

The Language of CLINGO: Choice, Intervals, Assignments, Constant Names

A rule of the form

$$P(\mathbf{t}) \leftarrow F \wedge \neg\neg P(\mathbf{t})$$

can be abbreviated as

$$\{P(\mathbf{t})\} \leftarrow F.$$

Rules of this form are called *choice rules*. For instance, the second program from Problem 27 can be written as

$$\begin{aligned} &P(a), \\ &P(b), \\ &P(c), \\ &\{Q(x)\} \leftarrow P(x), \end{aligned}$$

or, in the syntax of CLINGO, as

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

The input language of CLINGO allows us to abbreviate the first line as

```
p(a;b;c).
```

Similarly, the group of facts

```
p(5). p(6). p(7). p(8). p(9).
```

can be abbreviated as

```
p(5..9).
```

Problem 28^e. Consider the program

```
p(a;b;c). p(5..9).
{q(X)} :- p(X).
```

Guess how many stable models it has. Use CLINGO to check that your guess is correct.

Interval expressions, such as $5..9$, can be used also as part of “assignment formulas.” For instance, $X=5..9$ stands for the disjunction

$$x = 5 \vee x = 6 \vee x = 7 \vee x = 8 \vee x = 9.$$

Problem 29^e. Consider the one-rule program

```
{p(X,Y)} :- X=1..3, Y=1..4.
```

Guess how many stable models it has. Use CLINGO to check that your guess is correct.

The CLINGO symbol `#const` allows us to introduce names for constants. For instance, the program from Problem 29 can be written also as

```
#const m=3.  
#const n=4.  
{p(X,Y)} :- X=1..m, Y=1..n.
```