1 Introduction

TRAM (Term Rewriting Abstract Machine) is an abstract machine for order-sorted conditional term rewriting systems (OSCTRSs). The OSCTRSs [5] can serve as a general computation model for advanced algebraic specification languages such as OBJ [3] or CafeOBJ [4].

In TRAM, left-hand sides (LHSs) of rewrite programs (sets of rewrite rules) are encoded into discrimination nets [2]. Terms to be rewritten using the rules are compiled into pairs of self modifying programs (matching programs) and strategy lists. The matching programs match the original terms with the discrimination nets (i.e. LHSs). The strategy lists are sequences of labels (addresses) of the matching programs. They hold the information for controlling the order of rewritings. Right-hand sides (RHSs) of rewrite programs are compiled into pairs of matching program templates and strategy list templates. OBJ's evaluation strategies (E-strategies) [3], which allow us to control the order of rewritings, can be implemented easily and efficiently by using the strategy lists.

2 TRAM Programs

TRAM programs are rewrite programs with declarations of sorts, subsort relations, variables and operators. They are very similar to flat OBJ's modules. For example, the program defining a function that returns the n-th element of (infinite) natural numbers' lists is as follows:

```
sorts: Zero NzNat Nat List .
order: Zero < Nat NzNat < Nat .
ops: 0 : -> Zero s : Nat -> NzNat
     cons : Nat List -> List { strat: (1 0) }
     inf : Nat -> List
     nth : Nat List -> Nat .
rules: inf(X) -> cons(X, inf(s(X)))
       nth(0, cons(X, L)) -> X
       nth(s(X), cons(Y, L)) -> nth(X, L) .
```

The program declares four sorts, two subsort relations, three variables and five operators, and defines three rewrite rules. Figure 1 shows the reduction sequence for the term "nth(s(0), inf(0))" using the above program.
Fig. 1. Reduction sequence for the term “nth(s(0), inf(0))”

3 TRAM Architecture

TRAM consists of six regions and three processing units. The six regions are DNET, CR, CODE, SL, STACK and VAR. The three processing units are the rule compiler, the term compiler and the TRAM interpreter.

DNET is the region for discrimination nets [2] encoded from LHSs of rewrite programs. Compiled RHSs of rewrite programs are allocated on CR. Matching programs compiled from subject terms are allocated on CODE. SL contains a strategy list for a reduction strategy. STACK is the working place for pattern matching. VAR contains variable bindings for pattern matching.

Given a TRAM program, the rule compiler encodes the LHSs into a discrimination net and compiles the RHSs into pairs of matching program templates and strategy list templates. Inputing a subject term into TRAM, the term compiler compiles it into the matching program and generates the strategy list, and then the TRAM interpreter interprets the matching program to reduce the term according to the strategy list.

4 Matching Programs

In TRAM, subject terms are compiled into sequences of abstract instructions (matching programs) that are self modifying programs. The matching program $L_T$ for a term $T$ whose top operator $f$ is $N$-ary is as follows:

$$L_T: \text{match_sym } \textit{idx}_f$$

$$L_1,$$

$$\vdots,$$

$$L_N$$

where, $\textit{idx}_f$ is the index for $f$ and $L_i (i = 1, \ldots, N)$ is the label of the $i$th argument’s matching program. Figure 1 shows some terms and the corresponding matching programs.