The Functional Strategy and Transitive Term Rewriting Systems

Yoshihito Toyama[1*]
Sjaak Smetsers[2] Marko van Eekelen[2] Rinus Plasmeijer[2]

[1] NTT Laboratories, Inunidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan.
email: toyama@ntt-20.ntt.jp

[2] Department of Informatics, Faculty of Mathematics and Informatics, University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands.
email: marko@cs.kun.nl, rinus@cs.kun.nl, sjakie@cs.kun.nl

Abstract

The functional strategy has been widely used implicitly (Haskell, Miranda, Lazy ML) and explicitly (Clean) as an efficient, intuitively easy to understand reduction strategy for term (or graph) rewriting systems. However, little is known of its formal properties since the strategy deals with priority rewriting which significantly complicates the semantics. Nevertheless, this paper shows that some formal results about the functional strategy can be produced by studying the functional strategy entirely within the standard framework of orthogonal term rewriting systems. A concept is introduced that is one of the key aspects of the efficiency of the functional strategy: *transitive indexes*. The corresponding class of *transitive* term rewriting systems is characterized. An efficient normalizing strategy is given for these rewriting systems. It is shown that the functional strategy is normalizing for the class of *left-incompatible* term rewriting systems.

1. Introduction

An interesting common aspect of the functional languages Miranda[1] (Turner (1985)), Haskell (Hudak et al. (1992)), Lazy ML (Augustsson (1984)) and Clean (Brus et al. (1987),Nöcker et al. (1991)) is the similarity between their reduction strategies. The reduction order determined by these strategies can roughly be characterized as top-to-bottom left-to-right lazy pattern matching. This reduction order, in the following referred to as the *functional strategy*, is intuitively easy to understand and can efficiently be implemented. It is usually considered as an aspect of the language that is transformed during the compilation process to some standard reduction strategy (e.g. normal order reduction) in the underlying computational model (e.g. lambda-calculus). Several authors have pursued studies of this reduction order with different

*Partially supported by grants from NWO, the Vrije Universiteit Amsterdam and the Katholieke Universiteit Nijmegen

[1]Miranda^{TM} is a trademark of Research Software Limited

semantic transformations (Kennaway (1990),Laville (1987),Puel and Suárez (1990)). The language Clean is close to its underlying computational model (i.e. term graph rewriting (Barendregt et al. (1987))). Therefore, it seems natural to define the functional strategy *directly* in the computational model rather than using a transformation to an equivalent system with a well-known strategy.

An important efficiency aspect of the functional strategy lies in the fact that evaluation of an actual argument is always forced (by applying the strategy recursively to that actual argument) when this argument is tried to match a non-variable in the corresponding formal pattern. A possible analysis of properties of the functional strategy may be performed using some kind of priority semantics as in Baeten et al. (1987). A problem with these priority semantics is however the fact that important theoretical properties of standard term rewriting theory do not easily carry over to the priority world.

In this paper the functional strategy is investigated within the standard framework of orthogonal term rewriting systems. Thus we leave the overlapping situation between rules that usually appears in the functional strategy out of consideration. We believe that this approach is worth-while as a first step since by this restriction we can rely upon the well-known concept of indexes when we try to explain why the functional strategy works well for a wide class of orthogonal term rewriting systems. The concept of indexes was proposed by Huet and Lévy (1979). They introduced the subclass of strongly sequential orthogonal term rewriting systems for which index reduction is normalizing. However, for reasons of efficiency their approach is not very feasible in a practical sense. An important problem they had to cope with is the fact that indexes in general lack a certain transitivity property that seems to be essential for the efficiency of any reduction strategy.

This paper studies transitivity properties of indexes by introducing so-called *transitive indexes*. The *transitive term rewriting systems* are defined as a subclass of the strongly sequential term rewriting systems for which each term not in strong head normal form has a transitive index. Furthermore, the notion *transitive direction* is introduced that is used in two different ways. Firstly, it is shown that with the aid of these transitive directions a simple test on the left-hand-sides of the rewrite rules can be expressed that is sufficient to characterize transitive term rewriting systems. Secondly, transitive directions are the basis of a new strategy: the *transitive strategy*. This strategy is normalizing for transitive term rewriting systems. Finally it is shown, using the introduced concepts, that the functional strategy is normalizing for a subclass of transitive term rewriting systems: so-called *left-incompatible term rewriting systems*.

2. Preliminaries

In the sequel we will assume that the reader is familiar with the basic concepts concerning term rewriting systems as introduced by Dershowitz and Jouannaud (1990),Klop (1992) or Huet and Lévy (1979).

2.1. Term Rewriting Systems

The following definitions are based on definitions given in Klop (1992). In contrast with Klop (1992) we use the notion 'constant symbol' for a symbol that cannot be rewritten, instead of for a function symbol with arity 0.

2.1. DEFINITION. A *Term Rewriting System* (TRS) is a pair (Σ, R) of an *alphabet* or *signature* Σ and a set of *rewrite rules* R.

(i) The alphabet Σ consists of:

(1) A countable infinite set of variables $x, y, z, \ldots$.

(2) A non empty set Σ_0 of *function symbols* or *operator symbols* $f, g, \ldots$, each equipped with an 'arity' (a natural number), i.e. the number of 'arguments' it is supposed to have. We have 0-ary, unary, binary, ternary etc function symbols.

(ii) The set of *terms* (or *expressions*) 'over' Σ indicated by $T(\Sigma)$ or, if Σ is not relevant by T, is defined inductively:

(1) $x, y, z, \ldots \in T(\Sigma)$.

(2) If $f \in \Sigma_0$ is an n-ary symbol, and $t_1, \ldots, t_n \in T(\Sigma)$ $(n \geq 0)$, then

$$f(t_1, \ldots, t_n) \in T(\Sigma).$$

(iii) Terms not containing a variable are called *ground terms* (also: *closed terms*), and $T_0(\Sigma)$ is the set of ground terms. Terms in which no variable occurs twice or more, are called *linear*.

(iv) A *rewrite rule* $\in R$ is a pair (l, r) of terms $\in T(\Sigma)$such that l is not a variable, and all variables in r are contained in l. It will be written as $l \to r$. Often a rewrite rule will get a name, e.g. $\mathbf{r}$, and we write $\mathbf{r} : l \to r$.

When the signature Σ is not relevant, a TRS (Σ, R) is indicated by the rewrite rules R only.

2.2. DEFINITION. (i) Consider an extra 0-ary constant $\Box$ called a *hole* and the set $T(\Sigma \cup \{\Box\})$. Then $C \in T(\Sigma \cup \{\Box\})$ is called a *context*. We use the notation $C[\ ,\ldots,\ ]$ for the context containing n holes $(n \geq 1)$, and if $t_1, \ldots, t_n \in T(\Sigma)$, then $C[t_1, \ldots, t_n]$ denotes the result of placing $t_1, \ldots, t_n$ in the holes of $C[\ ,\ldots,\ ]$ from left to right. In particular, $C[\ ]$ denotes a context containing precisely one hole.

(ii) $t \equiv s$ indicates the identity of two terms t and s. s is called a *subterm* of t if $t \equiv C[s]$. We write $s \subseteq t$. s is a *proper subterm*, denoted by $s \subset t$, if $s \subseteq t$ and $t \not\equiv s$

(iii) If a term t has an occurrence of some (function or variable) symbol e, we write $e \in t$. The variable occurrence z in $C[z]$ is *fresh* if $z \notin C[\ ]$.

2.3. DEFINITION. (i) A *substitution* σ is a map from $T(\Sigma)$ to $T(\Sigma)$ satisfying

$$\sigma(f(t_1, \ldots, t_n)) \equiv f(\sigma(t_1), \ldots, \sigma(t_n))$$

for every n-ary function symbol f. We also write t^σ instead of $\sigma(t)$.

(ii) The set of rewrite rules R defines a *reduction relation* $\to$ on T as follows:

> $t \to s$ iff there exists a rule $\mathbf{r} : l \to r$, a context $C[\ ]$ and a substitution σ such that $t \equiv C[l^\sigma]$ and $s \equiv C[r^\sigma]$.

The term l^σ is called a *redex*, or more precisely an $\mathbf{r}$-redex. t itself *is* a redex if $t \equiv l^\sigma$.

(iii) $\twoheadrightarrow$ denotes the transitive reflexive closure of $\to$.

(iv) Two terms t and $s \in T$ are *overlapping* if there exist substitutions σ_1 and σ_2 such that $t^{\sigma_1} \equiv s^{\sigma_2}$.

(v) $t \in T$ is a *normal form* (with respect to $\rightarrow$) if there exists no $s \in T$ such that $t \rightarrow s$. NF denotes the set of normal forms of T.

(vi) A term t is in *head-normal form* if there exists no redex $s \in T$ such that $t \twoheadrightarrow s$.

2.4. Definition. A term rewriting system R is *orthogonal* if:

(i) For all rewrite rules $\mathbf{r} : l \rightarrow r \in R$, l is linear

(ii) For any two rewrite rules $\mathbf{r}_1 : l_1 \rightarrow r_1$ and $\mathbf{r}_2 : l_2 \rightarrow r_2 \in R$:

(1) If $\mathbf{r}_1$ and $\mathbf{r}_2$ are different then l_1 and l_2 are non-overlapping.

(2) For all $s \subset l_2$ such that s is not a single variable, l_1 and s are non-overlapping.

Note. From here on we assume that every term rewriting system R is orthogonal.

3. Strong Sequentiality

In Huet and Lévy (1979) a class of orthogonal TRS's is defined wherein needed redex are identified by looking at the left-hand-sides only. These so-called *strongly sequential TRS's* are based on the two notions *Ω-reduction* and *index* of which the definition is given in this section.

3.1. Definition (Ω-terms). (i) Consider an extra constant Ω. The set $T(\Sigma \cup \{\Omega\})$, also denoted by T_Ω, is called the set of Ω-terms. t_Ω indicates the Ω-term obtained from a term t by replacing each variable in t with Ω.

(ii) The preordering $\succeq$ on T_Ω is defined as follows:

(1) $t \succeq \Omega$ for all $t \in T_\Omega$,

(2) $f(t_1, \ldots, t_n) \succeq f(s_1, \ldots, s_n) \quad (n \geq 0) \quad$ if $t_i \succeq s_i$ for $i = 1, \cdots, n$.

We write $t \succ s$ if $t \succeq s$ and $t \not\equiv s$.

(iii) Two Ω-terms t and s are *compatible*, denoted by $t \uparrow s$, if there exists some Ω-term r such that $r \succeq t$ and $r \succeq s$; otherwise, t and s are *incompatible*, which is indicated by $t \# s$.

(iv) Let $S \subseteq T_\Omega$. Then $t \succeq S$ (resp. $t \uparrow S$) if there exists some $s \in S$ such that $t \succeq s$ (resp. $t \uparrow s$); otherwise, $t \not\succeq S$ (resp. $t \# S$).

3.2. Definition (Ω-systems). Let R be a term rewriting system.

(i) The set of *redex schemata* of R is $Red = \{ l_\Omega \mid l \rightarrow r \in R\}$.

(ii) $\Omega-$*reduction*, denoted by $\rightarrow_\Omega$, is defined on T_Ω as $C[s] \rightarrow_\Omega C[\Omega]$ where $s \uparrow Red$ and $s \not\equiv \Omega$.

(iii) The *Ω-system* R_Ω (corresponding to R) is defined as a reduction system on T_Ω having $\rightarrow_\Omega$ as reduction relation.

3.3. Lemma. *For any R, R_Ω is* complete *(i.e.* confluent *and* terminating*)*

Proof. Easy. See Klop (1992). □

3.4. Definition (Ω-normal form). (i) $\omega(t)$ denotes the normal form of t with respect to $\rightarrow_\Omega$. Note that due to lemma 3.3 $\omega(t)$ is well-defined. NF_Ω denotes the set of Ω-normal forms.

(ii) $\bar{\omega}(f(t_1, \ldots, t_n)) \equiv f(\omega(t_1), \ldots, \omega(t_n))$.

The next technical lemma concerns Ω-reduction and the related definition of $\bar{\omega}$. It will be used in the proofs later on in this paper.

3.5. LEMMA. (i) *If $t \succeq s$ then $\omega(t) \succeq \omega(s)$.*
(ii) *Let $C[\Omega] \in NF_\Omega$. Then for all $t \in NF_\Omega$, $C[t] \in NF_\Omega$*
(iii) *If $\bar{\omega}(t) \equiv C[\Omega]$ and $C[z] \# Red$ then $C[z] \in NF_\Omega$.*

PROOF. (i) By induction on the size of t.
(ii) Suppose $C[t] \notin NF_\Omega$. Then there exist a rule $r \in Red$ that is compatible with a subterm of $C[t]$. This subterm is a result of the combination of $C[\Omega]$ and t, i.e. $C[t] \equiv C'[C''[t]]$ such that $C''[t]$ is compatible with Red for some C' and C''. But, then $C''[\Omega]$ is also compatible with Red which is a contradiction to $C[\Omega] \in NF_\Omega$.
(iii) Obvious. $\square$

The intuitive idea of $\rightarrow_\Omega$ is that it 'approximates' ordinary reduction by considering left-hand-sides only. All right-hand-sides of rewrite rules in R_Ω are equal to Ω which represents any term. The 'approximation' is expressed in the following lemma:

3.6. LEMMA. *Let R be a TRS, and $t_1, t_2 \in T$. Then*

$$t_1 \twoheadrightarrow t_2 \quad \Rightarrow \quad \omega(t_1) \preceq t_2.$$

PROOF. By induction on the length of the reduction sequence from t_1 to t_2. $\square$

The head-normal form property (definition 2.3 (vi)) is in general undecidable. With the aid of Ω-reduction we can define a decidable variant of this property.

3.7. DEFINITION. A term t is in *strong head-normal form* if $\omega(t) \not\equiv \Omega$.

3.8. LEMMA. *If t is in strong head normal form then t is in head-normal form.*

PROOF. Let $t' \equiv \omega(t)$. Suppose t is not in head-normal form. Then there exists a term s such that $t \twoheadrightarrow s$ and $s \succeq Red$. Due to lemma 3.6 $t' \preceq s$ so $t' \uparrow Red$. But also $t' \not\equiv \Omega$ and therefore $t' \rightarrow_\Omega \Omega$ which is a contradiction to $t' \in NF_\Omega$. $\square$

3.9. DEFINITION (Index). Let $C[\ ]$ be a context such that $z \in \omega(C[z])$ where z is a fresh variable. Then the displayed occurence of Ω in $C[\Omega]$ is called an *index* and we write $C[\Omega_I]$. Let $C[\Omega_I]$ and Δ be a redex occurrence in $C[\Delta]$. This redex occurrence is also called an index and we write $C[\Delta_I]$.

3.10. DEFINITION (Strong Sequentiality). Let R be a term rewriting system.
(i) R is *strongly sequential* if for each term $t \notin NF$, t has an index (Huet and Lévy (1979),Klop (1992)).
(ii) If Δ is an index of t then $t \stackrel{\Delta}{\rightarrow} s$ is the index reduction.

3.11. PROPOSITION. *Let R be strongly sequential. Then index reduction is normalizing.*

PROOF. See Huet and Lévy (1979). $\square$

3.12. PROPOSITION. *For any strongly sequential TRS one has the following.*
(i) $C_1[C_2[\Omega_I]] \;\Rightarrow\; C_1[\Omega_I]$ *and* $C_2[\Omega_I]$.
(ii) *The reverse implication does not hold generally.*

PROOF. (i) See Klop (1992).
(ii) See example 3.13 $\square$

In Huet and Lévy (1979) an algorithm has been given that is capable of finding an index in a term t in $O(|t|)$ time. The main disadvantage of the algorithm is that after an index has been rewritten to a term t' the whole new term t' has to be considered again in order to determine the next index. So, in general, the search cannot be started locally, i.e. at the position where the last index was found. This is in fact a consequence of proposition 3.12 (ii). This problem is illustrated by the next example:

3.13. EXAMPLE. Let $Red = \{f(1,1), g(f(\Omega,2)), h\}$. Consider the term $g(h)$. Clearly, h is an index. Suppose h reduces to $f(\Delta_1, \Delta_2)$ where both Δ_1 and Δ_2 are redexes. Locally (i.e. when leaving the surrounding context out of consideration), both redexes are indexes. But for the whole term $g(f(\Delta_1, \Delta_2))$ only Δ_2 is an index.

3.14. LEMMA. (i) *If* $C_1[\Omega_I]$ *and* $C_1[z] \preceq C_2[z]$ *(where z is fresh) then* $C_2[\Omega_I]$.
(ii) *If* $C_1[\Omega] \in NF_\Omega$ *and* $C_2[\Omega_I]$ *then* $C_1[C_2[\Omega_I]]$

PROOF. (i) By lemma 3.5 (i), it follows that $\omega(C_1[z]) \leq \omega(C_2[z])$. Thus, we get $z \in \omega(C_2[z])$ as $z \in \omega(C_1[z])$
(ii) By lemma 3.5 (ii) and $C_1[\Omega] \in NF_\Omega$, for any t, $\omega(C_1[t]) \equiv C_1[\omega(t)]$. Hence, one has $\omega(C_1[C_2[z]]) \equiv C_1[\omega(C_2[z])]$. Since $z \in \omega(C_2[z])$ also $z \in C_1[\omega(C_2[z])]$. $\square$

4. Transitive Indexes

Example 3.13 indicates why indexes in strongly sequential system are not always transitive. A certain subterm t in a context $C[t]$ may reduce to a term t' without rewriting all indexes in t, but, resulting in a term $C[t']$ that is compatible with one of the elements of *Red*. In this section we formulate a restriction for TRS's that avoids this problem. As will be shown, this criterion is sufficient for the transitivity property for indexes.

We first introduce a new concept of transitive indexes.

4.1. DEFINITION (Transitive Index). The displayed index in $C_1[\Omega_I]$ is *transitive* if for any Ω-term $C_2[\Omega_I]$, $C_2[C_1[\Omega_I]]$. We indicate the transitive index with $C_1[\Omega_{TI}]$. We also call the redex occurrence Δ in $C_1[\Delta]$ a transitive index and indicate it with $C_1[\Delta_{TI}]$.

Note that replacing $C_2[C_1[\Omega_I]]$ by $C_1[C_2[\Omega_I]]$ in definition 4.1 would give a different notion. For example, let $Red = \{f(g(\Omega))\}$. Then $f(\Omega_{TI})$ by definition 4.1 and the fact that $C_2[f(\Omega_I)]$ holds for any $C_2[\Omega_I]$. But, if we exchange C_1 and C_2 in this definition the displayed Ω in $f(\Omega)$ is not transitive anymore. Take, for example, the context $C_2[\Omega] \equiv g(\Omega)$. Clearly, $C_2[\Omega_I]$. However, in $f(g(\Omega))$, Ω is not an index.

Transitive indexes have the following transitivity property.

4.2. Lemma. *If $C_1[\Omega_{TI}]$ and $C_2[\Omega_{TI}]$ then $C_1[C_2[\Omega_{TI}]]$.*

Proof. Let $C_3[\Omega_I]$. From $C_2[\Omega_{TI}]$ it follows that $C_3[C_2[\Omega_I]]$. By the definition of transitivity and $C_1[\Omega_{TI}]$, $C_3[C_2[C_1[\Omega_I]]]$. □

As with indexes, transitivity of indexes remains valid for larger contexts.

4.3. Lemma. *If $C_1[\Omega_{TI}]$ and $C_1[z] \preceq C_2[z]$ (where z is fresh) then $C_2[\Omega_{TI}]$.*

Proof. This lemma follows immediately from the definition of transitive indexes and lemma 3.14 (i). □

The importance of transitivity is that it allows to search locally for indexes. Once an index has been found and rewritten, the search for the next index may continue at the same location where the last index has been found. As a consequence, rewriting can be performed in an efficient depth-first way. However, requiring that each term not in normal form should have a transitive index (analogous to the way strongly sequential systems are defined) appears to be too restrictive as can be seen in the next example:

4.4. Example. Let R be a TRS with $Red = \{f(g(\Omega))\}$. Consider the term $g(\Delta)$ where $\Delta \equiv f(g(1))$. In this term Δ is not a transitive index, since Δ is not an index in $f(g(\Delta))$.

Now the question is: 'How to weaken the transitivity criterion for TRS's?'. The answer is given in the following reasoning. Suppose we have a TRS R and a strategy, for convenience called *hnf*, that delivers the redexes of a term t that should be reduced in order to obtain the head-normal form of t. Then it is easy to construct a normalising strategy, say *nf*, for R.

> First, reduce a term t to head-normal-form using *hnf* and then apply *nf* to all the arguments of the result.

The fact that the head-normal form property is undecidable makes it impossible for general TRS's to give such a *hnf* strategy. The next definition of transitive TRS's is based on the decidable strong head-normal form property.

4.5. Definition (Transitive Term Rewriting Systems). Let R be a term rewriting system. R is *transitive* if each term t not in strong head-normal form has a transitive index.

4.6. Proposition. *Let R be a TRS. If R is transitive then R is strongly sequential.*

Proof. We have to prove that every term t not in normal form contains an index. Therefore, we distinguish the following two cases:

$\omega(t) \equiv \Omega$: From the definition of transitivity of R it follows that t has a transitive index.

$\omega(t) \not\equiv \Omega$: . Since t is not a normal form there exists a context $C[,\cdots,]$ such that $t \equiv C[t_1,\cdots,t_n]$ and $\omega(t) \equiv C[\Omega,\cdots,\Omega]$ with every $t_i \succ \Omega$. Form the fact that R is transitive and $\omega(t_1) \equiv \Omega$, t_1 has an index. Applying lemma 3.14 (ii) , $C[t_1,\Omega,\cdots,\Omega]$ has an index and therefore (by lemma 3.14 (i)) $C[t_1,\cdots,t_n]$ has also an index. □

The reverse of the previous proposition does not hold generally, i.e. not every strongly sequential system is also transitive.

4.7. EXAMPLE. Let $Red = \{f(f(\Omega,0),1), f(2,f(3,\Omega))\}$. This TRS is strongly sequential. Now consider the term $f(\Delta_1,\Delta_2)$. Clearly, this term is not in strong head-normal form. But, Δ_1 is not a transitive index. Take, for instance, the context $f(\Omega_I,1)$. In $f(f(\Delta_1,\Delta_2),1)$ Δ_1 is not an index. For the same reason Δ_2 is not a transitive index.

The next problem is: 'How can we localize transitive indexes?'. The solution is given with the aid of the following definition of *transitive directions*.

4.8. DEFINITION (Transitive Direction). (i) Let $Q \subseteq T_\Omega$. The displayed Ω in $C[\Omega]$ is a *direction* for Q if $C[z] \# Q$. We indicate a direction for Q with $C[\Omega_Q]$.

(ii) Let $Red^* = \{p \mid \Omega \prec\ p \subseteq r$ for some $r \in Red\}$. A *transitive direction* is defined as a direction for Red^*. We denote a transitive direction with $C[\Omega_{TD}]$.

Transitive directions can be related to transitive indexes as follows.

4.9. LEMMA. *Let $C[\Omega_{TD}]$ and $C[z] \in NF_\Omega$. Then $C[\Omega_{TI}]$.*

PROOF. It is clear that $C[\Omega_I]$. We shall prove that the displayed index Ω is transitive, i.e. $C'[C[\Omega_I]]$ for any Ω-term $C'[\Omega_I]$. Let $\omega(C'[z]) \equiv C''[z]$. Note that $C''[z] \in NF_\Omega$ and that $\omega(C'[C[z]]) \equiv \omega(C''[C[z]])$. Now we show that $C''[C[z]] \in NF_\Omega$. Suppose $C''[C[z]] \notin NF_\Omega$. Then there exists some $r \in Red$ having a proper subterm r' not being Ω that is compatible with $C[z]$. However, this contradicts the assumption that $C[z] \# Red^*$. $\square$

The following lemma explains how to use the previous one for finding an index.

4.10. LEMMA. *Let $C[\Delta] \in T$. If there exists some $C'[z] \preceq C[z]$ (where z is fresh) such that $C'[z]$ is divided into $C'[z] \equiv C_1[C_2[\cdots C_n[z]\cdots]]$ $(n \geq 1)$ where $C_i[\Omega_{TD}]$ for $i = 2\cdots n$ and $C_i[z] \in NF_\Omega$ for $i = 1\cdots n$. Then $C[\Delta_I]$.*

PROOF. By lemma 4.9, $C_i[\Omega_{TI}]$ for $i = 2\cdots n$. Since $C_1[z] \in NF_\Omega$, we have $C_1[\Omega_I]$. By definition 4.1 and lemma 4.2, $C'[\Omega_I]$. From lemma 3.14 (i), it follows that $C[\Omega_I]$. $\square$

It seems that the problem of finding transitive indexes has been postponed since we need transitive directions to determine transitive indexes. Lemma 4.11 in combination with lemma 4.13 shows us where to look for transitive directions in a term that might be a candidate for being rewritten. Lemma 4.13 on its own, enables an efficient test for deciding whether or not a certain TRS is transitive.

4.11. LEMMA. *Let $Red^{\prec} = \{p \mid \Omega \prec\ p \prec\ r$ for some $r \in Red\}$ and let any $t \in Red^{\prec}$ have a transitive direction. Then for every $s \in T_\Omega$ such that $s \uparrow Red \wedge s \not\succeq Red$, s has a transitive direction.*

PROOF. Since $s \uparrow Red \wedge s \not\succeq Red$ there exists some $r \in Red$ such that $r \uparrow s \wedge s \not\succeq r$. Without loss of generality we may state that $r \equiv C[s_1, \cdots, s_m, \Omega, \cdots, \Omega]$ and $s \equiv C[\Omega, \cdots, \Omega, s_{m+1}, \cdots, s_{m+n}]$ where $s_i \succ \Omega$ for $i = 1 \cdots m+n, m > 0$ and $n \geq 0$. Since $C[\Omega, \cdots, \Omega, \Omega, \cdots, \Omega] \in Red^{\prec}$, $C[\Omega, \cdots, \Omega, \Omega, \cdots, \Omega]$ has a transitive direction. It is clear that this transitive direction must appear in the first m occurrences of Ω, say $C[\Omega_{TD}, \cdots, \Omega, \Omega, \cdots, \Omega]$. $C[z, \Omega, \cdots, \Omega, \Omega, \cdots, \Omega] \preceq C[z, \Omega, \cdots, \Omega, s_{m+1}, \cdots, s_{m+n}]$, hence $C[z, \Omega, \cdots, \Omega, s_{m+1}, \cdots, s_{m+n}] \# Red^*$. □

4.12. LEMMA. . *Let $C[s] \in Red$, $s \succ \Omega$. Then $C[\Omega_I]$.*

PROOF. From the non-overlapping property of R (definition 2.4) it follows that $C[z] \in NF_\Omega$ □

4.13. LEMMA. *A TRS R is transitive iff every $t \in Red^{\prec}$ has a transitive direction.*

PROOF. $\Rightarrow$: Let $t \in Red^{\prec}$. Then $\omega(t) \equiv \Omega$. By assumption, t has a transitive index, say $t \equiv C[\Omega_{TI}]$. We will prove that $C[z] \# Red^*$. Assume that $C[z] \uparrow Red^*$. Then there exists an $s \in Red^*$ such that $C[z] \uparrow s$. This means that there exists a $r \in Red$ such that $r \equiv C'[s]$. Now consider the term $C'[C[z]]$. Since $C'[C[z]] \uparrow r$, $\omega(C'[C[z]]) \equiv \Omega$. From lemma 4.12 it follows that $C'[\Omega_I]$. But then $\omega(C'[C[z]]) \equiv \Omega$ contradicts to $C[\Omega_{TI}]$. Hence it follows that $C[z] \# Red^*$.

$\Leftarrow$: By induction to the size of t we will prove that if $\omega(t) \equiv \Omega$ then t has a transitive index. The basis step is trivial. For the induction step we make a distinction between two cases:

$t \succeq Red$: We can take t itself as the transitive index.

$t \not\succeq Red$: Let $C[, \cdots,]$ be a context such that $t \equiv C[t_1, \cdots, t_n]$ with every $t_i \succ \Omega$ and $\bar{\omega}(t) \equiv C[\Omega, \cdots, \Omega]$ in which all Ω occurrences that correspond to subterms $s \succ \Omega$ of t are displayed. Since $C[\Omega, \cdots, \Omega] \not\succeq Red$ and $C[\Omega, \cdots, \Omega] \uparrow Red$, by lemma 4.11, $C[\Omega, \cdots, \Omega]$ has a transitive direction. Applying lemma 3.5 (iii) and lemma 4.9 it follows that this transitive direction is a transitive index. Again we distinguish two cases:

(a) The transitive index Ω is displayed in $C[\Omega, \cdots, \Omega]$. Without any loss of generality we may assume that the first displayed Ω is the transitive index, i.e. $C[\Omega_{TI}, \cdots, \Omega]$. Since $\omega(t_1) \equiv \Omega$ we can apply the I.H.: t_1 has a transitive index. Thus, by lemma 4.2, $C[t_1, \Omega, \cdots, \Omega]$ has a transitive index in t_1 and hence, by lemma 4.3, $C[t_1, t_2, \cdots, t_n]$ has a transitive index in t_1.

(b) The transitive index Ω is not displayed in $C[\Omega, \cdots, \Omega]$. This means that this transitive index corresponds to an Ω-occurrence in t. Now we can apply lemma 4.3 immediately so, $C[t_1, \cdots, t_n]$ has a transitive index. □

4.14. REMARK. (i) Strongly sequential orthogonal constructor systems (Huet and Lévy (1979),Klop (1992)) are clearly transitive. We will prove later on that left-normal orthogonal systems (Huet and Lévy (1979),Klop (1992),O'Donnell (1977)) are transitive too.

(ii) Huet and Lévy (Huet and Lévy (1979)) defined simple systems as orthogonal term rewriting systems satisfying $\forall t \in (Red^*)^{\prec} : \exists C[\,] : t \equiv C[\Omega_{TD}]$. Here $(Red^*)^{\prec} = \{p \mid \Omega \prec p \prec r$ for some $r \in Red^*\}$. It is clear that if R is simple then it is transitive, but the reverse direction is not the case from the following example. Let R have $Red = \{f(g(0,\Omega)), h(g(\Omega,0))\}$. It is clear that R is transitive. However, $g(\Omega,\Omega) \in (Red^*)^{\prec}$ cannot make an incompatible term to Red^* by replacing an occurrence of Ω with z. Thus, R is not simple.

5. Transitive Strategy

This section presents a method for searching indexes of transitive systems. The key idea of our method is a marking of occurrences of subterms which are known to be in strong head normal form. Of course, these marks are valid through reductions. Hence, we can repeatedly use the information indicated by marks for future searches of indexes.

5.1. DEFINITION. Let (Σ, R) be a TRS.

(i) *root* is a function from T_Ω to Σ_0 such that *root* $(f(t_1, \cdots, t_n)) = f$

(ii) Let $D = \{root(l) \mid l \to r \in R\}$ be the set of *defined function symbols.* $D^* = \{f^* \mid f \in D\}$ is the set of *marked function symbols* assumed that $D^* \cap \Sigma = \emptyset$ and f^* has the arity of f. It is clear that $f^* \in D^*$ is not a defined function symbol. $T^* = T(\Sigma \cup D^*)$ is the set of *marked terms.*

(iii) Let t be a marked term. $e(t)$ denotes the term obtained from t by erasing all marks. $\delta(t)$ denotes the Ω-term obtained from t by replacing all the maximal subterms having defined function symbols at the roots with Ω. $\bar{\delta}(f(t_1, \ldots, t_n) \equiv f(\delta(t_1), \cdots, \delta(t_n))$ for $f \in \Sigma \cup D^*$.

5.2. DEFINITION. $t \in T^*$ is *well-marked* if $\forall s \subseteq t\ [root(s) \in D^* \ \Rightarrow e(\delta(s)) \in NF_\Omega]$.

5.3. LEMMA. *If $t \in T^*$ is well-marked then $e(\delta(t)) \in NF_\Omega$.*

PROOF. Trivial. □

5.4. LEMMA. *Let $\forall s \subseteq t\ [root(s) \in D^* \ \Rightarrow e(\delta(s)) \# Red]$. Then t is well-marked.*

PROOF. We will prove the lemma by induction on the size of t. The basic step is trivial. Induction step: Let $t \equiv h(t_1, \cdots, t_n)$. From I.H., every t_i is well-marked. If $h \notin D^*$, t is well-marked. Assume that $h \in D^*$, say $h = f^*$. Then, $e(\delta(t)) \equiv f(e(\delta(t_1)), \cdots, e(\delta(t_n))) \# Red$. Since every $e(\delta(t_i)) \in NF_\Omega$, it follows that $e(\delta(t)) \in NF_\Omega$. □

5.5. LEMMA. *Let t be well-marked and let $e(\bar{\delta}(t)) = C[\Omega_{TD}]$. Then $C[z] \in NF_\Omega$.*

PROOF. It follows directly from $C[z] \# Red$ and lemma 5.3. □

5.6. DEFINITION. Let $t \equiv C[t_1, \cdots, t_p, \cdots, t_n] \in T^*$ and $t' \equiv e(C)[\Omega, \ldots, \Omega_{TD}, \ldots, \Omega]$. Then we say that t_p is a *directed subterm of t with respect to t'.*

5.7. DEFINITION (Transitive Reduction Strategy). The *transitive strategy* has as input a term $t \in T$. s indicates a subterm occurrence of t.

(1) If t has no defined function symbol, terminate with "$e(t)$ is a normal form".

(2) Take the leftmost-outermost subterm of t having a defined function at the root as s.

(3) If $e(\bar{\delta}(s)) \succeq Red$, terminate with "$e(s)$ is an index of $e(t)$".

(4) If $e(\bar{\delta}(s)) \uparrow Red$, take a directed subterm of s with respect to $e(\bar{\delta}(s))$ as s and go to (3).

(5) Mark the root of s and go to (1).

5.8. THEOREM. *Let R be transitive and let $t \in T$.*

(i) *The transitive strategy applied to t terminates with either "t is a normal form" (a) or with "s is an index of t" (b).*

(ii) *In case (a) t is a normal form. Otherwise (case (b)), s is an index of t.*

PROOF. A sketch of our proof is as follows. The loop consisting of (3)-(4) decreases the size of s. The loop consisting of (1)-(5) decreases the number of the defined function symbols in t. Thus, the transitive strategy eventually terminates at (1) or (3). If t is a normal form, the strategy cannot terminate at (3). Thus, it terminates at (1). Let t be not a normal form. Note that the root of a redex in t cannot be marked. Hence, the strategy eventually terminates at (3) with indicating "$e(s)$ is an index of $e(t)$" where "$e(t) \equiv e(C)[e(s)_I]$". From lemma 5.4, t is well-marked. If at (4) $e(\bar{\delta}(s)) \uparrow Red$ and $e(\bar{\delta}(s)) \equiv C'[\Omega_{TD}]$, then, by lemma 5.5 we obtain $C'[z] \in NF_\Omega$. If at (2) t has no defined function symbol at the root, then $e(\delta(t)) \in NF_\Omega$. Thus, by applying lemma 4.10 it can be easily proven that $e(s)$ is an index of $e(t)$. □

6. Functional Strategy

The reduction order determined by the functional strategy is obtained via top-to-bottom, left-to-right pattern matching. In this section we will identify those TRS's for which this way of pattern matching always delivers a transitive direction. Note that the fact that an Ω-occurrence in a term t is a transitive direction according to some rule R may not be affected by the rules 'below' R. We will show that this requirement is met if each rule R' 'below' R is *left-incompatible* with R.

6.1. DEFINITION (Left-Incompatibility). Let $s, t \in T_\Omega$. The *left-incompatibility* of s and t, indicated by $t \#_< s$, is defined as follows:

(i) $t \not\equiv s$, $t \not\equiv \Omega$ $s \not\equiv \Omega$, and

(ii) $f = g \Rightarrow \exists i[(\forall j < i, t_j \preceq s_j) \wedge t_i \#_< s_i]$
where $t \equiv f(t_1, \cdots, t_n)$ and $s \equiv g(s_1, \cdots, s_m)$.

Here, the above i is called the *left-incompatible point*.

6.2. EXAMPLE. Let $Red = \{f(\Omega, 1), f(1, 0)\}$. Then one has $f(\Omega, 1) \#_< f(1, 0)$, but not $f(1, 0) \#_< f(\Omega, 1)$. Furthermore, notice that in $f(\Delta_1, \Delta_2)$ only Δ_2 is an index. If the rule $f(\Omega, 1)$ is applied first then only Δ_2 is indicated as an index. This is not the case when $f(1, 0)$ is applied first; then both redexes are indicated.

6.3. LEMMA. *Let* $C[\Omega] \uparrow p$ *and let* $C[\Omega_{\{p\}}]$ *be the leftmost direction for* $\{p\}$. *Let* $p \#_{<} q$. *Then* $C[\Omega_{\{q\}}]$.

PROOF. By induction on the size of $C[\,]$. Basic step $C[\,] \equiv \Box$ is trivial. Induction step: Let $C[\Omega] \equiv f(t_1, \cdots, t_d, \cdots, t_n)$ where the indicated Ω occurs in t_d, say $t_d \equiv C_d[\Omega]$. Since $C[\Omega] \uparrow p$, $p \equiv f(p_1, \cdots, p_d, \cdots, p_n)$ and $p_i \uparrow t_i$ for $i = 1 \cdots n$. Since $p \#_{<} q$, we have the left-incompatible point k for p and q.

$d < k$: Then $p_d \preceq q_d$. Since $C_d[z] \# p_d$, we have $C[z] \# q_d$. Hence, $C[z] \# q$.

$d = k$: Since $C_d[\Omega_{\{p_d\}}]$ is the leftmost direction for $\{p_d\}$ and $p_d \#_{<} q_d$, we can apply I.H. to them. Thus, $C_d[\Omega_{\{q_d\}}]$ is obtained. Thus, $C[z] \# q$.

$d > k$: Since $C[\Omega_{\{p\}}]$ is the leftmost direction for $\{p\}$, we obtain $t_k \succeq p_k$. Since $p_k \#_{<} q_k$, we obtain that $t_k \# q_k$. Hence, $C[z] \# q$. $\Box$

6.4. DEFINITION. An orthogonal TRS $(\Sigma,\, R)$ is *left-incompatible* if it satisfies the following two conditions:

(i) *Red* can be expressed as a list $[p_1, \cdots, p_n]$ with $p_i \#_{<} p_j$ if $i < j$,
(ii) $\forall p_i \in Red,\ q \in Red^{+}\ [p_i \#_{<} q]$, where $Red^{+} = Red^{*} - Red$.

6.5. LEMMA. *Let* R *be a left-incompatible TRS with* $Red = [p_1, \cdots, p_n]$. *Let* $C[\,]$ *be a context such that* $C[\Omega] \uparrow p_d$, $C[\Omega] \# p_i$ $(1 \leq i < d)$ *and let* $C[\Omega_{\{p_d\}}]$ *display the leftmost direction for* $\{p_d\}$. *Then* $C[\Omega_{TD}]$.

PROOF. Since $C[\Omega] \# p_i$ $(1 \leq i < d)$, we have $C[\Omega_{\{p_i\}}]$ $(1 \leq i < d)$. From the left-incompatibility, it follows that $p_d \#_{<} p_j$ $(d < j \leq n)$ and $p_d \#_{<} q$ for $q \in Red^{+}$. Thus, by lemma 6.3 we can show that $C[\Omega_{\{q\}}]$ for any $q \in Red^{*}$. $\Box$

6.6. COROLLARY. *Every left-incompatible system is transitive.*

PROOF. According to lemma 4.13 it is sufficient to prove that each $t \in Red^{\prec}$ has a transitive direction. Let $t \in Red^{\prec}$. Then there exists some $p_d \in Red$ such that $t \# p_i$ $(i < d)$ and $t \uparrow p_d$. Since $t \not\succeq p_d$, t must have a direction for $\{p_d\}$. By lemma 6.5, the leftmost direction of t for $\{p_d\}$ is a transitive direction. $\Box$

6.7. DEFINITION. Let R be a left-incompatible TRS with $Red = [p_1, \cdots, p_n]$ and let $t \equiv C[t_1, \ldots, t_k, \ldots, t_n] \in T^{*}$ and $t' \equiv C[\Omega, \ldots, \Omega, \ldots, \Omega]$. Furthermore, let d be a number such that $e(C)[\Omega, \cdots, \Omega, \cdots, \Omega] \# p_i$ for $1 \leq i < d$ and $e(C)[\Omega, \cdots, \Omega, \cdots, \Omega] \uparrow p_d$ (which means that p_d is the first compatible pattern in the list), and let $e(C)[\Omega, \cdots, \Omega_{\{p_d\}}, \cdots, \Omega]$ display the leftmost direction for $\{p_d\}$. Then we say that t_k is the *leftmost directed subterm of* t *with respect to* t' *and* p_d.

6.8. DEFINITION (Functional Reduction Strategy). The *functional strategy* has as input a term $t \in T$ and a TRS R which is left-incompatible with $Red = [p_1, \cdots, p_n]$. s indicates a subterm occurrence of t.

(1) If t has no defined function symbol, terminate with "$e(t)$ is a normal form".

(2) Take the leftmost-outermost subterm of t having a defined function at the root as s.

(3) Find the first compatible pattern p_d to $e(\bar{\delta}(s))$ in the list *Red* if it exists; otherwise, mark the root of s and go to (1).

(4) If $e(\bar{\delta}(s)) \succeq p_d$, terminate with "$e(s)$ is an index of $e(t)$".

(5) Take as s the leftmost directed subterm of s with respect to $e(\bar{\delta}(s))$ and p_d, and go to (3).

6.9. THEOREM. *Let R be left-incompatible system and let $t \in T$.*

(i) *The functional strategy applied to t terminates with either "t is a normal form" (a) or with "s is an index of t" (b).*

(ii) *In case (a) t is a normal form. Otherwise (case (b)), s is an index of t.*

PROOF. Note that if R is left-incompatible, then by lemma 6.5 it is clear that the functional strategy is essentially same to the transitive strategy. Thus, by Theorem 5.8 we can easily prove the theorem. □

O'Donnell (O'Donnell (1977)) proved that if an orthogonal term rewriting system R is left-normal then R is strongly sequential and leftmost-outermost reduction is normalizing. We now show that his result is a special case of the above theorem.

6.10. DEFINITION (Left-normal TRS's). (i) The set T_L of the left-normal terms is inductively defined as follows:

(1) $x \in T_L$ if x is a variable,

(2) $f(t_1, \cdots, t_{p-1}, t_p, t_{p+1} \cdots, t_n) \in T_L \quad (0 \leq p \leq n)$

if $t_1, \cdots, t_{p-1} \in T_0$ (i.e. $t_1, \cdots, t_{p-1}$ are groud terms), $t_p \in T_L$, and $t_{p+1}, \cdots, t_n$ are variables.

(ii) The set of the *left-normal schemata* is $T_{L\Omega} = \{t_\Omega \mid t \in T_L\}$.

(iii) R is *left-normal* (O'Donnell (1977),Huet and Lévy (1979),Klop (1992)) iff for any rule $l \to r$ in R, l is a left-normal term, i.e. $Red \subseteq T_{L\Omega}$.

6.11. LEMMA. *Let $p, q \in T_{L\Omega}$ and $p \# q$. Then $p \#_< q$.*

PROOF. By induction on the size of q. Let $p \equiv f(p_1, \cdots, p_m, \Omega, \cdots \Omega)$ and $q \equiv f(q_1, \cdots, q_n, \Omega, \cdots ,\Omega)$ where p_i $(i < m)$ and q_j $(j < n)$ have no Ω occurrences. Since $p \# q$, there exists some k $(k \leq m, n)$ such that $p_i \equiv q_i$ $(i < k)$ and $p_k \# q_k$. Note that $p_k, q_k \in T_{L\Omega}$. Thus, from I.H., $p_k \#_< q_k$ follows. Therefore, $p \#_< q$. □

6.12. THEOREM. *Let R be a left-normal orthogonal term rewriting system. Then, R is a left-imcompatible system.*

PROOF. From $Red^* \subseteq T_{L\Omega}$, the orthogonality of R, and lemma 6.11, we can easily show that R is left-incompatible. □

6.13. COROLLARY. *Let R be a left-normal orthogonal term rewriting system. Then the functional strategy applied to $t \notin NF$ indicates the leftmost-outermost redex of t as an index.*

PROOF. Follows directly from the definition of the functional strategy. □

6.14. EXAMPLE. The following R is left-incompatible but not left-normal. Hence, the functional strategy is normalizing for R. However, the leftmost-outermost reduction strategy is not.

$$R \quad \begin{cases} f(c(x,0),c(0,x)) \to 1 \\ g \to 0 \\ \omega \to \omega \end{cases}$$

Now consider the term $f(c(\omega,g),c(g,\omega))$. It is clear that the functional strategy is normalizing and leftmost-outermost reduction not.

7. Future Work

With respect to the functional reduction strategy there exist two major problems that have to be solved. Firstly, since the functional strategy is initially intended as a strategy for Priority Rewriting Systems, the adequacy of this strategy for Priority Term Rewriting Systems has to be investigated. An additional problem comes from the fact that there exists not always a well-defined semantics for a Priority Term Rewriting System. Secondly, implementations of (lazy) functional languages that are using this strategy appear to be efficient. It should be investigated whether this practical efficiency can be founded theoretically.

References

Augustsson, L. (1984). A compiler for lazy ml, *Proc. of ACM Symposium on LISP and Functional Programming*, pp. 218–227.

Baeten, J.C.M., J.A. Bergstra and J.W. Klop (1987). Term rewriting systems with priorities, *Proc. of Conference on Rewriting Techniques and Applications*, Bordaux, Springer Verlag, LNCS 256, pp. 83–94.

Barendregt, H.P., M.C.J.D. van Eekelen, J.R.W. Glauert, J.R. Kennaway, M.J. Plasmeijer and M.R. Sleep (1987). Term graph reduction, *Proc. of Parallel Architectures and Languages Europe (PARLE)*, Eindhoven, The Netherlands, Springer Verlag, LNCS 259 II, pp. 141–158.

Brus, T., M.C.J.D. van Eekelen, M. van Leer, M.J. Plasmeijer and H.P. Barendregt (1987). Clean - a language for functional graph rewriting, *Proc. of Conference on Functional Programming Languages and Computer Architecture (FPCA '87)*, Portland, Oregon, USA, Springer Verlag, LNCS 274, pp. 364–384.

Dershowitz, N. and J.P. Jouannaud (1990). Rewrite systems, *in:* Leeuwen (ed.), *Formal Models and Semantics*, Elsevier, Handbook of Theoretical Computer Science B, pp. 243–320.

Hudak, P., S.L. Peyton Jones, P.L. Wadler, Arvind, B Boutel, J. Fairbairn, J. Fasel, K. Guzman, K. Hammond, J. Hughes, T. Johnsson, R. Kieburtz, R.S. Nikhil, W. Partain and J. Peterson (1992). Report on the functional programming language haskell, version 1.2, *Special Issue of SIGPLAN Notices*.

Huet, G. and J.J. Lévy (1979). Call by need computations in non-ambiguous linear term rewriting systems, *Technical Report 359*, INRIA.

Kennaway, J.R. (1990). The specificity rule for lazy pattern-matching in ambiguous term rewriting systems, *Proc. of 3rd European Symposium on Programming (ESOP)*, Springer Verlag, 432.

Klop, J.W. (1992). Term rewriting systems, *in:* Gabbay Abramsky and Maibaum (eds.), *Handbook of Logic in Computer Science*, Oxford University Press, I.

Laville, A. (1987). Lazy pattern matching in the ml language, *Proc. of 7th Conference on Software Technology and Theoretical Computer Science*, Pune, India, Springer Verlag, 287, pp. 400–419.

Nöcker, E.G.J.M.H., J.E.W. Smetsers, M.C.J.D. van Eekelen and M.J. Plasmeijer (1991). Concurrent clean, *Proc. of Parallel Architectures and Languages Europe (PARLE'91)*, Eindhoven, The Netherlands, Springer Verlag, LNCS 505, pp. 202–219.

O'Donnell, M.J. (1977). *Computing in Systems Described by Equations*, LNCS 58, Springer Verlag.

Puel, L. and A. Suárez (1990). Compiling pattern matching by term decomposition, *Proc. of ACM conference on LISP and Functional Programming*, pp. 273–281.

Turner, D.A. (1985). Miranda: A non-strict functional language with polymorphic types, *Proc. of Conference on Functional Programming Languages and Computer Architecture (FPCA '85)*, Nancy, France, Springer Verlag, LNCS 201, pp. 1–16.