Declarative Programming:
Lecture Notes and Exercises

Part 5

If the program that we are converting into a set of formulas contains arithmetical operations, intervals, or pools, then the process involves the additional step of eliminating these constructs. This step modifies even ground rules—rules that do not contain variables.

The general definition of translation $\tau$ refers to a transformation that turns any ground term $t$ into a set $[t]$ of precomputed terms. The recursive definition of $[t]$ is given in Section 4.2 of the paper Abstract Gringo. From that definition we see, for instance, that

\[
[5] = \{5\}, \quad [6/2] = \{3\}, \\
[6/0] = \emptyset, \quad [a + 1] = \emptyset, \\
[5..7] = \{5, 6, 7\}.
\]

The definition applies not only to ground terms, but also to ground pools. For example,

\[
a;b;c = \{a, b, c\}.
\]

5.1. Calculate
(a) $[-2],$
(b) $[2; 5..6],$
(c) $[2/6],$
(d) $[(2..4) \times 2],$
(e) $[(2..4) + (2..4)].$

Consider a ground rule with the head $p(t)$, where $t$ is a term containing arithmetical operations or intervals. To apply transformation $\tau$ to this rule, we replace $p(t)$ by the conjunction of the atoms $p(r)$ for all terms $r$ from the set $[r]$. For example, the result of applying $\tau$ to the rule

\[
p(5..7) \leftarrow q
\]

is the formula

\[
p(5) \land p(6) \land p(7) \leftarrow q.
\]
In application to the rule
\[ p(\overline{5}/\overline{0}) \leftarrow q \]
\( \tau \) gives
\[ \top \leftarrow q. \]
(The set \( [\overline{5}/\overline{0}] \) is empty, and the conjunction of the empty set of formulas is understood as \( \top \).)

If \( p \) in the head is followed not by a single term \( t \) but by a list of terms, or by a pool, the process is similar. For instance, \( \tau \) turns
\[ p(\overline{5..6}, \overline{10..11}) \leftarrow q \]
into the formula
\[ p(\overline{5}, \overline{10}) \land p(\overline{5}, \overline{11}) \land p(\overline{6}, \overline{10}) \land p(\overline{6}, \overline{11}) \leftarrow q, \]
and it turns
\[ t(austin, \overline{88}; dallas, \overline{95}; houston, \overline{90}; san\_antonio, \overline{85}) \]
into
\[ t(austin, \overline{88}) \land t(dallas, \overline{95}) \land t(houston, \overline{90}) \land t(san\_antonio, \overline{85}). \]

For rules containing arithmetical operations, intervals, or pools in the body, the process is similar, except that disjunctions are used instead of conjunctions. For instance, the result of applying \( \tau \) to the rule
\[ q \leftarrow p(a; b; c) \]
is the formula
\[ q \leftarrow p(a) \lor p(b) \lor p(c). \]

To apply \( \tau \) to a non-ground rule containing arithmetical operations, intervals, or pools in the head, we form the set of its instances by substituting precomputed terms for variables in all possible ways, and then process the instances as described above. Consider, for example, the rule
\[ q(X+\overline{1}) \leftarrow p(X). \] (1)
Its instances are the rules
\[ q(r+\overline{1}) \leftarrow p(r) \] (2)
for all precomputed terms \( r \). If \( r \) is a numeral \( n \) then 
\[
q(n+1) \leftarrow p(n).
\] (3)
If \( r \) is not a numeral then \([r+\top]\) is empty, so that (2) turns into
\[
\top \leftarrow p(r).
\] (4)
Consequently the set of formulas obtained by applying \( \tau \) to (1) consists of formulas (3) for all integers \( n \) and formulas (4) for all precomputed terms \( r \) other than numerals.

5.2. Find the minimal model of the set of formulas obtained by applying \( \tau \) to the program consisting of rule (1) and the facts \( p(a) \) and \( p(5) \). Check that the minimal model matches the actual output of CLINGO.

5.3. (a) Find the result of applying \( \tau \) to the program
\[
p(0),
q(X-\top..X+\top) \leftarrow p(X).
\]
(b) Find its minimal model and check that it matches the actual output of CLINGO.

Recall that in the process of applying \( \tau \) to a rule containing comparisons we replace each comparison by the symbol \( \top \) if it is true, and by \( \bot \) if it is false (see Part 4 of these lecture notes). If a comparison, say \( t_1 < t_2 \), in the body of a rule contains arithmetical operations or intervals then we understand the condition “the comparison is true” to mean that \( r_1 < r_2 \) is true for at least one pair of precomputed terms \( r_1 \in [t_1], r_2 \in [t_2] \). For example, condition \( 5..7 < 3+3 \) is true, because \( 5 < 3+3 \). Condition \( 4 = 5..7 \) is false, because \( 4 \) is different from each of the numbers 5, 6, 7.

5.4. Determine which of the given conditions are true.
(a) \( 3..5 < 3..5 \).
(b) \( 3..5 > 3..5 \).
(c) \( 4 < 3..5 \).
(d) \( 4 = 3..5 \).
(e) \( 4 > 3..5 \).

5.5. (a) Find the result of applying \( \tau \) to the one-rule program
\[
p(X+\top0) \leftarrow X = 2..5.
\]
(b) Find its minimal model and check that it matches the actual output of CLINGO.