Reduction Strategies and Acyclicity

Jan Willem Klop[1,2,3], Vincent van Oostrom[4], and Femke van Raamsdonk[1]

[1] Vrije Universiteit, Department of Theoretical Computer Science,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
[2] Radboud Universiteit Nijmegen, Department of Computer Science,
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
[3] CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[4] Universiteit Utrecht, Department of Philosophy,
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands
`jwk@cs.vu.nl`, `Vincent.vanOostrom@phil.uu.nl`, `femke@cs.vu.nl`

Abstract. In this paper we review some well-known theory about reduction strategies of various kinds: normalizing, outermost-fair, cofinal, Church-Rosser. A stumbling block in the definition of such strategies is the presence of reduction cycles that may 'trap' a strategy as it is memory-free. We exploit a recently (re)discovered fact that there are no reduction cycles in orthogonal rewrite systems when each term has a normal form, in order to enhance some of the theorems on strategies, both with respect to their scope and the proof of their correctness.

Dedicated to Jean-Pierre Jouannaud
on the occasion of his 60th birthday

1 Introduction

The general objective of our note is to survey some fundamental theorems about reduction strategies in term rewriting, both first-order and higher-order. Some of these theorems are by now rather classical, and some are of a more recent date. The strategies serve various purposes: they may be normalizing, or outermost-fair, or cofinal, or Church-Rosser strategies, to name some of their more important properties. One liability they have in common: they may fall into a trap formed by a *reduction cycle*. This is so because the usual notion of a reduction strategy is such that no memory is involved: a reduction strategy $\mathbb{F}$ is just a function from terms to terms, such that for every term t that is not in normal form we have $t \to \mathbb{F}(t)$ in case $\mathbb{F}$ is a one-step strategy, and $t \twoheadrightarrow \mathbb{F}(t)$ in case $\mathbb{F}$ is a many-step strategy. Thus, if we have for a one-step strategy $\mathbb{F}$ that e.g. $\mathbb{F}^3(t) \equiv t$ (see Figure 1), then $\mathbb{F}$ will be forever trapped in this reduction cycle

$$t \to \mathbb{F}(t) \to \mathbb{F}(\mathbb{F}(t)) \to t$$

The danger of a strategy being trapped in a reduction cycle is apparent in the proofs of several theorems about strategies. E.g. avoiding cycles is prominent in

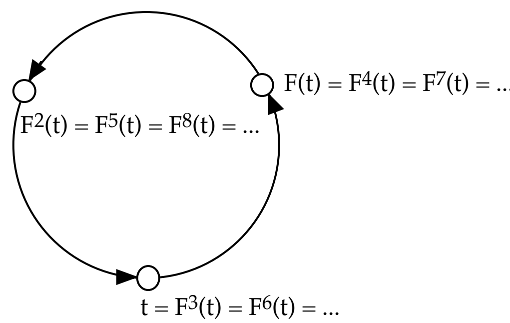

Fig. 1. trap

the proof that there exists a computable many-step Church-Rosser strategy for λ-calculus [5,3], discussed in Section 6. And avoiding cycles is a major complicating factor in Statman's definition of a cofinal strategy for combinators [14].

Now recently it was found by Ketema et al. [7] that under very natural conditions there are no reduction cycles at all. Of course, if the rewrite system has the property SN, strong normalization (or termination), there are no reduction cycles. But this is a bit too strong. Ketema et al. [7] proved that for orthogonal term rewriting systems that are WN, weakly normalizing, there are already no cycles, a property that we indicate by AC (acyclicity). This theorem generalizes to a wide class of rewrite systems, namely the weakly orthogonal fully extended higher-order rewriting systems. Actually, this theorem was for the case of almost orthogonal combinatory reduction systems already proved in a rather implicit way in Kennaway [6], but there it has remained unnoticed until now.

Combining our first observation of the 'cycle trap' for strategies with this recent insight that we have $\mathsf{WN} \Rightarrow \mathsf{AC}$ for orthogonal rewrite systems, there arises a natural question: does the acyclicity of orthogonal systems that are WN facilitate the proofs of the theorems for reduction strategies that are in the literature?

The goal of our note is to show that the answer is affirmative. In doing so, we obtain a strengthening of some of the theorems known for strategies, and a simplification of some of their proofs. We also present some new theorems. Finally, we employ in our formulation a notion of strategy, based on abstract rewriting notions, that is more expressive than the classical (functional) notion.

2 Basic notions

2.1 Abstract Rewriting

We start by introducing the necessary notions and notations concerning abstract rewriting, where the objects that are rewritten do not have any observable structure. One of the usual definitions of abstract rewriting is as a set equipped with a binary relation (as in [2, p.7], and as a special case of [15, Definition 1.1.1]). Here we take another approach, via the more expressive notion of ARS as in [15, Definition 8.2.2], where steps are first-class citizens.

Definition 1. An *abstract reduction system* (ARS) is defined as a quadruple

$$(A, \Phi, \mathsf{source}, \mathsf{target})$$

where A is a set of *objects*, Φ a set of *steps*, and $\mathsf{source} : \Phi \to A$ and $\mathsf{target} : \Phi \to A$ are functions mapping a step to its source and target. ARSs are denoted by $\to, \Rightarrow, \ldots$, and also by $\mathbb{A}, \mathbb{B}, \mathbb{S}, \mathbb{F}, \ldots$.

We can think of an ARS as a graph. There might be more than one step between two objects, so we can express more than binary relations on a set.

Example 1. See also Figure 2.

(i) The ARS $\mathbb{A}_1 = (A_1, \Phi_1, \mathsf{source}_1, \mathsf{target}_1)$ is defined by: $A_1 = \{a, b\}$, $\Phi_1 = \{\phi_1, \phi_2\}$, $\mathsf{source}_1(\phi_1) = a$, $\mathsf{target}_1(\phi_1) = a$, $\mathsf{source}_1(\phi_2) = a$, $\mathsf{target}_1(\phi_2) = b$.

(ii) The ARS $\mathbb{A}_2 = (A_2, \Phi_2, \mathsf{source}_2, \mathsf{target}_2)$ is defined by: $A_2 = \{a, b\}$, $\Phi_2 = \{\phi_1\}$, $\mathsf{source}_2(\phi_1) = a$, $\mathsf{target}_2(\phi_1) = a$.
So the underlying graph of an ARS is not necessarily connected.

(iii) The ARS $\mathbb{A}_3 = (A_3, \Phi_3, \mathsf{source}_3, \mathsf{target}_3)$ is defined by: $A_3 = \{a, b\}$, $\Phi_3 = \{\phi_2\}$, $\mathsf{source}_3(\phi_2) = a$, $\mathsf{target}_3(\phi_2) = b$.

(iv) The ARS $\mathbb{A}_4 = (A_4, \Phi_4, \mathsf{source}_4, \mathsf{target}_4)$ is defined by: $A_4 = \{a, b\}$, $\Phi_4 = \{\phi_2, \phi_3\}$, $\mathsf{source}_4(\phi_2) = \mathsf{source}_4(\phi_3) = a$, $\mathsf{target}_4(\phi_2) = \mathsf{target}_4(\phi_3) = b$.
In $\mathbb{A}_4$ there are two different steps from a to b, hence it cannot be given as a relation on a set. This phenomenon occurs for instance in λ-calculus with β-reduction: there are two different steps from $I\,(I\,I)$ to $I\,I$.

(v) The well-known counter-example against the implication 'local confluence implies confluence' is defined as the following ARS $\mathbb{A}_5$: $A = \{a, b, c, d\}$, $\Phi = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, $\mathsf{source}(\phi_1) = b$, $\mathsf{target}(\phi_1) = a$, $\mathsf{source}(\phi_2) = b$, $\mathsf{target}(\phi_2) = c$, $\mathsf{source}(\phi_3) = c$, $\mathsf{target}(\phi_3) = b$, $\mathsf{source}(\phi_4) = c$, $\mathsf{target}(\phi_4) = d$.

In some of the examples in the paper, we will write ARSs in a simplified notation, namely as a set of arrows with source and target. It is then understood that the set of objects is the set of objects mentioned, and that source and target are as suggested by the explicit enumeration of the arrows. For instance the ARS $\mathbb{A}_5$ from Example 1 is then written as $\{b \to a, b \to c, c \to b, c \to d\}$.

The definition of strategy below employs the following definition of sub-ARS.

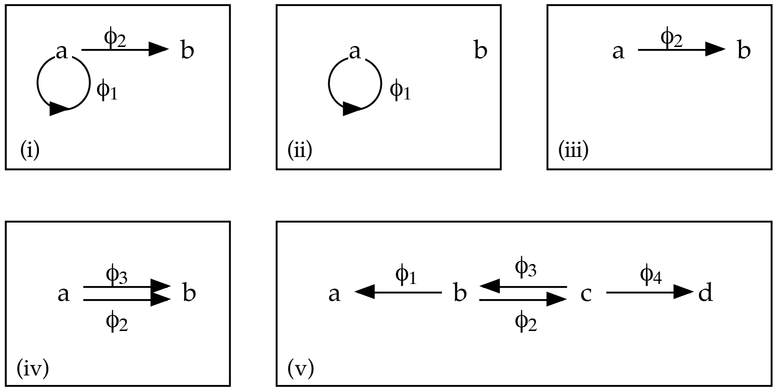

Fig. 2. The ARSs from Example 1

Definition 2. An ARS $\mathbb{A}_1 = (A_1, \Phi_1, \mathsf{source}_1, \mathsf{target}_1)$ is a *sub-ARS* of an ARS $\mathbb{A}_2 = (A_2, \Phi_2, \mathsf{source}_2, \mathsf{target}_2)$, notation $\mathbb{A}_1 \subseteq \mathbb{A}_2$, if the following holds:

(i) $A_1 \subseteq A_2$,
(ii) $\Phi_1 \subseteq \Phi_2$,
(iii) source_1 and target_1 are the restrictions of source_2 and target_2 to Φ_1.

If we think of an ARS as a graph, a sub-ARS is a sub-graph, with possibly less objects and less steps. The last condition of the definition of sub-ARS implies that $\mathsf{source}(\Phi_1) \subseteq A_1$ and also $\mathsf{target}(\Phi_1) \subseteq A_1$, that is, the sources and targets of steps in the small ARS must be present in its set of objects. Note that $\subseteq$ is a partial order on ARSs. The definition of sub-ARS used here is different from for instance [15, Definition 1.1.6]: there the set of objects of the 'small' ARS must be closed under the steps of the 'big' ARS. See also the example below.

Example 2.

(i) An ARS is a sub-ARS of itself: $\mathbb{A} \subseteq \mathbb{A}$.
(ii) For the ARSs defined of Example 1 we have: $\mathbb{A}_2 \subseteq \mathbb{A}_1$, $\mathbb{A}_3 \subseteq \mathbb{A}_1$, $\mathbb{A}_3 \subseteq \mathbb{A}_4$.
(iii) There is no sub-ARS of the ARS $\mathbb{A}_3$ from Example 1 with $\emptyset$ as set of objects, and $\{\phi_2\}$ as set of steps.
(iv) The ARS a without any steps is a sub-ARS of $a \to b$. So the set of objects of the 'small' ARS need not to be closed under the reduction of the 'big' ARS.

In the remainder of this subsection we assume an ARS $\mathbb{A} = (A, \Phi, \mathsf{source}, \mathsf{target})$.

A *reduction step* or *rewrite step* is an element of Φ, together with its source and target. Reduction steps are written as $\phi : a \to b$ or $a \to_\phi b$, for a step $\phi \in \Phi$. An object $a \in A$ is a *normal form* (or a is *in normal form*) if there is no reduction step $a \to$ with source a. A *reduction* or *rewrite sequence* starting in a_0 is a finite or infinite sequence of steps: $a_0 \to_{\phi_0} a_1 \to_{\phi_1} a_2 \to_{\phi_2} \ldots$. A finite reduction $a_0 \to_{\phi_0} \ldots \to_{\phi_{n-1}} a_n$ is said to be a *reduction from a_0 to a_n*. Such a reduction is also written as $a_0 \twoheadrightarrow a_n$. A reduction is said to be *maximal* if it cannot be extended, that is, either it ends in a normal form or it is infinite.

We sometimes omit irrelevant information from this notation, writing for instance $a \rightarrow$ to indicate that there is a step from a, or $a \twoheadrightarrow$ to indicate a finite or infinite reduction starting in a. If we want to stress that a step (or reduction) takes place in the ARS $\mathbb{A}$ we use $\mathbb{A}$ as subscript.

The *reduction graph of* a *in* $\mathbb{A}$, notation $\mathcal{G}(a, \mathbb{A})$, has as set of objects $\{b \mid a \twoheadrightarrow b\}$, and as set of steps all $\phi \in \Phi$ with $\mathsf{source}(\phi) \in \{b \mid a \twoheadrightarrow b\}$. Further, the source and target functions of $\mathcal{G}(a, \mathbb{A})$ are the appropriate restrictions of source and target. A reduction graph is a sub-ARS of $\mathbb{A}$.

An object $a \in A$ is *weakly normalizing in* $\mathbb{A}$, notation $a \in \mathsf{WN}(\mathbb{A})$, if there is a reduction sequence in $\mathbb{A}$ from a to a normal form of $\mathbb{A}$. The ARS $\mathbb{A}$ is *weakly normalizing*, notation $\mathbb{A} \in \mathsf{WN}$, if all objects $a \in A$ are weakly normalizing in $\mathbb{A}$. An object $a \in A$ is *strongly normalizing in* $\mathbb{A}$ or *terminating in* $\mathbb{A}$, notation $a \in \mathsf{SN}(\mathbb{A})$, if all reduction sequences starting in a are finite. The ARS $\mathbb{A}$ is *strongly normalizing* or *terminating*, notation $\mathbb{A} \in \mathsf{SN}$, if all objects $a \in A$ are strongly normalizing in $\mathbb{A}$. It may happen that an object is weakly normalizing in an ARS but not in a sub-ARS. For example, considering the ARSs of Example 1, we have $a \in \mathsf{WN}(\mathbb{A}_1)$, but a is not weakly normalizing in $\mathbb{A}_2 \subseteq \mathbb{A}_1$.

The ARS $\mathbb{A}$ is *confluent* if the endpoints of all pairs of co-initial reductions are joinable, so for all objects $a, b, b' \in A$: if $a \twoheadrightarrow b$ and $a \twoheadrightarrow b'$ then $b \downarrow_{\mathbb{A}} b'$.

2.2 Term rewriting

We use the standard notions and notations from [2,15] for first-order term rewriting systems (TRSs). We also consider higher-order rewriting systems with patterns (HRSs), defined by Nipkow [11]. At some places in the literature these are called PRSs. The terms are simply typed λ-terms with constants, that are considered modulo $\alpha\beta\eta$. Rules satisfy the patterns restriction that guarantees matching (and unification) modulo $\beta\eta$ to be decidable. See also [15, Chapter 11] for definitions and notations.

A term rewriting system is *orthogonal* if all rewrite rules are left-linear and there are no critical pairs. A term rewriting system is *weakly orthogonal* if all rewrite rules are left-linear and all critical pairs are trivial (so overlapping steps yield the same result). Almost orthogonal is in between orthogonal and weakly orthogonal: trivial critical pairs are allowed but only if the overlap is at the top.

Definition 3. A rewrite rule is *fully extended* if in the left-hand side every free variable has as arguments exactly all variables that are bound at that position.

In fully extended systems, whether a rule is applicable does not depend on whether or not a bound variable occurs. The rewrite rule $f(\lambda x.\, z) \rightarrow a$ with z a free variable is not fully extended. The rule can only be applied to a term of the form $f(\lambda x.s)$ if x does not occur in s. Another example of a rule that is not fully extended is the η-reduction rule: $\lambda x.M\, x \rightarrow M$, with the side-condition that x should not occur in the free variables of M.

2.3 Reduction strategies

A common approach in the literature is that a strategy, intuitively speaking, tells us how to reduce a term. That is, a strategy tells us how we *must* reduce a term. Then a strategy is a function from terms to terms, and hence deterministic, as in the Introduction. But we can also adopt a more general point of view: a strategy tells us how we *may* reduce a term. Following [15, Definition 9.1.1], see also [16], we take the latter approach.

Definition 4. A *strategy for an ARS* $\to$ is a sub-ARS of $\to$ having the same objects and normal forms.

So a strategy has the same objects as the original ARS, but possibly less arrows. Also, an object is a normal form in the strategy exactly if it is a normal form in the original ARS. The explanation for the requirement that a strategy has the same normal forms as the original ARS is that we want the strategy to be 'pushy': whenever there are still possibilities (steps, edges) to proceed, the strategy will select a positive number of them, and not discard them all. So a strategy can only stop in a normal form, but not before. Note that 'strategy for' is transitive: a strategy $\mathbb{S}$ for a strategy $\mathbb{S}'$ for an ARS $\mathbb{A}$ is again a strategy for $\mathbb{A}$. Thus we have a notion of *refinement* of strategies.

Example 3.

(i) An ARS is a strategy for itself. In fact, this is the strategy of *exhaustive search*. This also shows that a strategy may be non-deterministic.
(ii) We consider the following ARS $\mathbb{A}$ from [15, Example 9.1.2]: $a \leftrightarrows b \to c$. Then $a \to b \to c$ is a strategy for $\mathbb{A}$. The sub-ARS $b \to c$ is not a strategy for $\mathbb{A}$, because it does not have the same objects as $\mathbb{A}$. The sub-ARS $a \leftarrow b \to c$ is not a strategy for $\mathbb{A}$ either, because a is not a normal form in $\mathbb{A}$.
(iii) We reconsider the ARSs of Example 1. $\mathbb{A}_2$ and $\mathbb{A}_3$ are both strategies for $\mathbb{A}_1$. Also, $\mathbb{A}_3$ is a strategy for $\mathbb{A}_4$.
(iv) Consider following ARS: $\mathbb{B} = \{a \to b, a \to c, a \to d, d \to d\}$. Then $\mathbb{S} = \{a \to b, a \to c, d \to d\}$ is a non-deterministic strategy for $\mathbb{B}$.

Properties of strategies. A strategy is usually employed to obtain a reduction with a certain property.

Definition 5. Let $\mathbb{S}$ be a strategy for $\mathbb{A} = (A, \Phi, \mathsf{source}, \mathsf{target})$.

(i) $\mathbb{S}$ is *normalizing* if $a \in \mathsf{SN}(\mathbb{S})$ for every object $a \in \mathsf{WN}(\mathbb{A})$.
(ii) $\mathbb{S}$ is *cofinal* if the following holds: for every $a \in A$, for every finite reduction $\mathcal{R} : a \twoheadrightarrow a'$ in $\mathbb{A}$, for every maximal reduction $\mathcal{S} : a \twoheadrightarrow$ in $\mathbb{S}$, there is an object c in $\mathcal{S}$ such that $a' \twoheadrightarrow c$ in $\mathbb{A}$. See Figure 3.

In Example 1, the ARS $\mathbb{A}_3$ is a normalizing strategy for $\mathbb{A}_1$. An ARS is not necessarily a cofinal strategy for itself. For instance, in the ARS $\mathbb{A}_1$ of Example 1 the final object b of the reduction $a \to b$ cannot be reduced to an object in the maximal reduction $a \to a \to a \to \ldots$. Also in this possibly non-deterministic setting cofinal strategies are normalizing [15, Proposition 9.1.16].

Proposition 1 ([15]). *Cofinal strategies are normalizing.*

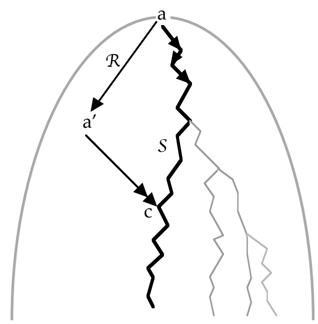

Fig. 3. a cofinal strategy

Church-Rosser strategies. We now focus on a subclass of strategies. A strategy is said to be *deterministic* or *functional* if every object is the source of at most one arrow. Reconsidering the ARSs of Example 1, both $\mathbb{A}_2$ and $\mathbb{A}_3$ are functional strategies for $\mathbb{A}$. The ARS $\mathbb{A}_1$ is also a strategy for itself, which is not functional.

A function from terms to terms that maps a normal form to itself, and a term that is not in normal form to one of its one-step reducts, naturally induces a deterministic (or functional) strategy. Therefore we use in the setting of functional strategies also the notation $\mathbb{F}^n(a)$ for the object reached by performing n steps according to $\mathbb{F}$, starting in a (with $\mathbb{F}(a) \equiv a$ if a is in normal form).

Definition 6. A deterministic (or functional) strategy $\mathbb{F}$ for an ARS $\mathbb{A}$ is said to be a *Church-Rosser strategy for* $\mathbb{A}$, also called a *CR-strategy for* $\mathbb{A}$, if the following holds: if $a =_{\mathbb{A}} b$ then $\exists n, m \in \mathbb{N} : \mathbb{F}^n(a) \equiv \mathbb{F}^m(b)$. See also Figure 4.

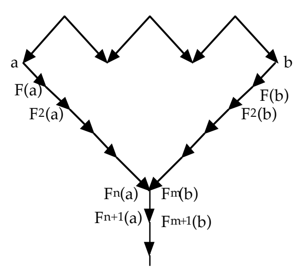

Fig. 4. one-step CR-strategy

In Section 6 we will also consider *many-step* functional Church-Rosser strategies, where intuitively a term not in normal form is mapped to some reduct (where the reduct is reached in at least one step).

Overview of properties of strategies. We conclude this section with a picture (Figure 5) of properties of reduction strategies, for abstract rewriting and term rewriting, and their interrelations. The notions fair and leftmost-fair are included for completeness and do not play a rôle in the remainder of the paper. The implication 'cofinal implies normalizing' is Proposition 1. The implication 'Church-Rosser implies cofinal' is Theorem 8, which is concerned with functional (or deterministic) strategies. The implication 'normalizing implies Church-Rosser' holds for functional strategies in weakly normalizing and confluent ARSs: a normalizing strategy then yields a common reduct of two convertible objects by computing their normal forms, which are the same. The implication 'outermost-fair implies normalizing' is Theorem 7, which is concerned with the more general setting of weakly orthogonal and fully extended HRSs.

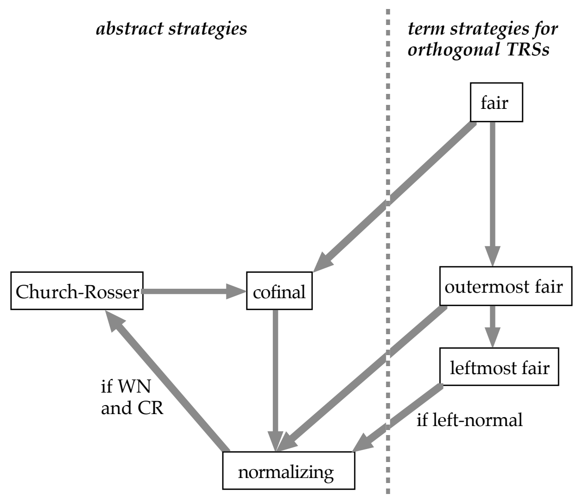

Fig. 5. overview of reduction strategies

3 A normalizing strategy

A normalizing strategy is of interest in the case that a rewriting system has terms that can be rewritten to normal form, but are also the starting point of an infinite reduction. This happens for instance to the term $\mathsf{K}\,\mathsf{I}\,((\mathsf{S}\,\mathsf{I}\,\mathsf{I})\,(\mathsf{S}\,\mathsf{I}\,\mathsf{I}))$ in Combinatory Logic (CL) which is defined by the rewrite rules

$$\begin{aligned} &\mathsf{I}\,x && \to x \\ &\mathsf{K}\,x\,y && \to x \\ &\mathsf{S}\,x\,y\,z && \to x\,z\,(y\,z) \end{aligned}$$

For (pure) CL the leftmost-outermost strategy is normalizing. But let us now consider CL extended with Gustave's term rewriting system which is defined by the following rules:

$$\begin{aligned} g(a,b,x) &\to c \\ g(x,a,b) &\to c \\ g(b,x,a) &\to c \end{aligned}$$

How should we evaluate a term of the form $g(l, m, r)$ with l, m, and r redexes of (pure) CL? Such a term intuitively seems to require parallel evaluation because it is undecidable whether a CL term has a (head) normal form. Nevertheless, Kennaway [6] proves, surprisingly, that there is a normalizing one-step strategy for the combination of CL and Gustave's TRS. More precisely, Kennaway proves that almost orthogonal Combinatory Reduction Systems (CRSs) have a computable one-step normalizing strategy. This means that also λ-calculus with parallel-or, defined by

$$\begin{array}{ll} (\lambda x.\, z)\, z' & \to z[x := z'] \\ \mathsf{por}(\mathsf{t}, z) & \to \mathsf{t} \\ \mathsf{por}(z, \mathsf{t}) & \to \mathsf{t} \\ \mathsf{por}(\mathsf{f}, \mathsf{f}) & \to \mathsf{f} \end{array}$$

has a computable one-step normalizing strategy, although intuitively it seems that a term $\mathsf{por}(M, N)$ with M and N λ-terms requires a parallel evaluation.

Antoy and Middeldorp [1] define a computable one-step strategy for almost orthogonal term rewriting systems, and show that it is normalizing via a reduction to normalization of outermost-fair rewriting. In this section we present two observations concerning their strategy. The first observation is that the definition of the strategy, and the proof of its normalization, can be extended to the case of fully extended weakly orthogonal higher-order rewriting. The second observation is that the definition of the strategy can be made substantially simpler for weakly normalizing systems, because then we do not have to deal with cycles.

The strategy $\mathbb{S}_F$. We define a non-deterministic strategy $\mathbb{S}_F$ for higher-order rewriting. The definition is a minor adaptation of the definition by Antoy and Middeldorp [1]. We use the notation $\mathbb{S}_F$ from [15]; the notation used in [1] is $\mathcal{S}_\omega$.

In the definition of $\mathbb{S}_F$ for higher-order rewriting we make use of the notion 'acyclicity-check'. In general, it is undecidable whether a term cycles. However, for finite systems, it is decidable whether a term t cycles within its own height, that is, is on a cycle where all terms are at most as big as t itself. More in particular, it is also decidable whether a terms cycles according to a strategy within its own height.

Definition 7. Assume a HRS $\mathbb{R}$. The strategy $\mathbb{S}_F$ and the property $\mathsf{AC}_{\mathbb{S}_F}$ of passing the $\mathbb{S}_F$-acyclicity-check are defined by induction on the structure of terms as follows:

(i) Given a term s, the strategy $\mathbb{S}_F$ can perform the following steps:
  (a) if s is a redex, then $\mathbb{S}_F$ can only contract a redex at the root,
  (a) otherwise, $\mathbb{S}_F$ is applied to a smallest direct argument satisfying $\mathsf{AC}_{\mathbb{S}_F}$,
  (a) otherwise, $\mathbb{S}_F$ can perform any step.

(ii) A term t satisfies $\mathsf{AC}_{\mathbb{S}_F}$, or otherwise said *passes the* $\mathbb{S}_F$*-acyclicity-check*, if it is not on a $\mathbb{S}_F$-cycle in which all terms are at most as big as t. Otherwise, it fails the acyclicity-check.

The strategy $\mathbb{S}_F$ is an outermost strategy that is only unfair to redexes that intuitively do not contribute to the normal form because they cycle. Below (Theorem 1) we show that $\mathbb{S}_F$ is normalizing.

Note that in the definition of $\mathbb{S}_F$-acyclicity-check for a term t, we only need the definition of $\mathbb{S}_F$ on terms smaller than t. Since overlap is allowed, a redex at the root is not necessarily unique.

Further, $\mathbb{S}_F$ is a non-deterministic strategy. If the first clause of the definition applies, then there is possibly a choice between overlapping root-redexes. If the second clause of the definition applies, there is a choice between all smallest reducible arguments that pass the $\mathbb{S}_F$-acyclicity-check. In addition, there might be a choice between overlapping redexes.

Example 4. We consider the orthogonal TRS defined by the following rules:

$$\begin{array}{ll} f(x, g'(y), h'(z)) & \to b \\ a & \to a \\ b & \to g(g(b)) \\ g(x) & \to g'(x) \\ h(x) & \to h'(x) \end{array}$$

An example of a reduction according to $\mathbb{S}_F$:

$f(a, b, h(b)) \to$
  the argument a fails the acyclicity-check
  b is the smallest reducible argument passing the acyclicity-check
$f(a, g(g(b)), h(b)) \to$
  $h(b)$ is the smallest reducible argument passing the acyclicity-check
$f(a, g(g(b)), h'(b)) \to$
  $h'(b)$ is the smallest reducible argument passing the acyclicity-check
$f(a, g(g(b)), h'(g(g(b)))) \to$
  $g(g(b))$ is the smallest reducible argument passing the acyclicity-check
$f(a, g'(g(b)), h'(b)) \to$
  there is a root-redex
b

The strategy $\mathbb{S}_F$ skips arguments that are $\mathbb{S}_F$-cyclic within their own height. One might wonder whether we could skip more arguments, while remaining normalizing. The following examples shows that skipping arguments that are cyclic (but not necessarily $\mathbb{S}_F$-cyclic) destroys normalization of $\mathbb{S}_F$.

Example 5. We consider the orthogonal TRS defined by the following rules:

$$\begin{array}{ll} a & \to b \\ c & \to c \\ g(b, y) & \to d \\ h(d, y) & \to e \end{array}$$

The term $h(g(a, c), c)$ has a normal form which is indeed found in the following $\mathbb{S}_F$-rewrite sequence:

$$h(g(a, c), c) \to h(g(b, c), c) \to h(d, c) \to e$$

However, if we adapt the definition of the strategy in the sense that we skip arguments that cycle within their own height, then we skip the argument $g(a, c)$ because $g(a, c) \to g(a, c)$, and we also skip the argument c because $c \to c$. In other words, both arguments are skipped, and the third clause of the definition of the strategy applies. That is, also

$$h(g(a, c), c) \to h(g(a, c), c) \to \dots$$

is a rewrite sequence according to the strategy. So this adaptation yields a strategy that is not normalizing.

One might also wonder whether we could skip fewer arguments, while remaining normalizing. The following example shows that skipping arguments if all their reductions are cyclic within their own height (so an argument that is $\mathbb{S}_F$-cyclic within its own height but also has another reduction is not skipped anymore) destroys normalization of $\mathbb{S}_F$.

Example 6. We consider the orthogonal TRS defined by the following rules:

$$\begin{array}{ll} a & \to f(a) \\ g(x) & \to g(x) \\ h(x) & \to h'(x) \\ j(h'(x), y) & \to b \end{array}$$

The term $j(h(h(x)), g(a))$ has a normal form. Indeed the following $\mathbb{S}_F$-rewrite sequence finds the normal form:

$$j(h(h(x)), g(a)) \to j(h'(h(x)), g(a)) \to b$$

Note that the argument $g(a)$ is skipped because it is $\mathbb{S}_F$-cyclic within its own height. However, if we adapt the definition of the strategy by skipping arguments if all their reductions are cyclic within their own height, then the argument $g(a)$ is no longer skipped. In that case we obtain the following rewrite sequence:

$$j(h(h(x)), g(a)) \to j(h(h(x)), g(a)) \to \dots$$

So this change in the condition on skipping arguments makes that the strategy is no longer normalizing.

Normalization via outermost-fair rewriting. One can show that if the strategy $\mathbb{S}_F$ fails to normalize a term s, then there is an infinite outermost-fair rewrite sequence starting from s. Section 5 is concerned with outermost-fair rewriting and in particular contains the theorem that outermost-fair rewriting is normalizing for HRSs that are fully extended and weakly orthogonal from [13].

Theorem 1. *Assume a fully extended and weakly orthogonal HRS. If there is an infinite $\mathbb{S}_F$-rewrite sequence starting from s, then there is an infinite outermost-fair rewrite sequence starting from s.*

Proof. We only give the idea of the proof. Suppose that there is an infinite $\mathbb{S}_F$-rewrite sequence starting in s. An outermost redex that is unfairly treated is on a $\mathbb{S}_F$-cycle. Also, it is disjoint from other such redexes. A $\mathbb{S}_F$-cycle can be made outermost-fair. Now interleave the original infinite $\mathbb{S}_F$-rewrite sequence repeatedly with extra reductions on the place of the cycles, to obtain an infinite outermost-fair reduction starting in s.

Corollary 1. *The strategy $\mathbb{S}_F$ is a computable one-step normalizing strategy for HRSs that are weakly orthogonal and fully extended.*

Acyclic rewriting. A second observation is that for weakly normalizing systems, the acyclicity-check can be omitted, thanks to Theorem 6. The definition of the strategy then becomes more simple, and applying the strategy becomes essentially easier.

Definition 8. Assume a weakly normalizing HRS $\mathbb{R}$. The strategy $\mathbb{S}_F^{AC}$ is defined by induction on the structure of terms as follows. Given a term s, the strategy $\mathbb{S}_F^{AC}$ can perform the following steps:

(i) if s is a redex, then $\mathbb{S}_F^{AC}$ can only contract a redex at the root,
(ii) otherwise, $\mathbb{S}_F^{AC}$ is applied to a smallest direct argument that is reducible.

We claim that $\mathbb{S}_F^{AC}$ is outermost-fair. Hence by Theorem 7 it is normalizing.

4 Acyclicity

As argued in this paper, for several strategies their computability is seen to rely on having a suitable cycle detection sub-routine available. As cycle detection is complex, it is interesting to look at rewrite systems which do not need it, i.e. which are acyclic. Then the sub-routine can simply answer: no, with a resulting simplification, as exemplified in Definition 8. In this section, we recapitulate some known acyclicity results and establish some new ones.

The following two classical acyclicity results for Combinatory Logic (CL) are due to [9] and [5] respectively. Here we consider CL with basis $\{\mathsf{S}, \mathsf{K}, \mathsf{I}\}$ and the following applicative rewrite rules:

$$\begin{aligned}
&\mathsf{I}\,x &&\to x\\
&\mathsf{K}\,x\,y &&\to x\\
&\mathsf{S}\,x\,y\,z &&\to x\,z\,(y\,z)
\end{aligned}$$

Theorem 2 ([9]). *In Combinatory Logic, every finite reduction graph is acyclic.*

This 'finite acyclicity' result is very much CL-specific. For instance, both the term a in the single-rule orthogonal TRS $\{a \to a\}$ as well as the term $\Omega = (\lambda x.xx)(\lambda x.xx)$ in the λ-calculus, have a single-node cyclic reduction graph. By the theorem, since the reduction graph of the direct translation $\mathsf{SII}(\mathsf{SII})$ of Ω into CL is cyclic: $\mathsf{SII}(\mathsf{SII}) \twoheadrightarrow_{CL} \mathsf{SII}(\mathsf{SII})$ it can no longer be finite, and indeed it is not: $\mathsf{SII}(\mathsf{SII}) \twoheadrightarrow_{CL} \mathsf{I}(\mathsf{SII}(\mathsf{SII})) \twoheadrightarrow_{CL} \mathsf{I}(\mathsf{I}(\mathsf{SII}(\mathsf{SII}))) \twoheadrightarrow_{CL} \ldots$.

Theorem 3 ([5]). *In Combinatory Logic,* S*-terms are acyclic.*

As it brings out nicely the minimal counter-example technique commonly employed in acyclicity proofs, we now present an alternative proof to this result.

Proof. For a proof by contradiction, suppose a cycle $\sigma : t \twoheadrightarrow t$ of minimal height would exist. By minimality σ is a *head* cycle, i.e. σ is of shape $t \twoheadrightarrow \mathsf{S}\, s\, u\, v \to s\, v\, (u\, v) \twoheadrightarrow t$. Observe no S-step decreases size and only doesn't increase size if its third argument is an S. But then in the next head-step, the term size increases, because no term other than S reduces to S, in particular, $u\, v$ does not reduce to S.

Remark 1. Neither of Theorems 2 and 3 entails the other; S-terms may not be terminating and thus have infinite reduction graphs, [5] and the example $\mathsf{SII}(\mathsf{SII})$ above shows that general CL-terms need not be acyclic. It could be difficult to find a common generalization, since the former result is even dependent on the particular basis chosen (here $\{\mathsf{S}, K, I\}$) as noted in [5].

Next we present two results allowing to infer acyclicity of a system from that same property for its components. The first result is a special case of a more general result due to Middeldorp and Ohsaki [10, Theorem 6.7].

Theorem 4 ([10]). *The disjoint union* $\mathbb{R}$ *of two TRSs is acyclic if both components are and either* $\mathbb{R}$ *is non-collapsing, or* $\mathbb{R}$ *is non-duplicating, or one of the components is both non-collapsing and non-duplicating.*

Here we complement the above result by showing that acyclicity is preserved when combining orthogonal acyclic TRSs.

Theorem 5. *Acyclicity is modular for orthogonal TRSs.*

Proof. See Appendix A.

The proof method also yields modularity of absence of non-empty *fixed-point* reductions of shape $t \twoheadrightarrow C[t]$ for orthogonal TRSs.

Remark 2. Both non-overlappingness and left-linearity are essential for acyclicity to be modular, answering questions by Middeldorp: Let $\mathcal{R}_b$ either be the overlapping left-linear TRS with rules $\{g(x,y) \to x, g(x,y) \to y\}$ or the non-overlapping non-left-linear TRS with rules $\{g(x,y,z,z) \to x, g(x,y,z,S(z)) \to y, \infty \to S(\infty)\}$. In either case, $\mathcal{R}_b$ is acyclic since applying a g-rule decreases the number of g-symbols. Combining either with the acyclic orthogonal TRS $\mathcal{R}_w$ $\{f(0,1,x) \to f(x,x,x)\}$ yields a cyclic combination $\mathcal{R}_b \uplus \mathcal{R}_w$, as can be seen from $f(0,1,g(0,1))$ or $f(0,1,g(0,1,\infty,\infty))$, respectively.

Remark 3. First-orderness is also essential for modularity of acyclicity. The HRS (higher-order pattern rewriting system in the sense of Nipkow) consisting of the single rule

$$f(xyz.Z(x,y,z), W, V) \to Z(W, Z(V, W, f(xyz.Z(x,y,z), W, V)), V)$$

[5] Remarkably termination *is* decidable for S-terms [17].

is acyclic (see below). The two-rule TRS

$$g(a, x, y) \to x$$
$$g(b, x, y) \to y$$

is trivially acyclic. However, their combination is not as witnessed by

$$\begin{aligned}&\underline{f(xyz.g(x, y, z), a, b)}\\ &\to \underline{g(a, g(b, a, f(xyz.g(x, y, z), a, b)), b)}\\ &\to \underline{g(b, a, f(xyz.g(x, y, z), a, b))}\\ &\to f(xyz.g(x, y, z), a, b)\end{aligned}$$

The main feat of the HRS is that the lhs of its rule is embeddable into its rhs, but only so in a non-empty context. If it were the case the context could be empty, then the term substituted for Z should collapse both to its second (y) and third (z) argument *only* depending on its first (x) argument, which is impossible using only the first rule (see Appendix B). It is exactly this feature which the two selection rules of the TRS bring.

Remark 4. We conjecture that the corresponding CRS rule (combinatory reduction system in the sense of Klop):

$$f([xyz]Z(x, y, z), W, V) \to Z(W, Z(V, W, f([xyz]Z(x, y, z), W, V)), V)$$

is acyclic as well. This does not follow directly from the above since the CRS contains 'spurious' terms, i.e. terms such as $f([x]f(x))$ which are not the image of a HRS term.

We conclude this section by discussing the relationship between normalization and acyclicity. Of course $\mathsf{SN} \Rightarrow \mathsf{AC}$, i.e. termination trivially implies acyclicity. More interesting is that in the presence of orthogonality weak normalization implies acyclicity. This result was prefigured in Kennaway [6] and recently rediscovered [7, Theorem 5.1], and extended [8, Theorem 3.1] by Ketema, Klop, and van Oostrom.

Theorem 6. *Assume a fully extended and weakly orthogonal HRS $\mathbb{R}$. Then:*

$$\mathbb{R} \in \mathsf{WN} \Rightarrow \mathbb{R} \in \mathsf{AC}$$

We give a sketch of the proof as it again nicely illustrates the minimal counter example technique, as well as the rôle of normalization of the outermost-fair strategy.

Proof. Assume that the system is not acyclic. Take a minimal term t with a cycle. That cycle contains a head step. As a consequence, t admits an infinite reduction with infinitely many head steps. Clearly such a reduction is outermost-fair, and since the outermost-fair strategy is normalizing and the system is WN by assumption, the reduction ends in a normal form. Contradiction.

Remark 5. Weak normalization cannot be relaxed from *system* to *term* level, as witnessed by the weakly normalizing but cyclic term $f(a)$ in the orthogonal (but not WN) TRS $\{f(x) \to b, a \to a\}$.

5 Outermost-fair rewriting

Normalization of $\mathbb{S}_F$ is proved by a reduction to normalization of outermost-fair rewriting. This section is concerned with normalization of outermost-fair rewriting; it does not contain new results.

A rewrite sequence is *outermost-fair* if every outermost redex is eventually contracted. For instance, in Example 5 the rewrite sequence $h(g(a,c),c) \to h(g(a,c),c) \to \ldots$ is not outermost-fair because it is unfair to the outermost redex a. The rewrite sequence $j(h(h(x)),g(a)) \to j(h(h(x)),g(a)) \to \ldots$ of Example 6 is not outermost-fair because it is unfair to the outermost redex $h(h(x))$.

O'Donnell [12] shows that outermost-fair rewriting is normalizing for almost orthogonal TRSs. Van Oostrom [13] extends this in two directions: from first-order to higher-order, and from almost orthogonal to weakly orthogonal.

Theorem 7 ([13]). *Outermost-fair strategies are normalizing for HRSs that are fully extended and weakly orthogonal.*

We collect some observations concerning the limitations of possible extensions.

Remark 6.

(i) In case rewrite rules with arbitrary overlapping patterns are allowed, then outermost-fair rewriting is not necessarily normalizing. Consider for instance the TRS from [13] defined by $\{a \to b, f(a) \to f(a)\}$. The term $f(a)$ can be reduced to the normal form $f(b)$, but the outermost-fair reduction $f(a) \to f(a) \to \ldots$ does not reach a normal form.

(ii) If non-fully-extended rewrite rules are allowed, then outermost-fair rewriting is not necessarily normalizing. Consider for example the HRS defined by $\{f(\lambda x.\, z) \to a, g(x) \to a, h(x) \to h(x)\}$. The term $f(\lambda x.\, h(g(x)))$ has a normal form: $f(\lambda x.\, h(g(x))) \to f(\lambda x.\, h(a)) \to a$. However, the outermost-fair reduction $f(\lambda x.\, h(g(x))) \to f(\lambda x.\, h(g(x))) \to \ldots$ contracting h-redexes does not reach the normal form. The reason is that the outermost f-redex is created only by contraction of the non-outermost g-redex.

(iii) Outermost-fair does not imply cofinal. Consider for instance the TRS defined by $\{f(x) \to f(x), a \to b\}$. The outermost-fair reduction $f(a) \to f(a) \to f(a) \to \ldots$ is not cofinal because the term $f(b)$ (which is a reduct of $f(a)$) cannot be reduced to a $f(a)$.

(iv) In [13] also head-normalization is considered. Using the proof method of Theorem 7, it is shown that outermost-fair strategies are head-normalizing for HRSs that are fully extended and almost orthogonal. Surprisingly, this cannot be extended to the *weakly* orthogonal case.

6 Church-Rosser strategies

In a research seminar in 1975 with as outcome the set of notes [4], the question was raised whether there exists a computable one-step Church-Rosser strategy for the λ-calculus? Here a strategy was defined in the classical sense, so as a functional strategy, and the notion of a (functional) CR-strategy was defined as in Definition 6. As far as we know, the question above is still unsolved. In [5] some partial results were obtained.

Church-Rosser strategies are important for the following reason, stated in Barendregt [3, Lemma 13.1.4] (the assumption of confluence is superfluous there; it follows from the assumption of a CR-strategy).

Theorem 8 ([3]). *Let $\mathbb{F}$ be a one-step or many-step functional strategy for the ARS $\mathbb{A}$. Then:*

$$\mathbb{F} \textit{ is a CR-strategy} \quad \Rightarrow \quad \mathbb{F} \textit{ is cofinal} \quad \Rightarrow \quad \mathbb{F} \textit{ is normalizing.}$$

Proof. The second implication is a special case of Proposition 1. We now prove the first implication. So suppose $\mathbb{F}$ is a CR-strategy for $\mathbb{A} = (A, \Phi, \mathsf{source}, \mathsf{target})$ and let $a \in A$. We claim that the reduction $a \to \mathbb{F}(a) \to \mathbb{F}^2(a) \to \ldots$ (in case $\mathbb{F}$ is a one-step strategy) or $a \twoheadrightarrow \mathbb{F}(a) \twoheadrightarrow \mathbb{F}^2(a) \twoheadrightarrow \ldots$ (in case $\mathbb{F}$ is a many-step strategy) is cofinal in $\mathcal{G}(a, \mathbb{A})$. Indeed, if $b \in \mathcal{G}(a, \mathbb{A})$, then $a =_{\mathbb{A}} b$, so since $\mathbb{F} \in \mathsf{CR}$, the trails $\mathbb{F}(a), \mathbb{F}^2(a), \mathbb{F}^3(a), \ldots$ and $\mathbb{F}(b), \mathbb{F}^2(b), \mathbb{F}^3(b), \ldots$ will intersect. This shows that $\mathbb{F}$ is a cofinal strategy.

Theorem 9 ([5]). *There exists a computable many-step CR-strategy for the lambda-calculus.*

One of the main problems in the proof of Theorem 9 is to avoid that the CR-strategy $\mathbb{F}$ to be defined, falls in the trap of a double $\mathbb{F}$-cycle which is linked, see Figure 6.

Suppose there are two different $\mathbb{F}$-cycles C_1 and C_2 containing points M_1 and M_2 such that $M_1 =_{\mathbb{R}} M_2$ (that is, M_1 and M_2 are convertible in $\mathbb{R}$).

Then we are in a contradictory situation: $\mathbb{F}$ cannot be a CR-strategy, because the $\mathbb{F}$-trails for M_1 and M_2 will keep cycling without ever intersecting.

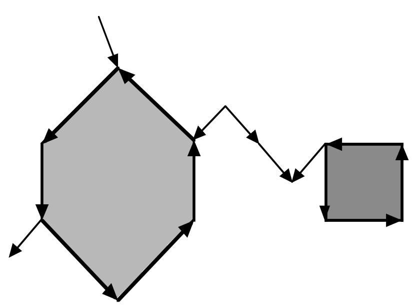

Fig. 6. linked cycles

An effective many-step CR strategy. We now show that under far more general circumstances a computable many-step CR-strategy exists.

Definition 9. *An* effective *ARS is an ARS* $\mathbb{A}$ *with an injective computable function* $\#$ *from objects to natural numbers, and where the reduction relation is computable.*

So if $\mathbb{A}$ is Church–Rosser, we can compute a common reduct for any given finite set of convertible objects. Effective ARSs occur frequently, e.g. λ-calculus, or CL, or any TRS or HRS based on a finite signature and with a finite set of reduction rules. We will identify an object a with its corresponding natural number $\#a$, and with $\rightarrow_n$ the restriction of the reduction relation $\rightarrow$ of $\mathbb{A}$ to objects less than or equal to n is denoted. We show that if such an effective ARS $\rightarrow$ has the Church–Rosser property, then it has an effective many-step Church–Rosser strategy.

The core of our strategy is constituted by the following procedure which, for a given equivalence class of terms less than a given bound n, attempts to find a common reduct greater than that bound. Formally, for an object a and bound n with $\#a \leq n$ the object a_n is defined to be a common reduct of all (finitely many!) objects which are $\rightarrow_n$-convertible to a, such that that common reduct

(i) is greater than n, or else
(ii) is the greatest object of a strongly connected component of $\rightarrow_n$.

Here 'strongly connected' means that every pair of objects is related by $\twoheadrightarrow_n$. Note that by effectiveness of the ARS and the Church–Rosser property it is easy to turn the construction of a reduct a' common to the equivalence class, into a deterministic procedure. Next, by deterministically searching from a', going through the finitely many objects in the equivalence class, we will find a unique common reduct greater than n if it exists. Otherwise, the second item applies and one notes that the strongly connected component, and hence the common reduct computed, is unique by the Church–Rosser property (the strongly connected component may consist of a single object; a normal form).

The idea is then to repeat this procedure. But using what bounds? Using the number of the object itself as bound, i.e. from a compute $a_{\#a}$, may fail:

$$
\begin{tikzcd}
0' \arrow[r] & 1' \arrow[r] & 2' \arrow[r] & 3' \arrow[r] & 4' \arrow[r] & 5' \arrow[r] & \cdots\cdots & {} \\
0 \arrow[u] \arrow[urr] & 1 \arrow[u] \arrow[urr] & 2 \arrow[u] \arrow[urr] & 3 \arrow[u] \arrow[urr] & 4 \arrow[u] \arrow[urr] & 5 \arrow[u] \arrow[urr] & &
\end{tikzcd}
$$

Fig. 7. Out-of-sync

Example 7. Consider the ARS of Figure 7, with the numbering $\#$ defined by $\#i = 2i$ and $\#i' = 2i + 1$. Then $0'_{\#0'} = 0'_1$ yields $2'$ (a common reduct of 0 and $0'$ and greater than both), $2'_{\#2'} = 2'_5$ yields $4'$, ... whereas $1'_{\#1'}$ yields $3'$, $3'_{\#3'}$ yields $5'$, etc.. A common reduct of $0'$ and $1'$ will never be reached.

One could think of the problem as arising from each object determining its own 'clock' and these staying 'out-of-sync' forever. To avoid this phenomenon, we inductively construct a (monotonically increasing) sequence of bounds $n(i)$, which one can think of as the 'ticks of a global clock'.

- $n(0) = 0$;
- $n(i+1)$ is the maximum of $n(i)+1$ and all objects occurring in the construction of the common reducts $a_{n(i)}$, for a such that $\#a \leq n(i)$.

Now define $\mathbb{CR}(a) = a_{n(i)}$, with i least such that $\#a \leq n(i)$. Per construction $a \twoheadrightarrow_{n(i+1)} \mathbb{CR}(a)$, and if the first item applies $n(i) < \#\mathbb{CR}(a) \leq n(i+1)$, or else $\mathbb{CR}(a)$ is the greatest object of a strongly connected component.

Example 8. Consider the ARS of Figure 7 with the numbering $\#$ of Example 7.

- $n(0) = 0$,
- $n(1) = 1$, the number of $0'$, a (common) reduct of 0 greater than 0,
- $n(2) = 5$, the number of $2'$, a common reduct of 0 and $0'$,
- $n(3) = 9$, the number of $4'$, a common reduct of $0, \ldots, 2'$,
- $n(4) = 13, \ldots$

Thus $\mathbb{CR}(0') = 0'_1 = 2'$ and $\mathbb{CR}(2') = 2'_5 = 4'$ which is the same object as $\mathbb{CR}(1') = 1'_5 = 4'$ (since $2'$ and $1'$ are convertible within $\rightarrow_5$), as desired.

Remark 7. 'Ticks' of the 'global clock' need not have 'equal duration'. This can be seen by considering the multiplication ARS having positive natural numbers as objects (having themselves as number) and multiplication by a number as steps. Then $n(i+1)$ will be, for $i \geq 3$, the least common multiple of *all* numbers up to $n(i)$. Thus, in general, the 'duration' of the 'ticks' grows very fast.

Theorem 10. *$\mathbb{CR}$ is an effective many-step Church–Rosser strategy.*

Proof. That $\mathbb{CR}$ is an effective many-step strategy is clear from the above. To show it is a Church–Rosser strategy suppose a conversion between a and b exists, and let $n(i)$ bound the largest object in this conversion. By the above, if the first item always applies, then $a \twoheadrightarrow_{n(i)} \mathbb{CR}^k(a)$ and $b \twoheadrightarrow_{n(i)} \mathbb{CR}^{k'}(b)$ for some k and k', and since a and b were assumed to be convertible within $\rightarrow_{n(i)}$, so are $\mathbb{CR}^k(a)$ and $\mathbb{CR}^{k'}(b)$ and we conclude per construction of $\mathbb{CR}$ that $\mathbb{CR}^{k+1}(a) = \mathbb{CR}^{k'+1}(b)$. Otherwise, either sequence reaches the greatest object of some strongly connected component, which eventually will be reached by the other sequence of steps as well.

Remark 8. The strategy $\mathbb{CR}$ only avoids cycles which do have 'an exit'; a cycle without an exit (a strongly connected component) need not be avoided as it will be a common reduct (the unique sink of its connected component) due to the Church–Rosser property.

As a corollary of Theorem 10 we also have a computable cofinal and normalizing many-step strategy for this class of systems. Kennaway's theorem delivers a computable normalizing *one-step* strategy for the sub-class of almost orthogonal fully extended combinatory reduction systems (CRSs).

7 Cofinal strategies

Using that CR-strategies are a fortiori cofinal, we have already some corollaries about the existence of cofinal strategies from the previous sections.

What we do not have as a corollary is the following noteworthy theorem:

Theorem 11 ([14]). *There exists a computable cofinal one-step strategy for Combinatory Logic.*

Also for CL, the question whether there is a computable one-step CR-strategy seems to be open.

In [15], Statman's strategy is given and shown to be cofinal in the setting of orthogonal TRSs. As remarked there, an extension to the higher-order case is not immediate, because the proof of cofinality essentially uses the fact that residuals of parallel redexes remain parallel, which is not true for the higher-order case. An extension to the weakly orthogonal case seems possible.

8 Concluding remarks

The reduction strategies studied here are memory-free, or history-free. It will be interesting to study strategies with some form of memory. An abstract initial approach to add memory is described in [15, p.480, Definition 9.1.6], there called 'history-aware'.

References

1. S. Antoy and A. Middeldorp. A sequential reduction strategy. *Theoretical Computer Science*, 165(1):75–95, 1996.
2. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
3. H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics.* North-Holland, 2nd revised edition edition, 1984.
4. H.P. Barendregt, J.A. Bergstra, J.W. Klop, and H. Volken. Degrees, reductions and representability in the lambda calculus. Technical Report 22, Utrecht University, February 1976.
5. J.A. Bergstra and J.W. Klop. Church-Rosser strategies in the lambda calculus. *Theoretical Computer Science*, 9(1):27–38, 1979.
6. J.R. Kennaway. Sequential evaluation strategies for parallel-or and related reduction systems. *Annals of Pure and Applied Logic*, 43:31–56, 1989.
7. J. Ketema, J.W. Klop, and V. van Oostrom. Vicious circles in orthogonal term rewriting systems. In S. Antoy and Y. Toyama, editors, *Proceedings of the 4th International Workshop on Reduction Strategies in Rewriting and Programming (WRS'04)*, volume 124 of *Electronic Notes in Theoretical Computer Science*, pages 65–77. Elsevier Science, 2005.
8. J. Ketema, J.W. Klop, and V. van Oostrom. Vicious circles in rewriting systems. In R. DiCosmo and Y. Toyama, editors, *Proceedings of the 5th International Workshop on Reduction Strategies in Rewriting and Programming (WRS'05)*, 2005.

9. J.W. Klop. Reduction cycles in Combinatory Logic. In J.R. Hindley and J.P. Seldin, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 193–214. Academic Press, 1980. Also available via `http://web.mac.com/janwillemklop/iWeb/Site/Bibliography.html`.
10. A. Middeldorp and H. Ohsaki. Type introduction for equational rewriting. *Acta Informatica*, 36(12):1007–1029, 2000.
11. T. Nipkow. Higher-order critical pairs. In *Proceedings of the 6th annual IEEE Symposium on Logic in Computer Science (LICS '91)*, pages 342–349, Amsterdam, The Netherlands, July 1991.
12. M.J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *LNCS*. Springer Verlag, 1977.
13. V. van Oostrom. Normalisation in weakly orthogonal rewriting. In P. Narendran and M. Rusinowitch, editors, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA '99)*, volume 1631 of *LNCS*, pages 60–74, Trento, Italy, July 1999. Springer Verlag.
14. R. Statman. Effective reduction and conversion strategies for combinators. In H. Comon, editor, *Proceedings of the 8th International Conference on Rewriting Techniques and Applications (RTA '97)*, volume 1232 of *LNCS*, pages 203–213, Sitges, Spain, June 1997. Springer Verlag.
15. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2003.
16. Y. Toyama. Reduction strategies for left-linear term rewriting systems. In *Processes, Terms and Cycles: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday*, volume 3838 of *LNCS*, pages 198–223. Springer Verlag, December 2005.
17. J. Waldmann. The combinator *S*. *Information and Computation*, 159:2–21, 2000.

A Proof of Modularity of Acyclicity

In this appendix we give a proof of Theorem 5 above. To that end, we first establish a lemma on standard reductions which is interesting in its own right.

Lemma 1 (Standard Prefix). *In an orthogonal TRS, the collection of standard reductions ending in a head step, is totally ordered by the prefix relation.*

Proof. In this proof we freely make use of Huet and Lévy's theory of standard/external reductions, for which we refer the reader to Sections 8.5 and 9.2.3 of [15].

First note that any standard reduction ending in a head step can be uniquely decomposed into a number of standard reductions such that *only* their final step is a head step. Thus to prove the lemma, it suffices to show that for any term there is *at most one* reduction of the latter type. We claim that a standard reduction of which only the final step is a head step is in fact an external reduction. The result then follows since supposing ρ and θ would be distinct such reductions from t, we may assume w.l.o.g. that they already differ in their first steps, say ϕ and ψ, and by totality of the textual order, we may assume w.l.o.g. that ϕ is to the left of ψ. Therefore, by externality of ϕ and standardness of θ, ϕ must have a unique residual up to the final step of θ. But that final step is a head step, so clearly it nests the residual of ϕ, contradicting externality of ϕ.

We prove the claim that any standard reduction of which only the final step is a head step, is an external reduction by induction on the lengh of the reduction. If the length is 1, it is trivial. Otherwise, we may write the reduction as $\phi \cdot \rho$ for some step ϕ. Since any suffix of a standard reduction is standard, the induction hypothesis yields that ρ is an external reduction.

For a proof by contradiction, suppose that ϕ contracting a redex-pattern at position p were not external. That is, a reduction θ co-initial to ρ would exist, consisting of steps disjoint from p and ending in a term allowing a step ψ at position q nesting p. By standardness of $\phi \cdot \rho$, the position p is in the redex-pattern of the first step above p in ρ (if any). As ρ ends in a head step such a step indeed exists, say it is ϕ' at position p' above p, and let ρ' be the prefix of ρ up to ϕ'.

Now consider the projections $\underline{\theta}$ of θ over ρ', and $\underline{\rho'}$ of ρ' over θ. Since by construction neither θ nor ρ' contracts redex-patterns on the path from the root to p, neither do their projections, hence the common reduct contains unique residuals of both ψ after $\underline{\rho'}$ and of ϕ' after $\underline{\theta}$, respectively at positions q and p' above p. Since the positions above p are totally ordered by the prefix relation, either of q and p' is above the other. We prove that neither is possible.

If q is properly above p', then the reduction $\underline{\theta}$ disjoint from p' and ending in a term containing a redex-pattern nesting p', shows that ϕ' is not external, contradicting the induction hypothesis. If p' is above q, then since q is above p and the redex-pattern at p' overlaps p, the redex-patterns of ϕ' and ψ must have overlap in the common reduct of $\overline{\theta}$ and $\overline{\rho'}$, contradicting orthogonality. $\square$

Remark 9. Although the lemma and its proof go through for higher-order rewriting systems, we have refrained from stating these since the next theorem doesn't generalise to the higher-order case (the result fails as shown in this paper, and its proof fails in it being based on the notion of rank).

Theorem 12. *Acyclicity is modular for orthogonal TRSs.*

Proof. In this proof we freely make use of the notions of modularity and tracing, for which we refer the reader to Sections 5.7.1 and 8.6.1 of [15].

Let $\mathcal{R}_b \uplus \mathcal{R}_w$ be the disjoint union of the orthogonal TRSs $\mathcal{R}_b$ and $\mathcal{R}_w$. To prove that acyclicity is a modular property is to prove that the underlying rewrite system $\rightarrow_{\mathcal{R}_b \uplus \mathcal{R}_w}$, which we will abbreviate to $\rightarrow$, is acyclic if both $\rightarrow_{\mathcal{R}_b}$ and $\rightarrow_{\mathcal{R}_w}$ are. For a proof by contradiction, assume that $\rightarrow_{\mathcal{R}_b \uplus \mathcal{R}_w}$ would allow a non-empty cycle σ on some term t, which we may w.l.o.g. assume to be of minimal rank. Since the rewrite systems $\rightarrow_{\mathcal{R}_b}$ and $\rightarrow_{\mathcal{R}_w}$ are acyclic by assumption, the rank of t must be positive, say it is $n + 1$. By minimality, at least a single step in σ must contract a redex-pattern in the top layer. Finally, w.l.o.g. we may assume t to have a minimal number, say $m + 1$, of principal subterms of maximal rank, i.e. of rank n.

Since rewriting does not increase the rank, the fact that σ is a cycle entails that the rank of all the terms along σ must be $n + 1$, so none of them has a principal subterm of rank greater than n. Now, let $\boldsymbol{p}$ be the vector of positions of principal subterms of maximal rank in t. We claim that for some index i and some positive k, p_i is its own origin when tracing p_i *back* along the k-fold repetition σ^k of σ. The claim holds true by the Pigeon Hole Principle and the fact that a principal subterm of maximal rank has another such subterm as origin.[6] Let s be the principal subterm of maximal rank at the position p_i given by the claim. We show that from t we can obtain a term t' which also allows a non-empty cycle but has at most m principal subterms of rank n, yielding a contradiction. The term t' is obtained from t by replacing a number of principal subterms, the subterm s at position p_i inclusive, by a term s' the rank of which is less than that of s.

The replacement term s' is defined as follows. If s allows some reduction having a destructive step in its top layer, then as standardisation preserves this and a destructive step in the top layer is a head step, the Standard Prefix Lemma yields some standard reduction ρ from s ending in a destructive step, which is least among such in the prefix order, and we let s' be the target of ρ. Otherwise, we let s' be a fresh variable. Either way, the rank of s' is less than the rank of s.

To see which principal subterms, other than s at position p_i, of t are to be replaced by s', we proceed as follows. Consider tracing p_i *forward* along an infinite repetition of σ^k, where we only let a position trace as long as it is the position of a principal subterm.[7] Then we let $\boldsymbol{p'}$ be the collection of all descendants which occur in t after some repetititon of σ^k, and let t' be obtained from t by replacing all subterms at positions in $\boldsymbol{p'}$ by s'. Note that by the above, p_i itself is among the $\boldsymbol{p'}$, and that by construction the subterms of t at positions in $\boldsymbol{p'}$ are reachable from s.

Next, we show the non-empty cycle σ^k on t can be *simulated* by a non-empty cycle σ' on t', by simulating each step $\phi : u \to v$ by a *reduction* $\phi' : u' \twoheadrightarrow v'$ depending on the relative positions of the redex-pattern contracted in ϕ and the (pairwise disjoint) descendants of $\boldsymbol{p'}$ in u. The invariant is that u' is obtained from u by replacing all subterms at positions of descendants of $\boldsymbol{p'}$ by s'.

- If ϕ contracts a redex-pattern *outside*, i.e. in the context of, the descendants of $\boldsymbol{p'}$ in u, then we let $\phi' : u' \to v'$ be obtained by contracting the same redex-pattern in u'.
- If ϕ contracts a redex-pattern *inside* some descendant of $\boldsymbol{p'}$ and ϕ is not destructive at its top layer, then we let $\phi' : u' \twoheadrightarrow u'$ be the empty reduction.
- If ϕ is a destructive step at a descendant p of $\boldsymbol{p'}$, then the subterm $v_{|p}$ is reachable from s as noted above, hence per construction of s' and the Standard Prefix Lemma, there also exists some reduction from s to $v_{|p}$ *via* s', thus using $u' = u'[s']_p$ we may set $\phi' : u'[s']_p \twoheadrightarrow u'[v_{|p}]_p$.

That σ' is non-empty follows from the fact that σ contains at least one step in its top layer, which will be simulated by exactly one step in σ' according to the

[6] Note that the claim need not hold when fixing k to 1. For instance, σ might swap two principal subterms (setting k to 2 then works).

[7] Per construction positions trace statically; a redex-pattern overlapping one would be polychrome *quod non*.

first item of the simulation. To show that σ' is a cycle, it suffices to show that each position p among $\boldsymbol{p'}$ in t traces *back* along σ^k to some position in that set again, which is trivial per construction of the set. $\square$

B Proof of Acyclicity of a PRS

Lemma 2. *The PRS consisting of the following rule is acyclic:*

$$f(xyz.Z(x,y,z),W,V) \to Z(W,Z(V,W,f(xyz.Z(x,y,z),W,V)),V)$$

Proof. By contradiction. To that end, suppose a non-empty cycle σ on t were to exist. W.l.o.g. we may assume such a t to be of minimal size among all terms admitting a non-empty cycle. Moreover, we may assume by the Standardisation Theorem for left-linear PRSs, that σ is standard.

We will employ the notion of *gripping* path due to Melliès. Say a position of an f-symbol in a term *grips* any position of an f-symbol which has a variable bound by the binders of the former, below it.

Note that a position only can grip positions inside its first argument, as only that argument has (three) binders. For instance, if we were to have a term $f^1(xyz.f^2(x'y'z'.a, f^3(x''y''z''.a,x,a),a,a),a,f^4(x'y'z'.a,a,a))$, where we have labelled the f-symbols for easy reference, then f^1 grips both f^2 and f^3 as both contain the variable x bound by f^1. Note that f^2 does not grip f^3.

Then a gripping path is a path w.r.t.. the gripping relation; in the example term there are just two non-trivial paths $f^1 f^2$ and $f^1 f^3$. Note that gripping paths are finite since successive positions are properly ordered by prefix. The property to be exploited is that gripping paths are preserved under taking their (position-wise) origin along any rewrite step, where we take as origin/descendant relation the usual (static) one [15], except that as origin of (all copies of) the f displayed in the right-hand side the head-f of the left-hand side is taken. The point is that if the path in the target of a step has some position below the contracted redex—which is the only interesting case—then the first such must either be the position of the head-symbol of the copy of the redex, or be in one of the copies of (the terms substituted for) the meta-variables in the rhs. In either case, the path must proceed completely inside that copy, by the variable convention. But then a corresponding path exists in the source of the step through the corresponding copy.

Now consider the origin of an arbitrary gripping path from the root of t along σ. This origin is again a gripping path in t by the above and by σ being a cycle. Distinguish cases depending on whether its first position, say p, is on some gripping path from the root of t or not.

If it is not, then consider the position q which is closest to the root and above p such that it is not on some gripping path from the root. Per construction, the subterm headed by q, say t', contains no variable bound by an f-symbol above it. Since p descends, also per construction, to the root of t along σ which was assumed to be standard, it follows that t reduces to t' somewhere along σ. Thus there would be a cylce on t' as well, contradicting minimality of t.

If it is, then a further case distinction is made depending on whether p is itself the root or it isn't.

If p is not the root, a contradiction is obtained since then for any gripping path from the root in t, one obtains a longer such path by prefixing its origin along σ by the (non-empty) gripping path from the root to p.

If p is the root, then first note that by the minimality assumption for t, σ must contain some head step. By standardness and the shape of the lhs—which is a pattern which cannot be created—also the first step of σ must be a head step, i.e.

$$\sigma : f([xyz]t', s, u) \to t'[x, y, z := s, t'[x, y, z := u, s, t], u] \twoheadrightarrow f([xyz]t', s, u)$$

and the root descends to the head f-symbol of a fresh copy of t in the rhs. Both y and z must occur in t' otherwise t would be erased. Therefore. by reasoning as above, the (instantiated) copies of t' above the f-symbol must be collapsable to their second respectively third arguments, depending on the respective substitutions s and u for its first argument. However, neither s nor u contains a variable bound outside it, and thus a collapsing reduction from t' could not depend on them. Contradiction. $\square$