

On the Complexity of Deciding Call-by-Need^{*}

Irène Durand^{1**} and Aart Middeldorp²

¹ Université de Bordeaux I
33405 Talence, France

`idurand@labri.u-bordeaux.fr`

² Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305-8573, Japan
`ami@is.tsukuba.ac.jp`

Abstract. In a recent paper we introduced a new framework for the study of call by need computations to normal form and root-stable form in term rewriting. Using elementary tree automata techniques and ground tree transducers we obtained simple decidability proofs for classes of rewrite systems that are much larger than earlier classes defined using the complicated sequentiality concept. In this paper we show that we can do without ground tree transducers in order to arrive at decidability proofs that are phrased in direct tree automata constructions. This allows us to derive better complexity bounds.

1 Introduction

The seminal work of Huet and Lévy [11] on optimal normalizing reduction strategies for orthogonal rewrite systems marks the beginning of the quest for decidable subclasses of (orthogonal) rewrite systems that admit a computable call by need strategy for deriving normal forms. Call by need means that the strategy may only contract *needed* redexes, i.e., redexes that are contracted in every normalizing rewrite sequence. Huet and Lévy showed that for the class of orthogonal rewrite systems every term not in normal form contains a needed redex and repeated contraction of needed redexes results in a normal form if the term under consideration has a normal form. However, neededness is in general undecidable. In order to obtain a decidable approximation to neededness Huet and Lévy introduced in the second part of [11] the subclass of *strongly sequential* systems. In a strongly sequential system at least one of the needed redexes in every reducible term can be effectively computed. Moreover, Huet and Lévy showed that strong sequentiality is a decidable property of orthogonal rewrite systems.

Strong sequentiality is determined by the left-hand sides of a rewrite system. By incorporating information of the right-hand sides, Oyamaguchi [17] showed that the class of strongly sequential systems can be enlarged without losing its good properties. The resulting class of NV-sequential systems was slightly

^{*} Submission category: regular research paper.

^{**} Corresponding author; phone: +33 556 846 914, fax: +33 556 846 669

extended by Nagaya *et al.* [16]. Comon [2] connected sequentiality notions with tree automata techniques, resulting in much shorter decidability proofs for larger classes of rewrite systems. Jacquemard [13] built upon the work of Comon. His class of growing-sequential rewrite systems extends all previously defined classes while still being decidable.

In a previous paper (Durand and Middeldorp [9]) we presented a new framework for decidable call by need. This framework, which we briefly recall in the next section, is simpler because complicated notions like sequentiality and index are avoided and *hence* more powerful. Moreover, we showed how to eliminate the difficult connection between tree automata and definability in weak second-order monadic logic in [2,13] by assigning a greater role to the concept of ground tree transducer (GTT, [6]). In this paper we show that we can do without GTTs as well.

Not much is known about the complexity of the problem of deciding membership in one of the classes that guarantees a computable call by need strategy to normal form. Comon [2] showed that strong sequentiality of a left-linear rewrite system can be decided in exponential time. Moreover, for left-linear rewrite systems satisfying the additional syntactic condition that whenever two proper subterms of left-hand sides are unifiable one of them matches the other, strong sequentiality can be decided in polynomial time. The class of forward-branching systems (Strandh [19]), a proper subclass of the class of orthogonal strongly sequential systems, coincides with the class of transitive systems (Toyama *et al.* [20]) and can be decided in quadratic time (Durand [8]). For classes higher in the hierarchy no non-elementary upperbounds are known, although Oyama-guchi [18] believes that the time complexity of his algorithm for deciding NV-sequentiality is at least double exponential.

In this paper we obtain a double exponential upperbound for the problem of deciding whether a left-linear rewrite system belongs to CBN-NF_g , the largest class in the hierarchy of [9]. This is better than the complexity of the decision procedure in [9] which, using the analysis presented in this paper, is at least triple exponential in the size of the rewrite system and much better than the non-elementary upperbound that is obtained via the weak second-order monadic logic connection.

The remainder of the paper is organized as follows. In the next section we briefly recall the framework of our earlier paper [9] for analyzing call by need computations in term rewriting. In Section 3 we show that $(\rightarrow_{\mathcal{R}}^*)[T]$ of ground terms that rewrite to a term in T is recognizable for every linear growing TRS \mathcal{R} and recognizable T . This result, essentially originating from [2] and [13], forms the basis of the explicit construction that we present in Section 4 of a tree automaton that decides whether a left-linear TRS belongs to CBN-NF_g . Section 4 also contains an example illustrating the various constructions. The complexity of the construction is analyzed in the next section. In Section 6 we consider call by need computations to root-stable form. As argued in Middeldorp [15], root-stable forms and root-neededness are the proper generalizations of normal forms and neededness when it comes to infinitary normalization. In this case we again

obtain a double exponential upperbound, which is a significant improvement over the non-elementary upperbound of the complexity of the decision procedure presented in [9].

2 Preliminaries

We assume the reader is familiar with the basics of term rewriting ([1,7,14]) and tree automata ([4,10]). We recall the following definitions from [9]. We refer to the latter paper for motivation and examples.

Let \mathcal{R} be a TRS over a signature \mathcal{F} . The sets of ground redexes, ground normal forms, and root-stable forms of \mathcal{R} are denoted by $\text{REDEX}_{\mathcal{R}}$, $\text{NF}_{\mathcal{R}}$, and $\text{RS}_{\mathcal{R}}$. Let \mathcal{R}_{\bullet} be the TRS $\mathcal{R} \cup \{\bullet \rightarrow \bullet\}$ over the extended signature $\mathcal{G} = \mathcal{F} \cup \{\bullet\}$. We say that redex Δ in $C[\Delta] \in \mathcal{T}(\mathcal{F})$ is \mathcal{R} -needed if there is no term $t \in \text{NF}_{\mathcal{R}_{\bullet}}$ such that $C[\Delta] \rightarrow_{\mathcal{R}}^* t$. For orthogonal TRSs \mathcal{R} -neededness coincides with neededness.

Let \mathcal{R} and \mathcal{S} be TRSs over the same signature. We say that \mathcal{S} approximates \mathcal{R} if $\rightarrow_{\mathcal{R}}^* \subseteq \rightarrow_{\mathcal{S}}^*$ and $\text{NF}_{\mathcal{R}} = \text{NF}_{\mathcal{S}}$. Next we define the approximations \mathcal{R}_s , \mathcal{R}_{nv} , and \mathcal{R}_g of a TRS \mathcal{R} . The TRS \mathcal{R}_s is obtained from \mathcal{R} by replacing the right-hand side of every rewrite rule by a variable that does not occur in the corresponding left-hand side. The TRS \mathcal{R}_{nv} is obtained from \mathcal{R} by replacing the variables in the right-hand sides of the rewrite rules by pairwise distinct variables that do not occur in the corresponding left-hand sides. A TRS \mathcal{R} is called growing if for every rewrite rule $l \rightarrow r \in \mathcal{R}$ the variables in $\text{Var}(l) \cap \text{Var}(r)$ occur at depth 1 in l . We define \mathcal{R}_g as any right-linear growing TRS that is obtained from \mathcal{R} by replacing variables in the right-hand sides of the rewrite rules by variables that do not occur in the corresponding left-hand sides.

An approximation mapping is a mapping α from TRSs to TRSs with the property that $\alpha(\mathcal{R})$ approximates \mathcal{R} , for every TRS \mathcal{R} . In the following we write \mathcal{R}_{α} instead of $\alpha(\mathcal{R})$. The class of TRSs \mathcal{R} such that every reducible term in $\mathcal{T}(\mathcal{F})$ has an \mathcal{R}_{α} -needed redex is denoted by CBN-NF_{α} . The above definitions of \mathcal{R}_s , \mathcal{R}_{nv} , and \mathcal{R}_g induce approximation mappings s , nv , and g . (We don't consider shallow approximations in this paper since they are a restricted case of growing approximations without resulting in any obvious lower complexity.) It is known that CBN-NF_s coincides with the class of strongly sequential TRSs but CBN-NF_{nv} and CBN-NF_g are much larger than the corresponding classes based on the sequentiality concept, see [9].

Let \mathcal{R} be a TRS over a signature \mathcal{F} . Let \mathcal{R}° be the TRS $\mathcal{R} \cup \{l^{\circ} \rightarrow r \mid l \rightarrow r \in \mathcal{R}\}$ over the extended signature $\mathcal{G} = \mathcal{F} \cup \{f^{\circ} \mid f \in \mathcal{F}\}$. Here l° denotes the term $f^{\circ}(l_1, \dots, l_n)$ for $l = f(l_1, \dots, l_n)$. For TRSs \mathcal{R} and \mathcal{S} over the same signature \mathcal{F} , we say that redex Δ in $C[\Delta] \in \mathcal{T}(\mathcal{F})$ is $(\mathcal{R}, \mathcal{S})$ -root-needed if there is no term $t \in \text{RS}_{\mathcal{S}^{\circ}}$ such that $C[\Delta^{\circ}] \rightarrow_{\mathcal{R}}^* t$. Let α and β be approximation mappings. The class of TRSs \mathcal{R} such that every non- \mathcal{R}_{β} -root-stable term in $\mathcal{T}(\mathcal{F})$ has an $(\mathcal{R}_{\alpha}, \mathcal{R}_{\beta})$ -root-needed redex is denoted by $\text{CBN-RS}_{\alpha, \beta}$.

3 Basic Construction

We consider finite bottom-up tree automata without ϵ -transitions. Let \mathcal{R} be a linear growing TRS over a signature \mathcal{F} . Let $T \subseteq \mathcal{T}(\mathcal{G})$ be a recognizable tree language with $\mathcal{F} \subseteq \mathcal{G}$ and $\mathcal{A}_T = (\mathcal{G}, Q_{\mathcal{A}}, Q_f, \Gamma_{\mathcal{A}})$ a tree automaton that recognizes T . We assume without loss of generality that all states of \mathcal{A}_T are accessible.

The goal of this section is to construct a tree automaton that recognizes the set $(\rightarrow_{\mathcal{R}}^*)[T]$ of ground terms that rewrite to a term in T . This result is not very new (Jacquemard [13] gives the construction for $T = \text{NF}_{\mathcal{R}}$ and Comon [3] for arbitrary T and shallow \mathcal{R}), but our presentation of the proof is a bit crisper. More importantly, we need the details of the construction for further analysis in subsequent sections.

3.1 Step 1

Let $A_{\mathcal{R}}$ be the set of arguments of the left-hand sides of \mathcal{R} and let $S_{\mathcal{R}}$ be the set of all subterms of terms in $A_{\mathcal{R}}$. Construct the tree automaton $\mathcal{B}(\mathcal{R}) = (\mathcal{G}, Q_{\mathcal{B}}, \emptyset, \Gamma_{\mathcal{B}})$ with $Q_{\mathcal{B}} = \{\langle t \rangle \mid t \in S_{\mathcal{R}}\} \cup \{\langle x \rangle\}$ and $\Gamma_{\mathcal{B}}$ consisting of the *matching* rules $f(\langle t_1 \rangle, \dots, \langle t_n \rangle) \rightarrow \langle t \rangle$ for every term $t = f(t_1, \dots, t_n)$ in $S_{\mathcal{R}}$ and *propagation* rules $f(\langle x \rangle, \dots, \langle x \rangle) \rightarrow \langle x \rangle$ for every $f \in \mathcal{G}$. Here $\langle t \rangle$ denotes the equivalence class of the term t with respect to literal similarity. (So we identify $\langle s \rangle$ and $\langle t \rangle$ whenever s and t differ a variable renaming.) Note that all states of $\mathcal{B}(\mathcal{R})$ are accessible. From now on we write \mathcal{B} for $\mathcal{B}(\mathcal{R})$ when the TRS \mathcal{R} can be inferred from the context. The set of ground instances of a term t is denoted by $\Sigma(t)$.

Lemma 1. *Let $t \in S_{\mathcal{R}} \cup \{x\}$. We have $s \in \Sigma(t)$ if and only if $s \rightarrow_{\mathcal{B}}^* \langle t \rangle$.* \square

3.2 Step 2

We assume that $\{t \mid t \rightarrow_{\mathcal{A}}^+ q\} = \{t \mid t \rightarrow_{\mathcal{B}}^+ q\}$ for every state $q \in Q_{\mathcal{A}} \cap Q_{\mathcal{B}}$. This can always be achieved by a renaming of states. Let $\mathcal{C}_T(\mathcal{R}) = (\mathcal{G}, Q, Q_f, \Gamma)$ be the union of \mathcal{A}_T and $\mathcal{B}(\mathcal{R})$, so $Q = Q_{\mathcal{A}} \cup Q_{\mathcal{B}}$ and $\Gamma = \Gamma_{\mathcal{A}} \cup \Gamma_{\mathcal{B}}$.

Lemma 2.

1. *Let $t \in S_{\mathcal{R}} \cup \{x\}$. We have $s \in \Sigma(t)$ if and only if $s \rightarrow_{\mathcal{C}_T(\mathcal{R})}^* \langle t \rangle$.*
2. *$L(\mathcal{C}_T(\mathcal{R})) = T$.*

\square

3.3 Step 3

We saturate the transition rules Γ of $\mathcal{C}_T(\mathcal{R})$ under the following inference rule:

$$\frac{f(l_1, \dots, l_n) \rightarrow r \in \mathcal{R} \quad r\theta \rightarrow_{\Gamma}^* q}{\Gamma = \Gamma \cup \{f(q_1, \dots, q_n) \rightarrow q\}} \quad (*)$$

with θ mapping the variables in r to states in Q and

$$q_i = \begin{cases} l_i\theta & \text{if } l_i \in \mathcal{V}\text{ar}(r), \\ \langle l_i \rangle & \text{otherwise.} \end{cases}$$

Because Q is finite and no new state is added by $(*)$, the saturation process terminates. We claim that $L(\mathcal{C}_T(\mathcal{R})) = (\rightarrow_{\mathcal{R}}^*)[T]$ upon termination.

Lemma 3. $(\rightarrow_{\mathcal{R}}^*)[T] \subseteq L(\mathcal{C}_T(\mathcal{R}))$.

Proof. Let $s \in (\rightarrow_{\mathcal{R}}^*)[T]$. So there exists a term $t \in T$ such that $s \rightarrow_{\mathcal{R}}^* t$. We show that $s \in L(\mathcal{C}_T(\mathcal{R}))$ by induction on the length of $s \rightarrow_{\mathcal{R}}^* t$. If $s = t$ then $s \in T \subseteq L(\mathcal{C}_T(\mathcal{R}))$ according to Lemma 2(2). Let $s = C[l\sigma] \rightarrow_{\mathcal{R}} C[r\sigma] \rightarrow_{\mathcal{R}}^* t$ with $l = f(l_1, \dots, l_n)$. The induction hypothesis yields $C[r\sigma] \in L(\mathcal{C}_T(\mathcal{R}))$. Hence there exists a final state q_f , a mapping θ from $\mathcal{V}\text{ar}(r)$ to Q , and a state q such that $C[r\sigma] \rightarrow_{\Gamma}^* C[r\theta] \rightarrow_{\Gamma}^* C[q] \rightarrow_{\Gamma}^* q_f$. By construction there exists a transition rule $f(q_1, \dots, q_n) \rightarrow q \in \Gamma$ such that $q_i = l_i\theta$ if $l_i \in \mathcal{V}\text{ar}(r)$ and $q_i = \langle l_i \rangle$ otherwise. We claim that $l\sigma \rightarrow_{\Gamma}^* f(q_1, \dots, q_n)$. Let $i \in \{1, \dots, n\}$. If $l_i \in \mathcal{V}\text{ar}(r)$ then $l_i\sigma \rightarrow_{\Gamma}^* l_i\theta = q_i$, otherwise $l_i\sigma \rightarrow_{\Gamma}^* \langle l_i \rangle = q_i$ by Lemma 2(1). Consequently $s \rightarrow_{\Gamma}^* C[f(q_1, \dots, q_n)] \rightarrow_{\Gamma} C[q] \rightarrow_{\Gamma}^* q_f$ and hence $s \in L(\mathcal{C}_T(\mathcal{R}))$. \square

Note that we don't use the growing assumption in the above proof; right-linearity of \mathcal{R} is sufficient.

Lemma 4. Let $s \in \mathcal{T}(\mathcal{G})$ with $s \rightarrow_{\mathcal{C}_T(\mathcal{R})}^* q$.

1. If $q = \langle t \rangle$ with $t \in S_{\mathcal{R}} \cup \{x\}$ then $s \in (\rightarrow_{\mathcal{R}}^*)[\Sigma(t)]$.
2. If $q \in Q_f$ then $s \in (\rightarrow_{\mathcal{R}}^*)[T]$.

Proof. Let Γ_k denote the value of Γ after the k -th transition rule has been added by $(*)$. We have $s \rightarrow_{\Gamma_k}^* q$ for some $k \geq 0$. We prove statements (1) and (2) by induction on k . If $k = 0$ then the result follows from Lemma 2. Let $s \rightarrow_{\Gamma_{k+1}}^* q$. We use a second induction on the number of steps that use the (unique) transition rule $f(q_1, \dots, q_n) \rightarrow q' \in \Gamma_{k+1} \setminus \Gamma_k$. Suppose this rule is created from $l = f(l_1, \dots, l_n) \rightarrow r \in \mathcal{R}$ and $r\theta \rightarrow_{\Gamma_k}^* q'$. If this number is zero then the result follows from the first induction hypothesis. Otherwise we may write $s = C[f(s_1, \dots, s_n)] \rightarrow_{\Gamma_k}^* C[f(q_1, \dots, q_n)] \rightarrow C[q'] \rightarrow_{\Gamma_{k+1}}^* q$. We will define a substitution τ such that $s \rightarrow_{\mathcal{R}}^* C[l\tau] \rightarrow_{\mathcal{R}} C[r\tau] \rightarrow_{\Gamma_k}^+ C[q']$. The second induction hypothesis applied to $C[r\tau] \rightarrow_{\Gamma_{k+1}}^+ q$ then yields the desired result. We define τ as the (disjoint) union of $\tau_1, \dots, \tau_n, \tau'$ such that $\text{Dom}(\tau_i) = \mathcal{V}\text{ar}(l_i)$ for $i = 1, \dots, n$ and $\text{Dom}(\tau') = \mathcal{V}\text{ar}(r) \setminus \mathcal{V}\text{ar}(l)$. Note that since l is a linear term, the union of τ_1, \dots, τ_n is well-defined. Fix $i \in \{1, \dots, n\}$. If $l_i \in \mathcal{V}\text{ar}(r)$ then we let $\tau_i = \{l_i \mapsto s_i\}$. Otherwise $q_i = \langle l_i \rangle$ and thus $s_i \rightarrow_{\Gamma_k}^+ \langle l_i \rangle$. Part (1) of the first induction hypothesis yields $s_i \in (\rightarrow_{\mathcal{R}}^*)[\Sigma(l_i)]$. Hence there exists a substitution τ_i such that $s_i \rightarrow_{\mathcal{R}}^* l_i\tau_i$. We assume without loss of generality that $\text{Dom}(\tau_i) = \mathcal{V}\text{ar}(l_i)$. The substitution τ' is defined as $\{x \mapsto u_x \mid x \in \mathcal{V}\text{ar}(r) \setminus \mathcal{V}\text{ar}(l)\}$ where u_x is an arbitrary but fixed ground term such that $u_x \rightarrow_{\Gamma_0}^* x\theta$. (This is possible

because all states of Q are accessible.) It remains to show that $s \rightarrow_{\mathcal{R}}^* C[l\tau]$ and $C[r\tau] \rightarrow_{\Gamma_k}^+ C[q']$. The former is an immediate consequence of the definitions of τ_1, \dots, τ_n . For the latter it is sufficient to show that $C[r\tau] \rightarrow_{\Gamma_k}^* C[r\theta]$. Let $x \in \mathcal{Var}(r)$. If $x \in \mathcal{Var}(l)$ then, because \mathcal{R} is growing and left-linear, there is a unique $i \in \{1, \dots, n\}$ such that $x = l_i$. We have $x\tau = l_i\tau_i = s_i$ by construction of τ_i and $q_i = l_i\theta = x\theta$ by definition. Hence $x\tau = s_i \rightarrow_{\Gamma_k}^+ q_i = x\theta$ by assumption. If $x \notin \mathcal{Var}(l)$ then $x\tau = x\tau' \rightarrow_{\Gamma_0}^* x\theta$ by construction of τ' . This completes the proof. The induction step is summarized in the following diagram:

$$\begin{array}{ccccc}
 s & \xrightarrow[\Gamma_k]{*} & C[f(q_1, \dots, q_n)] & \xrightarrow{\Gamma_{k+1}} & C[q'] & \xrightarrow[\Gamma_{k+1}]{*} & q \\
 \mathcal{R} \downarrow^* & & & & \uparrow^* \Gamma_k & & \\
 C[l\tau] & \xrightarrow{\mathcal{R}} & C[r\tau] & \xrightarrow[\Gamma_k]{*} & C[r\theta] & &
 \end{array}$$

□

Corollary 1. $L(\mathcal{C}_T(\mathcal{R})) = (\rightarrow_{\mathcal{R}}^*)[T]$. □

As a side remark we mention that the result described above remains true if we drop the restriction that the left-hand side of a rewrite rule is a non-variable term; just add the following saturation rule:

$$\frac{x \rightarrow r \in \mathcal{R} \quad r\theta \rightarrow_{\Gamma}^* q}{\Gamma = \Gamma \cup \{q' \rightarrow q\}} \quad (**)$$

with θ mapping the variables in r to states in Q and

$$q' = \begin{cases} x\theta & \text{if } x \in \mathcal{Var}(r), \\ \langle x \rangle & \text{otherwise.} \end{cases}$$

Although this extension is useless when it comes to call-by-need (because no TRS that has a rewrite rule whose left-hand side is a single variable has normal forms), it is interesting to note that it generalizes Theorem 5.1 of Coquidé *et al.* [5]—the preservation of recognizability for linear semi-monadic rewrite systems.

For an example of the above constructions we refer to the end of the next section.

4 Call by Need Computations to Normal Form

In this section we assume that $\mathcal{G} = \mathcal{F} \cup \{\bullet\}$ and $T = \mathbf{NF}_{\mathcal{R}, \bullet} \subseteq \mathcal{T}(\mathcal{F})$, the set of ground normal forms of the linear growing TRS \mathcal{R} . Based on the automaton $\mathcal{C}_T(\mathcal{R})$ of the previous section we construct an automaton $\mathcal{D}(\mathcal{R})$ (or simply \mathcal{D}) that accepts all reducible terms in $\mathcal{T}(\mathcal{F})$ that do not have an \mathcal{R} -needed redex.

First we extend $\mathcal{C}_T(\mathcal{R})$ in such a way that it can be used to identify redexes and reducible terms with respect to \mathcal{R} . This is essentially achieved by adding a fresh copy of the automaton $\mathcal{B}(\mathcal{R})$ of Section 3.1. More precisely, we

add for every term $t \in S_{\mathcal{R}}$ a new state $\langle t \rangle'$ to Q as well as a distinguished state q_r together with the transition rules $f(\langle t_1 \rangle', \dots, \langle t_n \rangle') \rightarrow \langle t \rangle'$ for every term $t = f(t_1, \dots, t_n) \in S_{\mathcal{R}}$ and $f(\langle l_1 \rangle', \dots, \langle l_n \rangle') \rightarrow q_r$ for every left-hand side $f(l_1, \dots, l_n)$ of a rewrite rule in \mathcal{R} , where we identify $\langle x \rangle'$ with $\langle x \rangle$, and all rules of the form $f(\langle x \rangle, \dots, q_r, \dots, \langle x \rangle) \rightarrow q_r$. In state q_r all reducible terms in $\mathcal{T}(\mathcal{G})$ are accepted. Let us denote the new automaton by $\mathcal{C}'_T(\mathcal{R}) = (\mathcal{G}, Q', Q_f, \Gamma')$. Note that $\mathcal{C}_T(\mathcal{R})$ and $\mathcal{C}'_T(\mathcal{R})$ accept the same language $(\rightarrow_{\mathcal{R}}^*)[T]$.

The construction of \mathcal{D} is related to the construction in Comon [3, Lemma 31]. The signature of \mathcal{D} is \mathcal{F} . The set $Q_{\mathcal{D}}$ of states of \mathcal{D} consists of all pairs $[S, P]$ of subsets S, P of states in Q' . The final states all are pairs $[S, P]$ such that $q_r \in S$ and $P \subseteq Q_f$. Intuitively, the first component S of a state $[S, P]$ records all reachable states with respect to the automaton $\mathcal{C}'_T(\mathcal{R})$ and the second component P will be a subset of S that can be divided in non-empty sets P_p for every redex position p such that all states in P_p are reachable if the redex at position p is not contracted. This will be made precise in Lemmata 6 and 7 below.

Before defining the transition rules of \mathcal{D} we introduce the following abbreviations. Given a term $t \in \mathcal{T}(\mathcal{G} \cup Q_{\mathcal{D}})$, let $Q(t)$ denote the set of all terms in $\mathcal{T}(\mathcal{G} \cup Q')$ that are obtained from t by replacing every state in $Q_{\mathcal{D}}$ by one of its elements. We denote the set $\{q \in Q' \mid s \rightarrow_{\Gamma'}^* q \text{ and } s \in Q(t)\}$ by $t \downarrow$. The set $\Gamma_{\mathcal{D}}$ consists of all transition rules of the form

$$f([S_1, P_1], \dots, [S_n, P_n]) \rightarrow [S, P]$$

where f is an n -ary function symbol in \mathcal{F} , $S = f(S_1, \dots, S_n) \downarrow$, and P is the union of P^1 and P^2 where P^1 is a subset of

$$\bigcup_{i=1}^n f(S_1, \dots, P_i, \dots, S_n) \downarrow$$

with the property that for all $i \in \{1, \dots, n\}$ and $q_i \in P_i$

$$P^1 \cap f(S_1, \dots, \{q_i\}, \dots, S_n) \downarrow \neq \emptyset$$

and $P^2 = \{\langle x \rangle\}$ if there exists a rewrite rule in \mathcal{R} with left-hand side $f(l_1, \dots, l_n)$ such that $\langle l_i \rangle' \in S_i$ for all $i \in \{1, \dots, n\}$ and $P^2 = \emptyset$ otherwise. Note that the resulting automaton is non-deterministic due to the freedom of choosing P^1 . This is essential for the correctness of the construction.

Lemma 5. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \rightarrow_{\Gamma_{\mathcal{D}}}^* [S, P]$ then $S = s \downarrow$.*

Proof. Straightforward. \square

Lemma 6. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \rightarrow_{\Gamma_{\mathcal{D}}}^* [S, P]$ then for every redex position p in s there exists a state $q \in P$ such that $q \in s[\bullet]_p \downarrow$.*

Proof. Let $s|_p = l\sigma$ for some left-hand side $l = f(l_1, \dots, l_n)$ of a rewrite rule in \mathcal{R} . We have $l_i\sigma \rightarrow_{\Gamma'}^* \langle l_i \rangle'$ for all $i \in \{1, \dots, n\}$. We show the statement by induction on the depth of the position p . If $p = \varepsilon$ then $s = l\sigma$. We may write

$$s = f(l_1\sigma, \dots, l_n\sigma) \rightarrow_{\Gamma_{\mathcal{D}}}^* f([S_1, P_1], \dots, [S_n, P_n]) \rightarrow_{\Gamma_{\mathcal{D}}} [S, P].$$

According to Lemma 5 $\langle l_i \rangle' \in S_i$ for all $i \in \{1, \dots, n\}$. By construction $P = P^1 \cup P^2$ for some P^1 and $P^2 = \{\langle x \rangle\}$. It is sufficient to show that $s[\bullet]_p = \bullet \rightarrow_{\Gamma'}^* \langle x \rangle$, which is obvious since by construction every ground term in $\mathcal{T}(\mathcal{G})$ can be rewritten to $\langle x \rangle$. For the induction step we suppose that $p = i \cdot p'$ for some $i \in \{1, \dots, n\}$. Write $s = f(s_1, \dots, s_n) \rightarrow_{\Gamma_D}^* f([S_1, P_1], \dots, [S_n, P_n]) \rightarrow_{\Gamma_D} [S, P]$. The induction hypothesis yields $q' \in s_i[\bullet]_{p'} \downarrow$ for some $q' \in P_i$. By construction $P = P^1 \cup P^2$ where P^1 has a non-empty intersection with $f(S_1, \dots, \{q'\}, \dots, S_n) \downarrow$. Let q be a state in this intersection. By definition there exist states q_j for $j \neq i$ such that $f(q_1, \dots, q', \dots, q_n) \rightarrow_{\Gamma'} q$. Lemma 5 yields $s_j \rightarrow_{\Gamma'}^* q_j$ for all $j \neq i$. Hence $s[\bullet]_p = f(s_1, \dots, s_i[\bullet]_{p'}, \dots, s_n) \rightarrow_{\Gamma'}^* f(q_1, \dots, q', \dots, q_n) \rightarrow_{\Gamma'} q$ and thus $q \in s[\bullet]_p \downarrow$. \square

We denote the set of redex positions in a term s by $\mathcal{R}(s)$.

Lemma 7. *Let $s \in \mathcal{T}(\mathcal{F})$. If*

$$P \subseteq \bigcup_{p \in \mathcal{R}(s)} s[\bullet]_p \downarrow$$

such that $P \cap s[\bullet]_p \downarrow \neq \emptyset$ for all $p \in \mathcal{R}(s)$ then $s \rightarrow_{\Gamma_D}^ [S, P]$ where $S = s \downarrow$.*

Proof. Induction on the structure of s . Suppose $s = f(s_1, \dots, s_n)$ for some $n \geq 0$. Define

$$P' = P \cap \bigcup_{\varepsilon \neq p \in \mathcal{R}(s)} s[\bullet]_p \downarrow.$$

Fix $i \in \{1, \dots, n\}$ and define

$$P_i = \bigcup_{p' \in \mathcal{R}(s_i)} \{q' \in s_i[\bullet]_{p'} \downarrow \text{ such that } f(S_1, \dots, \{q'\}, \dots, S_n) \downarrow \cap P' \neq \emptyset\}.$$

Clearly

$$P_i \subseteq \bigcup_{p' \in \mathcal{R}(s_i)} s_i[\bullet]_{p'} \downarrow.$$

We claim that $P_i \cap s_i[\bullet]_{p'} \downarrow \neq \emptyset$ for all $p' \in \mathcal{R}(s_i)$. Let $p = i \cdot p'$. By assumption $P \cap s[\bullet]_p \downarrow \neq \emptyset$. We have $s[\bullet]_p \downarrow = f(S_1, \dots, s_i[\bullet]_{p'} \downarrow, \dots, S_n) \downarrow$, so there exists a state $q' \in s_i[\bullet]_{p'} \downarrow$ such that $f(S_1, \dots, \{q'\}, \dots, S_n) \downarrow \cap P \neq \emptyset$. This implies that $f(S_1, \dots, \{q'\}, \dots, S_n) \downarrow \cap P' \neq \emptyset$. Hence $q' \in P_i$ by definition. The induction hypothesis yields $s_i \rightarrow_{\Gamma_D}^* [S_i, P_i]$ with $S_i = s_i \downarrow$. Since this holds for every $i \in \{1, \dots, n\}$ we obtain

$$s \rightarrow_{\Gamma_D}^* f([S_1, P_1], \dots, [S_n, P_n]).$$

It suffices to show that $f([S_1, P_1], \dots, [S_n, P_n]) \rightarrow [S, P]$ is a transition rule of \mathcal{D} , which amounts to showing that $S = f(S_1, \dots, S_n) \downarrow$ and $P = P^1 \cup P^2$ with P^1 a subset of

$$\bigcup_{i=1}^n f(S_1, \dots, P_i, \dots, S_n) \downarrow$$

with the property that for all $i \in \{1, \dots, n\}$ and $q_i \in P_i$

$$P^1 \cap f(S_1, \dots, \{q_i\}, \dots, S_n) \downarrow \neq \emptyset$$

and $P^2 = \{\langle x \rangle\}$ if there exists a rewrite rule in \mathcal{R} with left-hand side $f(l_1, \dots, l_n)$ such that $\langle l_1 \rangle' \in S_1, \dots, \langle l_n \rangle' \in S_n$ and $P^2 = \emptyset$ otherwise. Clearly

$$f(S_1, \dots, S_n) \downarrow = f(s_1 \downarrow, \dots, s_n \downarrow) \downarrow = s \downarrow = S.$$

For P^1 we take P' . By the definition of P_1, \dots, P_n , for all $i \in \{1, \dots, n\}$ and $q_i \in P_i$

$$P' \cap f(S_1, \dots, \{q_i\}, \dots, S_n) \downarrow \neq \emptyset,$$

so P' satisfies the conditions for P^1 . If s is a redex then there exist a left-hand side $f(l_1, \dots, l_n)$ of a rewrite rule in \mathcal{R} and a substitution σ such that $s = f(l_1\sigma, \dots, l_n\sigma)$. We have $P = P' \cup \bullet \downarrow = P' \cup \{\langle x \rangle\}$ and $P^2 = \{\langle x \rangle\}$ as desired since according to Lemma 5 $\langle l_i \rangle' \in S_i$ for all $i \in \{1, \dots, n\}$. If s is not a redex then $P' = P$ and indeed $P^2 = \emptyset$ in this case since there does not exist a left-hand side $f(l_1, \dots, l_n)$ such that $\langle l_i \rangle' \in S_i$ for all $i \in \{1, \dots, n\}$. \square

Lemma 8. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \rightarrow_{\Gamma_D}^* [S, P]$ and $P \subseteq Q_f$ then, for every redex position p in s , $s[\bullet]_p \in (\rightarrow_{\mathcal{R}}^+)[\text{NF}_{\mathcal{R}_\bullet}]$.*

Proof. Let p be a redex position in s . By Lemma 6 there exists a state $q \in Q_f$ such that $s[\bullet]_p \rightarrow_{\Gamma'}^* q$. Hence $s[\bullet]_p \in L(\mathcal{C}'_{\text{NF}_{\mathcal{R}_\bullet}}) = (\rightarrow_{\mathcal{R}}^*)[\text{NF}_{\mathcal{R}_\bullet}]$. \square

Lemma 9. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s[\bullet]_p \in (\rightarrow_{\mathcal{R}}^*)[\text{NF}_{\mathcal{R}_\bullet}]$ for every redex position p in s then there exists a set $P \subseteq Q_f$ such that $s \rightarrow_{\Gamma_D}^* [S, P]$ with $S = s \downarrow$.*

Proof. Define

$$P = Q_f \cap \bigcup_{p \in \mathcal{R}(s)} s[\bullet]_p \downarrow.$$

Since $(\rightarrow_{\mathcal{R}}^*)[\text{NF}_{\mathcal{R}_\bullet}] = L(\mathcal{C}'_{\text{NF}_{\mathcal{R}_\bullet}})$, for every redex position p in s there exists a state $q \in Q_f$ such that $s[\bullet]_p \rightarrow_{\Gamma'}^* q$ and thus $q \in P$ by definition. Hence we can apply Lemma 7 which yields the desired result. \square

Corollary 2. *Let $s \in \mathcal{T}(\mathcal{F})$. We have $s \in L(\mathcal{D}(\mathcal{R}))$ if and only if s is reducible and $s[\bullet]_p \in (\rightarrow_{\mathcal{R}}^*)[\text{NF}_{\mathcal{R}_\bullet}]$ for every redex position p in s .*

Proof. By Lemmata 5, 8, 9, and the observation that s is reducible if and only if $q_r \in s \downarrow$. \square

Theorem 1. *Let \mathcal{R} be a left-linear TRS. We have $\mathcal{R} \in \text{CBN-NF}_g$ if and only if $L(\mathcal{D}(\mathcal{R}_g)) = \emptyset$.*

Proof. Note that \mathcal{R}_g is a linear growing TRS. By definition of CBN-NF_g , $\mathcal{R} \notin \text{CBN-NF}_g$ if and only if there exists a reducible term s in $\mathcal{T}(\mathcal{F})$ without \mathcal{R}_g -needed redexes. The latter is equivalent to $s[\bullet]_p \in (\rightarrow_{\mathcal{R}_g}^*)[\text{NF}_{\mathcal{R}_g}]$ for every redex position p in s . (Note that $\text{NF}_{\mathcal{R}_g}$ and $\text{NF}_{\mathcal{R}_g}$ coincide.) According to the preceding corollary this is equivalent to $s \in L(\mathcal{D}(\mathcal{R}_g))$. Hence $\mathcal{R} \in \text{CBN-NF}_g$ if and only if $L(\mathcal{D}(\mathcal{R}_g)) = \emptyset$. \square

Let us illustrate the construction on a small example. Consider the orthogonal growing TRS \mathcal{R} consisting of the rewrite rules

$$\begin{array}{ll} f(a, g(x, a)) \rightarrow b & f(x, a) \rightarrow x \\ f(b, g(a, x)) \rightarrow b & g(b, b) \rightarrow a \end{array}$$

The automaton $\mathcal{B}(\mathcal{R})$ has states $Q_{\mathcal{B}} = \{\langle x \rangle, \langle a \rangle, \langle b \rangle, \langle g(x, a) \rangle, \langle g(a, x) \rangle\}$ and transition rules $\Gamma_{\mathcal{B}}$:

$$\begin{array}{llll} a \rightarrow \langle x \rangle & \bullet \rightarrow \langle x \rangle & a \rightarrow \langle a \rangle & g(\langle x \rangle, \langle a \rangle) \rightarrow \langle g(x, a) \rangle \\ b \rightarrow \langle x \rangle & f(\langle x \rangle, \langle x \rangle) \rightarrow \langle x \rangle & b \rightarrow \langle b \rangle & g(\langle a \rangle, \langle x \rangle) \rightarrow \langle g(a, x) \rangle \end{array}$$

The set $\text{NF}_{\mathcal{R}_g}$ is for instance accepted by the automaton $\mathcal{A}_{\text{NF}_{\mathcal{R}_g}}$ with states $Q_{\mathcal{A}} = Q_f = \{\langle X \rangle, \langle a \rangle, \langle b \rangle, \langle G(X, a) \rangle, \langle G(a, X) \rangle, \langle G(a, a) \rangle\}$ and transition rules $\Gamma_{\mathcal{A}}$ consisting of $a \rightarrow \langle a \rangle, b \rightarrow \langle b \rangle,$

$$g(q, q') \rightarrow \begin{cases} \langle G(a, a) \rangle & \text{if } q = q' = \langle a \rangle \\ \langle G(a, X) \rangle & \text{if } q = \langle a \rangle \text{ and } q' \in Q_{\mathcal{A}} \setminus \{\langle a \rangle\} \\ \langle G(X, a) \rangle & \text{if } q \in Q_{\mathcal{A}} \setminus \{\langle a \rangle\} \text{ and } q' = \langle a \rangle \\ \langle X \rangle & \text{for all other cases except } q = q' = \langle b \rangle \end{cases}$$

and $f(q, q') \rightarrow \langle X \rangle$ for all pairs (q, q') except those in $Q_{\mathcal{A}} \times \{\langle a \rangle\}, \{\langle a \rangle\} \times \{\langle G(X, a) \rangle, \langle G(a, a) \rangle\}$, and $\{\langle b \rangle\} \times \{\langle G(a, X) \rangle, \langle G(a, a) \rangle\}$. Note that $\mathcal{A}_{\text{NF}_{\mathcal{R}_g}}$ and $\mathcal{B}(\mathcal{R})$ share states $\langle a \rangle$ and $\langle b \rangle$, which is allowed since both automata accept the same set of terms in those states ($\{a\}$ and $\{b\}$ respectively). Let $Q = Q_{\mathcal{A}} \cup Q_{\mathcal{B}}$ and $\Gamma = \Gamma_{\mathcal{A}} \cup \Gamma_{\mathcal{B}}$. Let us compute the saturation rules for the various approximations of \mathcal{R} . For \mathcal{R}_s we get

$$\begin{array}{ll} f(\langle a \rangle, \langle g(x, a) \rangle) \rightarrow q & f(\langle x \rangle, \langle a \rangle) \rightarrow q \\ f(\langle b \rangle, \langle g(a, x) \rangle) \rightarrow q & g(\langle b \rangle, \langle b \rangle) \rightarrow q \end{array}$$

for all states $q \in Q$. For \mathcal{R}_{nv} we obtain

$$\begin{array}{ll} f(\langle a \rangle, \langle g(x, a) \rangle) \rightarrow q & f(\langle x \rangle, \langle a \rangle) \rightarrow q_1 \\ f(\langle b \rangle, \langle g(a, x) \rangle) \rightarrow q & g(\langle b \rangle, \langle b \rangle) \rightarrow q_2 \end{array}$$

for all $q \in \{\langle x \rangle, \langle b \rangle\}, q_1 \in Q$, and $q_2 \in \{\langle x \rangle, \langle a \rangle\}$. Finally, \mathcal{R}_g gives rise to

$$\begin{array}{ll} f(\langle a \rangle, \langle g(x, a) \rangle) \rightarrow q & f(q_1, \langle a \rangle) \rightarrow q_1 \\ f(\langle b \rangle, \langle g(a, x) \rangle) \rightarrow q & g(\langle b \rangle, \langle b \rangle) \rightarrow q_2 \end{array}$$

for all $q \in \{\langle x \rangle, \langle b \rangle\}$, $q_1 \in Q$, and $q_2 \in \{\langle x \rangle, \langle a \rangle\}$. Let us denote the resulting set of transition rules of the automaton $\mathcal{C}_{\text{NF}_{\mathcal{R}_\bullet}}(\mathcal{R}_\alpha)$ for $\alpha \in \{s, \text{nv}, g\}$ by Γ_α . Note the difference between Γ_{nv} and Γ_g . We obtain $\mathcal{C}'_{\text{NF}_{\mathcal{R}_\bullet}}(\mathcal{R}_\alpha)$ from $\mathcal{C}_{\text{NF}_{\mathcal{R}_\bullet}}(\mathcal{R}_\alpha)$ by adding states $\{\langle a \rangle', \langle b \rangle', \langle g(x, a) \rangle', \langle g(a, x) \rangle', q_r\}$ and transition rules

$$\begin{array}{lll} a \rightarrow \langle a \rangle' & f(\langle a \rangle', \langle g(x, a) \rangle') \rightarrow q_r & f(\langle x \rangle, q_r) \rightarrow q_r \\ b \rightarrow \langle b \rangle' & f(\langle b \rangle', \langle g(a, x) \rangle') \rightarrow q_r & f(q_r, \langle x \rangle) \rightarrow q_r \\ g(\langle x \rangle, \langle a \rangle') \rightarrow \langle g(x, a) \rangle' & f(\langle x \rangle, \langle a \rangle') \rightarrow q_r & g(\langle x \rangle, q_r) \rightarrow q_r \\ g(\langle a \rangle', \langle x \rangle) \rightarrow \langle g(a, x) \rangle' & g(\langle b \rangle', \langle b \rangle') \rightarrow q_r & g(q_r, \langle x \rangle) \rightarrow q_r \end{array}$$

We will not attempt to present the automata $\mathcal{D}(\mathcal{R}_\alpha)$ in detail. Rather, we show a possible computation of $\mathcal{D}(\mathcal{R}_{\text{nv}})$ for the term $t = f(\Delta, g(\Delta, \Delta))$ with $\Delta = g(a, a)$. We have $a \rightarrow [\{\langle x \rangle, \langle a \rangle, \langle a \rangle'\}, \emptyset]$ and thus

$$\Delta \rightarrow^* f([\{\langle x \rangle, \langle a \rangle, \langle a \rangle'\}, \emptyset], [\{\langle x \rangle, \langle a \rangle, \langle a \rangle'\}, \emptyset]) \rightarrow [S_1, P_1]$$

with $S_1 = Q' \setminus \{\langle b \rangle', \langle g(x, a) \rangle', \langle g(a, x) \rangle'\}$ and $P_1 = \{\langle x \rangle\}$. Consequently,

$$g(\Delta, \Delta) \rightarrow^* g([S_1, P_1], [S_1, P_1]) \rightarrow [S_2, P_2]$$

with $S_2 = \{\langle x \rangle, \langle a \rangle, \langle g(x, a) \rangle, \langle g(a, x) \rangle, \langle X \rangle, \langle G(X, a) \rangle, \langle G(a, X) \rangle, \langle G(a, a) \rangle, q_r\}$ and $P_2 = \{\langle g(x, a) \rangle, \langle g(a, x) \rangle\}$. Note that there are other possibilities for P_2 . Finally

$$t \rightarrow^* f([S_1, P_1], [S_2, P_2]) \rightarrow [S, P]$$

with $S = t \downarrow_{\Gamma_{\text{nv}}}$ and $P = \{\langle X \rangle\}$. Since $q_r \in S$ and $\langle X \rangle \in Q_f$, $[S, P]$ is a final state of $\mathcal{D}(\mathcal{R}_{\text{nv}})$ and hence t does not have an \mathcal{R}_{nv} -needed redex and thus $\mathcal{R} \notin \text{CBN-NF}_{\text{nv}}$. The reader is invited to verify that for every computation $t \rightarrow^* [S, P]$ in $\mathcal{D}(\mathcal{R}_g)$ we have $\langle x \rangle \in P$ and thus t has an \mathcal{R}_g -needed redex (the occurrence of Δ at position 1). Actually, it turns out that $L(\mathcal{D}(\mathcal{R}_g)) = \emptyset$, so $\mathcal{R} \in \text{CBN-NF}_g$.

5 Complexity Analysis

In this section we analyze the complexity of the decision procedure of the previous section. Given a term t , we denote its size (i.e., its total number of symbols) by $|t|$. Given a TRS \mathcal{R} , we denote the number of rewrite rules it contains by $\#\mathcal{R}$ and its size (the sum of the sizes of the left and right-hand sides) by $|\mathcal{R}|$. We assume that the signature \mathcal{F} of a \mathcal{R} does not contain function symbols that do not appear in \mathcal{R} , except possibly for a constant (to make the set $\mathcal{T}(\mathcal{F})$ of ground terms non-empty). This entails no loss of generality. Let m be the maximum arity of function symbols in \mathcal{F} .

Given an automaton \mathcal{A} , we denote the number of transitions rules of \mathcal{A} by $|\mathcal{A}|$. It is well-known that the number of states of an automaton $\mathcal{A}_{\text{NF}_{\mathcal{R}_\bullet}}$ that accepts the ground normal forms in $\mathcal{T}(\mathcal{F})$ with respect to the TRS \mathcal{R} is in $\mathcal{O}(2^{|\mathcal{R}|})$. Hence $|\mathcal{A}|$ (from now on we drop the subscript $\text{NF}_{\mathcal{R}_\bullet}$) is in $\mathcal{O}(2^{\mathcal{O}(|\mathcal{R}|^2)})$:

For every function symbol in \mathcal{F} there can be at most $|Q_{\mathcal{A}}|^{m+1}$ transition rules hence $|\mathcal{A}|$ is in $\mathcal{O}(|\mathcal{F}| \cdot 2^{|\mathcal{R}|(m+1)})$ and thus in $\mathcal{O}(2^{\mathcal{O}(|\mathcal{R}|^2)})$ since we can estimate $|\mathcal{F}|$ and m by $|\mathcal{R}|$. Compared to \mathcal{A} the size of the automaton $\mathcal{B}(\mathcal{R})$ is neglectable. Next we analyze the complexity of the saturation process of Sect. 3.3. (A similar analysis is reported in [12, chapitre IV].)

Lemma 10. *Let \mathcal{R} be a growing TRS. The saturation rules of $\mathcal{C}_T(\mathcal{R})$ can be computed in $\mathcal{O}(|\mathcal{R}|^5 \cdot |Q|^{\mathcal{O}(|\mathcal{R}|)})$ time.*

Proof. Let Ξ be the set of transition rules that may potentially appear in in the automaton $\mathcal{C}_T(\mathcal{R})$: $\Xi = \{f(q_1, \dots, q_n) \rightarrow q \mid f \in \mathcal{F} \text{ and } q_1, \dots, q_n, q \in Q\}$. There are at most $K = \#\mathcal{R} \cdot |Q|^{m+1}$ rules in Ξ . The saturation process may be described by the following algorithm:

```

 $\Xi_0 := \Xi \setminus \Gamma_0;$ 
 $k := 1;$ 
while
   $\exists f(q_1, \dots, q_n) \rightarrow q \in \Xi_{k-1}$ 
   $\exists f(l_1, \dots, l_n) \rightarrow r \in \mathcal{R}$ 
   $\exists \theta: \text{Var}(r) \rightarrow Q$ 
    such that
     $r\theta \rightarrow_{\Gamma_{k-1}}^* q$  and, for all  $1 \leq i \leq n$ ,
     $q_i = l_i\theta$  if  $l_i \in \text{Var}(r)$  and  $q_i = \langle l_i \rangle$ 
    otherwise
do
   $\Xi_k := \Xi_{k-1} \setminus \{f(q_1, \dots, q_n) \rightarrow q\};$ 
   $\Gamma_k := \Gamma_{k-1} \cup \{f(q_1, \dots, q_n) \rightarrow q\};$ 
   $k := k + 1$ 

```

Let us estimate the time to evaluate the condition of the while-loop. There are $K - |\Gamma_{k-1}|$ choices for $f(q_1, \dots, q_n) \rightarrow q$, $\#\mathcal{R}$ choices for $f(l_1, \dots, l_n) \rightarrow r$, and $|Q|^{\text{Var}(r)}$ choices for θ . For every choice we have to test whether $r\theta \rightarrow_{\Gamma_{k-1}}^* q$ is true. (The other requirements are neglectable.) This can be done in $\mathcal{O}(|r| \cdot |\Gamma_{k-1}|)$ time. So one iteration of the while-loop takes

$$\mathcal{O}((K - |\Gamma_{k-1}|) \cdot \#\mathcal{R} \cdot |Q|^{\text{Var}(r)} \cdot |r| \cdot |\Gamma_{k-1}|)$$

time. To obtain the time complexity of the algorithm we have to multiply this by the maximum number of iterations, which is $K - |\Gamma_0|$. Removing the negative terms and estimating $|\Gamma_{k-1}|$ by K and $\text{Var}(r)$ by $|r|$ yields

$$\mathcal{O}(K^3 \cdot \#\mathcal{R} \cdot |Q|^{|r|} \cdot |r|) = \mathcal{O}(\#\mathcal{R}^4 \cdot |Q|^{3m+3+|r|} \cdot |r|)$$

Estimating $\#\mathcal{R}$, m , and $|r|$ by $|\mathcal{R}|$ yields the complexity class $\mathcal{O}(|\mathcal{R}|^5 \cdot |Q|^{\mathcal{O}(|\mathcal{R}|)})$ in the statement of the lemma. \square

So for growing TRSs the time complexity of the saturation process is exponential in the size of the TRS. For \mathcal{R}_s and \mathcal{R}_{nv} we get a polynomial time complexity, but the space and time complexity of the automaton \mathcal{C} is still exponential in $|\mathcal{R}|$ due to the normal form automaton.

The number $|Q|$ of states of the automaton $\mathcal{C}(\mathcal{R}_g)$ is of the same order as the number of states of \mathcal{A} . The time to compute $\mathcal{C}(\mathcal{R}_g)$ is the sum of the times to compute the automaton \mathcal{A} ($\mathcal{O}(2^{\mathcal{O}(|\mathcal{R}|^2)})$), the automaton $\mathcal{B}(\mathcal{R})$ (neglectable),

and the saturation rules ($\mathcal{O}(2^{\mathcal{O}(|\mathcal{R}|^2)})$ from Lemma 10 with $|Q| \in \mathcal{O}(2^{|\mathcal{R}|})$). This yields an $\mathcal{O}(2^{\mathcal{O}(|\mathcal{R}|^2)})$ time complexity. Note that the complexity of \mathcal{C}' is of the same order.

Finally, let us consider the construction of \mathcal{D} . The number of states of \mathcal{D} is in $\mathcal{O}(2^{2^{|\mathcal{R}|}})$ and the number of transition rules in $\mathcal{O}(2^{2^{\mathcal{O}(|\mathcal{R}|^2)}})$. The time to build \mathcal{D} is the time to build \mathcal{C}' plus the time to compute the rules of \mathcal{D} . The former is neglectable with respect to the latter, which can be done in

$$\mathcal{O}(2^{2^{\mathcal{O}(|\mathcal{R}|^2)}} \cdot |\mathcal{C}'|) = \mathcal{O}(2^{2^{\mathcal{O}(|\mathcal{R}|^2)}})$$

time.

As emptiness can be decided in polynomial time with respect to the size of the automaton, we conclude with the following theorem.

Theorem 2. *It can be decided in double exponential time whether a left-linear TRS belongs to CBN-NF_g .* \square

Although the saturation process for \mathcal{R}_s and \mathcal{R}_{nv} is much simpler, our constructions do not give better complexity results for deciding membership in CBN-NF_s or CBN-NF_{nv} . Nevertheless, for CBN-NF_s (which coincides with the class of strongly sequential TRSs, see [9]) a lower complexity bound is known. Comon [2,3] showed that it can be decided in exponential time whether a left-linear TRS is strongly sequential. He uses an automaton for ω -reduction which plays the same role as our $\mathcal{C}_{\text{NF}_{\mathcal{R}}}$ automaton. Since there is no satisfactory notion of ω -reduction corresponding to the approximations mappings nv and g , it remains to be seen whether the result of Theorem 2 can be improved.

6 Call by Need Computations to Root-Stable Form

In this section we assume that \mathcal{R} and \mathcal{S} are linear growing TRSs over signature \mathcal{F} and $\mathcal{G} = \mathcal{F} \cup \{f^\circ \mid f \in \mathcal{F}\}$. Our first goal is to construct a tree automaton that recognizes the set $\text{RS}_{\mathcal{S}^\circ}$ of root-stable ground terms with respect to the TRS $\mathcal{S}^\circ = \mathcal{S} \cup \{l^\circ \rightarrow r \mid l \rightarrow r \in \mathcal{S}\}$. Consider the automaton $\mathcal{B}(\mathcal{S}^\circ)$ defined in Section 3.1. To this automaton we add a single final state q_f and transition rules $f(\langle l_1 \rangle, \dots, \langle l_n \rangle) \rightarrow q_f$ and $f^\circ(\langle l_1 \rangle, \dots, \langle l_n \rangle) \rightarrow q_f$ for every left-hand side $f(l_1, \dots, l_n)$ of a rewrite rule in \mathcal{S} . One easily verifies that the resulting automaton, which we denote by $\mathcal{A}_{\text{REDEX}_{\mathcal{S}^\circ}}$, accepts the set of ground redexes of \mathcal{S}° . Applying the construction in Section 3.3 to $\mathcal{A}_{\text{REDEX}_{\mathcal{S}^\circ}}$ and $\mathcal{B}(\mathcal{S}^\circ)$ results in an automaton $\mathcal{C}_{\text{REDEX}_{\mathcal{S}^\circ}}(\mathcal{S}^\circ)$ that accepts all ground terms in $\mathcal{T}(\mathcal{G})$ that rewrite in \mathcal{S}° to a term in $\text{REDEX}_{\mathcal{S}^\circ}$, in other words, all non-root-stable terms of \mathcal{S}° . From this we obtain the desired tree automaton $\mathcal{A}_{\text{RS}_{\mathcal{S}^\circ}}$ by a subset construction. Formally, the states of $\mathcal{A}_{\text{RS}_{\mathcal{S}^\circ}}$ are subsets of states of $\mathcal{C}_{\text{REDEX}_{\mathcal{S}^\circ}}(\mathcal{S}^\circ)$, the final states are those subsets that do not contain q_f (the unique final state of $\mathcal{C}_{\text{REDEX}_{\mathcal{S}^\circ}}(\mathcal{S}^\circ)$) and the transition rules of $\mathcal{A}_{\text{RS}_{\mathcal{S}^\circ}}$ are defined as expected.

Next we apply the constructions in Section 3.3 to $\mathcal{A}_{\text{RS}_{\mathcal{S}^\circ}}$ and $\mathcal{B}(\mathcal{R})$. This yields an automaton $\mathcal{C}_{\text{RS}_{\mathcal{S}^\circ}}(\mathcal{R})$ that accepts all ground terms that rewrite in \mathcal{R}

to a term in RS_{S° . The construction in Section 4 needs some modifications. We obtain $\mathcal{C}'_{\text{RS}_{S^\circ}}(\mathcal{R})$ from $\mathcal{C}_{\text{RS}_{S^\circ}}(\mathcal{R})$ by adding a fresh copy of $\mathcal{B}(\mathcal{R})$ (we don't need the state q_r here) as well as the automaton $\mathcal{C}_{\text{REDEX}_S}(\mathcal{S})$. Let Q'_f be the set of final states of $\mathcal{C}_{\text{REDEX}_S}(\mathcal{S})$. Concerning the construction of \mathcal{D} , instead of $P^2 = \{\langle x \rangle\}$ (in the case that there exists a rewrite rule in \mathcal{R} with left-hand side $f(l_1, \dots, l_n)$ such that $\langle l_i \rangle' \in S_i$ for all $i \in \{1, \dots, n\}$) we define P^2 as a non-empty subset of $f^\circ(S_1, \dots, S_n) \downarrow$. Let us denote the resulting automaton by $\mathcal{D}'(\mathcal{R}, \mathcal{S})$. The final states of $\mathcal{D}'(\mathcal{R}, \mathcal{S})$ are those pairs $[S, P]$ that satisfy $S \cap Q'_f \neq \emptyset$ and $P \subseteq Q_f$. The former condition ensures that only non- \mathcal{R} -root-stable terms are accepted. The proofs of the following statements are easy modifications of the corresponding ones in Section 4. (In the proof of Lemma 12 we take $P^2 = P \cap s^\circ \downarrow$ in the case that s is a redex.)

Lemma 11. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \rightarrow_{\Gamma_{\mathcal{D}'}}^* [S, P]$ then for every redex position p in s there exists a state $q \in P$ such that $q \in s[(s|_p)^\circ]_p \downarrow$. \square*

Lemma 12. *Let $s \in \mathcal{T}(\mathcal{F})$. If*

$$P \subseteq \bigcup_{p \in \mathcal{R}(s)} s[(s|_p)^\circ]_p \downarrow$$

such that $P \cap s[(s|_p)^\circ]_p \downarrow \neq \emptyset$ for all $p \in \mathcal{R}(s)$ then $s \rightarrow_{\Gamma_{\mathcal{D}'}}^ [S, P]$ where $S = s \downarrow$. \square*

Lemma 13. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \rightarrow_{\Gamma_{\mathcal{D}'}}^* [S, P]$ and $P \subseteq Q_f$ then, for every redex position p in s , $s[(s|_p)^\circ]_p \in (\rightarrow_{\mathcal{R}}^+) [\text{RS}_{S^\circ}]$. \square*

Lemma 14. *Let $s \in \mathcal{T}(\mathcal{F})$. If $s[(s|_p)^\circ]_p \in (\rightarrow_{\mathcal{R}}^+) [\text{RS}_{S^\circ}]$ for every redex position p in s then there exists a set $P \subseteq Q_f$ such that $s \rightarrow_{\Gamma_{\mathcal{D}'}}^* [S, P]$ with $S = s \downarrow$. \square*

Corollary 3. *Let $s \in \mathcal{T}(\mathcal{F})$. We have $s \in L(\mathcal{D}'(\mathcal{R}, \mathcal{S}))$ if and only if s is non- S -root-stable and $s[(s|_p)^\circ]_p \in (\rightarrow_{\mathcal{R}}^+) [\text{RS}_{S^\circ}]$ for every redex position p in s . \square*

Theorem 3. *Let \mathcal{R} be a left-linear TRS and $\alpha, \beta \in \{\text{s}, \text{nv}, \text{g}\}$. We have $\mathcal{R} \in \text{CBN-RS}_{\alpha, \beta}$ if and only if $L(\mathcal{D}'(\mathcal{R}_\alpha, \mathcal{R}_\beta)) = \emptyset$.*

Proof. Note that \mathcal{R}_α and \mathcal{R}_β are linear growing TRSs. We have $\mathcal{R} \notin \text{CBN-RS}_{\alpha, \beta}$ if and only if there exists a non- \mathcal{R}_β -root-stable term s in $\mathcal{T}(\mathcal{F})$ without $(\mathcal{R}_\alpha, \mathcal{R}_\beta)$ -root-needed redexes. The latter is equivalent to $s[(s|_p)^\circ]_p \in (\rightarrow_{\mathcal{R}_\alpha}^+) [\text{RS}_{\mathcal{R}_\beta}]$ for every redex position p in s . According to the preceding corollary this is equivalent to $s \in L(\mathcal{D}'(\mathcal{R}_\alpha, \mathcal{R}_\beta))$. We conclude that $\mathcal{R} \in \text{CBN-RS}_{\alpha, \beta}$ if and only if $L(\mathcal{D}'(\mathcal{R}_\alpha, \mathcal{R}_\beta)) = \emptyset$. \square

The following complexity result is obtained by a similar analysis to the one in Sect. 5. Note that the nested saturation process does not give rise to an extra exponential since saturation increases only the time but not the space complexity by an exponential.

Theorem 4. *It can be decided in double exponential time whether a left-linear TRS belongs to $\text{CBN-RS}_{\alpha, \beta}$ for all $\alpha, \beta \in \{\text{s}, \text{nv}, \text{g}\}$. \square*

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
2. H. Comon. Sequentiality, second-order monadic logic and tree automata. In *Proc. 10th LICS*, pages 508–517, 1995.
3. H. Comon. Sequentiality, monadic second-order logic and tree automata. *Information and Computation*, 1999. To appear. Full version of [2].
4. H. Comon, M. Dauchet, R. Gilleron, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 1998. Draft, available from <http://www.grappa.univ-lille3.fr/tata/>.
5. J.L. Coquidé, M. Dauchet, R. Gilleron, and S. Vágvölgyi. Bottom-up tree push-down automata: Classification and connection with rewrite systems. *Theoretical Computer Science*, 127:69–98, 1994.
6. M. Dauchet, T. Heuillard, P. Lescanne, and S. Tison. Decidability of the confluence of finite ground term rewriting systems and of other related term rewriting systems. *Information and Computation*, 88:187–201, 1990.
7. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science*, volume B, pages 243–320. Elsevier, 1990.
8. I. Durand. Bounded, strongly sequential and forward-branching term rewriting systems. *Journal of Symbolic Computation*, 18:319–352, 1994.
9. I. Durand and A. Middeldorp. Decidable decidable call by need computations in term rewriting (extended abstract). In *Proc. 14th CADE*, volume 1249 of *LNAI*, pages 4–18, 1997.
10. F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.
11. G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems, i and ii. In *Computational Logic, Essays in Honor of Alan Robinson*, pages 396–443. The MIT Press, 1991. Original version: Report 359, Inria, 1979.
12. F. Jacquemard. *Automates d’Arbres et Réécriture de Termes*. PhD thesis, Université Paris-Sud, 1996.
13. F. Jacquemard. Decidable approximations of term rewriting systems. In *Proc. 7th RTA*, volume 1103 of *LNCS*, pages 362–376, 1996.
14. J.W. Klop. Term rewriting systems. In *Handbook of Logic in Computer Science*, Vol. 2, pages 1–116. Oxford University Press, 1992.
15. A. Middeldorp. Call by need computations to root-stable form. In *Proc. 24th POPL*, pages 94–105, 1997.
16. T. Nagaya, M. Sakai, and Y. Toyama. NVNF-sequentiality of left-linear term rewriting systems. In *Proc. RIMS Workshop on Theory and Applications of Rewriting Systems*, pages 109–117, 1995.
17. M. Oyamaguchi. NV-sequentiality: A decidable condition for call-by-need computations in term rewriting systems. *SIAM Journal on Computation*, 22:114–135, 1993.
18. M. Oyamaguchi, November 1998. Personal communication.
19. R. Strandh. Classes of equational programs that compile into efficient machine code. In *Proc. 3rd RTA*, volume 355 of *LNCS*, pages 449–461, 1989.
20. Y. Toyama, S. Smetsers, M. van Eekelen, and R. Plasmeijer. The functional strategy and transitive term rewriting systems. In *Term Graph Rewriting: Theory and Practice*, pages 61–75. Wiley, 1993.