

On the derivational complexity of Kachinuki orderings

Dieter Hofbauer

ASW – Berufsakademie Saarland, Germany
d.hofbauer@asw-berufsakademie.de

Abstract

For string rewriting systems compatible with the standard Kachinuki ordering it is known that their derivational complexity is primitive recursively bounded. However, in case the definition of Kachinuki orderings comprises a possibly different lexicographic status for different letters, the standard upper bound proof method by monotone interpretations into the natural numbers fails, and primitive recursive complexity bounds are an open problem, to the best of our knowledge. In this talk, such an upper bound result is shown by examining worst case rewriting strategies.

Keywords and phrases Kachinuki ordering, string rewriting, termination, derivational complexity

1 Introduction

We assume familiarity with basic notations for string rewriting. Here, Σ^* is the set of strings over an alphabet Σ , $|x|$ denotes the length of a string x , $|x|_a$ denotes the number of occurrences of letter a in x , and ϵ is the empty string. A rewriting system R is *compatible* with a relation $>$ on Σ^* if $R \subseteq >$, and $\text{Nf}(>)$ denotes the set of normal forms modulo $>$. For a given ordering $>$ on a set A , let $>^{\triangleleft\text{-lex}}$ denote the *right-to-left* length-lexicographic extension of $>$ to A^* , analogously define the *left-to-right* variant $>^{\triangleright\text{-lex}}$. For $x = c_0 \dots c_n \in \Sigma^*$, $c_i \in \Sigma$, let $x^{\text{rev}} = c_n \dots c_0$ denote the *reversal* of x , and for a rewriting system R define $R^{\text{rev}} = \{\ell^{\text{rev}} \rightarrow r^{\text{rev}} \mid \ell \rightarrow r \in R\}$. Presently, we consider only finite alphabets and rewriting systems.

The *derivation height* of a string x modulo some finitely branching and well-founded relation $>$ on Σ^* is $\text{dh}_>(x) = \max\{n \mid \exists y : x >^n y\}$, and the *derivational complexity* of $>$ maps a natural number n to the maximal derivation height of strings of size at most n , i. e., $\text{dc}_>(n) = \max\{\text{dh}_>(x) \mid |x| \leq n\}$. For a string rewriting system R let dh_R and dc_R abbreviate $\text{dh}_{\rightarrow_R}$ and $\text{dc}_{\rightarrow_R}$, respectively.

2 Kachinuki orderings

Throughout, we use $\succ$ as an ordering on Σ , a so-called *precedence*. Since we consider chains of maximal length, we assume $\succ$ to be total. Further, each letter $a \in \Sigma$ has an assigned *status* $s(a) \in \{\triangleleft, \triangleright\}$, either *right-to-left* ($\triangleleft$) or *left-to-right* ($\triangleright$).

► **Definition 1.** The *Kachinuki ordering* with status function s on Σ^* is defined by $x \succ_s y$ provided one of the following holds, where a is the maximal letter modulo $\succ$ occurring in x or y :

- $|x|_a > |y|_a$, or
- $|x|_a = |y|_a = n$, $x = x_0 a x_1 \dots a x_n$, $y = y_0 a y_1 \dots a y_n$ for $x_i, y_i \in (\Sigma \setminus \{a\})^*$, and $(x_0, x_1, \dots, x_n) \succ_s^{s(a)\text{-lex}} (y_0, y_1, \dots, y_n)$.

Varying the status function s , there are $2^{|\Sigma|-1}$ many different Kachinuki orderings, as is easily seen, all of them being well-founded and closed under contexts. Let $\succ^\triangleleft$ denote the particular *standard* Kachinuki ordering where each letter has right-to-left status, cf. [7].

► **Example 2.** The rewriting system $R_0 = \{ac \rightarrow ca, cb \rightarrow bc\}$ is compatible with $\succ_s$ for $a \succ b \succ c$ and $s(a) = \triangleleft, s(b) = \triangleright$. For this example, also the standard Kachinuki ordering could be used as $R_0 \subseteq \succ^\triangleleft$ for $a \succ c \succ b$. In contrast, $R_1 = \{ac \rightarrow cca, cb \rightarrow bcc\} \subseteq \succ_s$ (as for R_0), but both R_1 and R_1^{rev} are incompatible with $\succ^\triangleleft$, whatever precedence is chosen.

► **Remark.** Under the usual isomorphism between Σ^* and the set of terms $\mathcal{T}_\Sigma(\{x\})$ via $a_0a_1 \dots a_n \mapsto a_0(a_1(\dots a_n(x) \dots))$, with variable x and Σ be seen as a set of unary function symbols, the Kachinuki ordering $\succ^\triangleleft$ corresponds to the precedence based path ordering on terms, i. e., the *multiset path ordering* without exploiting multisets. Therefore, if a finite string rewriting system R is compatible with $\succ^\triangleleft$, then dc_R is primitive recursively bounded, and the relationship between the size of the underlying alphabet and the corresponding level of the Grzegorzczuk hierarchy for dc_R is known, see [3, 4].

► **Remark.** The standard upper bound proof method by monotone interpretations into the natural numbers fails for Kachinuki orderings since no such interpretation exists for $\{cb \rightarrow bcc\}$, see [8]. Therefore, a monotone interpretation for system R_1 (and also for R_1^{rev}) from Example 2 doesn't exist.

Depending on the status function s , there is a simple order isomorphism $\lambda x.x^\triangleleft$ on Σ^* such that, for $x, y \in \Sigma^*$ and arbitrary precedence $\succ$,

$$x \succ_s y \quad \text{if, and only if,} \quad x^\triangleleft \succ^\triangleleft y^\triangleleft.$$

For a being the maximal letter modulo $\succ$ occurring in $x = x_0a \dots ax_n$ with $x_i \in (\Sigma \setminus \{a\})^*$, define the *straightening* of x as $x^\triangleleft = x_0^\triangleleft a \dots ax_n^\triangleleft$ for $s(a) = \triangleleft$, and $x^\triangleleft = x_n^\triangleleft a \dots ax_0^\triangleleft$ for $s(a) = \triangleright$.

Considering a rewriting system R compatible with some Kachinuki ordering $\succ_s$, at first glance it might be tempting to assume that derivations modulo R are isomorphic to derivations modulo $R^\triangleleft = \{\ell^\triangleleft \rightarrow r^\triangleleft \mid \ell \rightarrow r \in R\}$. Since $R \subseteq \succ_s$ holds if, and only if, $R^\triangleleft \subseteq \succ^\triangleleft$, upper bound results for $\succ^\triangleleft$ would then imply upper bound results for $\succ_s$. However, the crucial observation is that

$$x \rightarrow_R y \quad \text{does not imply} \quad x^\triangleleft \rightarrow_{R^\triangleleft} y^\triangleleft.$$

► **Example 3.** For $R = \{ab \rightarrow ba\}$ over alphabet $\{a, b, c\}$, precedence $a \succ b \succ c$, status function s with $s(a) = \triangleleft, s(b) = \triangleright$ and arbitrary $n \neq 0$, we have $R^\triangleleft = R$ and

$$c^n ab \rightarrow_R c^n ba, \text{ but } (c^n ab)^\triangleleft = c^n ab \not\rightarrow_{R^\triangleleft} bc^n a = (c^n ba)^\triangleleft.$$

This shows that in general rewriting modulo some finite system R translates into rewriting on straightened strings modulo some infinite system (or rule scheme) with unbounded size of right-hand sides of rules. For infinite rewriting systems compatible with the standard Kachinuki ordering, however, the derivational complexity is not necessarily primitive recursively bounded, as the following example shows.

► **Example 4.** The infinite string rewriting system $A = \{ab \rightarrow bba\} \cup \{b^i aab \rightarrow ab^i a \mid i \geq 0\}$ is compatible with $\succ^\triangleleft$ for $a \succ b$, but dc_A is not primitive recursively bounded [5]. (In fact, A is a string version of the term rewriting system from [5], slightly simplified.) Note that A has infinitely many right-hand sides, although $\rightarrow_A$ is finitely branching.

3 A worst case derivation strategy

For the rest of the exposition let $\Sigma = \{a_0, \dots, a_n\}$, and consider the total precedence $a_n \succ \dots \succ a_0$. First, we give a characterization of Kachinuki orderings on strings of bounded size in terms of rewriting relations.

► **Definition 5.** Let $P_0 = \{a_i \rightarrow \epsilon \mid 0 \leq i \leq n\}$ and, for $k > 0$,

$$P_k = \{a_0 \rightarrow \epsilon, a_{i+1} \rightarrow a_i^k, a_{i+1}a_0 \rightarrow a_i^{k-1}a_{i+1} \mid 0 \leq i < n\}.$$

Note that P_k is terminating because of $P_k \subseteq \succ^\triangleleft$. Further, $a_i \rightarrow_{P_k}^+ \epsilon$, thus $a_{i+1} \rightarrow_{P_k}^+ a_i$ for $k > 0$. Since $x \rightarrow_{P_k}^+ \epsilon$ for $x \in \Sigma^*$, normal forms modulo P_k are unique, thus P_k is confluent. For other characterizations of path orderings (on general terms) by rewriting systems see [6, 2, 1]; in contrast to P_k , these systems are non-terminating, however.

► **Lemma 6.** For $x, y \in \Sigma^*$ and $k \geq 0$, if $x \succ^\triangleleft y$ and $|y| \leq k$, then $x \rightarrow_{P_k}^+ y$.

Proof. By induction on x along $\succ^\triangleleft$. ◀

► **Corollary 7.** For a rewriting system R compatible with the standard Kachinuki ordering where $\{r \mid \ell \rightarrow r \in R\}$ is finite, dc_R is primitive recursively bounded.

Proof. Sketch (a special case of [4]): For $\max\{|\ell| \mid \ell \rightarrow r \in R\} = k$, define a strictly monotone interpretation for P_k as follows. Inductively define functions $\bar{a}_i$ on the natural numbers by $\bar{a}_0(m) = m + 1$, $\bar{a}_{i+1}(0) = \bar{a}_i^k(m) + 1$, and $\bar{a}_{i+1}(m + 1) = \bar{a}_i^{k-1}(\bar{a}_{i+1}(m)) + 1$. Then $\text{dc}_R(m) \leq \text{dh}_{P_k}(a_n^m) \leq \bar{a}_n^m(0)$. ◀

Next, we determine a worst case rewriting strategy for P_k in the sense that it induces maximal derivation lengths. The cases $k \leq 1$ are uninteresting as all derivations have the same length and the derivational complexity of P_k is linear, therefore we assume $k > 1$ from now on.

► **Definition 8.** Let $\rightarrow_k$ on Σ^* be defined as the following relation, where $u \in \Sigma^*$, $v \in \{a_0\}^*$, $0 \leq i < n$, $m \geq 0$:

$$\begin{aligned} a_0^{m+1} &\rightarrow_k a_0^m, \\ ua_{i+1} &\rightarrow_k ua_i^k, \\ ua_{i+1}a_0v &\rightarrow_k ua_i^{k-1}a_{i+1}v. \end{aligned}$$

Observe that $\rightarrow_k \subseteq \rightarrow_{P_k}$, that ϵ is the only normal form modulo $\rightarrow_k$, and that $\rightarrow_k$ is deterministic in the sense that every string except ϵ has exactly one successor modulo $\rightarrow_k$. Further note that $\rightarrow_k$ does not correspond to an innermost (i. e., rightmost) derivation strategy for P_k . The following commutation property is needed later.

► **Lemma 9.** For $x, y_1, y_2 \in \Sigma^*$, if $y_1 \leftarrow_{P_k} x \rightarrow_k y_2$, then $y_1 \rightarrow_k \circ \leftarrow_{P_k} y_2$ or $y_1 \leftarrow_{P_k}^* y_2$.

Proof. We proceed by case analysis, using the notations from Definition 8. Recall that we assume $k > 1$. Case 1: $x = a_0^{m+1}$, thus $y_1 = y_2 = a_0^m$. Case 2: $x = ua_{i+1}$, thus $y_2 = ua_i^k$. Then either $y_1 = y_2$, or $y_1 = u'a_{i+1}$ for $u \rightarrow_{P_k} u'$. In the latter case the underlying redexes do not overlap, thus $y_1 \rightarrow_k u'a_i^k \leftarrow_{P_k} y_2$. Case 3: $x = ua_{i+1}a_0v$, thus $y_2 = ua_i^{k-1}a_{i+1}v$. If $y_1 = y_2$ we are done. If $y_1 = u'a_{i+1}a_0v$ for $u \rightarrow_{P_k} u'$, then $y_1 \rightarrow_k u'a_i^{k-1}a_{i+1}v' \leftarrow_{P_k} y_2$. Similarly, if $y_1 = ua_{i+1}a_0v'$ for $v \rightarrow_{P_k} v'$, then $v' \in \{a_0\}^*$ and $y_1 \rightarrow_k ua_i^{k-1}a_{i+1}v' \leftarrow_{P_k} y_2$. If $y_1 = ua_{i+1}v$ using rule $a_0 \rightarrow \epsilon$, then $y_2 \rightarrow_{P_k}^* y_1$ since $a_i^{k-1} \rightarrow_{P_k}^* \epsilon$. Eventually, if $y_1 = ua_i^k v$ using rule $a_{i+1} \rightarrow a_i^k$, then $y_2 \rightarrow_{P_k}^* y_1$ since, for $k > 1$, $a_i^{k-1}a_{i+1} \rightarrow_{P_k} a_i^{k-1}a_i^k = a_i^k a_i^{k-1} \rightarrow_{P_k}^* a_i^k a_i \rightarrow_{P_k}^* a_i^k a_0$. ◀

► **Lemma 10.** *Let $\rightarrow$ and $\rightarrow$ be binary relations on the same set such that $\text{Nf}(\rightarrow) \subseteq \text{Nf}(\rightarrow)$ and $\leftarrow \circ \rightarrow \subseteq (\rightarrow \circ \leftarrow^+) \cup \leftarrow^*$. Then for every reduction sequence modulo $\rightarrow$ there is a reduction sequence modulo $\rightarrow$ of the same length and starting from the same element.*

Proof. First we show $\leftarrow^n \circ \rightarrow \subseteq \rightarrow^= \circ \leftarrow^{\geq n-1}$ for $n > 0$ by induction on n : We have $\leftarrow^1 \circ \rightarrow \subseteq (\rightarrow \circ \leftarrow^+) \cup \leftarrow^* \subseteq \rightarrow^= \circ \leftarrow^{\geq 0}$ and, using the induction hypothesis, $\leftarrow^{n+1} \circ \rightarrow = \leftarrow^n \circ \leftarrow \circ \rightarrow \subseteq \leftarrow^n \circ ((\rightarrow \circ \leftarrow^+) \cup \leftarrow^*) = (\leftarrow^n \circ \rightarrow \circ \leftarrow^+) \cup (\leftarrow^n \circ \leftarrow^*) \subseteq (\rightarrow^= \circ \leftarrow^{\geq n-1} \circ \leftarrow^+) \cup \leftarrow^{\geq n} = \rightarrow^= \circ \leftarrow^{\geq n}$.

Now for $n \geq 0$, if $x \rightarrow^n y$, then there is some element y' such that $x \rightarrow^n y'$. Again, this is shown by induction on n . The case $n = 0$ being trivial, assume $x \rightarrow \circ \rightarrow^n y$. By $\text{Nf}(\rightarrow) \subseteq \text{Nf}(\rightarrow)$ we obtain $x \rightarrow z$ for some z , and from $y \leftarrow^{n+1} x \rightarrow z$ we get $y \rightarrow^= \circ \leftarrow^{\geq n} z$ by the first claim. By induction hypothesis, $z \rightarrow^{\geq n} z'$ for some z' , so $x \rightarrow z \rightarrow^n y' \rightarrow^* z'$ for some y' , concluding the proof. ◀

Combining these two lemmas, we arrive at the announced worst case property of the derivation strategy $\rightarrow_k$.

► **Corollary 11.** *If R is a rewriting system compatible with $\succ^\triangleleft$ and $\max\{|r| \mid \ell \rightarrow r \in R\} = k$, then $\text{dh}_{\rightarrow_k}(x) = \text{dh}_{P_k}(x) \geq \text{dh}_R(x)$.*

This result can now be generalized to arbitrary Kachinuki orderings, as follows.

► **Theorem 12.** *If R is a rewriting system compatible with $\succ_s$ for status function s and $\max\{|r| \mid \ell \rightarrow r \in R\} = k$, then $\text{dc}_{\rightarrow_k}(m) \geq \text{dc}_R(m)$.*

Proof. Sketch: Analogously to P_k and $\rightarrow_k$, define a rewriting system $P_{s,k}$ and a corresponding worst case strategy $\rightarrow_{s,k}$. Let $L = \{x \in \Sigma^* \mid \exists m \geq 0 : a_n^m \rightarrow_{s,k} x\}$. We then show that $x \rightarrow_{s,k} y$ is equivalent to $x^\triangleleft \rightarrow_k y^\triangleleft$ for $x, y \in L$, and conclude by $\text{dc}_{\rightarrow_k}(m) = \text{dh}_{\rightarrow_k}(a_n^m) = \text{dh}_{\rightarrow_{s,k}}(a_n^m) = \text{dh}_{P_{s,k}}(a_n^m) \geq \text{dc}_R(m)$. ◀

► **Corollary 13.** *Finite rewriting systems that are compatible with a Kachinuki ordering have primitive recursively bounded derivational complexity.*

Acknowledgments: Thanks to Johannes Waldmann for posing the question addressed here.

References

- 1 N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Volume B: Formal Methods and Semantics, Chapter 6, pp. 243–320. North-Holland, 1990.
- 2 A. Geser. *Relative Termination*. Dissertation, Universität Passau, Germany, 1990.
- 3 D. Hofbauer. *Termination proofs and derivation lengths in term rewriting systems*. Dissertation, Technische Universität Berlin, Germany, 1991.
- 4 D. Hofbauer. Termination proofs by multiset path orderings imply primitive recursive derivation lengths. *Theoretical Computer Science*, 105(1):129–140, 1992.
- 5 D. Hofbauer and C. Lautemann. Termination proofs and the length of derivations. In N. Dershowitz (Ed.), *Proc. Rewriting Techniques and Applications (RTA)*, Volume 355 of *LNCS*, pp. 167–177. Springer, 1989.
- 6 J. W. Klop. Term rewriting systems. Report CS-R9073, Centrum voor Wiskunde en Informatica, Amsterdam, 1990.
- 7 K. Sakai. Knuth-Bendix algorithm for Thue system based on Kachinuki ordering. ICOT Technical Memorandum TM-0087, 1984.
- 8 H. Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, 1994.