Preliminary version. Final version in Information Processing Letters (Elsevier), Vol. 67, No. 1, pp. 1–8, 1998

Strongly sequential and inductively sequential term rewriting systems

Michael Hanus* Salvador Lucas† Aart Middeldorp‡

Abstract

The concept of definitional tree by Antoy serves to introduce control information into the bare set of rules of a constructor-based term rewriting system (TRS). TRSs whose rules can be arranged into a definitional tree are called *inductively sequential*. By relying on the existence of such a definitional tree, an optimal rewriting strategy, the *outermost-needed strategy* is defined. Optimality was proved w.r.t. the Huet and Lévy's theory of neededness. In this paper, we prove that strongly sequential and inductively sequential constructor-based TRSs coincide. We also show that outermost-needed rewriting only reduces strongly needed redexes.

Keywords: definitional trees, neededness, strong sequentiality, term rewriting, declarative programming.

1 Introduction

For orthogonal TRSs, the best normalizing strategy which avoid unnecessary reductions is *needed rewriting* [7]. Needed rewriting only considers *needed* redexes, i.e., redexes which are contracted (themselves or some residual) in every normalizing derivation. Unfortunately, it is undecidable whether a redex is needed. Strongly sequential redexes approximate needed redexes. However, reduction of strong redexes is costly to implement [10], and some efforts have been done to simplify the (strongly) needed reduction process.

Definitional trees [1] allow us to define optimal strategies both in rewriting [1] and narrowing [3]. A definitional tree consists of branches on the values of particular arguments and the rules of the TRS.

*Informatik II, RWTH Aachen, D-52056 Aachen, Germany, `hanus@informatik.rwth-aachen.de`. Work partially supported by DFG (under grant Ha 2457/1-1) and Acción Integrada.

†DSIC, U.P. de Valencia, Camino de la Vera s/n, Apdo. 22012, E-46071 Valencia, Spain, `slucas@dsic.upv.es`. Work partially supported by EEC-HCM grant ERBCHRXCT940624, Bancaixa (Bancaja-Europa grant), Acción Integrada (HA1997-0073) and CICYT (under grant TIC 95-0433-C03-03).

‡Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba 305-8573, Japan, `ami@score.is.tsukuba.ac.jp`.

Example 1.1 Consider the following rules:

```
first(0,x)→ []
first(s(x),y::z) → y::first(x,z)
```

We give a graphic representation of the definitional tree for the function `first` (Figure 1). The first branch is done on the first argument ($\underline{x_1}$) with alternative patterns `0` and `s(x)`, where a further branch on $\underline{x_2}$ (with only one alternative) is necessary for the latter pattern.

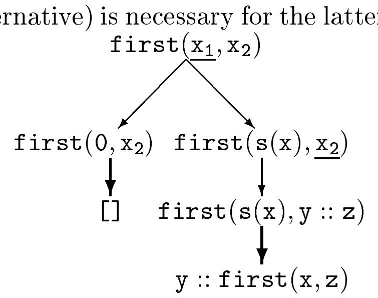

Figure 1: Definitional tree for the function `first`

A function f, defined by rules of a TRS $\mathcal{R}$ is *inductively sequential* if there exists a definitional tree $\mathcal{P}$ containing all rules defining f. $\mathcal{R}$ is *inductively sequential* if all defined symbols are inductively sequential.

Recently, definitional trees and inductively sequential TRS became important for declarative programming languages since they model lazy functional languages with pattern matching and are an adequate basis to implement optimal evaluation strategies for functional logic languages [2]. Moreover, they can be easily extended to more general classes of TRSs [4].

The relevance of inductively sequential TRSs for declarative programming languages raised the question about their relationship to the classical concept of strong sequentiality. In [3], the question was raised whether the classes of strongly sequential and inductively sequential constructor-based TRSs are the same. In this paper we formally show that the two classes coincide. Furthermore, we generalize the notion of a definitional tree to deal also with non-constructor-based TRSs. This allows us to compare definitional trees to other structures used to implement strong sequentiality, like *index trees* and *forward-branching index trees* [6, 11], and the *matching dags* of Huet and Lévy [7].

A definitional tree determines a rewriting strategy, namely the *outermost-needed strategy*. We prove that this strategy is an *index reduction* strategy, i.e., it only selects strong indices for reduction. Since we can use the properties of strong indices in constructor-based TRSs, our proof is easier than the one in [1].

In Section 2, we review the technical concepts used in the remainder of the paper. In Section 3, we introduce definitional trees. Section 4 explores the relation between strongly sequential and inductively sequential TRSs. Section 5 compares to other approaches.

2 Preliminaries

This section introduces our main notations (see [5, 8] for full definitions). Given a partial order $\leq$ on a set A and $a \in A$, $a{\uparrow} = \{b \in A \mid a \leq b\}$ is the *upward set associated with* a. $a \parallel b$ means $a \not\leq b$ and $b \not\leq a$.

V denotes a set of *variables* and Σ denotes a set of *function symbols* $\{\mathtt{f}, \mathtt{g}, \ldots\}$, each with a fixed *arity* given by a function $ar : \Sigma \to \mathbb{N}$. We denote the set of *terms* by $\mathcal{T}(\Sigma, V)$. A k-tuple $t_1, \ldots, t_k$ of terms is denoted by $\tilde{t}$, where k will be clarified by the context. The set of variables appearing in a term t is denoted by $\mathcal{V}ar(t)$.

Terms are viewed as labeled trees in the usual way. *Occurrences* $u, v, \ldots$ are represented by chains of positive natural numbers used to address subterms of t. Occurrences are ordered by the standard prefix ordering: $u \leq v$ iff $\exists v'$ such that $v = u.v'$. The empty occurrence is denoted by ϵ. $O(t)$ denotes the set of occurrences of t. The subterm at occurrence u of t is denoted by $t|_u$. The term t with the subterm at the occurrence u replaced with s is denoted by $t[s]_u$. The symbol labeling the root of t is denoted by $root(t)$.

A *rewrite rule* is an ordered pair (l, r), written $l \to r$, with $l, r \in \mathcal{T}(\Sigma, V)$, $l \notin V$ and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$. Given a rule $l \to r$, l is called the left-hand side (*lhs*) of the rule and r the right-hand side (*rhs*). A *TRS* is a pair $\mathcal{R} = (\Sigma, R)$ where R is a set of rewrite rules. A *left-linear* TRS is a TRS where every *lhs* is a linear term. An *orthogonal* TRS is a left-linear TRS without overlapping rules, i.e., given a rule $l \to r$, there is no non-variable occurrence $u \in O(l)$ such that $l|_u$ unifies with a *lhs* l' of a rule $l' \to r'$ in the TRS (where $l \to r$ and $l' \to r'$ are different in case of $u = \epsilon$).

A term t *rewrites* to a term s, written $t \to_{\mathcal{R}} s$, if $t|_u = \sigma(l)$ and $s = t[\sigma(r)]_u$, for some rule $l \to r$, $u \in O(t)$ and substitution σ. $O_{\mathcal{R}}(t) = \{u \in O(t) \mid \exists l \to r \in \mathcal{R}$ and σ with $t|_u = \sigma(l)\}$ is the set of *redex occurrences* in t.

Given a TRS, we split the signature into the disjoint union $\Sigma = \mathcal{C} \uplus \mathcal{F}$ of symbols $c \in \mathcal{C}$, called *constructors*, having no associated rule and symbols $f \in \mathcal{F}$, called *defined functions* or *operations*, which are defined by some rule $f(\tilde{l}) \to r \in R$. Each rule $f(\tilde{\delta}) \to r$ in a *constructor-based* TRS or *constructor system* (*CS*) must satisfy $f \in \mathcal{F}$ and $\tilde{\delta} \in \mathcal{T}(\mathcal{C}, V)^{ar(f)}$.

3 Definitional trees

We generalize the notion of (partial) definitional tree (*pdt*) by using its "declarative" definition (see [2]). To represent *unknown* parts of a term t we use the symbol Ω. Terms in $\mathcal{T}(\Sigma \cup \{\Omega\}, V)$ are called Ω-*terms*. To discuss about unknown portions of expressions, we use the ordering $\leq$ on Ω-terms given by: $\Omega \leq t$ for all $t \in \mathcal{T}(\Sigma \cup \{\Omega\}, V)$, $x \leq x$ for all $x \in V$, and $f(\tilde{t}) \leq f(\tilde{s})$ if $t_i \leq s_i$ for $1 \leq i \leq ar(f)$. In this way, $t \leq s$ means "t is less or equally defined than s". $O_\Omega(t) = \{u \in O(t) \mid t|_u = \Omega\}$ is the set of occurrences of Ω in t.

A *definitional tree of a finite set of (incomparable)* Ω*-terms* $S \subseteq \mathcal{T}(\Sigma \cup \{\Omega\})$ *with pattern* $\pi \in \mathcal{T}(\Sigma \cup \{\Omega\})$ is a non-empty, ordered set $\mathcal{P}$ of Ω-terms having the following properties:

- There is a minimum element which is the pattern of the *pdt*: $min(\mathcal{P}) = \pi$ (*minimum property*).
- The maximal elements are the elements of S: $maximal(\mathcal{P}) = S$ (*leaves property*).
- If $\pi' \in \mathcal{P}$, $\pi' \neq \pi$, there is a unique $\pi'' \in \mathcal{P}$, $\pi'' < \pi'$, such that there is no $\pi''' \in \mathcal{T}(\Sigma \cup \{\Omega\})$ with $\pi'' < \pi''' < \pi'$ (*parent property*).
- Given $\pi' \in \mathcal{P} \backslash S$, there is an occurrence $u \in O_\Omega(\pi')$ (called the *inductive* occurrence), and symbols[1] $f_1, \ldots, f_n \in \Sigma$ with $f_i \neq f_j$ for $i \neq j$, such that, for all $\pi_1, \ldots, \pi_n$ which are immediately below π', $\pi_i = \pi'[f_i(\widetilde{\Omega})]_u$ for all $1 \leq i \leq n$ (*induction property*).

These properties entail $S \subseteq \mathcal{P} \subseteq \pi{\uparrow}$. Given a TRS $\mathcal{R} = (\mathcal{F} \uplus \mathcal{C}, R)$, a *redex scheme* of $\mathcal{R}$ is a *lhs* of a rule $l \to r$ where all variables are replaced by Ω. Let $L_\Omega(\mathcal{R})$ be the set of redex schemes of $\mathcal{R}$. Since we deal with orthogonal TRSs, we assume that a bijective function $\rho : L_\Omega(\mathcal{R}) \to R$ associates the rule which corresponds to each redex scheme. A *preredex* of a redex scheme l is an Ω-term π such that $\pi \leq l$. Let $L_\Omega^<(\mathcal{R}) = \{\pi \mid \exists l \in L_\Omega(\mathcal{R})$ with $\pi < l\}$. For $f \in \mathcal{F}$, let $L_\Omega^f(\mathcal{R}) = \{l \in L_\Omega(\mathcal{R}) \mid root(l) = f\}$. f is called *inductively sequential* if there exists a definitional tree $\mathcal{P}_f$ which is a *pdt* of $L_\Omega^f(\mathcal{R})$ with pattern $f(\widetilde{\Omega})$. $\mathcal{R}$ is called *inductively sequential* if all defined symbols $f \in \mathcal{F}$ are inductively sequential. An inductively sequential TRS

[1] In the original definition of definitional trees, only constructor symbols are allowed.

can be viewed as a set of definitional trees, each defining a function symbol. By using a representation function pdt, we can represent a *pdt* $\mathcal{P}$ of a set $S \subseteq L_\Omega(\mathcal{R})$ with pattern π as a term $\mathsf{pdt}(\mathcal{P})$ as follows:

$\mathsf{pdt}(\mathcal{P}) = \mathit{rule}(\rho(\pi))$ if $\mathcal{P} = \{\pi\} = S$.

$\mathsf{pdt}(\mathcal{P}) = \mathit{branch}(\pi, u, \mathsf{pdt}(\mathcal{P}_1), \ldots, \mathsf{pdt}(\mathcal{P}_n))$ if $\mathcal{P}$ is not a singleton, where $\pi = \mathit{min}(\mathcal{P})$, u is the inductive occurrence of π, $f_1, \ldots, f_n \in \Sigma$, $f_i \neq f_j$ if $i \neq j$, and, for all i, $1 \leq i \leq n$, $\mathcal{P}_i$ is a *pdt* with pattern $\pi_i = \pi[f_i(\widetilde{\Omega})]_u$ of the set $S_i = S \cap \pi_i\!\uparrow$.

Example 3.1 Consider the program of Example 1.1. Then (we use Ω's instead of variables),

$$\begin{array}{l} \mathit{branch}(\mathtt{first(}\Omega\mathtt{,}\Omega\mathtt{)}, 1, \\ \quad \mathit{rule}(\mathtt{first(0,y)} \rightarrow \mathtt{[]}), \\ \quad \mathit{branch}(\mathtt{first(s(}\Omega\mathtt{),}\Omega\mathtt{)}, 2, \\ \qquad \mathit{rule}(\mathtt{first(s(x),y::z)} \rightarrow \mathtt{y::first(x,z)}))) \end{array}$$

is a definitional tree for the function `first` (Figure 1).

4 Strong sequentiality and inductively sequential TRSs

Regarding normalization strategies, the main result of Huet and Lévy [7, 9] is the following: reduction of needed redexes is normalizing for orthogonal TRSs. In general, the occurrences of such needed redexes are undecidable, but Huet and Lévy define a computable approximation, the (strong) *indices*. To obtain such an approximation, they use Ω-terms.

To calculate indices a function ω is used. It is defined by means of a reduction relation $\rightarrow_\Omega$ [9]: $C[t] \rightarrow_\Omega C[\Omega]$ if $t \neq \Omega$ and there exists $l \in L_\Omega(\mathcal{R})$ such that $t \uparrow l$, i.e., there exists an Ω-term s such that $t \leq s$ and $l \leq s$. The relation $\rightarrow_\Omega$ is confluent and terminating (see [7, 9]). Let $\omega(t)$ be the $\rightarrow_\Omega$-normal form of t. Instead of the usual definition of index, based on the notion of sequential predicate, we use an equivalent, simpler characterization (see [7, 9]).

Definition 4.1 *Let $t \in \mathcal{T}(\Sigma \cup \{\Omega\}, V)$ and $u \in O_\Omega(t)$. Let $\bullet$ be a fresh constant symbol, and $t' = t[\bullet]_u$. Then u is an index of t iff $\omega(t')|_u = \bullet$ (sometimes we write $\bullet \in \omega(t')$ for short). The set of indices of t is denoted by $\mathcal{I}(t)$.*

Proposition 4.2 ([9]) *If $u.v \in \mathcal{I}(t[s]_u)$, then $u \in \mathcal{I}(t[\Omega]_u)$ and $v \in \mathcal{I}(s)$.*

Proposition 4.3 ([9]) *If $u \in \mathcal{I}(t)$ and $t \leq t'$, then $u \in \mathcal{I}(t'[\Omega]_u)$.*

An Ω-normal form is an Ω-term t such that $O_\mathcal{R}(t) = \varnothing$ and $O_\Omega(t) \neq \varnothing$. Strongly sequential TRSs are defined as follows.

Definition 4.4 ([8]) *An orthogonal TRS is strongly sequential if every Ω-normal form has an index.*

When considering CSs, things are simpler.

Proposition 4.5 ([9]) *An orthogonal CS $\mathcal{R}$ is strongly sequential iff $\forall \pi \in L_\Omega^<(\mathcal{R}) \backslash \{\Omega\}, \mathcal{I}(\pi) \neq \varnothing$.*

We use the following property of indices in CSs.

Proposition 4.6 ([9]) *Let $\mathcal{R}$ be an orthogonal CS. Let $u \in \mathcal{I}(t)$ and s such that $\mathit{root}(s) \in \mathcal{F}$ and $v \in \mathcal{I}(s)$. Then $u.v \in \mathcal{I}(t[s]_u)$.*

A strategy which always reduces redexes pointed by indices is called *index reduction*.

Theorem 4.7 ([7]) *Index reduction is normalizing for orthogonal, strongly sequential TRSs.*

4.1 Inductive sequentiality of strongly sequential TRSs

Let $t \in \mathcal{T}(\Sigma \cup \{\Omega\})$. $t\!\uparrow^<$ is the set of terms which are greater than t: $t\!\uparrow^<= \{s \in \mathcal{T}(\Sigma \cup \{\Omega\}) \mid t < s\}$. Given $u \in O(t)$, $t\!\downarrow^u_<$ is the set of terms which are smaller than t and whose subterm at occurrence u is not Ω: $t\!\downarrow^u_<= \{s \in \mathcal{T}(\Sigma \cup \{\Omega\}) \mid s < t \wedge u \in O(s) \wedge s|_u \neq \Omega\}$.

Given a set of terms $S \subseteq L_\Omega(\mathcal{R})$ and an occurrence $u \in O(s)$ for all $s \in S$, we define the equivalence relation $\equiv_u$ by $s \equiv_u s'$ iff $\mathit{root}(s|_u) = \mathit{root}(s'|_u)$, i.e., the terms have the same symbol rooting the subterm at the occurrence u.

In the remainder of the paper, given Ω-terms π and $l \in \pi\!\uparrow$ and an occurrence $u \in O(\pi)$, we define $\Pi(l, \pi, u) = \pi\!\uparrow^< \cap\, l\!\downarrow^u_<$. The function `nodes` builds a *pdt* for a given function definition:

$$\begin{array}{l} \mathtt{nodes}(S, \pi, u) \ \mathtt{=} \\ \quad \mathtt{if}\ S = \{l\}\ \mathtt{and}\ \Pi(l, \pi, u) = \varnothing\ \mathtt{then} \\ \qquad \mathit{rule}(\rho(l)) \\ \quad \mathtt{else\ let}\ \pi' = \mathit{min}(\cup_{l \in S} \Pi(l, \pi, u)) \\ \qquad\qquad u' \in \mathcal{I}(\pi') \\ \qquad\qquad \{S_1, \ldots, S_n\} = S/\!\equiv_{u'} \\ \qquad\ \mathtt{in}\ \ \mathit{branch}(\pi', u', \mathtt{nodes}(S_1, \pi', u'), \ldots, \\ \qquad\qquad\qquad \mathtt{nodes}(S_n, \pi', u')) \end{array}$$

Lemma 4.8 *Let $\pi \in \mathcal{T}(\Sigma \cup \{\Omega\})$, $u \in O_\Omega(\pi)$ and $S \subseteq \pi\!\uparrow$, such that, $\exists f \in \Sigma. \forall l \in S$, $\mathit{root}(l|_u) = f$ and there exists $l \in S$ with $l\!\downarrow^u_< \neq \varnothing$. Let $\Pi_S = \cup_{l \in S} \Pi(l, \pi, u)$. Then, $\mathit{min}(\Pi_S) = \pi[f(\widetilde{\Omega})]_u$.*

PROOF. Let $\pi' = \pi[f(\widetilde{\Omega})]_u$. Clearly, $\pi' \in \pi\!\uparrow^<$ and it is minimal in $\pi\!\uparrow^<$. Let $l \in S$ be such that $l\!\downarrow^u_< \neq \varnothing$. Clearly, $\Pi(l, \pi, u) \neq \varnothing$ because, since $l \in \pi\!\uparrow$, $l\!\downarrow^u_< \neq \varnothing$, and $l|_u \neq \Omega$, we have that $\pi' \in l\!\downarrow^u_<$. Therefore, π' is minimal in $\Pi(l, \pi, u)$. Let $\pi'' \in \Pi(l, \pi, u)$. Since $root(l|_u) = f$ and $root(\pi''|_u) \neq \Omega$, it must be that $root(\pi''|_u) = f$. Thus $\pi' \leq \pi''$. Since π'' is arbitrary, it follows that $\Pi(l, \pi, u)$ has a minimum element $min(\Pi(l, \pi, u)) = \pi'$. Since this holds for every $l \in S$ with $l\!\downarrow^u_< \neq \varnothing$ and the elements $l \in S$ with $l\!\downarrow^u_< = \varnothing$ do not introduce new elements in Π_S, we obtain $\min(\Pi_S) = \pi'$. □

The *height* h of a finite ordered set is the number of elements n of the largest strict chain $a = a_1 < a_2 < \cdots < a_n = b$ going from a minimal element a to a maximal element b. We define $h = 0$ if the set is empty. Then, we can prove the following result.

Theorem 4.9 *Let $\mathcal{R}$ be an orthogonal, strongly sequential TRS. Then, for all defined symbols f, $\mathtt{nodes}(L^f_\Omega(\mathcal{R}), \Omega, \epsilon)$ is a definitional tree for f.*

PROOF. We consider a generic call $\mathtt{nodes}(S, \pi, u)$ under the restrictions $\pi \in \mathcal{T}(\Sigma \cup \{\Omega\})$, $u \in O_\Omega(\pi)$, and $S \subseteq \pi\!\uparrow \cap\, L_\Omega(\mathcal{R})$ non-empty and such that $\exists f \in \Sigma, \forall l \in S$, $root(l|_u) = f$. First, we prove that `nodes` builds a *pdt* for S with pattern $\pi[f(\widetilde{\Omega})]_u$. Given $l \in S$, let h_l denote the height of $\Pi(l, \pi, u)$. We proceed by induction on the height $h_S = max_{l \in S}(h_l)$ of $\Pi_S = \cup_{l \in S}\Pi(l, \pi, u)$.

$h_S = 0$: Note that $h_S = 0$ implies that, for all $l \in S$, $h_l = 0$, i.e., $\Pi(l, \pi, u) = \varnothing$ for all $l \in S$. Moreover, orthogonality implies that $S = \{l\}$. Otherwise, since for all (distinct) $l, l' \in S$, we have $root(l|_u) = root(l'|_u) = f$, it holds that $\pi[f(\widetilde{\Omega})]_u \leq l$ and $\pi[f(\widetilde{\Omega})]_u \leq l'$. Orthogonality implies that $\pi[f(\widetilde{\Omega})]_u < l$. This means that $\Pi(l, \pi, u) \neq \varnothing$, a contradiction. Therefore, we are in the `if` part of `nodes` and the conclusion is immediate.

$h_S > 0$: Since $h_S > 0$ implies that there is $l \in S$ with $h_l > 0$, this means that $\Pi(l, \pi, u) \neq \varnothing$ for this l. Hence, we are in the `else` part of `nodes`. By definition of $\Pi(l, \pi, u)$, we have $l\!\downarrow^u_< \neq \varnothing$. Then, by Lemma 4.8, π' in the algorithm is correctly defined as $\pi' = min(\Pi_S) = \pi[f(\tilde{\Omega})]_u$. Since $\pi' < l$ and $l \in L_\Omega(\mathcal{R})$, by orthogonality, π' is an Ω-normal form. By strong sequentiality, there exists $u' \in \mathcal{I}(\pi')$. Strong sequentiality ensures that, for each $l \in S$, $l|_{u'} \neq \Omega$, i.e., $root(l|_{u'}) = g_l \in \Sigma$. Otherwise, u' is not an index, since $\pi'[\bullet]_{u'}$ can be refined to a redex of l, and hence $\bullet \notin \omega(\pi'[\bullet]_{u'}) = \Omega$. Since $\pi < \pi'$, the height h'_l of $\Pi(l, \pi', u')$ is less than h_l, for each $l \in S$. Thus, we apply the I.H.: each $\mathcal{P}_i = \mathtt{nodes}(S_i, \pi', u')$, $1 \leq i \leq n$, is a *pdt* for S_i with pattern $\pi''_i = \pi'[g_i(\widetilde{\Omega})]_{u'}$, where g_i is the common symbol at occurrence u' of each $l \in S_i$. Thus, $\mathcal{P} = branch(\pi', u', \mathtt{nodes}(S_1, \pi', u'), \ldots, \mathtt{nodes}(S_n, \pi', u'))$ immediately satisfies the *minimum* and *leaves* property. By minimality of each π''_i in $\mathcal{P}_i$, and by the definition of $\mathcal{P}$, it satisfies the *parent* and *induction* properties too.

Now we apply `nodes` to the arguments $L^f_\Omega(\mathcal{R}), \Omega, \epsilon$ in the hypothesis and we obtain the desired result. □

$$
\begin{array}{ccc}
\mathtt{f}(\underline{\Omega}, \Omega) & \qquad & \mathtt{f}(\underline{\Omega}, \Omega) \\
\downarrow & & \downarrow \\
\mathtt{f}(\mathtt{g}(\underline{\Omega}, \Omega), \Omega) & & \mathtt{f}(\mathtt{g}(\Omega, \underline{\Omega}), \Omega) \\
\downarrow & & \downarrow \\
\mathtt{f}(\mathtt{g}(\mathtt{h}(\Omega), \underline{\Omega}), \Omega) & & \mathtt{f}(\mathtt{g}(\underline{\Omega}, \mathtt{a}), \Omega) \\
\downarrow & & \downarrow \\
\mathtt{f}(\mathtt{g}(\mathtt{h}(\Omega), \mathtt{a}), \Omega) & & \mathtt{f}(\mathtt{g}(\mathtt{h}(\Omega), \mathtt{a}), \Omega) \\
\Downarrow & & \Downarrow
\end{array}
$$

Figure 2: Two *pdt*s for `f`

The previous definition of `nodes` was closer to our definition of a *pdt* as a kind of ordered set which simplified the proofs. However, using the previous results, we can give a more readable version of the algorithm. Remember that $\rho(\pi)$ associates with $\pi \in L_\Omega(\mathcal{R})$ the rule $l \to r$ such that π is a redex scheme of l.

```
nodes(S, π, u) =
  if S = {π} then rule(ρ(π))
  else let f = root(l|_u) for some l ∈ S
           π' = π[f(Ω̃)]_u
           u' ∈ I(π')
           {S_1, ..., S_n} = S/≡_u'
       in  branch(π', u', nodes(S_1, π', u'), ...,
                          nodes(S_n, π', u'))
```

4.2 Strong sequentiality of inductively sequential TRSs

Proposition 4.10 *Let $\mathcal{P}_f$ be a pdt for the function f of a CS. Then every inductive occurrence u in a branch node $branch(\pi, u, \widetilde{\mathcal{P}})$ of $\mathcal{P}_f$ satisfies $u \in \mathcal{I}(\pi)$.*

PROOF. Let $\mathcal{P}_f = branch(\pi, u, \widetilde{\mathcal{P}})$. Note that $u \in O_\Omega(\pi)$. By contradiction: Since we consider CSs, if $u \notin \mathcal{I}(\pi)$, then $\pi[\bullet]_u \uparrow l$ for some redex scheme l in $maximal(\cup_{\mathcal{P} \in \widetilde{\mathcal{P}}}\mathcal{P})$. Since $\pi < l$, this means that the pattern π' of $\mathcal{P}$, the *pdt* in $\widetilde{\mathcal{P}}$ which contains l, verifies $root(\pi'|_u) = \Omega$, contradicting the definition of a *pdt*, since it must be that $root(\pi'|_u) \in \Sigma$. □

This result does not hold for arbitrary *pdt*s.

Example 4.11 *Consider the orthogonal TRS*

$\mathtt{f(g(h(x),a),y)} \to \mathtt{x} \qquad \mathtt{c} \to \mathtt{a}$
$\mathtt{g(x,b)} \to \mathtt{g(h(x),a)}$

Partial definitional trees for f *are drawn in Figure 2.*

The inductive occurrence 1.1 *is* not *an index in the pattern* $f(g(\Omega,\Omega),\Omega)$ *of the first pdt. By using* nodes*, we obtain the second definitional tree for which every inductive occurrence is an index.*

Theorem 4.12 *Let* $\mathcal{R} = (\Sigma, R)$ *be an inductively sequential CS. Then,* $\mathcal{R}$ *is strongly sequential.*

Proof. From Proposition 4.5, we prove by contradiction that every proper prefix $\Omega < p < l$ of a redex scheme l has an index. Assume $\mathcal{I}(p) = \varnothing$. Let $\Omega < \pi_1 < \cdots < \pi_n < l$ be the chain of patterns in the branch nodes of a *pdt* for the function $f = root(p)$ which contains l. It is not possible to have a π_j, $1 \le j \le n$ such that $p \le \pi_j$. Otherwise, by Proposition 4.10 and Proposition 4.2, p also has an index. Thus, $\pi_i \parallel p$ for some i, $1 \le i \le n$. Let us consider the maximal $\pi \in \{\pi_1, \ldots, \pi_n\}$ such that $\pi \le p$ and $\pi \le \pi_i$. π exists, because $\pi_1 = f(\widetilde{\Omega})$, and $\Omega < f(\widetilde{\Omega}) \le p < l$. Let u be the inductive occurrence for the branch node with pattern π. By Proposition 4.10, $u \in \mathcal{I}(\pi)$. We have $p|_u = \Omega$. Otherwise, since $p < l$ and $root(l|_u) = root(p|_u)$, there is $\pi' > \pi$ such that $\pi' \le p$ and $\pi' \le \pi_i$, thus contradicting the maximality of π. By Proposition 4.3, $u \in \mathcal{I}(p)$. □

This theorem does not hold for general strongly sequential TRSs, as the following example shows.

Example 4.13 *Consider the following TRS which is not strongly sequential (from [7]):*

$\mathtt{f(g(a,x),f(b,y))} \to \mathtt{x} \qquad \mathtt{g(d,d)} \to \mathtt{d}$
$\mathtt{f(g(x,a),f(c,y))} \to \mathtt{x}$

f *and* g *admit definitional trees, and* nodes *can build them, because every redex scheme has some index.*

Theorems 4.9 and 4.12 entail our main result.

Theorem 4.14 *An orthogonal CS is strongly sequential iff it is inductively sequential.*

4.3 Outermost-needed reduction

A definitional tree determines a rewriting strategy, namely the *outermost-needed rewriting* strategy[2]:

Definition 4.15 ([1]) *The (partial) function* φ *takes arguments* $t = f(\tilde{t})$, $f \in \mathcal{F}$ *and a pdt* $\mathcal{P}$ *such that* $min(\mathcal{P}) \le t$*, and yields a redex occurrence* $u \in O_{\mathcal{R}}(t)$*:*

$$\varphi(t,\mathcal{P}) = \begin{cases} \epsilon & \textit{if } \mathcal{P} = rule(\alpha) \\ \varphi(t,\mathcal{P}_i) & \textit{if } \mathcal{P} = branch(\pi,u,\mathcal{P}_1,\ldots,\mathcal{P}_n) \\ & \quad \textit{and } min(\mathcal{P}_i) \le t \textit{ for some } i \\ u.\varphi(t|_u,\mathcal{P}_g) & \textit{if } \mathcal{P} = branch(\pi,u,\mathcal{P}_1,\ldots,\mathcal{P}_n), \qquad (*) \\ & \quad root(t|_u) = g \in \mathcal{F}, \textit{ and} \\ & \quad \mathcal{P}_g \textit{ is a definitional tree for } g. \end{cases}$$

[2] This is a slightly different definition because we do not allow for *exempt* nodes as in [1]. However, it is equivalent.

Note that, dealing with CSs, the second and third cases are disjoint. This is because if $root(t|_u) = g \in \mathcal{F}$, then since $\pi \le min(\mathcal{P}_i)$ for all sub*pdt* $\mathcal{P}_i$ of $\mathcal{P} = branch(\pi,u,\mathcal{P}_1,\ldots,\mathcal{P}_n)$, it is not possible to have $min(\mathcal{P}_i) \le t$ since $root(min(\mathcal{P}_i)|_u) \in \mathcal{C}$. We show that φ is equivalent to *index reduction*.

Theorem 4.16 *Let* $\mathcal{R}$ *be an inductively sequential CS and* $u = \varphi(t,\mathcal{P})$*. Then* u *is an index of* $t[\Omega]_u$*.*

Proof. Induction on the number of visited definitional trees. In the case base ($n = 1$), u is ϵ and the conclusion easily follows. Otherwise ($n > 1$), the occurrence u can be split up into $u = v.w$, where v is an occurrence of the *pdt* $\mathcal{P}$ and w has been used to reduce $t|_v$ and, because $\mathcal{R}$ is a CS, $root(t|_v) \in \mathcal{F}$, with v the inductive occurrence for some pattern in a branch node of $\mathcal{P}$. By Proposition 4.10, $v \in \mathcal{I}(\pi)$. Since $\pi \le t$, by Proposition 4.3, $v \in \mathcal{I}(t[\Omega]_v)$. Then, by I.H., $w \in \mathcal{I}(t|_v[\Omega]_w)$ and the conclusion follows by Proposition 4.6. □

Theorem 4.9 suggests that definitional trees can be used with general strongly sequential TRSs. The outermost-needed strategy, as given in Definition 4.15 cannot be used to successfully evaluate a term in general (i.e., non-constructor-based) TRSs. For instance, consider the TRS in Example 4.11, $t = \mathtt{f(g(x,b),y)}$ and let $\mathcal{P}_f$ be a *pdt* for the function f if we try to compute $\varphi(t,\mathcal{P}_f)$. Then $\varphi(t,\mathcal{P}_f) = \varphi(t,\mathcal{P}_1)$ are undefined, i.e., the strategy cannot proceed. This can be solved by changing $(*)$ in Definition 4.15 as follows: $\varphi(t,\mathcal{P}) = u'.v$ if $\mathcal{P} = branch(\pi,u,\mathcal{P}_1,\ldots,\mathcal{P}_n)$, $root(\pi_j|_u) \ne root(t|_u) = f' \in \mathcal{F}$, for all j, $1 \le j \le n$; $\pi' = \pi[f'(\widetilde{\Omega})]_u$, and $u' \in O(\pi')$, $\epsilon < u' \le u$ is the minimal occurrence such that $\pi'|_{u'}$ is compatible with some redex scheme, $\mathcal{P}_g$ is a definitional tree for $g = root(\pi'|_{u'})$, and $\varphi(t|_{u'},\mathcal{P}_g) = v$.

This works well when considering CSs (it is equivalent to Definition 4.15). However, this does not ensure that φ is index reduction when considering general TRSs. This can be clarified by comparing the strategy with the standard Huet and Lévy procedure, as discussed in the following section.

5 Definitional trees and matching dags

To implement normalizing strategies without look-ahead, the *matching dags* (directed acyclic graphs) of Huet and Lévy can be used with any strongly sequential TRS. Simpler structures are the *index trees* of Strandh [11] which have been proved equivalent to matching dags by Durand [6]. An index tree is a finite state automaton which has, in addition to the usual transfer function, also a *failure function*. The set of final states is $L_\Omega(\mathcal{R})$. Non-final states are *index points*, pairs $\langle \pi, u\rangle$, where $\pi \in L_\Omega^{<}(\mathcal{R})$, and u is an index of π and both satisfy some special conditions (see [6]). The initial state is $\langle \Omega, \epsilon\rangle$. The *transfer function*, written $\delta(\mathsf{s}, f)$, yields a new state of the automaton, given a state s and a function symbol f: $\delta(\langle \pi, u\rangle, f) = \langle \pi[f(\widetilde{\Omega})]_u, v\rangle$ (or just $\delta(\langle \pi, u\rangle, f) = \pi[f(\widetilde{\Omega})]_u$ if $\pi[f(\widetilde{\Omega})]_u \in L_\Omega(\mathcal{R})$). The *failure function*, ϕ, is $\phi(\mathsf{s}) = \mathsf{s}'$ iff s' is an *immediate failure point* of s. Failure points are states of the automaton which are expected to deal with a failing partial matching, by resuming the matching of a subterm of the currently inspected term. In the most general definition of an index tree, some states may *not* be reachable from the initial state $\langle \Omega, \epsilon\rangle$ via transfer transitions (using δ) only. Thus, only the failure function can provide access to these nodes of the tree. Orthogonal TRSs which can be given such an index tree are called *bounded* TRSs. Durand proves that the class of bounded TRSs and strongly sequential TRSs coincide. The proof is given by showing that there is an immediate correspondence between the matching dag of Huet and Lévy and the index trees of Strandh.

Strandh defines the *forward-branching* index trees, for which all states of the index tree can be reached via the transfer function δ from the initial state.

We provide a simple connection between *index trees* and *definitional trees*: transitions $\delta(\langle \pi, u\rangle, f_1) = \langle \pi[f_1(\widetilde{\Omega})]_u, v_1\rangle, \ldots, \delta(\langle \pi, u\rangle, f_n) = \langle \pi[f_n(\widetilde{\Omega})]_u, v_n\rangle$ can be written as $branch(\pi, u, \mathcal{P}_1, \ldots, \mathcal{P}_n)$, where the pattern of each $\mathcal{P}_i$ is $\pi_i = \pi[f_i(\widetilde{\Omega})]_u$. Each initial transition $\delta(\langle \Omega, \epsilon\rangle, f) = \langle f(\widetilde{\Omega}), u\rangle$ can be seen as the starting point of the *pdt* for the function f. When considering forward-branching index trees, the correspondence is even closer. However, *pdt*s are not equivalent to the previous structures. For instance, consider the *bounded* TRS (from [6])

$\mathtt{f(g(x,a),a)} \to \mathtt{a} \qquad \mathtt{g(b,b)} \to \mathtt{a}$

$\mathtt{f(g(a,x),b)} \to \mathtt{a}$

and the *pdt*s in Figure 3.

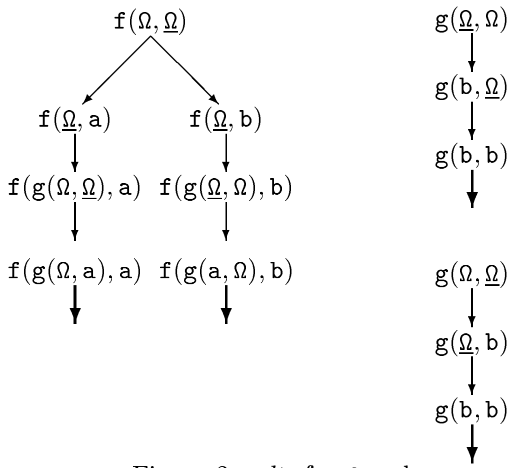

Figure 3: *pdt*s for $\mathtt{f}$ and $\mathtt{g}$

The patterns and inductive occurrences of these *pdt*s are taken by following the index tree for the TRS, as given in [6]. Of course, if we do not do this, we cannot ensure that the composition of the occurrences considered for the partial matchings (which are indices of the corresponding preredexes) is an index. This means that, even if we use `nodes`, that always selects (arbitrary) indices for inductive occurrences, we cannot ensure index reduction. But we have more involved situations. For instance, if we reduce $t = \mathtt{f}(\mathtt{g}(\Delta_1, \Delta_2), \mathtt{a})$, where Δ_1, Δ_2 are redexes, it is not difficult to see that φ (modified) reduces the redex Δ_1 which is *not* a needed redex. This is because, when we fail in matching $\mathtt{f}(\mathtt{g}(\Omega, \underline{\mathtt{a}}), \mathtt{a})$ (we underline the last considered occurrence, 1.2) against our term, we try to reduce $\mathtt{g}(\Delta_1, \Delta_2)$ and we must jump to a *pdt* for $\mathtt{g}$, in order to continue the matching. If we have the first *pdt* (recall that, in our strategy, only *one pdt* is available), then we choose to reduce Δ_1. However, Δ_1 is not needed in the whole context, since $\mathtt{f}(\mathtt{g}(\bullet, \Omega), \mathtt{a}) \to_\Omega \Omega$. With an index tree, the failure function selects the *other pdt* for $\mathtt{g}$, which properly continues the matching tasks by looking at the occurrence 2 of $\mathtt{g}(\Delta_1, \Delta_2)$. This is consistent with the situation before the jump.

The modification of φ would work if we consider forward branching TRSs (because they do not have such additional nodes) and we use *pdt*s having the same pattern and inductive occurrences as the corresponding forward-branching index tree. The previous TRS is *not* forward-branching. However, even with such modifications, we loose efficiency, because, having a very simple definition of the *failure* function (we always jump to the *root node* of a new *pdt*), we would read more than once some symbols. Therefore, it seems that there is no advantage in using *pdt*s with general TRSs.

References

[1] S. Antoy. Definitional Trees. In *Proc. of ALP'92*, LNCS 632:143-157, Springer-Verlag, 1992.

[2] S. Antoy. Needed Narrowing in Prolog (Extended Abstract). In *Proc. of PLILP'96*, LNCS 1140:473-474, Springer-Verlag, 1996.

[3] S. Antoy, R. Echahed and M. Hanus. A needed narrowing strategy. In *Proc of POPL'94*, pages 268-279. ACM Press, 1994.

[4] S. Antoy, R. Echahed, and M. Hanus. Parallel evaluation strategies for functional logic languages. In *Proc. of ICLP'97*, pages 138–152. The MIT Press, 1997.

[5] N. Dershowitz and J.P. Jouannaud. Rewrite Systems. In *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 243-320. Elsevier and The MIT Press, 1990.

[6] I. Durand. Bounded, Strongly Sequential and Forward Branching Term Rewriting Systems. Journal of Symbolic Computation, 18:319-352, 1994.

[7] G. Huet and J.J. Lévy. Computations in orthogonal term rewriting systems, I and II. In *Computational logic: essays in honour of J. Alan Robinson*, pages 396–443. The MIT Press, 1991.

[8] J.W. Klop. Term Rewriting Systems. In *Handbook of Logic in Computer Science*, volume 2, pages 1-116. Oxford University Press, 1992.

[9] J.W. Klop and A. Middeldorp. Sequentiality in Orthogonal Term Rewriting Systems. *Journal of Symbolic Computation* 12:161-195, 1991.

[10] M.J. O'Donnell. Equational Logic as a Programming Language. The MIT Press, 1985.

[11] R. Strandh. Classes of Equational Programs that Compile into Efficient Machine Code. In *Proc. of RTA'89*, LNCS 355:449-461, Springer-Verlag, 1989.