$	L_is are atoms, $k = 0$, and $m = n$	P-complete

- For program with functional symbols, we need to ground it first. The grounded program may have a infinite set of rules.

Answer set solvers

- Smodels:
  - Command line:
    `lparse AnsPrologInputFile | smodels [number of stable models`
  - Input: Smodels only accepts $AnsProlog^{\perp}$.
  - Output: If no stable models are found false is returned, if no number of stable models is specified then only the first stable model is returned.
- DLV:
  - Commend line:
    `dlv [options] AnsProlog InputFile`.
  - Input: *dlv* accepts $AnsProlog^{\perp,or}$.

Some examples

Example 1 *The input program file:*

```
p :- a.
a :- b.
b.
```

The stable model returned by smodels is $\{b, a, p\}$.

Example 2 `a :- not b. b :- not a.`
which returned two models in smodels, the first one being $\{a\}$, *the second being* $\{b\}$.

Example 3 *The input program is as follows:*

```
anc(X, Y) :- par(X, Z), anc(Z, Y), dom(X), dom(Y), dom(Z).
anc(X, Y) :- par(X, Y), dom(X), dom(Y).
par(a, b).
par(b, c).
par(c, d).
dom(a).
dom(b).
dom(c).
doc(d).
hide dom(X).
```

Which returned $par(c, d)$, $par(b, c)$, $par(a, b)$, $anc(b, d)$, $anc(b, c)$, $anc(a, d)$, $anc(a, c)$, $anc(a, b)$ *in smodels.*
The program lparse that passes it's output to smodels, grounds all the variables in the program used as input.