Declarative Programming:
Lecture Notes and Exercises

Part 4

The paper *Abstract Gringo* defines a translation $\tau$ that turns rules and programs into sets of formulas. This translation plays an important role in the theory of answer set programming.

As discussed in Part 3 of these lecture notes, some rules become “easy formulas” when we write them in theoretical notation. Such rules are not affected by the translation $\tau$ at all.

When we apply $\tau$ to a rule containing global variables, the first step is to form the set of instances of this rule by substituting precomputed terms for the variables in all possible ways. For example, the instances of the rule

$$warm(C) \leftarrow t(C, T1) \land t(austin, T2) \land T1 > T2$$

are the rules

$$warm(r3) \leftarrow t(r3, r1) \land t(austin, r2) \land r1 > r2$$

for arbitrary precomputed terms $r_1, r_2, r_3$. At the second step, we replace each comparison by the symbol $\top$ if it is true, and by $\bot$ if it is false. So the result of applying $\tau$ to rule (1) is the set consisting of the formulas

$$\begin{align*}
\text{warm}(r3) & \leftarrow t(r3, r1) \land t(austin, r2) \land \top \quad \text{if } r1 > r2, \\
\text{warm}(r3) & \leftarrow t(r3, r1) \land t(austin, r2) \land \bot \quad \text{if } r1 \leq r2.
\end{align*}$$

In many cases, the answers produced by clingo for a program $\Pi$ are identical to the minimal models of the translation $\tau\Pi$. Consider, for example, the program $\Pi$ consisting of rule (1) and the facts

$$t(austin, 88), \ t(dallas, 95), \ t(houston, 90), \ t(san\_antonio, 85).$$

The translation $\tau\Pi$ consists of formulas (2) and (3). We would like to find the minimal models of $\tau\Pi$. Set (2) can be simplified: it is equivalent to the set consisting of the formulas

$$warm(r3) \leftarrow t(r3, r1) \land t(austin, r2)$$

for all precomputed terms $r_1, r_2, r_3$ such that $r_1 > r_2$. Formulas (3), (4) are definite (see Part 2 of these lecture notes); consequently $\tau\Pi$ has a unique minimal model, which can be found by accumulating the atoms that are needed to satisfy all these formulas. To satisfy formulas (3), we should
include all of them in the minimal model. Which atoms need to be added to satisfy formulas (4)? There are 4 combinations of values of r₁, r₂, r₃ for which both t(r₃, r₁) and t(austin, r₂) are already included:

\[ r₁ = \text{88}, \ r₂ = \text{88}, \ r₃ = \text{austin}, \]
\[ r₁ = \text{95}, \ r₂ = \text{88}, \ r₃ = \text{dallas}, \]
\[ r₁ = \text{90}, \ r₂ = \text{88}, \ r₃ = \text{houston}, \]
\[ r₁ = \text{85}, \ r₂ = \text{88}, \ r₃ = \text{san_antonio}. \]

The first and the last of them do not satisfy the inequality \( r₁ > r₂ \), and we shouldn’t worry about them. To satisfy the formulas (4) corresponding to the other two combinations, we need to include

\[ \text{warm(dallas)}, \ \text{warm(houston)}. \]  \hspace{1cm} (5)

The set of atoms (3), (5) satisfies all formulas of \( \tau \Pi \), and consequently it is the minimal model of \( \tau \Pi \). This set is the only solution returned by CLINGO for program II.

For each of the following programs,

(a) calculate the result of applying the transformation \( \tau \),

(b) find its minimal model,

(c) check that the minimal model matches the actual output of CLINGO.

4.1.

\[ p(a), \]
\[ p(b), \]
\[ q(X,Y) \leftarrow p(X), p(Y). \]

4.2.

\[ p(a), \]
\[ p(b), \]
\[ q(X) \leftarrow p(X), X \neq a. \]

4.3.

\[ p(a), \]
\[ p(5), \]
\[ q \leftarrow p(X), p(Y), X > Y. \]

4.4.

\( \text{father(abraham, isaac)}, \)
\( \text{father(isaac, jacob)}, \)
\( \text{father(jacob, benjamin)}, \)
\( \text{ancestor}(X,Y) \leftarrow \text{father}(X,Y), \)
\( \text{ancestor}(X,Z) \leftarrow \text{ancestor}(X,Y), \text{ancestor}(Y,Z). \)