CS395T, Spring 2011

Answer Set Solvers

Instead of Prolog, we will be using now software systems called *answer set solvers*, such as the system CLINGO created at the University of Potsdam. In terms of their functionality, these systems differ from Prolog in several ways. In the examples below, `clingo` is aliased to `/projects/tag/clingo-3.0.90/clingo`.

1. An answer set solver does not reply to queries; instead, it outputs all ground atoms to which Prolog would have replied `yes`.

```
% File family
person(s).  person(w).  person(a).  person(m).
male(a).  male(m).
parent(s,w).  parent(s,a).  parent(w,m).
female(X) :- not male(X), person(X).
brother(X,Y) :- parent(Z,X), parent(Z,Y), male(X), X!=Y.
```

```
> clingo family
Answer: 1
person(s) person(w) person(a) person(m) male(a) male(m)
parent(s,w) parent(s,a) parent(w,m) female(w) female(s) brother(a,w)
```

2. The functionality of answer set solvers is based on the concept of a "stable model," or an "answer set," of a logic program. In many cases, the stable models of a program are identical to the models of its completion, but there are exceptions. When a program has several stable models, an answer set solver can generate all of them. In principle, an answer set solver always terminates.

```
% File problem15.c
p :- not q.
q :- not p.
```

```
> clingo problem15c 0
Answer: 1
p
Answer: 2
q
```

```
% File problem19
q(X,Y) :- p(X,Y).
q(X,Y) :- p(X,Z), q(Z,Y).
p(a,b).  p(b,c).  p(c,b).
```

```
> clingo problem19 0
Answer: 1
p(a,b) p(b,c) p(c,b) q(c,b) q(b,c) q(a,b) q(c,c) q(a,c) q(b,b)
```

3. Answer set solvers can handle syntactic constructs that are not allowed in Prolog input files, such as “choice rules” and “constraints.”

```
% File choice-and-constraint
p(a).  p(b).  p(c).
{q(X)} :- p(X).
:- q(a), not q(b).
```

```
> clingo choice-and-constraint 0
Answer: 1
p(a) p(b) p(c)
Answer: 2
p(a) p(b) p(c) q(c)
Answer: 3
p(a) p(b) p(c) q(b)
Answer: 4
p(a) p(b) p(c) q(b) q(a)
Answer: 5
p(a) p(b) p(c) q(c) q(b)
Answer: 6
p(a) p(b) p(c) q(c) q(b) q(a)
```