Theoretical Computer Science **165**:75-95, 1996.

A Sequential Reduction Strategy

Sergio Antoy
Portland State University, U.S.A.
antoy@cs.pdx.edu

Aart Middeldorp
University of Tsukuba, Japan
ami@is.tsukuba.ac.jp

ABSTRACT

Kennaway proved the remarkable result that every (almost) orthogonal term rewriting system admits a computable sequential normalizing reduction strategy. In this paper we present a computable sequential reduction strategy similar in scope, but simpler and more general. Our strategy can be thought of as an outermost-fair-like strategy that is allowed to be unfair to some redex of a term when contracting the redex is useless for the normalization of the term. Unlike the strategy of Kennaway, our strategy does not rely on syntactic restrictions that imply confluence. On the contrary, it can easily be applied to any term rewriting system, and we show that the class of term rewriting systems for which our strategy is normalizing properly includes all (almost) orthogonal systems. Our strategy is more versatile; in case of (almost) orthogonal term rewriting systems, it can be used to detect certain cases of non-termination. Our normalization proof is more accessible than Kennaway's. We also show that our sequential strategy sometimes succeeds where the parallel-outermost strategy fails.

1. Introduction

This paper is concerned with the fundamental question how to compute normal forms with respect to a given term rewriting system (TRS for short). Consider the TRS

$$\begin{array}{rcl} 0+x & \to & x \\ S(x)+y & \to & S(x+y) \\ 0\times x & \to & 0 \\ S(x)\times y & \to & x\times y+y \end{array}$$

specifying addition and multiplication on natural numbers. Suppose we want to normalize the term $(0+(0\times S(0)))+(0\times 0)$. This term contains three instances of left-hand sides of rewrite rules (so-called *redexes*):

$$\overbrace{(0+\underbrace{(0\times S(0))}_{(2)})}^{(1)}+\overbrace{(0\times 0)}^{(3)}$$

Redexes (1) and (3) are not contained in larger redexes. We call them *outermost* redexes. Redexes (2) and (3) are *innermost* redexes; they do not contain smaller redexes. If we select the

This work has been supported in part by the National Science Foundation under grant CCR-9406751.

leftmost-outermost redex (1) we obtain the term $(0 \times S(0)) + (0 \times 0)$ which contains two redexes. Selecting again the leftmost-outermost redex we obtain the term $0 + (0 \times 0)$. After two further contractions of the leftmost-outermost redex we arrive at the normal form 0. Instead of selecting a single redex in each step, we can also contract redexes in parallel. For instance, redexes (1) and (3) do not interfere, i.e., after contraction of redex (1), redex (3) is not affected, and vice-versa. Also redexes (2) and (3) can be contracted in parallel. If we contract all outermost redexes in parallel we also obtain the normal form 0: $(0+(0\times S(0)))+(0\times 0) \to^* (0\times S(0))+0 \to 0+0 \to 0$.

Does it matter which redexes we select for contraction? In the above example the answer is no since the TRS doesn't admit infinite reduction sequences. Hence, no matter which redexes we select for contraction, we are guaranteed to find a normal form. In general, however, terms may have a normal form but also admit infinite reduction sequences. Hence it is important to adopt a good reduction strategy. Given a TRS and a term, a reduction strategy tells you which redex(es) to contract. The desirable property of reduction strategies is *normalization*: repeated contraction of the redex(es) selected by the reduction strategy leads to a normal form, if the term under consideration has a normal form. For the important subclass of (almost) orthogonal TRSs several positive results are known. An *orthogonal* TRS is left-linear and non-overlapping. If the non-overlapping restriction is relaxed by allowing trivial overlays then we speak of *almost* orthogonal TRSs. A typical example of an almost orthogonal TRS that is not orthogonal is the two-rule system $\{\top \vee x \to \top,\ x \vee \top \to \top\}$. O'Donnell [8] showed that the *parallel-outermost* strategy—which evaluates all outermost redexes in parallel—is normalizing for all almost orthogonal TRSs.

The question whether there exists a computable normalizing *sequential* reduction strategy for all (almost) orthogonal TRSs has received quite a bit of attention. A sequential strategy must choose a single redex for contraction. The main results are summarized below.

(1) Contrary to the situation for λ-calculus, there are orthogonal TRSs for which the *leftmost-outermost* strategy is not normalizing. Consider for instance the orthogonal TRS (from [4])

$$\begin{array}{rcl} a & \to & b \\ c & \to & c \\ f(x,b) & \to & d \end{array}$$

and the term $t = f(c,a)$. The leftmost-outermost strategy will select redex c in t and hence produce the infinite reduction sequence $t \to t \to t \to \cdots$. Nevertheless, parallel-outermost succeeds in normalizing t: $t \to^* f(c,b) \to d$. O'Donnell [8] showed that the leftmost-outermost strategy is normalizing for every *left-normal* orthogonal TRS.[1] Left-normality means that variables do not precede function symbols in the left-hand sides of the rewrite rules. A typical example of a left-normal orthogonal TRS is *combinatory logic*.

(2) Huet and Lévy, in a landmark paper [4], formulated the strong sequentiality criterion and showed that every strongly sequential orthogonal TRS admits a computable normalizing sequential reduction strategy. Every left-normal orthogonal TRS is strongly sequential, but there are many strongly sequential orthogonal TRSs that are not left-normal, one example

[1] Actually, O'Donnell showed that the leftmost-outermost strategy is normalizing for all left-normal *almost* orthogonal TRSs. However, it is easy to see that if two rewrite rules in a left-normal almost orthogonal TRS overlap, then one of the rules is an instance of the other. Hence for every left-normal almost orthogonal TRS there exists a left-normal orthogonal TRS with the same rewrite relation. Thus nothing is really added by allowing trivial overlays, rather, it would unnecessarily confuse Figure 1 below.

being the TRS of the preceding paragraph. The strategy of Huet and Lévy is based on the fact that for orthogonal TRSs

- every term not in normal form contains a needed redex, and
- repeated contraction of needed redexes leads to the normal form, if it exists.

Here a *needed* redex is a redex with the property that in every reduction sequence (from the term in which the redex occurs) to normal form one of its descendants is contracted. The second statement above can be strengthened to the *hypernormalization* of needed reduction: there are no infinite reduction sequences starting from a term that has a normal form in which infinitely many needed redexes are contracted. Since needed redexes are in general not computable, this does not give a normalizing sequential strategy for all orthogonal TRSs. Huet and Lévy showed that for strongly sequential orthogonal TRSs at least one of the needed redexes in a term not in normal form can be (efficiently) computed. The orthogonality requirement cannot be weakened to almost orthogonality, simply because needed redexes may not exist for almost orthogonal TRSs, as observed by Sekar and Ramakrishnan [13].

(3) Kennaway [5] showed the remarkable fact that *every* almost orthogonal TRS admits a computable normalizing sequential reduction strategy. Actually, Kennaway doesn't restrict himself to TRSs but obtains his result in the very general setting of *combinatory reduction systems* of Klop [6]. The result of Kennaway is remarkable since it covers TRSs like combinatory logic extended with *Berry's* TRS

$$\begin{array}{lcl} f(a,b,x) & \to & c \\ f(b,x,a) & \to & c \\ f(x,a,b) & \to & c \end{array}$$

This TRS seems to require parallel evaluation of the arguments t_1, t_2, t_3 when faced with a term $f(t_1,t_2,t_3)$ since in this TRS it is undecidable whether a term reduces to a or b. Berry's TRS is an example of a TRS that is not strongly sequential.

(4) Toyama [14] showed that the sequential strategy of Huet and Lévy is normalizing for the larger class of strongly sequential left-linear *root-balanced joinable* TRSs. The root-balanced joinability requirement is less restrictive than the trivial overlays allowed by Kennaway, but because of the strong sequentiality requirement Toyama's result does not cover the result of Kennaway. For instance, the almost orthogonal TRS $\{\top \vee x \to \top,\ x \vee \top \to \top\}$ is not strongly sequential.

(5) Oyamaguchi [10] extended the work of Huet and Lévy in a different direction. Strong sequentiality is a property based on the left-hand sides of the rewrite rules. Oyamaguchi showed that by incorporating information in the right-hand sides of the rewrite rules, a less restrictive notion of sequentiality can be obtained, while retaining the good properties of strong sequentiality. The resulting class of *NV-sequential* orthogonal TRSs properly contains all strongly sequential orthogonal TRSs. Although Kennaway's result applies to all systems covered by Oyamaguchi's result, the result of Oyamaguchi is of interest because his strategy can be computed in polynomial time, which is not the case for the strategy of Kennaway.[2]

[2] Moreover, Oyamaguchi obtained his results in 1987, two years before Kennaway's paper was published.

Figure 1 shows the relationships between the above-mentioned results. The area enclosed in the dotted rectangle denotes the class of TRSs for which the parallel-outermost strategy is normalizing. Areas enclosed in numbered solid rectangles correspond to the sequential strategies described above. Observe that parallel-outermost encompasses all known sequential reduction strategies. (The fact that parallel-outermost is normalizing for every strongly sequential left-linear root-balanced joinable TRS follows from the work of Toyama [14].) All systems in area (3) are confluent. Systems in area (4) are not necessarily confluent, but they do have unique normal forms. (Slightly stronger, every term that has a normal form is confluent.)

Recently, Comon [2] showed that strong and NV-sequentiality are decidable properties of left-linear, possible overlapping, TRSs, and van Raamsdonk [12] showed that outermost-fair rewriting is normalizing for weakly orthogonal higher-order rewriting systems.

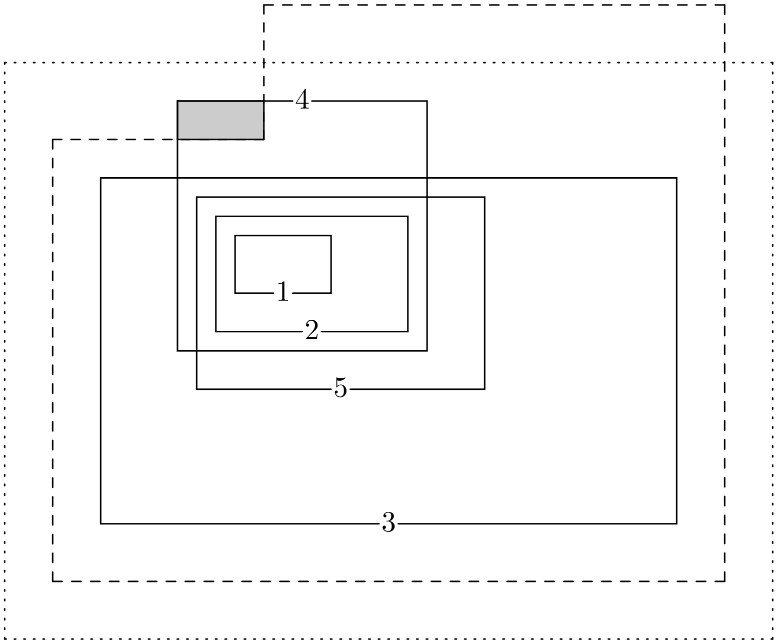

Figure 1.

The starting point of the present paper is Kennaway's result. The strategy of Kennaway is complicated and relies on the confluence property. We define a simpler and intuitive sequential strategy for *every* TRS and we show that the class of TRSs for which our strategy is normalizing properly includes the class of TRSs for which Kennaway's strategy is defined. This class is indicated in Figure 1 as the area between the dashed lines. At present it is unclear whether the shaded area is inhabited. All other areas are inhabited. In particular, we will see that there exist TRSs which cannot be normalized by means of the parallel-outermost strategy but for which our sequential strategy succeeds.

Our strategy is versatile. We will show that in case of almost orthogonal TRSs, it can be used as a sufficient condition for the property of having no normal form. This means that for certain terms our strategy will signal that it is useless to contract any redex as the term under consideration has no normal form.

The remainder of the paper is organized as follows. In the next section we formally define the notion of sequential reduction strategy. In Section 3 we describe an abstract game of Klop which is closely related to our strategy. It is well-known that in general it is undecidable whether a term cycles. In Section 4 we give a sufficient and decidable criterion for cyclicity. Using this result, our computable sequential strategy is defined in Section 5. In Section 6 we present our main theorem: if our strategy fails to normalize a term then there exists an infinite so-called *outermost-fair* reduction sequence starting from that term. This result does not rely on orthogonality, quite to the contrary, it holds for every TRS. Our proof is non-trivial but considerably simpler than the normalization proof of Kennaway. Since for almost orthogonal TRSs it is known (O'Donnell [8]) that terms that admit infinite outermost-fair reduction sequences do not have a normal form, an immediate consequence is the normalization of our strategy for all almost orthogonal TRSs. In Section 7 we explain how, in case of almost orthogonal systems, our strategy can be used to detect certain cases of non-termination. Furthermore, we exhibit a non-left-linear TRS which can be normalized by our sequential strategy, but for which parallel-outermost fails. We also mention some directions for further research. In particular, we address the question how to enhance our strategy in order to enlarge the class of TRSs for which it is normalizing.

2. Preliminaries

A signature is a set $\mathcal{F}$ of function symbols. Associated with every $f \in \mathcal{F}$ is a natural number denoting its arity. Function symbols of arity 0 are called constants. The set $\mathcal{T}(\mathcal{F},\mathcal{V})$ of terms built from a signature $\mathcal{F}$ and a countably infinite set of variables $\mathcal{V}$ is the smallest set containing $\mathcal{V}$ such that $f(t_1,\dots,t_n) \in \mathcal{T}(\mathcal{F},\mathcal{V})$ whenever $f \in \mathcal{F}$ has arity n and $t_1,\dots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{V})$. We write c instead of $c()$ whenever c is a constant. Let $\Box$ be a fresh constant symbol. A context C is a term in $\mathcal{C}(\mathcal{F},\mathcal{V})$ containing precisely one hole. The designation term is restricted to members of $\mathcal{T}(\mathcal{F},\mathcal{V})$. If C is a context and t a term then $C[t]$ denotes the result of replacing the hole in C by t. We say that t is a subterm of $C[t]$. A substitution is a map σ from $\mathcal{V}$ to $\mathcal{T}(\mathcal{F},\mathcal{V})$ with the property that the set $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ is finite. If σ is a substitution and t a term then $t\sigma$ denotes the result of applying σ to t. We call $t\sigma$ an instance of t.

A rewrite rule $l \to r$ is a pair of terms such that the left-hand side l is not a variable and variables which occur in the right-hand side r occur also in l. A TRS is a pair $(\mathcal{F},\mathcal{R})$ consisting of a signature $\mathcal{F}$ and a set $\mathcal{R}$ of rewrite rules between terms in $\mathcal{T}(\mathcal{F},\mathcal{V})$. We often present a TRS as a set of rewrite rules, without making explicit its signature, assuming that the signature consists of the function symbols occurring in the rewrite rules. A TRS $\mathcal{R}$ defines a rewrite relation $\to_{\mathcal{R}}$ on terms as follows: $s \to_{\mathcal{R}} t$ if there exists a rewrite rule $l \to r$ in $\mathcal{R}$, a substitution σ, and a context C such that $s = C[l\sigma]$ and $t = C[r\sigma]$. The subterm $l\sigma$ of s is called a redex and we say that s rewrites to t by contracting the redex $l\sigma$. The subterm $r\sigma$ of t is called a contractum of the redex $l\sigma$. We call $s \to_{\mathcal{R}} t$ a rewrite step. We usually omit the subscript $\mathcal{R}$. A term without redexes is called a normal form. The set of all normal forms of $\mathcal{R}$ is denoted by $\mathit{NF}(\mathcal{R})$. We say that a term t has a normal form if there exists a reduction sequence starting from t that ends in a normal form. A TRS is confluent if for all terms t_1, t_2, t_3 with $t_1 \to^* t_2$ and $t_1 \to^* t_3$ there exists a term t_4 such that $t_2 \to^* t_4$ and $t_3 \to^* t_4$. In a confluent TRS every term has at most one normal form.

A position is a sequence of natural numbers identifying a subterm in a term. The set $\mathcal{P}os(t)$ of positions in a term t is inductively defined as follows: $\mathcal{P}os(t) = \{\varepsilon\}$ if t is a variable and $\mathcal{P}os(t) = \{\varepsilon\} \cup \{i{\cdot}p \mid 1 \leqslant i \leqslant n \text{ and } p \in \mathcal{P}os(t_i)\}$ if $t = f(t_1,\dots,t_n)$. We say that a position p is above a position q if there exists a position r such that $p \cdot r = q$. If p is above q we also say that q is below p and we write $p \leqslant q$. We write $p < q$ if $p \leqslant q$ and $p \neq q$. Positions p, q are disjoint, denoted by $p \parallel q$, if neither $p \leqslant q$ nor $q \leqslant p$. If $p \in \mathcal{P}os(t)$ then $t_{|p}$ denotes the subterm of t at position t and $t[s]_p$ denotes the term that is obtained from t by replacing the subterm at position p by the term s. The size $|t|$ of a term t is the number of symbols occurring in it.

A TRS $\mathcal{R}$ is left-linear if the left-hand side l of every rewrite rule $l \to r \in \mathcal{R}$ doesn't contain multiple occurrences of the same variable. Let $l_1 \to r_1$ and $l_2 \to r_2$ be renamed versions of rewrite rules of a TRS $\mathcal{R}$ such that they have no variables in common. Suppose $l_{1|p}$, for some $p \in \mathcal{P}os(l_1)$ such that $l_{1|p}$ is not a variable, and l_2 are unifiable with most general unifier σ. The pair of terms $\langle l_1[r_2]_p\sigma, r_1\sigma\rangle$ is called a critical pair of $\mathcal{R}$. If $l_1 \to r_1$ and $l_2 \to r_2$ are renamed versions of the same rewrite rule, we do not consider the case $p = \varepsilon$. A critical pair $\langle l_1[r_2]_p\sigma, r_1\sigma\rangle$ with $p = \varepsilon$ is an overlay. A critical pair $\langle s, t\rangle$ is trivial if $s = t$. A left-linear TRS without critical pairs is called orthogonal. An almost orthogonal TRS is a left-linear TRS with the property that all its critical pairs are trivial overlays. Almost orthogonal TRSs are called weakly orthogonal in Kennaway [5], but nowadays weak orthogonality is used to denote the larger class of left-linear TRSs with only trivial critical pairs. A typical example of a weakly orthogonal TRS that is not (almost) orthogonal is the two-rule system $\{s(p(x)) \to x,\ p(s(x)) \to x\}$. Almost all results obtained for orthogonal TRSs (Klop [7] contains a good overview) carry over to almost orthogonal TRSs, with literally the same proofs, the notable exception being the theory of Huet and Lévy [4] on needed reductions. Weakly orthogonal TRSs are more complicated. Much more information on term rewriting can be found in Dershowitz and Jouannaud [3] and Klop [7]. The latter contains an introduction to the study of reduction strategies.

What is a sequential reduction strategy? In the literature one often finds the following definition:

> A sequential reduction strategy is a mapping $\mathcal{S}$ that assigns to every term t not in normal form a redex position in t.

This definition serves if the contractum of a redex is unique, which is the case for any TRS that lacks non-trivial overlays, in particular for all weakly orthogonal TRSs. However, if the TRS under consideration does have non-trivial overlays, we are faced with the problem that there are redexes with more than one contractum. Hence we must also supply the rewrite rule according to which the redex has to be contracted. Given a redex and a rule, the contractum is again uniquely determined. How to choose the rewrite rule? There are two possibilities:

- The rewrite rule depends only on the redex. This means that we are given a mapping that assigns to every redex a rewrite rule of whose left-hand side the redex is an instance.
- The rewrite rule depends on the redex and the surrounding context.

Clearly the latter option is more general. For instance, a (confluent) TRS like

$$\begin{array}{rcl} a & \to & b \\ b & \to & a \\ a & \to & c \\ c & \to & a \\ f(b,c) & \to & d \end{array}$$

cannot be normalized if we opt for the former. For example, the term $f(a,a)$ can only be normalized if we contract the first occurrence of redex a by the rewrite rule $a \to b$ and the second occurrence of the same redex by the rewrite rule $a \to c$. (This example also shows that both options lead to *unfairness* in the sense of Porat and Francez [11].) The simplicity of the former option however makes it possible to reason effectively about the strategy we will present later. Therefore we arrive at the following formal definition.

Definition 2.1. Let $\mathcal{R}$ be an arbitrary TRS. A *sequential reduction strategy* for $\mathcal{R}$ consists of the following two components:

- a mapping *rule* that assigns to every redex of $\mathcal{R}$ a rewrite rule of whose left-hand side the redex is an instance, and
- a mapping $\mathcal{S}$ that assigns to every term t not in normal form one of its redex positions.

We require that both *rule* and $\mathcal{S}$ are computable. The mapping *rule* induces a mapping *contract* from redexes to instances of right-hand sides of rewrite rules of $\mathcal{R}$ as follows: if $\mathit{rule}(t) = l \to r$ then $\mathit{contract}(t) = r\sigma$ where σ is any substitution satisfying $t = l\sigma$. If t is not in normal form then $t\mathcal{S}$ denotes the term $t[\mathit{contract}(t_{|\mathcal{S}(t)})]_{\mathcal{S}(t)}$, i.e. the result of applying the strategy to t. We write $t \to_{\mathcal{S}} t'$ if and only if $t' = t\mathcal{S}$.

In the remainder of this paper we identify a sequential reduction strategy with its mapping $\mathcal{S}$, that is, we assume that the mapping *rule* is given but our main results do not depend on its definition. This implies that we cannot prove that our sequential reduction strategy is capable of normalizing the rather simple TRS $\mathcal{R} = \{a \to a,\, a \to b\}$ since this depends on *rule*: a sequential reduction strategy normalizes $\mathcal{R}$ if and only if $\mathit{rule}(a) = a \to b$. For TRSs that do not admit non-trivial overlays, in particular for almost and weakly orthogonal TRSs, the identification of a sequential reduction strategy with its mapping $\mathcal{S}$ entails no loss of generality since the contractum of a redex is independent of the mapping *rule*.

It should be stressed that a sequential reduction strategy has no memory. This means that the decision concerning which redex to contract must be solely based on the term at hand and the given TRS, but the reduction sequence leading from the starting term to the present term may not be used. In particular, parallel-outermost cannot be simulated by a sequential reduction strategy since after contracting one outermost redex there is no way of telling which redexes in the resulting term coincide with the remaining outermost redexes in the starting term.

Definition 2.2. A sequential reduction strategy $\mathcal{S}$ for a TRS $\mathcal{R}$ is *normalizing* if for every term t that has a normal form there exists a reduction sequence $t = t_1 \to_{\mathcal{S}} t_2 \to_{\mathcal{S}} \cdots \to_{\mathcal{S}} t_n$ (with $n \geqslant 1$) ending in a normal form t_n.

Throughout the following we assume that we are dealing with TRSs that satisfy the following two properties:

- it is decidable whether a term is a redex (and hence it is also decidable whether a term is a normal form),
- in every infinite reduction sequence $t_1 \to t_2 \to \cdots$ in which the size of all terms is uniformly bounded, only finitely many different function symbols and variables occur (and hence the reduction sequence contains a repetition).

These properties are in particular true for all finite TRSs.

We conclude this preliminary section by introducing some notation for manipulating reduction sequences. Let $\mathcal{D}$: $t_1 \to t_2 \to t_3 \to \cdots$ be an arbitrary infinite reduction sequence. We write $\mathcal{D}[i,j]$ $(1 \leqslant i < j)$ to denote the finite portion $t_i \to t_{i+1} \to \cdots \to t_j$ and $\mathcal{D}[i]$ $(i \geqslant 1)$ to denote the term t_i. The concatenation $t_1 \to^+ t_2 \to^+ t_3$ of two finite reduction sequences $\mathcal{D}_1$: $t_1 \to^+ t_2$ and $\mathcal{D}_2$: $t_2 \to^+ t_3$ is denoted by $\mathcal{D}_1;\mathcal{D}_2$. If $\mathcal{D}$: $t \to^+ t$ is a cycle then $\mathcal{D}^\infty$ denotes the infinite reduction sequence $\mathcal{D};\mathcal{D};\mathcal{D};\ldots$. Let $\mathcal{D}$ be an arbitrary (finite or infinite) reduction sequence.

If C is a context then $C[\mathcal{D}]$ denotes the reduction sequence obtained from $\mathcal{D}$ by replacing every term t in $\mathcal{D}$ with $C[t]$. Finally, if p is a position in $\mathcal{D}[1]$ such that no redex at a position $q < p$ is contracted in $\mathcal{D}$, then $\mathcal{D}_{|p}$ denotes the reduction sequence extracted from $\mathcal{D}$ by replacing every term t in $\mathcal{D}$ with $t_{|p}$. In this case the resulting sequence may contain fewer rewrite steps.

3. Klop's Game

The result of Kennaway [5] covers the TRS combinatory logic extended with the parallel-or rules $\{\top \vee x \to \top,\ x \vee \top \to \top\}$, which is rather surprising since the term $t_1 \vee t_2$ seems to require parallel evaluation of its arguments t_1 and t_2. In Kennaway [5] (see also Klop [7]) the following abstract game is described which captures the essence of the difficulty of evaluating a term like $t_1 \vee t_2$ with a sequential strategy.

Suppose we are given two total functions f and g from $\mathbb{N}_+$, the set of positive integers, to $\mathbb{N}$. The objects of the game are pairs (x, y) of natural numbers. The search space is defined as a relation between these pairs:

$$\begin{array}{lcl} (x,y) & \to & (f(x),y), \\ (x,y) & \to & (x,g(y)), \\ (x,0) & \to & (0,0), \\ (0,y) & \to & (0,0) \end{array}$$

for all $x, y > 0$. The goal of the game is to reach $(0,0)$ from a given pair of natural numbers by adopting a sequential strategy. Sequential means here that, given a pair of positive integers, we must choose between applying function f to the first argument and applying function g to the second argument. Moreover, the choice can only be based on the pair of positive integers at hand (and the functions f and g of course). In particular, the strategy of alternatively applying f and g until one of the numbers become zero—which is guaranteed to produce $(0,0)$ if this is at all possible—is not allowed. Formally, a *sequential strategy* is a function $\mathcal{S}$ from $\mathbb{N}_+ \times \mathbb{N}_+$ to the set $\{L, R\}$. Every sequential strategy $\mathcal{S}$ defines a subset of the search space as follows:

$$\begin{array}{lcll} (x,y) & \to_{\mathcal{S}} & (f(x),y) & \text{if } \mathcal{S}(x,y) = L, \\ (x,y) & \to_{\mathcal{S}} & (x,g(y)) & \text{if } \mathcal{S}(x,y) = R, \\ (x,0) & \to_{\mathcal{S}} & (0,0), & \\ (0,y) & \to_{\mathcal{S}} & (0,0) & \end{array}$$

for all $x, y > 0$. We say that a sequential strategy $\mathcal{S}$ is *good* if $(x,y) \to_{\mathcal{S}}^* (0,0)$ whenever $(x,y) \to^* (0,0)$, for all natural numbers x and y.

In Kennaway [5] it is shown that there exists a good strategy for this game.[3] The sequential reduction strategy (for almost orthogonal TRSs) defined in [5] however is not related to the solution of this abstract game. We present a slightly different solution. Our solution is very close to the sequential reduction strategy that we define in Section 5.

We define a strategy $\mathcal{S}$ by distinguishing two cases. If $x \leqslant y$ then we compute the sequence $x,\ f(x),\ f^2(x),\ \ldots$ until we reach an $f^n(x)$ with $n \geqslant 1$ such that

- $f^n(x) = 0$, or

[3] The solution described in [5] is attributed to M. van Leeuwen and H. Mulder.

- $f^n(x) > x$, or
- $f^n(x) = f^m(x)$ for some $0 \leqslant m < n$.

Since there are only finitely many different natural numbers less than x, eventually one of these alternatives is satisfied. We define $\mathcal{S}(x, y) = R$ if the last alternative holds with $m = 0$, and $\mathcal{S}(x, y) = L$ in all other cases. If $x > y$ then we compute the sequence y, $g(y)$, $g^2(y)$, $\ldots$ until we reach a $g^n(y)$ with $n \geqslant 1$ such that

- $g^n(y) = 0$, or
- $g^n(y) > y$, or
- $g^n(y) = g^m(y)$ for some $0 \leqslant m < n$,

and we define $\mathcal{S}(x, y) = L$ if the last alternative holds with $m = 0$, and $\mathcal{S}(x, y) = R$ in all other cases. Before showing that $\mathcal{S}$ is a good strategy, we illustrate the strategy by means of a simple example. Consider the functions

$$f(x) = \begin{cases} x+1 & \text{if } x \neq 7, \\ 6 & \text{if } x = 7 \end{cases}$$

and

$$g(x) = \begin{cases} 2x & \text{if } x < 5, \\ 0 & \text{if } x = 5, \\ x-1 & \text{if } x > 5. \end{cases}$$

Starting from the pair $(1, 1)$, the strategy $\mathcal{S}$ produces the following sequence:

$$\begin{aligned} (1,1) &\to_{\mathcal{S}} (2,1) \to_{\mathcal{S}} (2,2) \to_{\mathcal{S}} (3,2) \to_{\mathcal{S}} (3,4) \to_{\mathcal{S}} (4,4) \to_{\mathcal{S}} (5,4) \\ &\to_{\mathcal{S}} (5,8) \to_{\mathcal{S}} (6,8) \to_{\mathcal{S}} (7,8) \to_{\mathcal{S}} (7,7) \to_{\mathcal{S}} (7,6) \to_{\mathcal{S}} (7,5) \\ &\to_{\mathcal{S}} (7,0) \to_{\mathcal{S}} (0,0). \end{aligned}$$

Theorem 3.1. *The strategy $\mathcal{S}$ just defined is a good strategy.*

Proof. Suppose the contrary. There exist some functions $f, g\colon \mathbb{N}_+ \to \mathbb{N}$ and a pair $(x, y) \in \mathbb{N} \times \mathbb{N}$ such that the game terminates for (x, y), but the strategy $\mathcal{S}$ computes an infinite sequence from (x, y), rather than reaching $(0, 0)$. Thus, there exists a sequence of moves $(x, y) \to^* (0, 0)$ and without loss of generality we can assume that the last step in this sequence is $(0, \tilde{y}) \to (0, 0)$, for some positive integer $\tilde{y}$. Hence there exists an $n > 0$ such that $f^n(x) = 0$. Since $\mathcal{S}$ fails to reach $(0, 0)$ from (x, y), the sequence of moves computed by $\mathcal{S}$ must reach (with less than n L choices) a state (x', y') after which only R choices are made and for all $i \geqslant 0$, $g^i(y') > 0$. That is, the infinite $\to_{\mathcal{S}}$-sequence starting from (x, y) must have the following form:

$$(x,y) \to_{\mathcal{S}}^* (x', y') \to_{\mathcal{S}} (x', g(y')) \to_{\mathcal{S}} (x', g^2(y')) \to_{\mathcal{S}} (x', g^3(y')) \to_{\mathcal{S}} \cdots,$$

Suppose that for all $i \geqslant 0$ $g^i(y')$ is smaller than x'. This implies that the infinite sequence y', $g(y')$, $g^2(y')$, $\ldots$ contains a repetition, say $g^p(y') = g^q(y')$ with $0 \leqslant p < q$. Let $g^l(y')$ be a maximal element in the cycle $g^p(y')$, $g^{p+1}(y')$, $\ldots$, $g^q(y')$. Then, $\mathcal{S}(x', g^l(y')) = L$, since neither 0 nor a value greater than $g^l(y')$ is found in the cycle $g^p(y')$, $g^{p+1}(y')$, $\ldots$, $g^q(y')$. This contradicts that $\mathcal{S}$ always chooses R after reaching (x', y'). Hence, there exists an $i \geqslant 0$ such that $g^i(y') \geqslant x'$. From $\mathcal{S}(x', g^i(y')) = R$ we infer that the sequence x', $f(x')$, $f^2(x')$, $\ldots$ contains a repetition and no term in it is 0. This contradicts that $f^n(x) = 0$, since $x' = f^k(x)$ for some $k \geqslant 0$. We can only conclude that $\mathcal{S}$ is a good strategy. $\square$

This solution, which easily generalizes to games with n-tuples ($n > 2$) of natural numbers, epitomizes the problems of a sequential strategy. A sequential strategy has no past memory, it must look at a term and decide which of its redexes to contract—contracting all redexes in turn, which requires past memory, since some redexes may reduce to themselves, is against the rules. Sequential strategies are natural for strongly sequential (NV-sequential) TRSs, since within this class every term t not in normal has a computable redex that must eventually be contracted to compute the normal form of t.

However, sequential strategies are much less natural for non-strongly sequential TRSs and, in fact, they have been thought impossible for some time. The interest in a computable sequential normalizing strategy for arbitrary (almost) orthogonal TRSs lies in its nature and in its very existence. The strategy that we discuss next is not intended to be efficient, though it is conceptually simple enough to be practical. To ensure normalization, parallel strategies rely on past memory, whereas sequential strategies rely on look ahead. An implementation of our strategy could be given past memory, in particular, it could record which redexes have been discovered to cycle, without destroying its sequential essence. An implementation with this feature would in some cases be more time efficient than a parallel strategy at the possible expense of some space efficiency.

4. Cycle Detection

Let $\mathcal{S}$ be a (sequential) reduction strategy for a TRS $\mathcal{R}$. In general it is undecidable whether a term t cycles with respect to $\mathcal{S}$. However, it is decidable whether there exists a cycle $t \to_{\mathcal{S}}^{+} t$ in which all terms have size less than or equal to some positive integer n.

Definition 4.1. Let $\mathcal{S}$ be a sequential reduction strategy for a TRS $\mathcal{R}$. We say that a term t *$\mathcal{S}$-cycles within size* $n > 0$, denoted by *cyclic*$(t, n, \mathcal{S})$, if there exists a cycle

$$t = t_1 \to_{\mathcal{S}} t_2 \to_{\mathcal{S}} \cdots \to_{\mathcal{S}} t_m = t$$

with $m > 1$ such that $|t_i| \leqslant n$ for all $i \in \{1, \ldots, m\}$.

Lemma 4.2. *Let $\mathcal{S}$ be a sequential reduction strategy for a TRS $\mathcal{R}$. Let t be a term and suppose $n > 0$. It is decidable whether t $\mathcal{S}$-cycles within size n.*

Proof. Since there are no infinite reduction sequences consisting of pairwise different terms whose size does not exceed n, there exists a finite reduction sequence $t = t_1 \to_{\mathcal{S}} t_2 \to_{\mathcal{S}} \cdots \to_{\mathcal{S}} t_m$ with $m \geqslant 1$ such that $t_1, \ldots, t_{m-1}$ are pairwise different terms whose size does not exceed n, and one of the following alternatives holds:

- $|t_m| > n$,
- $|t_m| \leqslant n$ and $t_m \in NF(\mathcal{R})$, or
- $|t_m| \leqslant n$ and $t_m = t_k$ for some $k \in \{1, \ldots, m-1\}$.

The term t $\mathcal{S}$-cycles within size n if and only if the last alternative holds with $k = 1$. Hence we can decide whether t $\mathcal{S}$-cycles within size n by simply applying the strategy and check which of the above alternatives holds. □

For deciding whether a term $\mathcal{S}$-cycles within size n, we only need to know what the strategy $\mathcal{S}$ does to terms whose size does not exceed n. This observation will be used in the next section where we define our strategy by induction on the size of terms. The proof of Lemma 4.2 suggests the 'algorithm' of Figure 2 to compute the predicate *cyclic*.

$$\mathit{cyclic}(t,n,\mathcal{S}) = \begin{cases} \mathit{false} & \text{if } |t| > n \text{ or } t \in \mathit{NF}(\mathcal{R}), \\ \varphi(t,n,\mathcal{S},\varnothing,t') & \text{if } |t| \leqslant n \text{ and } t \to_{\mathcal{S}} t' \end{cases}$$

$$\varphi(t,n,\mathcal{S},T,t') = \begin{cases} \mathit{true} & \text{if } t' = t, \\ \mathit{false} & \text{if } |t'| > n, \text{ or } t' \in \mathit{NF}(\mathcal{R}), \text{ or } t' \in T, \\ \varphi(t,n,\mathcal{S},T \cup \{t'\},t'') & \text{if } |t'| \leqslant n, \text{ and } t' \notin T \cup \{t\}, \text{ and } t' \to_{\mathcal{S}} t''. \end{cases}$$

Figure 2.

5. The Strategy

In this section we define our sequential strategy which will be denoted by $\mathcal{S}_\omega$. The strategy $\mathcal{S}_\omega$ only needs to know whether a term t $\mathcal{S}_\omega$-cycles within its own size. From now on we abbreviate $\mathit{cyclic}(t,|t|,\mathcal{S}_\omega)$ with $\odot(t)$.

Definition 5.1. Let t be a reducible term. By induction on $|t|$ we define a redex position $\mathcal{S}_\omega(t) \in \mathcal{P}os(t)$. If $|t| = 1$ then t is a reducible constant and we define $\mathcal{S}_\omega(t) = \varepsilon$. Let $t = f(s_1,\ldots,s_n)$. If t is a redex then we define $\mathcal{S}_\omega(t) = \varepsilon$. Suppose t is not a redex. Let m be the number of reducible terms among $s_1,\ldots,s_n$. We have $1 \leqslant m \leqslant n$. The reducible arguments of t are ordered according to their size. If two reducible arguments have the same size, the one to the left is considered first. Formally, there exists an injective map π from $\{1,\ldots,m\}$ to $\{1,\ldots,n\}$ such that

- $s_{\pi(i)}$ is reducible for all $i \in \{1,\ldots,m\}$,
- $|s_{\pi(1)}| \leqslant |s_{\pi(2)}| \leqslant \cdots \leqslant |s_{\pi(m)}|$, and
- if $|s_{\pi(i)}| = |s_{\pi(j)}|$ with $1 \leqslant i < j \leqslant m$ then $\pi(i) < \pi(j)$.

According to the induction hypothesis, $\mathcal{S}_\omega(s)$ is already defined for all terms s with $|s| < |t|$. Hence we can determine the validity of $\odot(s_{\pi(i)})$ for all $i \in \{1,\ldots,m\}$. If there is an $i \in \{1,\ldots,m\}$ such that $\odot(s_{\pi(i)})$ does not hold then we take the smallest such i and define $\mathcal{S}_\omega(t) = \pi(i) \cdot \mathcal{S}_\omega(s_{\pi(i)})$. If there is no such i then we let $\mathcal{S}_\omega(t) = \pi(m) \cdot \mathcal{S}_\omega(s_{\pi(m)})$.

It is easy to see that $\mathcal{S}_\omega$ only selects outermost redexes. We find it convenient to introduce some terminology relating to the selection of the arguments $s_{\pi(1)},\ldots,s_{\pi(m)}$ of t in Definition 5.1. If $1 \leqslant i < j \leqslant m$ then we call $s_{\pi(i)}$ a *predecessor* of $s_{\pi(j)}$ and we call $s_{\pi(j)}$ a *successor* of $s_{\pi(i)}$. If $1 \leqslant i < m$ and $\odot(s_{\pi(i)})$ holds then we say that $s_{\pi(i)}$ is *useless* in t. So if $\mathcal{S}_\omega(t) \geqslant \pi(k)$ then all predecessors of $s_{\pi(k)}$ are useless in t. We conclude this section by illustrating the workings of our strategy $\mathcal{S}_\omega$ on a small example.

Example 5.2. Consider the (almost orthogonal) TRS

$$\mathcal{R} = \begin{cases} \top \vee x & \to & \top \\ x \vee \top & \to & \top \\ \bot \vee \bot & \to & \bot \\ \infty & \to & \infty \end{cases}$$

and the term $t_1 = (\bot\vee\infty)\vee(\infty\vee(\top\vee\bot))$. We have $\mathcal{S}_\omega(\bot\vee\infty) = 2$ and hence $\bot\vee\infty \to_{\mathcal{S}_\omega} \bot\vee\infty$. So the first argument of t_1 $\mathcal{S}_\omega$-cycles within its own size. Therefore $\mathcal{S}_\omega(t_1) = 2\cdot\mathcal{S}_\omega(\infty\vee(\top\vee\bot))$. Because $\odot(\infty)$ we have $\mathcal{S}_\omega(\infty\vee(\top\vee\bot)) = 2\cdot\mathcal{S}_\omega(\top\vee\bot)$. The term $\top\vee\bot$ is a redex, so $\mathcal{S}_\omega(\top\vee\bot) = \varepsilon$. We conclude that $\mathcal{S}_\omega(t_1) = 2{\cdot}2$ and consequently $t_1 \to_{\mathcal{S}_\omega} (\bot\vee\infty)\vee(\infty\vee\top) = t_2$. We already know that $\odot(\bot\vee\infty)$. Hence $\mathcal{S}_\omega(t_2) = 2\cdot\mathcal{S}_\omega(\infty\vee\top) = 2$ since $\infty\vee\top$ is a redex. Therefore $t_2 \to_{\mathcal{S}_\omega} (\bot\vee\infty)\vee\top = t_3$. The term t_3 is a redex and thus $t_3 \to_{\mathcal{S}_\omega} \top$. So our strategy needs three steps to normalize the term t_1.

6. Normalization

In this section we establish a relationship between our strategy $\mathcal{S}_\omega$ and so-called outermost-fair reductions. In O'Donnell [8] this concept—O'Donnell uses the terminology *eventually outermost*—is defined for almost orthogonal TRSs. The definition we give below applies to all TRSs. It is equivalent to the one in Klop [7], except that we only consider *infinite* reduction sequences.

Definition 6.1. An infinite reduction sequence $\mathcal{D}$ is called *outermost-fair* if there do not exist a position p and an index $n \geqslant 1$ such that for all $i \geqslant n$, $\mathcal{D}[i]_{|p}$ is an outermost redex in $\mathcal{D}[i]$ which is not contracted in the reduction step $\mathcal{D}[i, i+1]$. If $\mathcal{D}$ is not outermost-fair then every position p satisfying the above condition is said to be *unfairly treated* by $\mathcal{D}$. If a position is unfairly treated by $\mathcal{D}$ we also say that the corresponding outermost redex is unfairly treated. These notions carry over to a cycle $\mathcal{D}$ via the associated infinite reduction sequence $\mathcal{D}^\infty$.

Let us illustrate the concept of outermost-fairness by means of two examples. The infinite reduction sequence $\mathcal{D}_1$: $f(a) \to g(f(a)) \to g(g(f(a))) \to \cdots$ with respect to the TRS

$$\begin{array}{rcl} a & \to & b \\ f(x) & \to & g(f(x)) \end{array}$$

is outermost-fair since every term in $\mathcal{D}_1$ contains a single outermost redex which is immediately contracted. The infinite reduction sequence $\mathcal{D}_2$: $f(a,c) \to f(a,d) \to f(a,c) \to f(a,d) \to \cdots$ with respect to the TRS

$$\begin{array}{rcl} a & \to & b \\ c & \to & d \\ f(x,d) & \to & f(x,c) \end{array}$$

is also outermost-fair. Observe that redex a in $\mathcal{D}_2$ is only half of the time an outermost redex, even though it is never contracted.

Next we show that if $\mathcal{S}_\omega$ fails to normalize a term t then there exists an (infinite) outermost-fair reduction sequence starting from t. The proof proceeds in three steps. First we show that every outermost redex unfairly treated in an $\mathcal{S}_\omega$-reduction sequence is a subterm of an $\mathcal{S}_\omega$-cyclic subterm (Lemma 6.3). This result is used to show that if a term admits an $\mathcal{S}_\omega$-cycle then it has a cycle in which no position is unfairly treated (Lemma 6.6). Finally we transform a presupposed infinite $\mathcal{S}_\omega$-reduction sequence into an outermost-fair reduction sequence by simply inserting enough cycles (Theorem 6.7).

Definition 6.2. Let $\mathcal{D}$ be an infinite reduction sequence. We say that a property $\mathcal{P}$ of terms *eventually* holds for $\mathcal{D}$ if $\mathcal{P}$ holds for all but a finite number of terms in $\mathcal{D}$.

Observe that a position p is unfairly treated by an infinite $\mathcal{S}_\omega$-reduction sequence $\mathcal{D}$ if and only if the property "$t_{|p}$ is an outermost redex in t and $\mathcal{S}_\omega(t) \parallel p$" eventually holds for $\mathcal{D}$.

Lemma 6.3. *If a position p is unfairly treated in an $\mathcal{S}_\omega$-reduction sequence $\mathcal{D}$ then there exist a term t, an index $N \geqslant 1$, and a position $\varepsilon < q \leqslant p$ such that t is $\mathcal{S}_\omega$-cyclic and $\mathcal{D}[i]_{|q} = t$ for all $i \geqslant N$.*

Proof. Suppose $\mathcal{D}$: $t_1 \rightarrow_{\mathcal{S}_\omega} t_2 \rightarrow_{\mathcal{S}_\omega} \cdots$ is an infinite reduction sequence in which position p is unfairly treated. This implies the existence of an index M in $\mathcal{D}$ and a position p such that for all $i \geqslant M$ p is the position of an outermost redex of t_i which is never chosen by $\mathcal{S}_\omega$. Since $\mathcal{S}_\omega$ is an outermost strategy, it cannot be unfair to position ε, consequently $p \neq \varepsilon$. Thus, $\mathcal{S}_\omega$ treats p unfairly only by continuously choosing from M onwards some position disjoint from p, despite the fact that p is the position of an outermost redex. In $\mathcal{D}$, infinitely many choices of $\mathcal{S}_\omega$ must fall under some position q (possibly ε) above p, but only finitely many can fall under some other position (possibly p itself) below q. More precisely, there exist a position q and a positive integer k such that

- $q{\cdot}k \leqslant p$,
- $\mathcal{S}_\omega(t_i) > q$ for infinitely many values of $i \geqslant M$, and
- $\mathcal{S}_\omega(t_i) > q{\cdot}k$ for finitely many values of $i \geqslant M$.

Hence there exists an index N in $\mathcal{D}$, $N \geqslant M$, such that no choice of $\mathcal{S}_\omega$ falls under $q{\cdot}k$ from N onwards, i.e., there is no $i \geqslant N$ such that $\mathcal{S}_\omega(t_i) > q{\cdot}k$. The subterm of t_N at q is compound, since q is strictly above p. Let $t_{N|q} = f(s_1, \ldots, s_n)$. Since for all $i \geqslant M$, p is the position of an outermost redex of t_i which is never chosen by $\mathcal{S}_\omega$, no reduction ever takes place above p from M onwards. Thus, for all $i \geqslant N$, $t_{i|q}$ is of the form $f(s^i_1, \ldots, s^i_n)$. Since for all $i \geqslant N$, no reduction takes place below $q{\cdot}k$, $s^i_k = s_k$ for all $i \geqslant N$. We show that s_k is $\mathcal{S}_\omega$-cyclic. Since infinitely many reductions take place below position q, according to the pigeon-hole principle there exists a $j \in \{1, \ldots, n\}$ such that $\mathcal{S}_\omega(t_i) \geqslant q{\cdot}j$ for infinitely many values of $i \geqslant N$. Let I be the set of these values. For any $i \in I$, either s_k is a predecessor of s^i_j, or s_k is a successor of s^i_j. First suppose that s_k is a successor of s^i_j for all $i \in I$. This implies that the sequence

$$s^{i_1}_j \rightarrow_{\mathcal{S}_\omega} s^{i_2}_j \rightarrow_{\mathcal{S}_\omega} s^{i_3}_j \rightarrow_{\mathcal{S}_\omega} \cdots$$

where $I = \{i_1, i_2, i_3, \ldots\}$ with $i_1 < i_2 < i_3 < \cdots$, contains a cycle, for otherwise there would be an $l \in I$ such that $|s^l_j| > |s_k|$ and consequently s_k would be a predecessor of s^l_j. Let s^m_j (with $m \in I$) be a biggest term in such a cycle. Then, s^m_j cycles within its own size. Hence s^m_j is useless in $t_{m|q}$, which is impossible since $m \in I$. We conclude that there exists an $i \in I$ such that s_k is a predecessor of s^i_j. This implies that s_k is useless in $t_{i|q}$ since a redex in s^i_j is selected by $\mathcal{S}_\omega$. Therefore s_k is $\mathcal{S}_\omega$-cyclic, which proves the lemma. □

The following technical result states that if $\mathcal{S}_\omega$ selects a redex in subterm $t_{|p}$ of t, then instead of applying $\mathcal{S}_\omega$ to t we can also replace its subterm $t_{|p}$ by the result of applying $\mathcal{S}_\omega$ to $t_{|p}$, without changing the final outcome. This does not hold if the contractum of a redex depends on the term in which the redex occurs (cf. the remarks in Section 2).

Lemma 6.4. *If $\mathcal{S}_\omega(t) \geqslant p$ then $t\mathcal{S}_\omega = t[t_{|p}\mathcal{S}_\omega]_p$.*

Proof. We use induction on the position p. If $p = \varepsilon$ then the result trivially holds. Suppose

$p = i{\cdot}q$. In the following we write t_i instead of $t_{|i}$. From the definition of $\mathcal{S}_\omega$ we infer that $\mathcal{S}_\omega(t) = i \cdot \mathcal{S}_\omega(t_i)$. Hence

$$\begin{aligned} t\mathcal{S}_\omega &= t[\mathit{contract}(t_{|\mathcal{S}_\omega(t)})]_{\mathcal{S}_\omega(t)} \\ &= t[\mathit{contract}(t_{|i\cdot\mathcal{S}_\omega(t_i)})]_{i\cdot\mathcal{S}_\omega(t_i)} \\ &= t[t_i[\mathit{contract}(t_{|i\cdot\mathcal{S}_\omega(t_i)})]_{\mathcal{S}_\omega(t_i)}]_i \\ &= t[t_i[\mathit{contract}(t_{i|\mathcal{S}_\omega(t_i)})]_{\mathcal{S}_\omega(t_i)}]_i \\ &= t[t_i\mathcal{S}_\omega]_i. \end{aligned}$$

Since $\mathcal{S}_\omega(t_i) \geqslant q$, the induction hypothesis yields $t_i\mathcal{S}_\omega = t_i[(t_{i|q})\mathcal{S}_\omega]_q$ and therefore

$$\begin{aligned} t\mathcal{S}_\omega &= t[t_i[t_{i|q}\mathcal{S}_\omega]_q]_i \\ &= t[t_{i|q}\mathcal{S}_\omega]_{i\cdot q} \\ &= t[t_{|p}\mathcal{S}_\omega]_p. \end{aligned}$$

□

Lemma 6.5. *Let $\mathcal{D}$ be any $\mathcal{S}_\omega$-reduction sequence. If p is a position in $\mathcal{D}[1]$ such that no redex at a position $q < p$ is contracted in $\mathcal{D}$, then $\mathcal{D}_{|p}$ is an $\mathcal{S}_\omega$-reduction sequence.*

Proof. Let $t \to t\mathcal{S}_\omega$ be an arbitrary reduction step in $\mathcal{D}$. We have to show that $t_{|p} \to^*_{\mathcal{S}_\omega} t\mathcal{S}_{\omega|p}$. If $\mathcal{S}_\omega(t) \parallel p$ then $t\mathcal{S}_{\omega|p} = (t[\mathit{contract}(t_{|\mathcal{S}_\omega(t)})]_{\mathcal{S}_\omega(t)})_{|p} = t_{|p}$. If $\mathcal{S}_\omega(t) \geqslant p$ then $t\mathcal{S}_{\omega|p} = (t[t_{|p}\mathcal{S}_\omega]_p)_{|p} = t_{|p}\mathcal{S}_\omega$ by using Lemma 6.4, and hence $t_{|p} \to_{\mathcal{S}_\omega} t\mathcal{S}_{\omega|p}$. □

Lemma 6.6. *If a term t $\mathcal{S}_\omega$-cycles then there exists a cycle of t in which every position is fairly treated.*

Proof. From an $\mathcal{S}_\omega$-cycle of t we construct a new cycle of t in which every position of t is fairly treated. We prove the lemma by structural induction on t. Let $\mathcal{D}$ be an $\mathcal{S}_\omega$-cycle of t. If t is a constant then it is contracted in $\mathcal{D}$ and the only position of t, ε, is treated fairly. Suppose $t = f(s_1, \ldots, s_n)$ and let P be the set of all positions that are unfairly treated by $\mathcal{D}$. Observe that $P \subset \mathcal{P}os(t)$. Let $p \in P$. We show that there exists a cycle $\mathcal{D}_p$ of t in which position p is fairly treated. Since an outermost strategy cannot treat unfairly position ε, $p \neq \varepsilon$. Hence $p = k \cdot p'$ for some $k \in \{1, \ldots, n\}$. We show that s_k $\mathcal{S}_\omega$-cycles. If a redex in s_k is contracted in $\mathcal{D}$ then Lemma 6.5 gives us an $\mathcal{S}_\omega$-cycle of s_k, namely $\mathcal{D}_{|k}$: $s_k \to^+_{\mathcal{S}_\omega} s_k$. If no redex in s_k is contracted in $\mathcal{D}$, then p' is a position unfairly treated in $\mathcal{D}^\infty$, the $\mathcal{S}_\omega$-reduction sequence of t. In this situation, Lemma 6.3 proves the existence of an $\mathcal{S}_\omega$-cyclic term at or below position k. Thus, s_k is $\mathcal{S}_\omega$-cyclic also if no redex in s_k is contracted in $\mathcal{D}$. Since s_k $\mathcal{S}_\omega$-cycles and is a proper subterm of t, according to the induction hypothesis there exists a cycle of s_k, say $\mathcal{D}_{p'}$, in which position p' is fairly treated. By adding context, we construct a cycle of t in which position p is fairly treated. More precisely, let C be the context $t[\square]_k$ and define $\mathcal{D}_p = C[\mathcal{D}_{p'}]$. Now $\mathcal{D}_p$ is a cycle of t in which position p is fairly treated. The concatenation of $\mathcal{D}$ and $\mathcal{D}_p$ is a cycle of t that is fair to p. Thus, all that remains to be done is to extend $\mathcal{D}$ with a "corrective" cycle for each position unfairly treated. Suppose $P = \{p_1, \ldots, p_m\}$. The reduction sequence $\mathcal{D}; \mathcal{D}_{p_1}; \ldots; \mathcal{D}_{p_m}$ is a cycle of t in which every position is fairly treated. □

Our main theorem is easily derived from the preceding lemma.

Theorem 6.7. *If $\mathcal{S}_\omega$ fails to normalize some term then there exists an outermost-fair reduction sequence starting from that term.*

Proof. Suppose $\mathcal{D}$ is an infinite $\mathcal{S}_\omega$-reduction sequence. For every $i \geqslant 1$ we define a reduction sequence $\mathcal{D}_i$: $\mathcal{D}[i] \to^* \mathcal{D}[i]$ as follows. Let $P_i \subseteq \mathcal{P}os(\mathcal{D}[i])$ be the set of all positions p with the property that $\mathcal{D}[i]_{|p}$ $\mathcal{S}_\omega$-cycles. From the preceding lemma we learn that for every $p \in P_i$ there exists a cycle $\mathcal{D}_p$ of $\mathcal{D}[i]$ in which every position is fairly treated. We define $\mathcal{D}_i$ as the concatenation of all those cycles. (If $P_i = \varnothing$ then $\mathcal{D}_i$ is the empty reduction sequence from $\mathcal{D}[i]$ to $\mathcal{D}[i]$, otherwise $\mathcal{D}_i$ is a cycle of $\mathcal{D}[i]$.) Now it is easy to see that the sequence $\mathcal{D}_1; \mathcal{D}[1,2]; \mathcal{D}_2; \mathcal{D}[2,3]; \mathcal{D}_3; \mathcal{D}[3,4]; \ldots$ is outermost-fair. □

O'Donnell [8] obtained the following result.

Theorem 6.8. *Let $\mathcal{R}$ be an almost orthogonal TRS. If a term admits an outermost-fair reduction sequence then it does not have a normal form.* □

The normalization of the parallel-outermost strategy for almost orthogonal TRSs is an immediate consequence of Theorem 6.8. Actually from Theorem 6.8 it follows that parallel-outermost is *hypernormalizing*, i.e., we are able to find normal forms even if we perform (finitely many) arbitrary reductions between successive applications of the parallel-outermost strategy. This in turn implies the hypernormalization of the *full-substitution* or *Gross-Knuth* strategy for almost orthogonal TRSs (cf. O'Donnell [8]).

Combining Theorem 6.8 with Theorem 6.7 yields the normalization of $\mathcal{S}_\omega$ for almost orthogonal TRSs.

Corollary 6.9. *The strategy $\mathcal{S}_\omega$ is normalizing for almost orthogonal TRSs.* □

It should be noted that, unlike parallel-outermost, $\mathcal{S}_\omega$ is not hypernormalizing. An example is provided by the TRS

$$\begin{array}{rcl} a & \to & f(b) \\ f(b) & \to & a \\ c & \to & d \\ g(x,d) & \to & d \end{array}$$

The term $g(a,c)$ is normalized by $\mathcal{S}_\omega$: $g(a,c) \to g(f(b),c) \to g(f(b),d) \to d$. However, if we contract the leftmost-outermost redex $f(b)$ after the first $\mathcal{S}_\omega$-step $g(a,c) \to g(f(b),c)$, we obtain a cycle.

7. Concluding Remarks

The ingenious proof of Theorem 6.8 in O'Donnell [8] is not easily digested. In the appendix of Bergstra and Klop [1] O'Donnell's proof is presented in a more accessible setting and is also generalized to combinatory reduction systems. (Although only orthogonal systems are considered in [1], the presence of trivial overlays doesn't cause any problems.) We would like to remark that for orthogonal TRSs Theorem 6.8 is easily derived from the work of Huet and Lévy [4] on needed reduction: it is not difficult to show that in every outermost-fair reduction sequence infinitely many needed redexes are contracted. In Section 2 we already remarked that the restriction to orthogonal TRSs is essential in the work of Huet and Lévy. Sekar and Ramakrishnan [13] showed that by generalizing the concept of needed redex to *necessary set of redexes* the main results of [4] carry over to the almost orthogonal case. In particular, contraction of necessary sets of redexes is hypernormalizing.

It is interesting to note that (in the case of almost orthogonal TRSs) the normalization of both the parallel-outermost strategy and our sequential strategy $\mathcal{S}_\omega$ is based on Theorem 6.8. So it is worthwhile to try to extend Theorem 6.8 to a more general class of TRSs. Van Raamsdonk [12] recently obtained the following result.

Theorem 7.1. *Let $\mathcal{R}$ be a weakly orthogonal TRS. If a term admits an outermost-fair reduction sequence then it does not have a normal form.* □

Corollary 7.2. *The strategy $\mathcal{S}_\omega$ is normalizing for weakly orthogonal TRSs.* □

As a matter of fact, van Raamsdonk proves her result for the much larger class of (weakly orthogonal) Higher-Order Rewriting Systems of van Oostrom [9]. We believe that also the class of strongly sequential left-linear root-balanced joinable TRSs considered by Toyama [14] should fall within the scope of Theorem 6.8.

Whereas all infinite reduction sequences produced by the parallel-outermost strategy are outermost-fair, this is not the case for $\mathcal{S}_\omega$. The following example shows that $\mathcal{S}_\omega$ ignores an outermost redex when it 'believes' that there is no reason to contract it.

Example 7.3. Consider the TRS

$$\begin{array}{rcl} a & \to & b \\ c & \to & c \\ f(b,x) & \to & f(a,x) \end{array}$$

and the term $t = h(f(a,b), g(g(c)))$. We have $f(a,b) \to_{\mathcal{S}_\omega} f(b,b) \to_{\mathcal{S}_\omega} f(a,b)$, so the first argument of t $\mathcal{S}_\omega$-cycles within its own size. Hence the second argument of t is selected for reduction, which of course gives $t \to_{\mathcal{S}_\omega} t$. Thus $\mathcal{S}_\omega$ produces an infinite reduction sequence in which position 1·1 (corresponding to outermost redex a) is unfairly treated.

In the above example both arguments of the term under consideration $\mathcal{S}_\omega$-cycle within their own size. Since the TRS is (almost) orthogonal, we might as well stop the evaluation since the term cannot have a normal form. This is justified by our next result.

Theorem 7.4. *Let $\mathcal{R}$ be an almost orthogonal TRS and $t = f(s_1, \ldots, s_n)$ a term that is neither a redex nor a normal form. If for all $i \in \{1, \ldots, n\}$ s_i is a normal form or $\odot(s_i)$ then $\odot(t)$.*

Proof. Since all reducible arguments of t are $\mathcal{S}_\omega$-cyclic within their own size, t itself is $\mathcal{S}_\omega$-cyclic within its own size, unless after some contractions in its $\mathcal{S}_\omega$-cyclic arguments a redex is created, say $t \to^+_{\mathcal{S}_\omega} f(s'_1, \ldots, s'_n) = t'$ with t' a redex. For all $i \in \{1, \ldots, n\}$ we have $s_i \to^*_{\mathcal{S}_\omega} s'_i \to^*_{\mathcal{S}_\omega} s_i$, hence redex t' can be rewritten to t by internal (i.e. at positions different from ε) contractions. Because $\mathcal{R}$ is almost orthogonal, redexes cannot be destroyed by internal contractions. Therefore t must be a redex, contrary to the assumption. We conclude that t $\mathcal{S}_\omega$-cycles within its own size. □

According to Corollary 6.9, in the case of almost orthogonal TRSs, $\odot(t)$ implies that t has no normal form. Hence Theorem 7.4 shows how our strategy can be used to detect non-termination. If we are faced with a redex t, we can of course also check whether $\odot(t)$ holds in order to avoid unnecessary infinite computations. These considerations do not generalize to arbitrary TRSs, as shown in the next example.

Example 7.5. Consider the TRS

$$\begin{array}{rcl} a & \to & b \\ b & \to & a \\ c & \to & c \\ f(b,x) & \to & d \end{array}$$

and the term $t = f(a,c)$. Both arguments of t $\mathcal{S}_\omega$-cycle within their own size. Nevertheless t has a normal form which can be reached by means of the outermost reduction sequence

$$f(a,c) \to f(b,c) \to d.$$

Observe that $\mathcal{S}_\omega$ will select redex c in t and hence produce an infinite reduction sequence that is not outermost-fair. It should be noted that the above TRS is not root-balanced joinable (in the terminology of Toyama [14]).

One possibility to extend the class of systems for which our strategy is normalizing is that in the case all reducible arguments of a term that is not a redex $\mathcal{S}_\omega$-cycle within their own size, we may look for a combination of terms in these cycles that creates a redex upwards. For instance, in the above example we see that the combination b and c in the cycles $a \to_{\mathcal{S}_\omega} b \to_{\mathcal{S}_\omega} a$ and $c \to_{\mathcal{S}_\omega} c$ creates the redex $f(b,c)$ which contracts to normal form d. Even if such a combination does not exist we can still test whether it is possible to redefine the mapping *rule*—which assigns rewrite rules to redexes—in such a way that the present deadlock situation disappears, at least temporarily. These ideas should be investigated further.

In the final example of the paper we justify our earlier claim that our sequential strategy $\mathcal{S}_\omega$ sometimes succeeds where parallel-outermost fails.

Example 7.6. Consider the non-left-linear TRS

$$\mathcal{R} = \left\{ \begin{array}{rcl} f(x,x) & \to & a \\ g(x) & \to & f(x,g(x)) \\ c & \to & g(c) \end{array} \right.$$

of Klop [6]. This well-known TRS shows that non-left-linearity destroys confluence, even in the absence of critical pairs. The following reduction sequence shows that our strategy $\mathcal{S}_\omega$ normalizes the constant c:

$$c \to_{\mathcal{S}_\omega} g(c) \to_{\mathcal{S}_\omega} f(c,g(c)) \to_{\mathcal{S}_\omega} f(g(c),g(c)) \to_{\mathcal{S}_\omega} a,$$

whereas the parallel-outermost strategy produces an infinite outermost-fair sequence:

$$c \to^* g(c) \to^* f(c,g(c)) \to^* f(g(c),f(c,g(c))) \to^* \cdots$$

Unfortunately, our strategy $\mathcal{S}_\omega$ does not normalize all terms of $\mathcal{R}$. For instance, a term like $f(g(c),g(g(c)))$ can only be normalized if we allow contraction of redexes that are not outermost.

Nevertheless, it can be shown that the set of terms which can be normalized by $\mathcal{S}_\omega$ properly includes those for which parallel-outermost succeeds. Of course, if we drop the rewrite rule $g(x) \to f(x, g(x))$ from $\mathcal{R}$ then we obtain a confluent TRS for which our strategy $\mathcal{S}_\omega$, unlike parallel-outermost, is normalizing.

This example shows in particular that Theorem 6.8 doesn't hold for non-left-linear TRSs (without critical pairs).

References

1. J.A. Bergstra and J.W. Klop, *Conditional Rewrite Rules: Confluence and Termination*, Journal of Computer and System Sciences **32**(3), pp. 323–362, 1986.
2. H. Comon, *Sequentiality, Second Order Monadic Logic and Tree Automata*, Proceedings of the 10th IEEE Symposium on Logic in Computer Science, San Diego, pp. 508–517, 1995.
3. N. Dershowitz and J.-P. Jouannaud, *Rewrite Systems*, in: Handbook of Theoretical Computer Science, Vol. B (ed. J. van Leeuwen), North-Holland, pp. 243–320, 1990.
4. G. Huet and J.-J. Lévy, *Computations in Orthogonal Rewriting Systems, I and II*, in: Computational Logic, Essays in Honor of Alan Robinson (eds. J.-L. Lassez and G. Plotkin), The MIT Press, pp. 396–443, 1991. Previous version: *Call by Need Computations in Non-Ambiguous Linear Term Rewriting Systems*, report 359, INRIA, 1979.
5. J.R. Kennaway, *Sequential Evaluation Strategies for Parallel-Or and Related Reduction Systems*, Annals of Pure and Applied Logic **43**, pp. 31–56, 1989.
6. J.W. Klop, *Combinatory Reduction Systems*, Ph.D. thesis, Mathematical Centre Tracts **127**, Centre for Mathematics and Computer Science, Amsterdam, 1980.
7. J.W. Klop, *Term Rewriting Systems*, in: Handbook of Logic in Computer Science, Vol. II (eds. S. Abramsky, D. Gabbay and T. Maibaum), Oxford University Press, pp. 1–116, 1992.
8. M.J. O'Donnell, *Computing in Systems Described by Equations*, Lecture Notes in Computer Science **58**, 1977.
9. V. van Oostrom, *Confluence for Abstract and Higher-Order Rewriting*, Ph.D. thesis, Vrije Universiteit, Amsterdam, 1994.
10. M. Oyamaguchi, *NV-Sequentiality: A Decidable Condition for Call-by-Need Computations in Term Rewriting Systems*, SIAM Journal on Computation **22**(1), pp. 114-135, 1993.
11. S. Porat and N. Francez, *Fairness in Term Rewriting Systems*, Methods of Logic in Computer Science **1**, pp. 141–181, 1994.
12. F. van Raamsdonk, *On Normalising Strategies*, Proceedings of the 2nd International Workshop on Higher-Order Algebra, Logic and Term Rewriting, Paderborn, Lecture Notes in Computer Science, 1995. To appear.
13. R.C. Sekar and I.V. Ramakrishnan, *Programming in Equational Logic: Beyond Strong Sequentiality*, Information and Computation **104**(1), pp. 78–109, 1993.
14. Y. Toyama, *Strong Sequentiality of Left-Linear Overlapping Term Rewriting Systems*, Proceedings of the 7th IEEE Symposium on Logic in Computer Science, Santa Cruz, pp. 274–284, 1992.