Logic Programs with Constraints

According to our definition of a rule, the head of a rule is an atomic formula $P(\mathbf{t})$. We will now relax this condition and allow the head of a rule to be also the symbol $\bot$. Rules of the form

$$\bot \leftarrow F \tag{1}$$

are called *constraints*. Including constraints in a logic program does not affect its stability formulas. The definition of a stable model in the presence of constraints is stated as follows: A *stable model* of a logic program is an Herbrand interpretation that satisfies all its stability formulas <u>and the universal closures of all its constraints</u>. Since implication (1) is equivalent to $\neg F$, we could say alternatively: "and the universal closure of $\neg F$ for each of its constraints (1)."

Consider, for example, the program

$$\begin{array}{l} P(a), \\ P(b), \\ P(c), \\ \{Q(x)\} \leftarrow P(x), \\ \{R(x)\} \leftarrow P(x), \\ \bot \leftarrow Q(x) \wedge R(x). \end{array}$$

Its stable models are the Herbrand interpretations such that

(i) the extent of P is the universe $\{a, b, c\}$,

(ii) Q is a subset of P,

(iii) R is a subset of P,

(iv) Q is disjoint from R.

Here is how we can write this program in the language of CLINGO:

```
p(a;b;c).
{q(X)} :- p(X).
{r(X)} :- p(X).
:- q(X), r(X).
```

Problem 30^e. We would like to replace condition (iv) by the weaker condition: Q and R have at most one common element. Modify the CLINGO program above accordingly.

Problem 31^e. A *clique* in a graph is a subset of its vertices such that every two vertices in the subset are connected by an edge. Write a CLINGO program such that its stable models represent all cliques in a given graph. To test your program, use the graph with the vertices a, b, c, d and the edges

$$\{a, b\},\ \{b, c\},\ \{c, d\},\ \{d, a\},\ \{a, c\}.$$

Satisfying interpretations for a propositional formula in conjunctive normal form can be computed by running CLINGO on a program consisting of choice rules and constraints. For instance, the formula

$$(\neg p \vee q) \wedge (\neg p \vee r) \wedge (q \vee r) \wedge (\neg q \vee \neg r)$$

can be represented by the file

```
{p,q,r}.
:- p, not q.
:- p, not r.
:- not q, not r.
:- q, r.
```