The evaluation strategy for head normal form with and without on-demand flags

Masaki Nakamura and Kazuhiro Ogata

School of Information Science
Japan Advanced Institute of Science and Technology
Email: {masaki-n, ogata}@jaist.ac.jp

Abstract

We propose two conditions of the E-strategy with and without on-demand flags on which an evaluated term is always in head normal form. In rewriting with the E-strategy without (or with) on-demand flags, terms are evaluated according to a list of natural numbers (or integers) given to each function symbol. The first (or second) condition is that if there exists a rule such that a function symbol f occurs in its left-hand side and its i-th argument is not a variable, a list of f must contain i (or $-i$), and if f is also a defined one, a list of f must contain 0 at the end. While there is no restriction w.r.t. the first condition, the second one can only be applied to left-linear constructor TRSs. But, There are cases in which rewriting with the E-strategy with on-demand flags terminates properly while that with the E-strategy without on-demand flags does not. We also propose a method of obtaining normal forms if a way to get head normal forms is given.

1 Introduction

Reduction strategies play an important role for the term rewriting. Standard strategies such as the eager evaluation and the lazy evaluation decide a redex in a given term which has to be reduced next according to the structure of the whole term. The evaluation strategy (the E-strategy for short) [4][5][6], adopted by OBJ languages such as OBJ3 [3] and CafeOBJ [2][7], searches a redex position according to local strategies given to symbols, not the structure of a whole term. Local strategies are given to every symbol as integer lists which mean order in which terms are evaluated. Since we can choose local strategies flexibly, the E-strategy can express various strategies. However, it is difficult to simulate the lazy evaluation properly by the E-strategy, since the redex which has to be evaluated next is not determined by the structure of a whole term. To solve this matter the on-demand E-strategy has been proposed [6][7]. In the on-demand E-strategy, symbols have on-demand flags. While it is being examined whether a term matches with the left-hand side

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

of a rewrite rule, a subterm may be evaluated on demand if the on-demand flag of the root symbol is up. As example is given so that you can intuitively understand how to rewrite terms with the on-demand E-strategy.

Example 1.1 In CafeOBJ, local strategies are operator attributes which are given as integer lists. Each non-negative integer in the lists represents an argument which has to be evaluate: positive i stands for the i-th argument and 0 the whole term. For negative integers $-i$, the i-th arguments are not evaluated until so forced. Evaluation may be forced when the arguments are involved in matching. The following example is a specification of infinite lists in CafeOBJ with the on-demand E-strategy.

```
mod! TEST {
  [ T ]
  op _::_ : T T -> T      {strat: (-1 -2)}
  op 1st  : T   -> T      {strat: (1 0)}
  op 2nd  : T   -> T      {strat: (1 0)}
  op inf  : T   -> T      {strat: (0)}

  vars L M N : T

  eq inf(N) = N :: inf(N).
  eq 1st(N :: L) = N .
  eq 2nd(N :: (M :: L)) = M .
}
```

Local strategies are given as integer lists after "`strat:`". In the above example, the local strategy of the symbols $1st$ and $2nd$ is the integer list (1 0), which requires the argument be evaluated before the whole term. With the declaration (-1 -2) for the symbol ::, both arguments are not evaluated immediately but their on-demand flags are raised. The term $inf(N)$ is the infinite list whose elements are N. For example, the term $2nd(inf(a))$ is evaluated as follows:

$$2nd(inf(a)) = 2nd(a :: inf(a)) = 2nd(a :: (a :: inf(a))) = a.$$

Because of the list (1 0) of $2nd$, the E-strategy tries to evaluate the argument first. The term $inf(a)$ is rewritten to $a :: inf(a)$. Since the list of :: is (-1 -2), the on-demand flags of both arguments are raised and no evaluation is done. So the term $a :: inf(a)$ is the result of evaluating $inf(a)$. Next, the E-strategy tries to evaluate the whole term $2nd(a :: inf(a))$. It is failed at the second argument of :: to match the term to $2nd(N :: (M :: L))$. Since the on-demand flag is up, the subterm $inf(a)$ is evaluated to $a :: inf(a)$. The result term is $2nd(a :: (a :: inf(a)))$, which successfully matches with $2nd(N :: (M :: L))$. Finally we get the term a.

In this paper, we show two conditions of the E-strategy with and without on-demand flags, respectively, such that if the E-strategy succeeds in evaluat-

ing a term, the evaluated term is in head normal form. In the next section, we briefly review some basic notions of the term rewriting [1] and introduce the rewrite relation which simulates the E-strategy without on-demand flags [4]. We show the first condition for head normal form through an analysis of the structures of the left-hand side of each rules. In sections 4 and 5, we define the rewrite relation for the on-demand E-strategy and show the condition of local strategies for head normal form under the restriction of left-linear constructor term rewriting systems. We give examples such that the E-strategy without the on-demand flags can not be applied but the on-demand E-strategy goes well.

2 Preliminaries

In this section, we briefly review some basic notions of the term rewriting [1] and the E-strategy without on-demand flags [4].

2.1 The term rewriting

A signature Σ is a finite set of function symbols where every $f \in \Sigma$ has a fixed arity $ar(f) \in \mathbb{N}$. A countably infinite set of variables V is defined as $\Sigma \cap V = \emptyset$. A set of terms $T(\Sigma, V)$ (or T) is the smallest set defined as follows: $V \subset T(\Sigma, V)$ and $f(t_1, \ldots, t_n) \in T(\Sigma, V)$ for $t_1, \ldots, t_n \in T(\Sigma, V)$, $f \in \Sigma$ and $ar(f) = n$. For a set D, the set of all sequences whose elements are in D is denoted by D^*. The empty sequence is denoted by ε. We write a sequence of D^* as $a \cdot b \cdot c$ or $a \cdot p$ for $a, b, c \in D$ and $p \in D^*$. A set of positions $O(t) \subset \mathbb{N}_+^*$ of a term t is defined as follows:

$$O(t) = \begin{cases} \{\varepsilon\} & \text{if } t \in V \\ \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{i \cdot p \mid p \in O(t_i)\} & \text{if } t \equiv f(t_1, \ldots, t_n). \end{cases}$$

The subterm of t at a position $p \in O(t)$, denoted by $t|_p$, is defined as $t|_\varepsilon \equiv t$, $f(t_1, \ldots, t_n)|_{i \cdot p} \equiv t_i|_p$. The term $t[s]_p$ is obtained from t by replacing the subterm at position p by s. The symbol at a position p of a term t is denoted by $(t)_p$. Especially the symbol at the root position $(t)_\varepsilon$ of t is called the root symbol of t. The set of variable positions in t is denoted by $O_V(t) = \{p \in O(t) \mid t|_p \in V\}$ and that of non-variable positions in t is $O_\Sigma(t) = O(t) \setminus O_V(t)$. The lexicographic order $_{lex}$ on $\mathbb{N}_+^*$ is defined as

$$p <_{lex} q \overset{\text{def}}{\iff} \begin{cases} p = \varepsilon \neq q \text{ or} \\ p = i \cdot p' \text{ and } q = j \cdot q' \text{ where } i < j \text{ or} \\ \qquad\qquad\qquad\qquad i = j \text{ and } p' <_{lex} q'. \end{cases}$$

A term t is called linear if $(t)_p \neq (t)_{p'}$ for any $p, p' \in O_V(t)$ such that $p \neq p'$. A map from variables to terms is called a substitution. A substitution over

terms is defined as a homomorphic extension. For a term t and a substitution θ, $t\theta$ is written instead of $\theta(t)$. A term t is called an instance of a term s if there exists θ such that $t = s\theta$.

Let $\rightarrow$ be a binary relation on a set D. We write $a \rightarrow b$ if $\langle a, b\rangle \in \rightarrow$ and $a, b \in D$. The transitive-reflexive closure of $\rightarrow$ is denoted by $\rightarrow^*$. An element $a \in D$ is in normal form w.r.t.$\rightarrow$ if there is no $b \in D$ such that $a \rightarrow b$. A set of all elements which is in normal form w.r.t.$\rightarrow$ is denoted by $NF_{\rightarrow}$ or only NF if no confusion exists. A term rewriting system (TRS for short) is a set R of rewrite rules. A rewrite rule is a pair of terms, denoted by $l \rightarrow r$, such that the left-hand side l is not a variable and any variable in the right-hand side r occurs in the left-hand side l. The term t is a redex if t is an instance of the left-hand side of a rewrite rule. A rewrite relation $\rightarrow_R$ is a binary relation on T defined as follows:

$$t \rightarrow_R s \stackrel{\text{def}}{\iff} \begin{array}{l}\text{there exist } l \rightarrow r \in R,\ p \in O(t) \text{ and } \theta \\ \text{such that } t|_p \equiv l\theta \text{ and } s \equiv t[r\theta]_p.\end{array}$$

A term t is in head normal form w.r.t. $\rightarrow_R$ if there is no redex u such that $t \rightarrow_R^* u$. If a term is in head normal form, the root symbol cannot be modified in any reduction sequence from the term. Therefore, a term of which all subterms are in head normal form is in normal form. A set of defined symbols $D(R)$ and constructor symbols $C(R)$ are defined as follows:

$$D(R) = \{f \in \Sigma \mid (l)_\varepsilon = f, l \rightarrow r \in R\},$$
$$C(R) = \Sigma \setminus D(R).$$

A TRS R is a constructor TRS if $(l)_p \notin D(R)$ for each $p \neq \varepsilon$ and $l \rightarrow r \in R$. If l is a linear term for each $l \rightarrow r \in R$, a TRS R is a left-linear TRS.

2.2 The evaluation strategy

The set of all lists of D, denoted by $L(D)$, is defined as $L(D) = \{nil\} \cup \{a :: l \mid a \in D, l \in L(D)\}$ where nil is the empty list. $[a_1, a_2, \ldots, a_n]$ is written instead of $a_1 :: (a_2 :: (\cdots (a_n :: nil)))$ and $l1@l2$ is the list appended $l1$ to $l2$. We define Σ_L, V_L and T_L as $\Sigma_L = \{f_l \mid f \in \Sigma, l \in L(\{0, \ldots, ar(f)\})\}$, $V_L = \{x_{nil} \mid x \in V\}$ and $T_L = T(\Sigma_L, V_L)$.

Definition 2.1 An E-strategy map φ is a map from $\Sigma \cup V$ to $L(\mathbb{N})$ such that $\varphi(f) \in L(\{0, \ldots, ar(f)\})$ for every $f \in \Sigma$ and $\varphi(x) = nil$ for every $x \in V$. We extend the E-strategy map from T to T_L as

$$\varphi(f(t_1, \ldots t_n)) \equiv f_{\varphi(f)}(\varphi(t_1), \ldots, \varphi(t_n)).$$

The map $erase : T_L \rightarrow T$ erases all lists of function symbols and variables in a term defined as follows:

$$erase(f_l(t'_1, \ldots t'_n)) \equiv f(erase(t'_1), \ldots, erase(t'_n)).$$

Note that although $erase(\varphi(t)) \equiv t$, it may be possible that $\varphi(erase(t')) \not\equiv t'$.

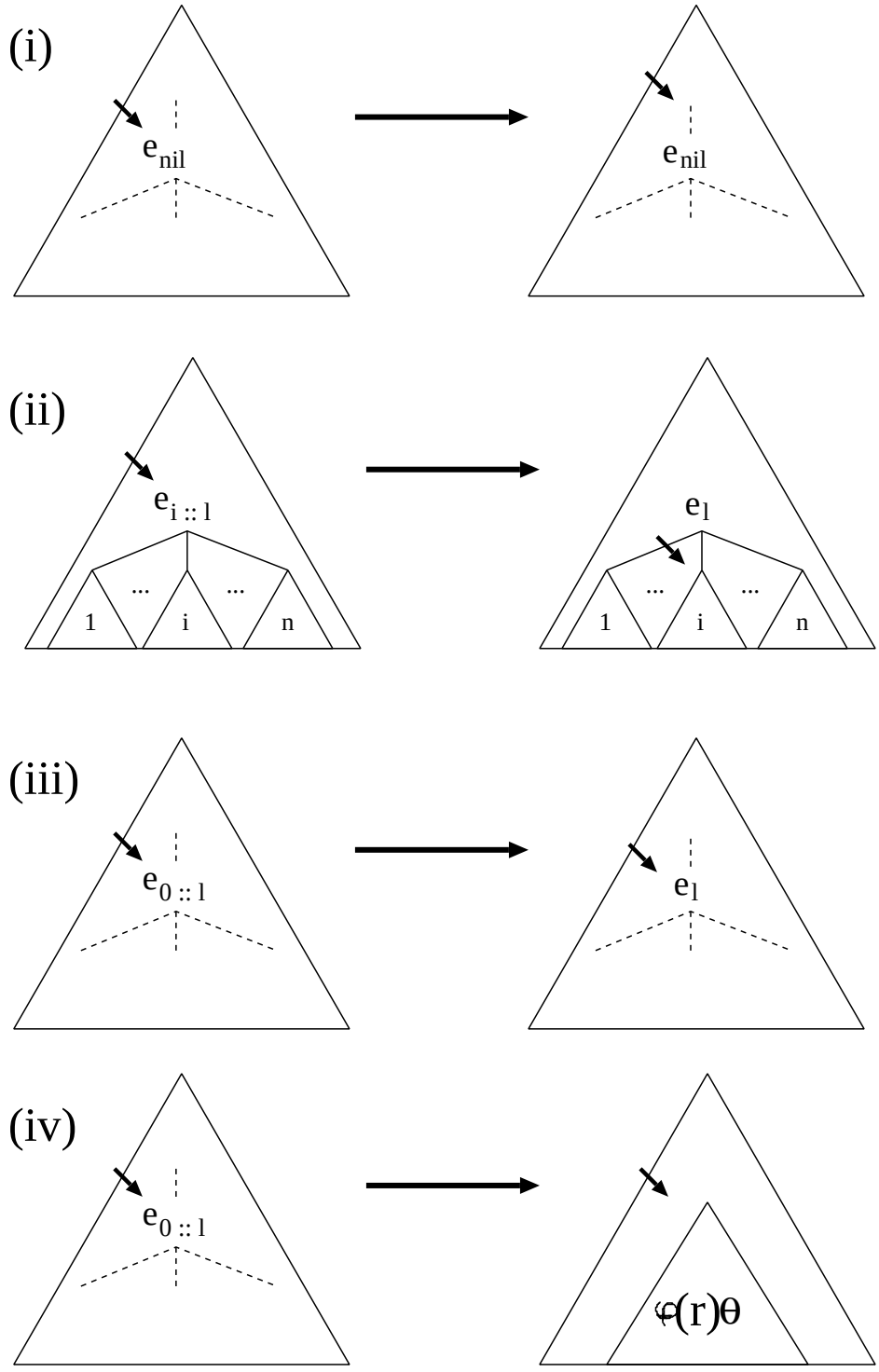

Fig. 1. Depiction of the four conditions of Definition 2.2

Definition 2.2 An evaluation map $eval_\varphi : T \to \mathcal{P}(T)$ of φ is defined as $eval_\varphi(s) = \{erase(t) \in T \mid \langle \varphi(s), \varepsilon \rangle \to_\varphi^* \langle t, \varepsilon \rangle \in NF\}$. An E-strategy rewrite relation (or φ-rewrite relation) $\to_\varphi$ is a binary relation on $T_L \times \mathbb{N}_+^*$ defined

as follows: $\langle t, p\rangle \to_\varphi \langle s, q\rangle$ if and only if $p \in O(t)$ and one of the following conditions is satisfied:

(i) $(t)_p = e_{nil}$, $s \equiv t$ and $p = q \cdot i$ for some i,

(ii) $t|_p \equiv e_{i::l}(t_1, \ldots, t_n)$ with $i > 0$, $s \equiv t[e_l(t_1, \ldots, t_n)]_p$ and $q = p \cdot i$,

(iii) $t|_p \equiv e_{0::l}(t_1, \ldots, t_n)$, $erase(t|_p)$ is not a redex, $s \equiv t[e_l(t_1, \ldots, t_n)]_p$, $q = p$,

(iv) $t|_p \equiv e_{0::l}(t_1, \ldots, t_n) \equiv l'\theta$, $erase(l') \equiv l$, $s \equiv t[\varphi(r)\theta]_p$ for some θ and $l \to r \in R$, $q = p$.

Roughly speaking, $\langle t, p\rangle$ means that $t|_p$ has to be evaluated next. Figure 1 depicts the four condition in Definition 2.2. A short arrow represents the second element of $\langle t, p\rangle$.

Example 2.3 Let $\varphi(head) = \varphi(tail) = [1, 0]$, $\varphi(cons) = nil$ and

$$R = \begin{cases} head(cons(x, y)) \to x \\ tail(cons(x, y)) \to y. \end{cases}$$

The E-strategy evaluates the term $t \equiv head(cons(tail(cons(x, z)), y))$ as follows:

$$\begin{aligned}
& \langle\, head_{[1,0]}(cons_{nil}(tail_{[1,0]}(cons_{nil}(x_{nil}, y_{nil})), z_{nil})),\ \varepsilon\rangle \\
\to_\varphi\ & \langle\, head_{[0]}(cons_{nil}(tail_{[1,0]}(cons_{nil}(x_{nil}, y_{nil})), z_{nil})),\ 1\rangle \quad \text{by (ii.)} \\
\to_\varphi\ & \langle\, head_{[0]}(cons_{nil}(tail_{[1,0]}(cons_{nil}(x_{nil}, y_{nil})), z_{nil})),\ \varepsilon\rangle \quad \text{by (i.)} \\
\to_\varphi\ & \langle\, tail_{[1,0]}(cons_{nil}(x_{nil}, y_{nil})),\ \varepsilon\rangle \quad \text{by (iv.)} \\
\to_\varphi\ & \langle\, tail_{[0]}(cons_{nil}(x_{nil}, y_{nil})),\ 1\rangle \quad \text{by (ii.)} \\
\to_\varphi\ & \langle\, tail_{[0]}(cons_{nil}(x_{nil}, y_{nil})),\ \varepsilon\rangle \quad \text{by (i.)} \\
\to_\varphi\ & \langle\, y_{nil},\ \varepsilon\rangle \quad \text{by (iv.).}
\end{aligned}$$

Since a pair $\langle y_{nil}, \varepsilon\rangle$ is in normal form, $y \in eval_\varphi(t)$. We can easily see that for any pair in normal form $\langle t, p\rangle \in NF$, $p = \varepsilon$ and $(t)_\varepsilon = e_{nil}$ for some $e \in \Sigma \cup V$. Because if $(t)_\varepsilon \neq e_{nil}$, the pair can be rewritten by (ii),(iii) or (iv) in Definition 2.2. If $p \neq \varepsilon$ and $(t)_\varepsilon = e_{nil}$, it can be rewritten by (i) in Definition 2.2.

3 The evaluation strategy for head normal form

Since the search for the next redex of a term does not depend on the structure of a whole term in the E-strategy, not all evaluated term are in normal form. For example, if we choose every local strategy the empty list, i.e. $\varphi(f) = nil$ for any $f \in \Sigma$, any term, even a redex itself, cannot be rewritten. For this reason, it is important to find conditions of E-strategy maps on which any evaluated term is in normal form. Such a condition is proposed in [4].

Proposition 3.1 (from [4]) *Let φ be an E-strategy map which satisfies following conditions:*

- *$\varphi(f)$ contains $1, \ldots, ar(f)$ if $f \in \Sigma$,*
- *the last element of $\varphi(f)$ is 0 if $f \in D(R)$.*

If $t \in eval_\varphi(t')$ for some $t' \in T$, the term t is in normal form w.r.t. $\to_R$.

An E-strategy map satisfing the above condition evaluates all arguments before the whole term. Inductively, the arguments are in normal form. If the head symbol of the term is not in $D(R)$, the whole term is in normal form. Otherwise, the element 0 of the head symbol's list should be removed in the last step of the E-strategy, which means that the term is not a redex and in normal form.

In this paper, we propose two similar conditions w.r.t. head normal forms with and without on-demand flags. If a condition w.r.t. head normal form can be gained, we can deduce a condition w.r.t. normal form easily.

Theorem 3.2 *Let φ be an E-strategy map such that any evaluated term is in head normal form. We define an E-strategy map φ' as $\varphi'(f) = \varphi(f)@[i_1, \ldots, i_n]$ where for any $i \in \{1, \ldots, ar(f)\} \setminus \varphi(f)$ there exists $i \in [i_1, \ldots, i_n]$. If $t \in eval_{\varphi'}(t')$ for some $t' \in T$, the term t is in normal form w.r.t. $\to_R$.*

Proof. We prove the claim by the induction of the structure of t. The case where $t \in V$ is trivial. We assume that $t \equiv f(t_1, \ldots, t_n)$. From Definition 2.2, the list $\varphi'(f) = \varphi(f)@[i_1, \ldots, i_n]$ of the root symbol f should transform to *nil* in the reduction sequence. From the assumption, we can easily verify that the whole term is in head normal form when the list of the root symbol has become $[i_1, \ldots, i_n]$ which is appended to the original list of the symbol to get φ'. Since a term reduced from a term in head normal form is also in head normal form, the evaluated term t is in head normal form. From the induction hypothesis, all subterms of t is in normal form since all arguments are in the list $\varphi'(f)$. Clearly a term in head normal form is not a redex. The evaluated term t is in normal form. □

In this section the one of the two conditions w.r.t. head normal form is proposed. The other is discussed in the next section.

Definition 3.3 A set of the linear variable positions of a term, which is a subset of the variable positions, and its complement are defined as follows:

$$LV(t) = \{p \in O_V(t) \mid (t)_p = (t)_q \Rightarrow p = q\},$$
$$\overline{LV(t)} = O(t) \setminus LV(t).$$

Lemma 3.4 *If a term t is an instance of a term l, i.e. $t \equiv l\theta$ for a substitution θ, the term $t[s]_p$ is also an instance of the term l for any term s and any position p under linear variable positions of l, i.e. $p \in O(t) \setminus \overline{LV(l)}$.*

Proof. We can easily see that there is a position $q \in LV(l)$ such that $p = q \cdot q'$. Let θ' be the substitution defined as follows:

$$\theta'(x) = \begin{cases} t|_q[s]_{q'} \text{ if } x \equiv (l)_q \\ \theta(x) \quad \text{if otherwise.} \end{cases}$$

The term $t[s]_p$ is the instance of the term l by the substitution θ'. □

Suppose that a term s is rewritten to a term s' at a position p. If the term s' is a redex by a rule $l \to r \in R$ and a position $p \in LV(l)$, the original term s is a redex by the same rule. Through the analysis of the linear variable positions of the left-hand side of each rule, we obtain the condition of an E-strategy map on which any evaluated term is in head normal form.

Definition 3.5 A set of the linear variable arguments of a function symbol and its complement are defined as follows:

$$LV_R(f) = \{i \mid p \cdot i \in LV(l) \text{ for all } l \to r \in R \text{ and } (l)_p = f\}.$$
$$\overline{LV_R(f)} = \{1, \ldots, ar(f)\} \setminus LV_R(f).$$

Theorem 3.6 *Let φ be an E-strategy map which satisfies following conditions:*

- *$i \in \varphi(f)$ if $i \in \overline{LV_R(f)}$,*
- *the last element of $\varphi(f)$ is 0 if $f \in D(R)$.*

If $t \in eval_\varphi(t')$ for some $t' \in T$, the term t is in head normal form w.r.t. $\to_R$.

Proof. We prove the claim by induction on the structure of t. It is trivial if $t \in V$ or $(t)_\varepsilon \in C(R)$. In the case where $(t)_\varepsilon \in D(R)$, we assume that t is not in head normal form. There must be a rule $l \to r \in R$ and a substitution θ such that $t \to_R^* l\theta$. From the first condition and the induction hypothesis, the subterm $t|_p$ is in head normal form for any $p \in \overline{LV(l)}$. Hence, only a subterm at a position $p \in O(t) \setminus \overline{LV(l)}$ can be reduced. From Lemma 3.4, the term t is a redex itself and should be rewritten from the second condition. □

Example 3.7 Consider the following TRS:

$$R = \begin{cases} head(cons(x, y)) \to x \\ tail(cons(x, y)) \to y \\ from(x) \to cons(x, from(s(x))). \end{cases}$$

Let φ be an E-strategy map such that $\varphi(head) = \varphi(tail) = [1, 0]$, $\varphi(from) = [0]$ and $\varphi(cons) = \varphi(s) = nil$, which satisfies the condition of Theorem 3.6. We first show the reduction sequence from the term $t_1 \equiv from(x)$:

$$\begin{aligned} \langle \varphi(t_1), \varepsilon \rangle \equiv \quad & \langle\ from_{[0]}(x_{nil}), && \varepsilon \rangle \\ \to_\varphi \ & \langle\ cons_{nil}(x_{nil}, from_{[0]}(s_{nil}(x_{nil}))), && \varepsilon \rangle. \end{aligned}$$

The result term $cons(x, from(s(x)))$ is not in normal form but in head normal form. The reduction sequence from the term $t_2 \equiv head(tail(from(x)))$ is given next:

$$\begin{aligned}
\langle \varphi(t_2), \varepsilon \rangle \equiv \quad & \langle\ head_{[1,0]}(tail_{[1,0]}(from_{[0]}(x_{nil}))), && \varepsilon \rangle \\
\to_\varphi^* \ & \langle\ head_{[0]}(tail_{[0]}(from_{[0]}(x_{nil}))), && 1 \cdot 1 \rangle \\
\to_\varphi^* \ & \langle\ head_{[0]}(tail_{[0]}(cons_{nil}(x_{nil}, from_{[0]}(s_{nil}(x_{nil}))))), && 1 \rangle \\
\to_\varphi \ & \langle\ head_{[0]}(from_{[0]}(s_{nil}(x_{nil}))), && 1 \rangle \\
\to_\varphi^* \ & \langle\ head_{[0]}(cons_{nil}(s_{nil}(x_{nil}), from_{[0]}(s_{nil}(s_{nil}(x_{nil}))))), && \varepsilon \rangle \\
\to_\varphi \ & \langle\ s_{nil}(x_{nil}), && \varepsilon \rangle
\end{aligned}$$

The result is the term $s(x)$ which is the second element of $[0, 1, 2, \ldots]$.

However, if an E-strategy map satisfying the conditions of Proposition 3.1 is given, evaluation of t_2 does not terminate.

From Theorems 3.2 and 3.6, we obtain the following new condition w.r.t. normal form.

Corollary 3.8 *Let φ be an E-strategy map which satisfies the following conditions:*

- *$\varphi(f)$ contains $1, \ldots, ar(f)$,*
- *$\varphi(f) = [i_1, \ldots, i_n, 0, \ldots]$ if $f \in D(R)$, where, for any $i \in \overline{LV_R(f)}$, $i = i_k$ for some $1 \leq k \leq n$.*

If $t \in eval_\varphi(t')$ for some $t' \in T$, the term t is in normal form w.r.t. $\to_R$.

Proof. From Theorems 3.2 and 3.6. □

Example 3.9 Let R be a TRS such that

$$R = \begin{cases} +(x, 0) \to x \\ +(x, s(y)) \to s(+(x, y)). \end{cases}$$

The E-strategy map φ such that $\varphi(0) = nil$, $\varphi(s) = [1]$ and $\varphi(+) = [2, 0, 1]$ satisfies the condition of Corollary 3.8. Hence, any evaluated term is in normal form. Unlike Proposition 3.1, the first argument of f is not evaluated eagerly.

There are cases in which evaluation does not go well on the condition of Theorem 3.6.

Example 3.10 Let R be a TRS such that

$$R = \begin{cases} 2nd(cons(x, cons(y, z))) \to y \\ from(x) \to cons(x, from(s(x))), \end{cases}$$

and φ an E-strategy map where $\varphi(2nd) = [1, 0]$, $\varphi(cons) = [2]$ and $\varphi(from) = [0]$.

$\varphi(cons)$ must have 2 so as to satisfy the condition of Theorem 3.6, because in the left-hand side of the first rule, the second argument of the first *cons* is not a variable.

For the term $t \equiv 2nd(from(x))$, the E-strategy first rewrites the subterm $from(x)$ to $cons(x, from(s(x)))$. Since $\varphi(cons)$ has 2, the subterm $from(s(x))$ should be rewritten and the E-strategy rewriting does not terminate. If $\varphi(cons) = nil$, the term $2nd(cons(x, from(s(x))))$ cannot be rewritten. For the success of evaluation of this term, the second argument of *cons* has to be rewritten, but the rewriting must not continue forever. For example, the second argument of *cons* appearing after the second rewrite must not be rewritten. That is, we have to define an E-strategy map φ such that for some f, $\varphi(f)$ can change according to contexts in which f occurs. However we cannot define such an E-strategy map since the list of each function symbol is fixed.

The on-demand E-strategy has been proposed for solving the above matter [7][6]. In the next section, we define a rewrite relation for the on-demand E-strategy and show the another condition on which any evaluated term is in head normal form.

4 The on-demand evaluation strategy

For defining the on-demand E-strategy reduction, on-demand flags are added to symbols. When it is being examined whether a term t matches to the left-hand side l of some rule, we compare the symbols of t and l in a top-to-bottom and left-to-right manner. If we meet a position $p \in O_\Sigma(l)$ such that $(t)_p \neq (l)_p$ and the flag of $(t)_p$ is up, we should evaluate the subterm $t|_p$ and continue the pattern matching.

We re-define Σ_L, V_L and T_L as $\Sigma_L = \{f_l^b \mid f \in \Sigma, l \in L(\{-n, \ldots, n\}), n = ar(f) \text{ and } b \in \{0, 1\}\}$, $V_L = \{x_{nil}^0 \mid x \in V\}$ and $T_L = T(\Sigma_L, V_L)$.

Definition 4.1 An on-demand E-strategy map φ is a map from $\Sigma \cup V$ to $L(\mathbb{Z})$ such that $\varphi(f) \in L(\{-ar(f), \ldots, ar(f)\})$ for all $f \in \Sigma$ and $\varphi(x) = nil$ for all $x \in V$. We extend the on-demand E-strategy map from T to T_L as

$$\varphi(f(t_1, \ldots t_n)) \equiv f_{\varphi(f)}^0(\varphi(t_1), \ldots, \varphi(t_n)).$$

The map $erase : T_L \to T$ erases the list and flag of each function symbol in a term:

$$erase(f_l^b(t'_1, \ldots t'_n)) \equiv f(erase(t'_1), \ldots, erase(t'_n)).$$

Definition 4.2 A map $up : T_L \to T_L$ ($dn : T_L \to T_L$) raises (lowers) the

on-demand flag of each function symbol in a term:

$$up(x^0_{nil}) \equiv dn(x^0_{nil}) \equiv x^0_{nil},$$
$$up(f^b_l(t_1, \ldots t_n)) \equiv f^1_l(up(t_1), \ldots up(t_n)),$$
$$dn(f^b_l(t_1, \ldots t_n)) \equiv f^0_l(dn(t_1), \ldots dn(t_n)).$$

A map $flag : T_L \times \mathbb{N}^*_+ \to \{0,1\}$ returns the flag of the function symbol at a position p of a term t:

$$flag(t,p) = b \text{ if } (t)_p = e^b_l.$$

We may omit the list or flag of a symbol if there is no confusion.

Definition 4.3 For a term l, a map $df_l : T \to O_\Sigma(l) \cup \{\bot, \top\}$ returns the first position $p \in O_\Sigma(t) \cap O_\Sigma(l)$ in top-to-bottom and left-to-right order such that the symbol of t at p differs from that of l, $\top$ if the function symbol of t at each position of $O_\Sigma(l)$ coincides with that of l, and $\bot$ otherwise:

$$df_l(t) = \begin{cases} p \text{ if } p \in O_\Sigma(t) \cap O_\Sigma(l), (t)_p \neq (l)_p \text{ and } (t)_{p'} = (l)_{p'} \\ \quad \text{for all } p' <_{lex} p \text{ and } p' \in O_\Sigma(l) \\ \top \text{ if } (t)_p = (l)_p \text{ for all } p \in O_\Sigma(l) \\ \bot \text{ otherwise.} \end{cases}$$

For a TRS R, a map $DF_R : T \to O_\Sigma(l) \cup \{\bot, \top\}$ basically returns the maximal position at which the function symbol of t differs from the corresponding symbol of the left-hand side of rules in R:

$$DF_R(t) = \begin{cases} \top \text{ if } df_l(t) = \top \text{ for some } l \to r \in R \\ \bot \text{ if } df_l(t) = \bot \text{ for all } l \to r \in R \\ max_{<_{lex}}\{p \in O_\Sigma(t) \mid p = df_l(t), l \to r \in R\} \text{ otherwise.} \end{cases}$$

Definition 4.4 An evaluation map $eval_\varphi : T \to \mathcal{P}(T)$ is re-defined as follows: $eval_\varphi(s) = \{t \in T \mid \langle \varphi(s), \varepsilon\rangle \to^*_\varphi \langle t, \varepsilon\rangle \in NF, (t)_\varepsilon = e_{nil}\}$. An on-demand E-strategy rewrite relation (a φ-rewrite relation) $\to_\varphi$ is a binary relation on $T_L \times \mathbb{N}^*_+$ defined as follows: $\langle t, p\rangle \to_\varphi \langle s, q\rangle$ if and only if $p \in O(t)$ and one of the following conditions is satisfied:

(i) $(t)_p = e_{nil}$, $s \equiv t$ and $p = q \cdot i$ for some i,

(ii) $t|_p \equiv e_{i::l}(t_1, \ldots, t_n)$ with $i > 0$, $s \equiv t[e_l(t_1, \ldots, t_n)]_p$ and $q = p \cdot i$,

(iii) $t|_p \equiv e_{-i::l}(t_1, \ldots, t_n)$ with $i > 0$, $s \equiv t[e_l(t_1, \ldots, up(t_i), \ldots, t_n)]_p$ and $q = p$,

(iv) $t|_p \equiv e_{0::l}(t_1, \ldots, t_n)$, $s \equiv t[t']_p$, $q = p$ where t' is a term such that
  (a) $t' \equiv e_l(t_1, \ldots, t_n)$ if
    $DF_R(erase(t|_p)) = \bot$ or
    $DF_R(erase(t|_p)) = \top$, $erase(t|_p)$ is not a redex or

$DF_R(erase(t|_p)) = \varepsilon$ or
$DF_R(erase(t|_p)) = p' \neq \varepsilon$, $flag(t, p \cdot p') = 0$,

(b) $t' \equiv e_l(t_1, \ldots, t_i[up(u)]_{p''}, \ldots, t_n)$ if
$DF_R(erase(t|_p)) = p' = i \cdot p''$, $flag(t, p \cdot p') = 1$, $\langle dn(t|_{p \cdot p'}), \varepsilon\rangle \rightarrow_\varphi^* \langle u, \varepsilon\rangle \in NF$, $(u)_\varepsilon = e'_{nil}$, $DF_R(erase(t|_p[u]_{p'})) = p'$ or $\bot$,

(c) $t' \equiv t[up(u)]_{p \cdot p'}$ if
$DF_R(erase(t|_p)) = p' \neq \varepsilon$, $flag(t, p \cdot p') = 1$, $\langle dn(t|_{p \cdot p'}), \varepsilon\rangle \rightarrow_\varphi^* \langle u, \varepsilon\rangle \in NF$, $(u)_\varepsilon = e'_{nil}$, $p' <_{lex} DF_R(erase(t|_p[u]_{p'}))$ or $DF_R(erase(t|_p[u]_{p'})) = \top$,

(d) $t' \equiv \varphi(r)\theta$ if
$DF_R(erase(t|_p)) = \top$, $t|_p \equiv l'\theta$, $erase(l') \equiv l$ and $l \rightarrow r \in R$.

The cases (i) and (ii) are similar to the definition without the on-demand flags. In the case (iii) of a negative integer $-i$, the on-demand flags of all symbols in the argument i are raised. The case (iv) is the definition of the matching action of the on-demand E-strategy. If there exists a position at which function symbols of the term to be rewritten and the left-hand side of some rule are different, and the on-demand flag of the function symbol of the term is up, the E-strategy evaluates the subterm at the position. If not so, in the case (a), the matching action fails. According to the evaluated term, the case is divided into two sub-cases (b) and (c). The case (b) means that if the root symbol of the evaluated term is not equivalent to the corresponding symbol of the left-hand side either, the matching is failed and the first element 0 is removed. The case (c) means that if both symbols are equivalent, the E-strategy continues the matching action. The case (d) means that the term is rewritten when the matching action succeeds. Figure 2 depicts the four conditions (a)–(d) in Definition 4.4 (iv).

Note that in the on-demand E-strategy a term $erace(s)$ is not always an element of $eval_\varphi(t)$ even if $\langle s, p\rangle$ is a normal form of $\langle \varphi(t), \varepsilon\rangle$. For example, let φ be an E-strategy map such that $\varphi(f) = [-1, 0]$, $\varphi(a) = [0]$, $\varphi(b) = nil$ and $R = \{f(b) \rightarrow a, a \rightarrow a\}$. Although $\langle \varphi(f(a)), \varepsilon\rangle \rightarrow_\varphi^* \langle f^0_{[0]}(a^1_{[0]}), \varepsilon\rangle \in NF$, $eval_\varphi(f(a)) = \emptyset$ because $DF_R(f(a)) = 1$, $flag(f(a), 1) = 1$, but evaluation of $\langle a_{[0]}, \varepsilon\rangle$ does not terminate.

5 The on-demand evaluation strategy for head normal form

If a symbol of l at each position of $O_\Sigma(l)$ is equivalent to that of t, the term t is not always an instance of l. For example, $f(a, b)$ is not an instance of $f(x, x)$, but $df_{f(x,x)}(f(a, b)) = \top$.

Lemma 5.1 *Let R be a left-linear TRS. A term t is a redex if $DF_R(t) = \top$.*

Proof. Trivial. □

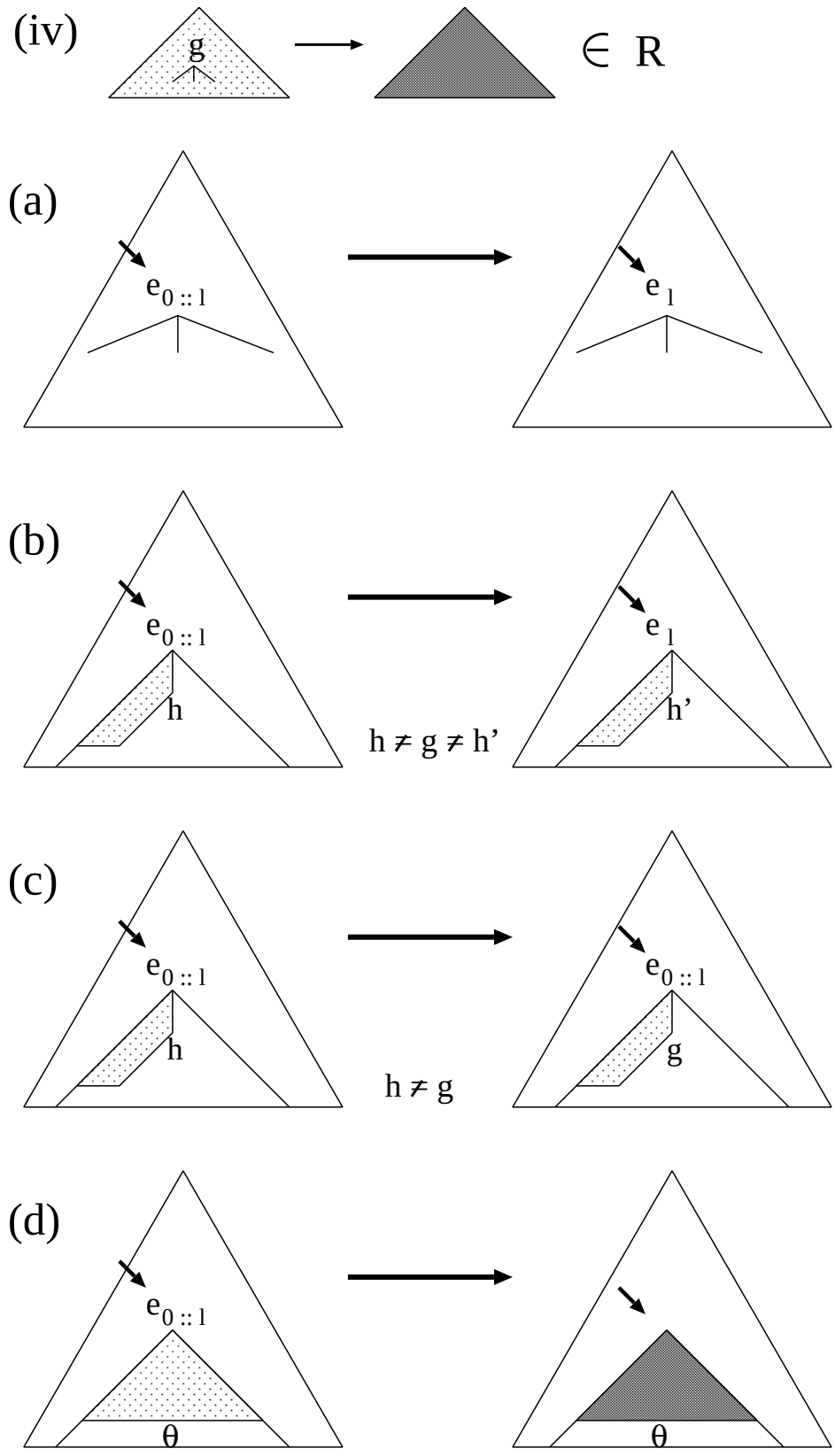

Fig. 2. Depiction of the four conditions of Definition 4.4(iv)

Lemma 5.2 *Let R be a constructor TRS. A term t is in head normal form w.r.t. $\to_R$ if $DF_R(t) = \bot$, or $DF_R(t) = p$ and $t|_p$ is in head normal form w.r.t. $\to_R$.*

Proof. If $DF_R(t) = \bot$ or p, from the definition of DF_R, there is the first position $p \in O(t)$ in top-to-bottom and left-to-right order such that a symbol of t at p differs from that of l. $(t)_p$ is a function symbol if $DF_R(t) = p$ or a variable if $DF_R(t) = \bot$. There is no defined symbol at a position q where $\varepsilon \neq q <_{lex} p$ because R is a constructor TRS. Since $t|_p$ is a variable or in head normal form, $(t)_p$ cannot be modified by any reduction and t cannot be reduced into a redex. Hence t is in head normal form. □

Lemma 5.3 *Let R be a left-linear constructor TRS and φ an on-demand E-strategy map such that $\varphi(f)$ contains $-i$ for any $i \notin LV_R(f)$ and the last element is 0 for each $f \in D(R)$. If $\langle \varphi(s'), \varepsilon \rangle \to^*_\varphi \langle s, p \rangle$, $(s)_p = e^b_{nil}$, $e \in \Sigma \cup V$, $b \in \{0, 1\}$ and $s' \in T$, $erase(s|_p)$ is in head normal form w.r.t. $\to_R$.*

Proof. We prove this claim by induction of the definition of $\to_\varphi$. The term $s|_p$ is trivially in head normal form if the symbol $(s)_p$ is a variable or a constructor symbol. If $(s)_p = e^b_{nil}$ is a defined symbol, from the assumption of φ, the last element of $\varphi(e)$ is 0. A rewrite step to modify the list $[0]$ into nil is only iv.(a) or (b). Hereafter, we use the notation of Definition 4.4 and let $s|_p \equiv t'$. In the case where $DF_R(erase(t|_p)) = \bot$, because of Lemma 5.2, the term $t|_p$ is in head normal form. Because of Lemma 5.1, $erase(t|_p)$ is a redex if $DF_R(erase(t|_p)) = \top$. For $e \in D(R)$ there is a rule $l \to r \in R$ where $(l)_\varepsilon = e$. Hence $DF_R(erase(t|_p)) \neq \varepsilon$. From the assumption of φ and the definitions of DF_R and LV_R, the on-demand flag of $(t)_{p \cdot p'}$ should be up. Since $erase(s|_p) \equiv erase(t') \equiv erase(t|_p)$, the term $erase(s|_p)$ is in head normal form in the case iv.(a). We consider the other case iv.(b). From the induction hypothesis, the term u is in head normal form since $(u)_\varepsilon = e'_{nil}$. From $DF_R(erase(t|_p[u]_{p'})) = p'$ and Lemma 5.2, $erase(t|_p[u]_{p'})$ is in head normal form. Since $erase(s|_p) \equiv erase(t') \equiv erase(t|_p[u]_{p'})$, the term $erase(s|_p)$ is also in head normal form in the case iv.(b). □

Theorem 5.4 *Let R be a left-linear constructor TRS and φ an on-demand E-strategy map such that $\varphi(f)$ contains $-i$ for any $i \notin LV_R(f)$ and the last element is 0 for each $f \in D(R)$. A term t is in head normal form w.r.t. $\to_R$ if $t \in eval_\varphi(t')$ for some $t' \in T$.*

Proof. From the definition of $eval_\varphi$, there is a term $s \in T_L$ such that $\langle \varphi(t'), \varepsilon \rangle \to^*_\varphi \langle s, \varepsilon \rangle$, $(s)_\varepsilon = e^b_{nil}$ and $t \equiv erase(s)$. From Lemma 5.3, the term t is in head normal form. □

Example 5.5 We consider the following TRS again:

$$R = \begin{cases} 2nd(cons(x, cons(y, z))) \to y \\ from(x) \to cons(x, from(s(x))). \end{cases}$$

Let φ be an on-demand E-strategy map such that $\varphi(2nd) = [-1, 0]$, $\varphi(from) = [0]$ and $\varphi(cons) = \varphi(s) = nil$, which satisfies the condition of Theorem 5.4.

We first show the reduction sequence from the term $t_1 \equiv from(x)$:

$$\begin{aligned}\langle \varphi(t_1), \varepsilon\rangle \equiv \quad & \langle\ from^0_{[0]}(x^0_{nil}), && \varepsilon\rangle \\ \to_\varphi\ & \langle\ cons^0_{nil}(x^0_{nil}, from^0_{[0]}(s^0_{nil}(x^0_{nil}))),\ \varepsilon\rangle.\end{aligned}$$

The result term $cons(x, from(s(x)))$ is not in normal form but in head normal form. The reduction sequence from the term $t_2 \equiv head(tail(from(x)))$ is given next:

$$\begin{aligned}& \langle\ 2nd^0_{[-1,0]}(from^0_{[0]}(x^0_{nil})), && \varepsilon\rangle \\ \to_\varphi\ & \langle\ 2nd^0_{[0]}(from^1_{[0]}(x^1_{nil})), && \varepsilon\rangle \\ \to_\varphi\ & \langle\ 2nd^0_{[0]}(cons^1_{nil}(x_{nil}, from^1_{[0]}(s^1_{nil}(x^1_{nil})))), && \varepsilon\rangle \\ \to_\varphi\ & \langle\ 2nd^0_{[0]}(cons^1_{nil}(x_{nil}, cons(s^1_{nil}(x^1_{nil}), from^1_{[0]}(s^1_{nil}(s^1_{nil}(x^1_{nil})))))), && \varepsilon\rangle \\ \to_\varphi\ & \langle\ s^1_{nil}(x^1_{nil}), && \varepsilon\rangle.\end{aligned}$$

First, all flags under $2nd$ are raised. Since $DF_R(2nd(from(x))) = 1$, the subterm at a position 1 is replaced by the evaluated one obtained as follows:

$$\begin{aligned}&\langle from^1_{[0]}(x^1_{nil}), \varepsilon\rangle \\ &\to_\varphi \langle cons^1_{nil}(x_{nil}, from^1_{[0]}(s^1_{nil}(x^1_{nil}))), \varepsilon\rangle.\end{aligned}$$

Next, $DF_R(2nd(cons(x, from(s(x))))) = 1{\cdot}2$. Again the subterm at a position $1 \cdot 2$ is evaluated as follows:

$$\begin{aligned}&\langle from^1_{[-1,0]}(s^1_{nil}(x^1_{nil})), \varepsilon\rangle \\ &\to_\varphi \langle cons^1_{nil}(s^1_{nil}(x_{nil}), from^1_{[0]}(s^1_{nil}(s^1_{nil}(x^1_{nil})))), \varepsilon\rangle.\end{aligned}$$

Finally, $DF_R(2nd(cons(x, cons(s(x), from(s(s(x)))))))) = \top$ and it is a redex. The on-demand E-strategy rewrites the whole term into $s(x)$.

Without the restriction of the left-linear and constructor TRS, Theorem 5.4 does not hold. We show two examples.

Example 5.6 Let $R = \{f(x, x) \to a, a \to b\}$ and $\varphi(f) = [-1, -2, 0]$, $\varphi(a) = [0]$ and $\varphi(b) = nil$. For the term $f(a, b)$, there is a reduction sequence

$$\langle \varphi(f(a, b), \varepsilon\rangle \equiv \langle f^0_{[-1,-2,0]}(a^0_{[0]}, b^0_{nil}), \varepsilon\rangle \to^*_\varphi \langle f^0_{nil}(a^1_{[0]}, b^1_{nil}), \varepsilon\rangle.$$

The pair $\langle f^0_{nil}(a^1_{[0]}, b^1_{nil}), \varepsilon\rangle$ is in normal form w.r.t. $\to_\varphi$. Although $f(a, b) \in eval_\varphi(f(a, b))$, the term $f(a, b)$ is not in head normal form since $f(a, b) \to_R f(b, b)$. The reason why the subterm a can not be evaluated is that $1 \notin O_\Sigma(f(x, x))$.

Example 5.7 Let $R = \{f(f(a)) \to a, f(b) \to f(a)\}$ and $\varphi(f) = [-1, 0]$,

$\varphi(a) = \varphi(b) = nil$. For the term $f(f(b))$, there is a reduction sequence

$$\begin{aligned}\langle\varphi(f(f(b))),\varepsilon\rangle &\equiv \langle f^0_{[-1,0]}(f^0_{[-1,0]}(b^0_{nil})),\varepsilon\rangle\\ &\to_\varphi \langle f^0_{[0]}(f^1_{[-1,0]}(b^1_{nil})),\varepsilon\rangle\\ &\to_\varphi \langle f^0_{nil}(f^1_{[-1,0]}(b^1_{nil})),\varepsilon\rangle.\end{aligned}$$

The pair $\langle f_{nil}(f(b)),\varepsilon\rangle$ is in normal form w.r.t. $\to_\varphi$. However, $f(f(b))$ is not in head normal form since $f(f(b)) \to_R f(f(a))$. Note that the subterm $f(b)$ cannot be evaluated because $(f(f(b)))_1 = f = (f(f(a)))_1$. There is no such a case in a constructor TRS.

6 Concluding remarks

We have shown the two conditions of E-strategy such that an evaluated term is in head normal form. One of methods finding a normal form for a given term is that the term is first reduced to a head normal form, all arguments are next reduced to head normal forms. Repeating these reductions, we finally get a term in normal form as a result of reducing the term if the reduction terminates. However, it is not easy to get an E-strategy map which simulates this operation. For example, let us consider the following term rewriting system:

$$R = \begin{cases} 2nd(cons(x, cons(y, z))) \to y \\ from(x) \to cons(x, from(s(x))). \end{cases}$$

For the E-strategy map φ in Example 5.5, i.e. $\varphi(2nd) = \varphi(from) = [-1, 0]$ and $\varphi(cons) = \varphi(s) = nil$, an evaluated term is in head normal form. We define an E-strategy map φ' such that $\varphi'(f) = \varphi(f) :: [1, \ldots, n]$. The term $2nd(from(x))$ is reduced with φ' as follows:

$$\begin{aligned}2nd(from(x)) &\to 2nd(cons(x, from(s(x))))\\ &\to 2nd(cons(x, cons(s(x), from(s(s(x))))))\\ &\to \cdots\end{aligned}$$

This is because the list of $cons$ is $[1, 2]$ and the arguments of $cons$ are forced to be reduced. Hence we need a meta operation as above for reducing a term to a normal form.

References

[1] Baader, F., and T. Nipkow, "Term rewriting and all that," Cambridge University Press, 1998.

[2] Diaconescu, R., and K. Futatsugi, "CafeOBJ report," World Scientific, 1998.

[3] Goguen, J. A. T. Winkler, J. Meseguer, K. Futatsugi and J. P. Jouannaud, *Introducig OBJ,* Software Engineering with OBJ: algebraic specification in action, edited by Goguen and Malcolm, Kluwer, 2000.

[4] Nagaya, T., "Reduction strategy for term rewriting system," Ph.D. thesis, Japan Advanced Institute of Science and Technology, 1999.

[5] Nagaya, T. M. Matsumoto, K. Ogata, and K. Futatsugi, *How to give local strategies to function symbols for equality of two implementations of the E-strategy with and without evaluated flags,* Proceedings of Asian Symposium on Computer Mathematics, 1998, 71-81.

[6] Ogata, T., and K. Futatsugi, *Operational semantics of rewriting with the on-demand evaluation strategy,* Proceedings of the 2000 ACM Symposium on Applied Computing, 2000, 756-763.

[7] Nakagawa, A., T. Sawada and K. Futatsugi, CafeOBJ user's manual –ver.1.4.–, `http://www.ldl.jaist.ac.jp/cafeobj/`, 1998.