DECIDING TERMINATION FOR ANCESTOR MATCH-BOUNDED STRING REWRITING SYSTEMS*

Alfons Geser†, Dieter Hofbauer‡, and Johannes Waldmann§

ABSTRACT

Termination of a string rewriting system can be characterized by termination on suitable recursively defined languages. This kind of termination criteria has been criticized for its lack of automation. In an earlier paper we have shown how to construct an automated termination criterion if the recursion is aligned with the rewrite relation. We have demonstrated the technique with Dershowitz's forward closure criterion. In this paper we show that a different approach is suitable when the recursion is aligned with the inverse of the rewrite relation. We apply this idea to Kurth's ancestor graphs and obtain ancestor match-bounded string rewriting systems. Termination is shown to be decidable for this class. The resulting method improves upon those based on match-boundedness or inverse match-boundedness.

1 INTRODUCTION

Automated termination criteria are vital for the processing, by machine, of string rewriting systems or similar formalisms. The limitations of the existing automated criteria show up clearly by their inability to prove, or disprove, termination of certain small string rewriting systems like Zantema's System $\{aabb \to bbbaaa\}$.

There are strong termination criteria which are not automated. Some of them characterize termination through termination on a suitable recursively defined subset of the set of all strings. For instance, Dershowitz [3] shows that termination is equivalent to termination on the set of right hand sides of forward closures. Kurth [11] shows that termination is equivalent to the absence of infinite paths in the ancestor graph. Usually the constructed subset is infinite, so the construction cannot be automated directly. In order to enable automation, we seek circumstances under which these infinite sets become regular languages, as for the class of regular languages many useful properties are decidable.

A string rewriting system R is called *deleting*, if for each letter that occurs in the right hand side of a rule, there is a greater letter in the corresponding left hand side modulo some partial order on the underlying alphabet. For a deleting system R and a regular language L, the set $R^*(L)$ of descendants of L modulo R is effectively regular [10]. One can go a step further by annotating each letter by a natural number that records its "recycling" history. If this number is bounded for all derivations starting from strings in L annotated with zeros,

*This work was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046.

†Senior Staff Scientist, National Institute of Aerospace (NIA), 144 Research Drive, Hampton, VA 23666. Email: `geser@nianet.org`, Web: `http://research.nianet.org/~geser/`.

‡Assistant Professor, Dept. of Mathematics/Informatics, University of Kassel, D-34109 Kassel, Germany. Email: `dieter@theory.informatik.uni-kassel.de`.

§Full Professor, Department of IMN, Polytechnic University of Leipzig, D-04251 Leipzig, Germany. Email: `waldmann@imn.htwk-leipzig.de`.

we say that the system is *match-bounded* for L. The annotated system of a match-bounded system is deleting; so match-bounded systems also preserve regularity.

We have shown in a previous paper [7] that RFC-match-boundedness, which is essentially the same as match-boundedness for the set of right hand sides of forward closures, is an automated criterion for termination. In the present paper we show that a different approach is capable to turn Kurth's ancestor graph criterion into an automated criterion.

The basic difference between criteria like Dershowitz's and Kurth's is the direction of recursion. Right hand sides of forward closures are aligned with the rewrite relation: the more complex ones are essentially descendants of the simpler ones. In contrast, the nodes of ancestor graphs are aligned with the *inverse* of the rewrite relation: the more complex ones are essentially ancestors of the simpler ones.

It seems appropriate to use inverse match-boundedness for the task, a property that we introduced as a (non-)termination criterion in [8] for the special case $L = \Sigma^*$. It turns out that the main formal language theoretic result presented there is not general enough for our present purpose. We show how it has to be extended so as to serve as a basis for automating ancestor-graph like criteria.

The paper is organized as follows. Section 3 and Section 4 contain a summary of definitions and basic results on deleting and match-bounded rewriting systems. In Section 5 we develop the formal language theoretic ingredients of our method. Kurth's ancestor graphs are defined in Section 6, and in Section 7 we describe how they can be characterized through inverse rewriting relations, leading to our main result. Finally, we present a few examples and refer to an implementation of our algorithm.

2 PRELIMINARIES

A *string rewriting system* over an alphabet Σ is a relation $R \subseteq \Sigma^* \times \Sigma^*$ that induces the *rewrite relation* $\to_R = \{(x\ell y, xry) \mid x, y \in \Sigma^*, (\ell, r) \in R\}$ on Σ^*. Unless indicated otherwise, all rewriting systems are finite. Pairs (ℓ, r) from R are frequently referred to as *rules* $\ell \to r$, and by $\mathrm{lhs}(R)$ and $\mathrm{rhs}(R)$ we denote the sets of left (right, resp.) hand sides of R. The reflexive and transitive closure of $\to_R$ is $\to_R^*$, often abbreviated as R^*, and $\to_R^+$ or R^+ denotes the transitive closure. An *R-derivation* is a (finite or infinite) sequence $x_0, x_1, \ldots$ with $x_i \to_R x_{i+1}$ for all i. We call R *terminating on* $L \subseteq \Sigma^*$ if there is no infinite R-derivation starting from a string $x_0 \in L$. We say that R is *terminating* if R is terminating on Σ^*, otherwise R is *non-terminating*. A particular kind of non-termination is caused by *loops*, i.e., derivations of the form $x \to_R^+ uxv$ for strings x, u, v. For surveys on termination of rewriting we refer to [5, 15]. Further standard notations for strings and string rewriting can be found, for instance, in [2].

For a relation $\rho \subseteq A \times B$ let $\rho(a) = \{b \in B \mid (a, b) \in \rho\}$ for $a \in A$ and $\rho(A') = \bigcup_{a \in A'} \rho(a)$ for $A' \subseteq A$. The inverse of ρ is $\rho^- = \{(b, a) \mid (a, b) \in \rho\} \subseteq B \times A$, and we say that ρ satisfies the property *inverse* P if ρ^- satisfies P. Define $\mathrm{Inf}(\rho) = \{a \in A \mid \rho(a) \text{ is infinite}\}$. The relation ρ is *finitely branching* if $\mathrm{Inf}(\rho) = \emptyset$. For the case $A = B$ let $\mathrm{Im}(\rho)$ denote the set of all "immortal" elements, i.e., those $a \in A$ that initiate an infinite sequence $a = a_0, a_1, \ldots$ with $(a_i, a_{i+1}) \in \rho$ for all i.

For a relation $\rho \subseteq \Sigma^* \times \Sigma^*$ on strings and a set $\Delta \subseteq \Sigma$ let $\rho|_\Delta = \rho \cap (\Delta^* \times \Delta^*)$. Note the difference between $R^*|_\Delta$ and $(R|_\Delta)^*$ for a string rewriting system R. For $R = \{a \to b, b \to c\}$ over $\Sigma = \{a, b, c\}$ and $\Delta = \{a, c\}$, e.g., we have $(a, c) \in R^*|_\Delta$, but $(a, c) \notin (R|_\Delta)^*$.

The set of *descendants* of a language $L \subseteq \Sigma^*$ modulo some string rewriting system R is $R^*(L)$. The system R is said to *preserve regularity (context-freeness)* if $R^*(L)$ is a regular (context-free) language whenever L is. For standard results on *rational transductions* we refer to [1].

A rewriting rule $\ell \to r$ is *context-free* if $|\ell| \leq 1$, and a rewriting system is context-free if all its rules are. Throughout we use ϵ for the *empty string* and $|x|$ for the *length* of a string x.

A relation $s \subseteq \Sigma^* \times \Gamma^*$ is a *substitution* if $s(\epsilon) = \{\epsilon\}$ and $s(xy) = s(x)s(y)$ for $x, y \in \Sigma^*$, therefore s is uniquely determined by the languages $s(a)$ for $a \in \Sigma$. For a family of languages $\mathcal{L}$ over Γ, the substitution s is an *$\mathcal{L}$-substitution* if $s(a) \in \mathcal{L}$ for all $a \in \Sigma$. For instance, if $\mathcal{L}$ is the family of finite (context-free) languages, then s is a *finite* (*context-free*, resp.) *substitution*. If $\epsilon \notin s(a)$ for every $a \in \Sigma$, then s is *epsilon-free*. Finite substitutions are finitely branching, and the same holds for the inverses of finite and epsilon-free substitutions.

3 DELETING STRING REWRITING SYSTEMS

Here we recall definitions and results on *deleting* string rewriting systems [10], a topic that can be traced back to Hibbard [9]. For detailed proofs and algorithms we refer to [10].

Definition 1. A string rewriting system R over an alphabet Σ is *$>$-deleting* for an irreflexive partial ordering $>$ on Σ (a *precedence*) if $\epsilon \notin \mathrm{lhs}(R)$, and if for each rule $\ell \to r$ in R and for each letter a in r, there is some letter b in ℓ with $b > a$. The system R is *deleting* if it is $>$-deleting for some precedence $>$.

Proposition 1 ([10]). *Every deleting string rewriting system is terminating, and has linear derivational complexity.*

This class of string rewriting systems enjoys a strong effective decomposition property. As an immediate consequence, deleting systems preserve regularity of languages, and inverse deleting systems preserve context-freeness.

Theorem 1 ([10]). *Let R be a deleting string rewriting system over Σ. Then there are an extended alphabet $\Gamma \supseteq \Sigma$, a finite substitution $s \subseteq \Sigma^* \times \Gamma^*$, and a context-free string rewriting system C over Γ such that $R^* = (s \circ C^{-*})|_\Sigma$.*

Corollary 1 ([10]). *Every deleting string rewriting system effectively preserves regularity.*

Corollary 2 ([9, 10]). *Every inverse deleting string rewriting system effectively preserves context-free languages.*

In the present paper, we will need a more specialized version of the above decomposition theorem.

Corollary 3. *Let R be a deleting string rewriting system over Σ such that $\epsilon \notin \mathrm{rhs}(R)$. Then there are an extended alphabet $\Gamma \supseteq \Sigma$, an epsilon-free finite substitution $s \subseteq \Sigma^* \times \Gamma^*$, and an epsilon-free context-free substitution $c \subseteq \Sigma^* \times \Gamma^*$ such that*

$$R^* = s \circ c^-.$$

Note that this implies $R^{-*} = R^{*-} = (s \circ c^-)^- = c \circ s^-$.

4 MATCH-BOUNDED STRING REWRITING SYSTEMS

Many properties of deleting string rewriting systems carry over to the more general case of match-bounded systems. In this section, we summarize essential results from [6, 7].

In order to record information about the history of letters during a derivation, we annotate letters by natural numbers, called *match heights*. The match heights in a reduct will be $h+1$ if the minimal height in the corresponding redex is h. Formally, for a given alphabet Σ define the morphisms $\mathrm{lift}_c : \Sigma^* \to (\Sigma \times \mathbb{N})^*$ for $c \in \mathbb{N}$ by $\mathrm{lift}_c : a \mapsto (a, c)$, $\mathrm{base} : (\Sigma \times \mathbb{N})^* \to \Sigma^*$ by $\mathrm{base} : (a, c) \mapsto a$, and $\mathrm{height} : (\Sigma \times \mathbb{N})^* \to \mathbb{N}^*$ by $\mathrm{height} : (a, c) \mapsto c$. For a string rewriting system R over Σ with $\epsilon \notin \mathrm{lhs}(R)$ define the rewriting system

$$\mathrm{match}(R) = \{\ell' \to \mathrm{lift}_c(r) \mid (\ell \to r) \in R,\ \mathrm{base}(\ell') = \ell,\ c = 1 + \min(\mathrm{height}(\ell'))\}$$

over alphabet $\Sigma \times \mathbb{N}$. For instance, the system $\mathrm{match}(\{ab \to bc\})$ contains the rules $a_0b_0 \to b_1c_1$, $a_0b_1 \to b_1c_1$, $a_1b_0 \to b_1c_1$, $a_1b_1 \to b_2c_2$, $a_0b_2 \to b_1c_1$, ..., writing x_c as abbreviation for (x, c). For non-empty R, the system $\mathrm{match}(R)$ is always infinite.

Every $\mathrm{match}(R)$-derivation corresponds to an R-derivation (i.e., for $x, y \in (\Sigma \times \mathbb{N})^*$, if $x \to_{\mathrm{match}(R)} y$ then $\mathrm{base}(x) \to_R \mathrm{base}(y)$) and vice versa (i.e., for $v, w \in \Sigma^*$ and $x \in (\Sigma \times \mathbb{N})^*$, if $v \to_R w$ and $\mathrm{base}(x) = v$, then there is $y \in (\Sigma \times \mathbb{N})^*$ such that $\mathrm{base}(y) = w$ and $x \to_{\mathrm{match}(R)} y$). In particular,

$$R^* = \mathrm{lift}_0 \circ \mathrm{match}(R)^* \circ \mathrm{base}\,.$$

Definition 2. A string rewriting system R over Σ is called *match-bounded* for $L \subseteq \Sigma^*$ by $c \in \mathbb{N}$ if $\epsilon \notin \mathrm{lhs}(R)$ and $\max(\mathrm{height}(x)) \leq c$ for every $x \in \mathrm{match}(R)^*(\mathrm{lift}_0(L))$. If we omit L, then it is understood that $L = \Sigma^*$.

The number $\max(\mathrm{height}(x))$ in Definition 2 (and $\min(\mathrm{height}(\ell'))$ in the definition of $\mathrm{match}(R)$) denotes the maximum (minimum, resp.) over the corresponding sequences of heights; we set $\max(\epsilon) = 0$, and we leave $\min(\epsilon)$ undefined as this case is excluded in the definition of $\mathrm{match}(R)$. Obviously, a system that is match-bounded for L is also match-bounded for any subset of L by the same bound. Further, by the remark before Definition 2, if R is match-bounded for L then R is match-bounded for $R^*(L)$, again by the same bound.

For a match-bounded system R, the infinite system $\mathrm{match}(R)$ may be replaced by a finite restriction. Denote by $\mathrm{match}_c(R)$ the restriction of $\mathrm{match}(R)$ to the alphabet $\Sigma \times \{0, 1, \ldots, c\}$. Then, if R is match-bounded for L by c,

$$R^* \cap (L \times \Sigma^*) = (\mathrm{lift}_0 \circ \mathrm{match}_c(R)^* \circ \mathrm{base}) \cap (L \times \Sigma^*).$$

Lemma 1. *For all $c \in \mathbb{N}$, the string rewriting system $\mathrm{match}_c(R)$ is deleting.*

Dually every deleting system is match-bounded. This connection with deleting rewriting makes results from Section 3 applicable. By Proposition 1, match-bounded systems terminate and have linearly bounded derivation lengths, and Corollary 1 implies that match-bounded systems effectively preserve regularity; as a consequence, match-boundedness by a given bound is decidable.

Corollary 4. *If R is match-bounded for a regular language L, then $R^*(L)$ is effectively regular.*

Finally note that if R is inverse match-bounded, then $\to_R^+$ is irreflexive, therefore $\mathrm{Im}(\to_R)$ and $\mathrm{Inf}(R^*)$ coincide.

5 RESTRICTED INVERSE MATCH-BOUNDEDNESS

In this section, we generalize our termination criterion [6, 8] from inverse match-bounded systems (for Σ^*) to systems that are inverse match-bounded for a suitable language L.

Consider the set $\mathrm{Inf}(R^* \cap (\Sigma^* \times R^{-*}(L)))$, consisting of all strings $x \in \Sigma^*$ where infinitely many strings $y \in \Sigma^*$ exist such that $x \to_R^* y \to_R^* z$ for some $z \in L$.[1] We start by showing that this set is regular for inverse deleting R and regular L. As we show later, emptiness of this set characterizes termination of R on $R^{-*}(L)$. The result can be carried over to systems R that are inverse match-bounded for L.

Lemma 2 ([8]). *Let Σ, Γ, Δ be alphabets, let $c \subseteq \Sigma^* \times \Gamma^*$ be a substitution, and let $T \subseteq \Gamma^* \times \Delta^*$ be a finitely branching rational transduction such that also T^- is finitely branching. Then $\mathrm{Inf}(c \circ T)$ is regular; and if c is a context-free substitution then effectively so.*

Lemma 3. *For any inverse deleting string rewriting system R with $\epsilon \notin \mathrm{lhs}(R)$ and any regular language L, both over Σ, the set $\mathrm{Inf}(R^* \cap (\Sigma^* \times R^{-*}(L)))$ is effectively regular.*

Note that $R^* \cap (\Sigma^* \times L') = R^* \cap (L' \times L') = (\to_R \cap (\Sigma^* \times L'))^*$ for $L' = R^{-*}(L)$.

Proof. We have $R^* = c \circ s^-$ by Corollary 3, where $c \subseteq \Sigma^* \times \Gamma^*$ is a context-free substitution and $s \subseteq \Sigma^* \times \Gamma^*$ is a finite epsilon-free substitution. Thus $R^* \cap (\Sigma^* \times R^{-*}(L)) = c \circ s'^-$, where $s' = s \cap (R^{-*}(L) \times \Gamma^*)$. Both s' and s'^- are finitely branching since both s and s^- have this property. As $R^{-*}(L)$ is regular by Corollary 1, the claim follows by Lemma 2. □

Lemma 4. *Let R over Σ be inverse match-bounded for $L \subseteq \Sigma^*$ by c, and let $S = \mathrm{match}_c(R^-)^-$. Then to every infinite derivation $x_0 \to_R x_1 \to_R \cdots$ such that $x_i \in R^{-*}(L)$ for all $i \geq 0$ there is an infinite derivation $x'_0 \to_S x'_1 \to_S \cdots$ such that $x'_i \in S^{-*}(\mathrm{lift}_0(L))$ and $\mathrm{base}(x'_i) = x_i$ for all $i \geq 0$. Therefore, $\mathrm{Inf}(R^* \cap (\Sigma^* \times R^{-*}(L))) = \mathrm{base}(\mathrm{Inf}(S^* \cap ((\Sigma \times \mathbb{N})^* \times S^{-*}(\mathrm{lift}_0(L)))))$.*

Proof. For every finite segment $x_i \to_R x_{i+1} \to_R \cdots \to_R x_j$, $0 \leq i \leq j$, of the given derivation and every witnessing derivation $x_j \to_R^* y \in L$ for $x_j \in R^{-*}(L)$, we construct (by the remark before Definition 2) a derivation $x''_i \to_S x''_{i+1} \to_S \cdots \to_S x''_j \to_S^* \mathrm{lift}_0(y)$ such that $\mathrm{base}(x''_k) = x_k$ for $i \leq k \leq j$. Let the set A_{ij} comprise all strings x''_i that can be obtained this way.

Every set A_{ij} is a subset of $\mathrm{base}^{-1}(x_i) \cap \mathrm{height}^{-1}(\{0, \ldots, c\}^*)$ by inverse match-boundedness, thus finite, and non-empty. We claim that $A_{ij} \supseteq A_{ij'}$ for all $i \leq j \leq j'$. For, every derivation $x''_i \to_S^* x''_{j'} \to_S^* \mathrm{lift}_0(y)$ such that $y \in L$, which witnesses that $x''_i \in A_{ij'}$, can also be read as a derivation $x''_i \to_S^* x''_j \to_S^* \mathrm{lift}_0(y)$, whence it witnesses that $x''_i \in A_{ij}$. Let $A_i = \bigcap_{j \geq i} A_{ij}$, which is therefore a non-empty set. Let us call v a *successor* of u if $u \to_S v$. By definition, every $u \in A_{ij}$ has a successor $v \in A_{i+1,j}$. We now show that every $u \in A_i$ also has a successor

[1] z may depend on y

$v \in A_{i+1}$. There are only finitely many successors of u since $\to_S$ is finitely branching. By the pigeonhole principle one of them, say v, is in infinitely many sets $A_{i+1,j}$, $j > i$. In fact v is then in all of them, hence $v \in A_{i+1}$. Now we can finish the proof by choosing an arbitrary $x'_0 \in A_0$, and using the just proven existence of a successor x'_{i+1} to every x'_i. This yields the wanted infinite derivation. □

Theorem 2. *Let the string rewriting system R over Σ with $\epsilon \notin \mathrm{lhs}(R)$ be inverse match-bounded for the regular language $L \subseteq \Sigma^*$. Then $\mathrm{Inf}(R^* \cap (\Sigma^* \times R^{-*}(L)))$ is effectively regular.*

Proof. Let R be inverse match-bounded for L by c. By Lemma 1 the system $S = \mathrm{match}_c(R^-)^-$ is inverse deleting. Therefore $\mathrm{Inf}(S^* \cap ((\Sigma \times \mathbb{N})^* \times S^{-*}(\mathrm{lift}_0(L))))$ is effectively regular by Lemma 3. By Lemma 4, $\mathrm{Inf}(R^* \cap (\Sigma^* \times R^{-*}(L))) = \mathrm{base}(\mathrm{Inf}(S^* \cap ((\Sigma \times \mathbb{N})^* \times S^{-*}(\mathrm{lift}_0(L)))))$, so $\mathrm{Inf}(R^* \cap (\Sigma^* \times R^{-*}(L)))$ is effectively regular. □

We obtain our previous result as the special case $L = \Sigma^*$. Note that $\epsilon \in \mathrm{lhs}(R)$ and $\epsilon \notin \mathrm{rhs}(R)$ implies $\mathrm{Inf}(R^*) = \Sigma^*$.

Corollary 5 ([6, 8]). *For an inverse match-bounded string rewriting system R, the set $\mathrm{Inf}(R^*)$ is effectively regular.*

6 ANCESTOR GRAPHS AND TERMINATION PROOFS

Kurth introduced in his thesis a criterion that characterizes termination by the well-foundedness of a recursively defined relation on strings. In this section we briefly review this ancestor graph criterion.

Definition 3 ([11]). For a string rewriting system R over alphabet Σ we define the *ancestor graph* (V_R, E_R), where the set of vertices $V_R \subseteq \Sigma^*$ and the set of edges $E_R \subseteq \Sigma^* \times \Sigma^*$ are the least sets such that $\mathrm{lhs}(R) \subseteq V_R$ and, for $u, v \in \Sigma^*$,

- if $v \in V_R$ and $u \to_R v$, then $u \in V_R$ and $(u, v) \in E_R$ *(reduction)*,
- if $r_2 v \in V_R$ and $(\ell \to r_1 r_2) \in R$ for strings $r_1, r_2 \in \Sigma^+$, then $\ell v \in V_R$ and $(\ell v, r_2 v) \in E_R$ *(left extension)*,
- if $v r_1 \in V_R$ and $(\ell \to r_1 r_2) \in R$ for strings $r_1, r_2 \in \Sigma^+$, then $v\ell \in V_R$ and $(v\ell, v r_1) \in E_R$ *(right extension)*,
- if $r_2 \in V_R$ and $(\ell \to r_1 r_2 r_3) \in R$ for strings $r_1, r_2, r_3 \in \Sigma^+$, then $(\ell, r_2) \in E_R$ *(left-right extension)*.

Remark 1. Observe that $\epsilon \in V_R$ is equivalent to $\epsilon \in \mathrm{lhs}(R)$. In this case R is trivially non-terminating, so we can exlude this case without loss of generality.

Remark 2. Kurth's definition addresses the case of one-rule systems $\{\ell \to r\}$, and does not include left-right extension. The latter is needed only in the trivial case where ℓ is a factor of r. For many-rule systems there is no such trivial case, therefore it is necessary to add left-right extension.

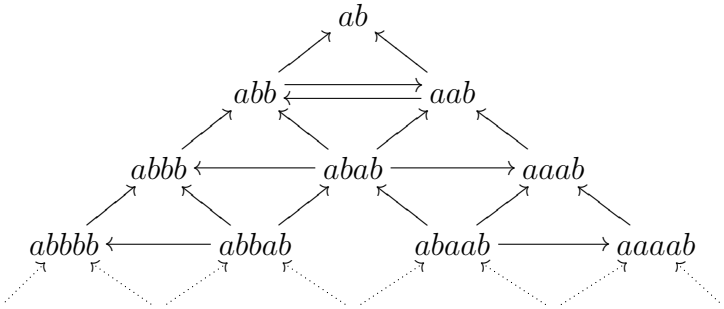
Theorem 3 ([11, Satz 4.24]). *A string rewriting system R is terminating if and only if both $\epsilon \notin \mathrm{lhs}(R)$ and from every node in V_R the length of directed paths in E_R to nodes in $\mathrm{lhs}(R)$ is bounded.*

By a König's Lemma argument for the finitely branching relation E_R we get:

Corollary 6. *A string rewriting system R is terminating if and only if $\epsilon \notin \mathrm{lhs}(R)$ and $\mathrm{Im}(E_R) = \emptyset$.*

Remark 3. Kurth stated and proved Theorem 3 for one-rule string rewriting and claimed that it can be extended to the case of many rules. For a rendering of Kurth's proof in the many-rule case see Appendix A.

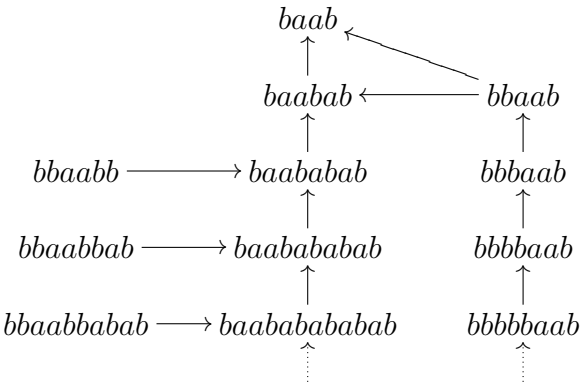
Example 1. The non-terminating system $R = \{ab \to bbaa\}$ has the following ancestor graph, cf Example 3. It contains a cycle, therefore R is non-terminating by Corollary 6.



Example 2 *([11]).* The system $R = \{baab \to aababa\}$ has the ancestor graph (V_R, E_R) with $V_R = ba(ab)^+ \cup b^2a^2b^2(ab)^* \cup b^+ba^2b$ and

$$E_R = \{(b^{i+2}a^2b, b^{i+1}a^2b), (ba(ab)^{i+2}, ba(ab)^{i+1}),$$
$$(b^2a^2b^2(ab)^i, ba(ab)^{i+3}) \mid i \geq 0\} \cup \{b^2a^2b, ba(ab)^2\},$$

illustrated in the following figure. It shows termination of R by Corollary 6.



7 ANCESTOR MATCH-BOUNDED SYSTEMS

In this section we derive a new automated criterion for (non-)termination. We do it by instantiating Theorem 2 with L essentially being the vertex set of Kurth's ancestor graph.

We obtain V_R by inverse rewriting w.r.t. a rewriting system $R_{\langle\rangle}$ constructed from R, using markers "$\langle$" and "$\rangle$" for the left and right end of strings respectively. In the following, let $\Sigma_{\langle\rangle}$ denote $\Sigma \cup \{\langle, \rangle\}$ for alphabets Σ with $\langle, \rangle \notin \Sigma$.

Definition 4. For a string rewriting system R over alphabet Σ with $\langle, \rangle \notin \Sigma$ define the system

$$\begin{aligned} R_{\langle\rangle} = R &\cup \{\ell \to \langle r_2 \mid (\ell \to r_1 r_2) \in R, r_1, r_2 \in \Sigma^+\} \\ &\cup \{\ell \to r_1\rangle \mid (\ell \to r_1 r_2) \in R, r_1, r_2 \in \Sigma^+\} \\ &\cup \{\ell \to \langle r_2\rangle \mid (\ell \to r_1 r_2 r_3) \in R, r_1, r_2, r_3 \in \Sigma^+\} \end{aligned}$$

over alphabet $\Sigma_{\langle\rangle}$. Let $R^-_{\langle\rangle}$ denote $(R_{\langle\rangle})^-$.

The relation E_R and the rewrite relation induced by $R_{\langle\rangle}$ correspond to each other as we elaborate next. Let $h : \Sigma^*_{\langle\rangle} \to \Sigma^*$ be the morphism that deletes the end markers "$\langle$" and "$\rangle$", and keeps all other letters.

Lemma 5. *Let $\epsilon \notin \mathrm{rhs}(R)$. For $x \in \Sigma^*_{\langle\rangle}$ and $y \in \langle^* \cdot V_R \cdot \rangle^*$, if $x \to_{R_{\langle\rangle}} y$ then $x \in \langle^* \cdot V_R \cdot \rangle^*$ and $(h(x), h(y)) \in E_R$.*

Proof. Case analysis on the definition of $R_{\langle\rangle}$. The case $x \to_R y$ is trivial. If $x = u\ell v$ and $y = u\langle r_2 v$ for some $u, v \in \Sigma^*_{\langle\rangle}$ and $(\ell, r_1 r_2) \in R$ then $u \in \langle^*$ and $v \in v'\rangle^*$ for some $v' \in \Sigma^*$. From $h(y) = r_2 v' \in V_R$ we get $h(x) = \ell v' \in V_R$ and $(r_2 v', \ell v') \in E_R$ by left extension. The case where $x = u\ell v$ and $y = u r_1\rangle v$ for some $u, v \in \Sigma^*_{\langle\rangle}$ and $(\ell, r_1 r_2) \in R$ is symmetric by right extension. This leaves the case where $x = u\ell v$ and $y = u\langle r_2\rangle v$ for some $u, v \in \Sigma^*_{\langle\rangle}$ and $(\ell, r_1 r_2) \in R$. We get $u \in \langle^*$ and $v \in \rangle^*$. From $h(y) = r_2 \in V_R$ we get $h(x) = \ell \in V_R$ and $(r_2, \ell) \in E_R$ by left-right extension. □

Lemma 6. *For $x \in \langle^* \cdot \Sigma^+ \cdot \rangle^*$ and $y' \in \Sigma^+$, if $(h(x), y') \in E_R$, then there is $y \in \langle^* \cdot y' \cdot \rangle^*$ such that $x \to_{R_{\langle\rangle}} y$.*

Proof. Case analysis on the definition of the ancestor graph. Let $x \in u \cdot h(x) \cdot v$ for some $u \in \langle^*$ and $v \in \rangle^*$. In each case we have $h(x), y' \in V_R$. We get $x \to_{R_{\langle\rangle}} y$ and $y \in \langle^* \cdot y' \cdot \rangle^*$ by defining

- $y = uy'v$ in the reduction case $h(x) \to_R y'$,
- $y = uy'\rangle v$ in the left extension case $h(x) = \ell v$, $y' = r_2 v$,
- $y = u\langle y' v$ in the right extension case $h(x) = \ell v$, $y' = v r_1$,
- $y = u\langle y'\rangle v$ in the left-right extension case $h(x) = \ell$, $y' = r_2$. □

Lemma 7. *If $\epsilon \notin \mathrm{rhs}(R)$, then $\langle^* \cdot V_R \cdot \rangle^* = R^{-*}_{\langle\rangle}(\langle^* \cdot \mathrm{lhs}(R) \cdot \rangle^*)$.*

Proof. For "$\supseteq$", we note that $\mathrm{lhs}(R) \subseteq V_R$ and that $\langle^* \cdot V_R \cdot \rangle^*$ is closed under rewriting modulo $R^-_{\langle\rangle}$. For "$\subseteq$", we prove that $x \in R^{-*}_{\langle\rangle}(\langle^* \cdot \mathrm{lhs}(R) \cdot \rangle^*)$ for all $x \in V_R$ by induction on the recursive definition of the ancestor graph. □

Lemma 8. *If $\epsilon \notin \mathrm{rhs}(R)$, then*

$$\mathrm{Im}(E_R) = \mathrm{Im}(\to_{R_{\langle\rangle}} \cap (\Sigma^*_{\langle\rangle} \times R^{-*}_{\langle\rangle}(\langle^* \cdot \mathrm{lhs}(R) \cdot \rangle^*))) \cap \Sigma^*.$$

Proof. We show $\langle^* \cdot \mathrm{Im}(E_R) \cdot \rangle^* = \mathrm{Im}(\to_{R_{\langle\rangle}} \cap (\Sigma^*_{\langle\rangle} \times \langle^* \cdot V_R \cdot \rangle^*))$.

To show "$\subseteq$", let $x'_0, x'_1, \ldots$ satisfy $(x'_i, x'_{i+1}) \in E_R$ for all $i \geq 0$. By definition of the ancestor graph we get $x'_i \in V_R$. Starting from $x_0 = x'_0$ one can construct $x_0, x_1, \ldots$ iteratively by Lemma 5 such that $x_i \in \langle^* x'_i \rangle^*$ and $x_i \to_{R_{\langle\rangle}} x_{i+1}$ for all $i \geq 0$.

To show "$\supseteq$", let $x_0, x_1, \ldots$ satisfy $x_{i+1} \in \langle^* \cdot V_R \cdot \rangle^*$ and $x_i \to_{R_{\langle\rangle}} x_{i+1}$ for all $i \geq 0$. By Lemma 5 then $(h(x_i), h(x_{i+1})) \in E_R$. □

Remark 4. As Example 1 shows, E_R may admit cycles even though the rewrite relation induced by $R_{\langle\rangle}$ does not. Hence usually $\mathrm{Inf}(E^*_R) = \mathrm{Inf}(R^*_{\langle\rangle} \cap (\Sigma^*_{\langle\rangle} \times \langle^* \cdot V_R \cdot \rangle^*)) \cap \Sigma^*$ fails to hold.

Definition 5. A string rewriting system R is called *ancestor match-bounded* if $R_{\langle\rangle}$ is inverse match-bounded for $\langle^* \cdot \mathrm{lhs}(R) \cdot \rangle^*$.

Theorem 4. *Uniform termination is decidable for ancestor match-bounded string rewriting systems.*

Proof. Let $R^-_{\langle\rangle}$ be match-bounded for $L = \langle^* \cdot \mathrm{lhs}(R) \cdot \rangle^*$. Match-boundedness implies $\epsilon \notin \mathrm{lhs}(R^-_{\langle\rangle})$, thus $\epsilon \notin \mathrm{rhs}(R_{\langle\rangle})$. And by Remark 1 we may assume $\epsilon \notin \mathrm{lhs}(R)$, which is equivalent to $\epsilon \notin \mathrm{lhs}(R_{\langle\rangle})$.

By Corollary 6, the system R is terminating if and only if $\mathrm{Im}(E_R) = \emptyset$. We know $\mathrm{Im}(E_R) = \mathrm{Im}(\rho) \cap \Sigma^*$ from Lemma 8 for the relation

$$\rho = \to_{R_{\langle\rangle}} \cap (\Sigma^*_{\langle\rangle} \times R^{-*}_{\langle\rangle}(L))$$

on $\Sigma^*_{\langle\rangle}$. Since $R^-_{\langle\rangle}$ is terminating on L, the relation $R^{-+}_{\langle\rangle}$ restricted to $R^{-*}_{\langle\rangle}(L)$ is irreflexive, so the same holds for $R^+_{\langle\rangle}$; as $R_{\langle\rangle}$ is finitely branching, we obtain $\mathrm{Im}(\rho) = \mathrm{Inf}(\rho^*)$. By Theorem 2, $\mathrm{Inf}(\rho^*)$ is effectively regular; note that $\rho^* = R^*_{\langle\rangle} \cap (\Sigma^*_{\langle\rangle} \times R^{-*}_{\langle\rangle}(L))$. Therefore emptiness of $\mathrm{Im}(E_R)$ is decidable. □

Example 3. Here we consider a few one-rule string rewriting systems of the form $\{a^m b^n \to b^p a^q\}$, see [14, 12]. The systems $\{ab \to ba\}$ and $\{ab \to bba\}$ are not ancestor match-bounded since they are not match-bounded on $V_R = a(a+b)^*b$. In contrast, the systems $R = \{ab \to b^n a^m\}$ for $m, n \geq 2$ are ancestor match-bounded by 2, for we have in each case $\mathrm{match}_3(R_{\langle\rangle})^*(\mathrm{lift}_0(\langle^* \cdot V_R \cdot \rangle^*)) = \langle^*_0 \cdot L \cdot \rangle^*_0$ where

$$L = a_0 b_0 + (a_0 + a_1) a_1^+ b_1 + a_1 b_1^+ (b_0 + b_1) + a_1 b_1^* (a_2 + b_1)(a_1 + b_2) a_1^* b_1.$$

Dropping the annotations yields the node set $V_R = ab^*a^*b$. We have $\mathrm{Im}(E^*_R) = ab^*(a+b)a^*b = V_R \setminus \{ab\}$. None of these systems is inverse match-bounded.

For string rewriting systems R over Σ define $>_R = \{(x, y) \mid \exists s, t \in \Sigma^* : x \to_R syt\}$. Then we get:

Lemma 9. *Let $\epsilon \notin \mathrm{rhs}(R)$, $x \in \Sigma^*$, $y \in \Sigma^+$, $u \in \langle^*$, and $v \in \rangle^*$. If $x \to_{R_{\langle\rangle}}^* uyv$ then $x >_R^* y$.*

Proof. By induction on the length k of the given derivation. The case $k = 0$ is trivial, so let $k > 0$ and $x \to_{R_{\langle\rangle}}^{k-1} u'y'v' \to_{R_{\langle\rangle}} uyv$ for some $u' \in \langle^*$, $v' \in \rangle^*$, $y' \in \Sigma^+$. By induction hypothesis, $x >_R^{k-1} y'$. We get $y' \to_R syt$ and so $y' >_R y$ by defining

- $s = \epsilon$, $t = \epsilon$ if the $R_{\langle\rangle}$-rule is from R,
- $s = r_1$, $t = \epsilon$ if the $R_{\langle\rangle}$-rule $\ell \to \langle r_2$ stems from the R-rule $\ell \to r_1 r_2$,
- $s = \epsilon$, $t = r_2$ if the $R_{\langle\rangle}$-rule $\ell \to r_1\rangle$ stems from the R-rule $\ell \to r_1 r_2$,
- $s = r_1$, $t = r_3$ if the $R_{\langle\rangle}$-rule $\ell \to \langle r_2\rangle$ stems from the R-rule $\ell \to r_1 r_2 r_3$. □

Theorem 5. *Every inverse match-bounded system is ancestor match-bounded by at most the same bound.*

Proof. Let $S = \mathrm{match}(R^-)^-$. Note that $S_{\langle\rangle} = \mathrm{match}(R_{\langle\rangle}^-)^-$ where we assume for $S_{\langle\rangle}$ the angular brackets $\langle_0$ and $\rangle_0$. We prove that $x >_S^* y$ for all $x \in (\Sigma \times \mathbb{N})^*$, $y \in (\Sigma \times \{0\})^+$, $u \in \langle_0^*$, and $v \in \rangle_0{}^*$ such that $x \to_{S_{\langle\rangle}}^* uyv$. For this purpose we use Lemma 9 for the system S, the alphabet $\Sigma \times \mathbb{N}$, and the angular brackets $\langle_0$ and $\rangle_0$. □

Example 4. The system $\{babbab \to abbabbba\}$ admits a loop of length 3. It is inverse match-bounded by 3 and ancestor match-bounded by 2.

The converse is false as the following examples show.

Example 5. The system $\{aaab \to aababaa\}$ admits a loop of length 4. It is not inverse match-bounded but ancestor match-bounded by 3.

Example 6. The system $\{baba \to ababbbba\}$ admits a loop of length 4. It is ancestor match-bounded by 2 but not inverse match-bounded.

Example 7. The system $\{babaa \to aaababbab\}$ admits a loop of length 4. It is ancestor match-bounded by 2 but not inverse match-bounded.

Example 8. The system $\{baaa \to aaabba\}$ admits a loop of length 8 but no shorter loop [4]. It is ancestor match-bounded by 3 but not inverse match-bounded.

All one-rule, terminating, ancestor match-bounded string rewriting systems that we have encountered are also inverse match-bounded and RFC-match-bounded. However, the bounds may differ.

Example 9. The system $\{baababa \to aababbaab\}$ is terminating. It is ancestor match-bounded by 2; inverse match-bounded by 3 but not by 2; and RFC-match-bounded by 3 but not by 2.

The following problems are still open:

Problem 1. Are all terminating, ancestor match-bounded string rewriting systems also inverse match-bounded and RFC-match-bounded?

Problem 2. Are all non-terminating, ancestor match-bounded string rewriting systems looping?

8 CONCLUSION

Kurth's ancestor graph criterion characterizes termination by termination on a specific recursively defined language. Such termination criteria have been criticized for their lack of automation. We add automation to this non-automated criterion. In doing so we introduce a new general technique that allows to lift similar characterizations to automated termination criteria. The new technique is amenable if the recursion is aligned with the inverse rewrite relation, like in Kurth's ancestor graph criterion. We have done a similar construction earlier, where the recursion had to be aligned with the rewrite relation. That construction turns Dershowitz's characterization of termination on right hand sides of forward closures into the RFC-match-boundedness criterion.

The new ancestor match-bound criterion complements three related termination criteria:

- match-boundedness for Σ^* [6],
- inverse match-boundedness for Σ^* [8],
- RFC-match-boundedness [7].

All four criteria are implemented in `Matchbox` [13], see

`http://theo1.informatik.uni-leipzig.de/matchbox/`.

The four criteria have in common that they are strong: each applies to string rewriting systems, like Zantema's System, that could not be proven (non-)terminating by automated methods earlier. Each is decidable for a given bound and, where applicable, allows to decide uniform termination. The match-boundedness criteria entail termination whereas the inverse ones also cover some non-terminating systems. This may justify the somewhat greater effort for their correctness proofs. The criteria for Σ^* have decidable termination and normalization problems, and they entail linear upper bounds on the derivational complexity in the terminating case. The restricted criteria (RFC-match-boundedness and ancestor match-boundedness) on the other hand apply to a greater class of systems.

The ancestor match-bound criterion improves upon the inverse match-bound criterion. It covers more non-terminating systems, and obviously the computation for ancestor match-bounds is cheaper. Every terminating ancestor match-bounded system we found appears to be RFC-match-bounded, while the converse is not true. We trace this weakness back to the presence of both the left-extension and the right-extension branch in the definition of ancestor graphs. The definition of forward closures has only one such branch. We do not take the weakness of the ancestor match-bound as a weakness of the underlying technique. It is conceivable to design restrictions of ancestor graphs that characterize termination for certain classes of string rewriting systems.

REFERENCES

[1] J. Berstel. *Transductions and Context-Free Languages*. Teubner, Stuttgart, 1979.

[2] R. V. Book and F. Otto. *String-Rewriting Systems*. Texts Monogr. Comput. Sci.. Springer-Verlag, New York, 1993.

[3] N. Dershowitz. Termination of linear rewriting systems. In S. Even and O. Kariv (Eds.), *Proc. 8th Int. Coll. Automata, Languages and Programming ICALP-81*, *Lecture Notes in Comput. Sci.* Vol. 115, pp. 448–458. Springer-Verlag, 1981.

[4] A. Geser. Loops of superexponential lengths in one-rule string rewriting. In S. Tison (Ed.), *Proc. 13th Int. Conf. Rewriting Techniques and Applications RTA-02*, *Lecture Notes in Comput. Sci.* Vol. 2378, pp. 267–280. Springer-Verlag, 2002.

[5] N. Dershowitz. Termination of rewriting. *J. Symbolic Comput.*, 3(1–2):69–115, 1987.

[6] A. Geser, D. Hofbauer and J. Waldmann. Match-bounded string rewriting systems. In B. Rovan and P. Vojtas (Eds.), *Proc. 28th Int. Symp. Mathematical Foundations of Computer Science MFCS-03*, *Lecture Notes in Comput. Sci.* Vol. 2747, pp. 449-459. Springer-Verlag, 2003.

[7] A. Geser, D. Hofbauer and J. Waldmann. Match-bounded string rewriting systems. NIA Report 2003-09, National Institute of Aerospace, Hampton, VA, USA. Available at `http://research.nianet.org/~geser/papers/nia-matchbounded.html`.

[8] A. Geser, D. Hofbauer and J. Waldmann. Termination proofs for string rewriting systems via inverse match-bounds. NIA Report 2003-XX, National Institute of Aerospace, Hampton, VA, USA. Available at `http://research.nianet.org/~geser/papers/nia-inverse.html`.

[9] T. N. Hibbard. Context-limited grammars. *J. ACM*, 21(3):446–453, 1974.

[10] D. Hofbauer and J. Waldmann. Deleting string rewriting systems preserve regularity. In Z. Ésik and Z. Fülöp (Eds.), *Proc. 7th Int. Conf. Developments in Language Theory DLT-03*, *Lecture Notes in Comput. Sci.* Vol. 2710, pp. 337-348. Springer-Verlag, 2003.

[11] W. Kurth. Termination und Konfluenz von Semi-Thue-Systemen mit nur einer Regel. Dissertation, Technische Universität Clausthal, Germany, 1990.

[12] G. Sénizergues. On the termination problem for one-rule semi-Thue systems. In H. Ganzinger (Ed.), *Proc. 7th Int. Conf. Rewriting Techniques and Applications RTA-96*, *Lecture Notes in Comput. Sci.* Vol. 1103, pp. 302–316. Springer-Verlag, 1996.

[13] J. Waldmann. `Matchbox`: A tool for match-bounded string rewriting. System description submitted to RTA-04.

[14] H. Zantema and A. Geser. A complete characterization of termination of $0^p1^q \to 1^r0^s$. *Appl. Algebra Engrg. Comm. Comput.*, 11(1):1–25, 2000.

[15] H. Zantema. Termination. In Terese, *Term Rewriting Systems*, pp. 181–259, Cambridge University Press, 2003.

A KURTH'S PROOF OF THE ANCESTOR GRAPH CRITERION

We render Kurth's proof of Theorem 3 with small changes to account for many-rule systems and to improve the presentation.

Let an infinite derivation $w_0 \to_R w_1 \to_R \cdots$ be given. We designate positions in this derivation by pairs (p, i) where p is a position in w_i. Positions (p, i) and $(p', i+1)$ are called residuals of each other if there are $x, y \in \Sigma^*$ and $(\ell \to r) \in R$ such that $w_i = x\ell y$ and $w_{i+1} = xry$, and either $0 \leq p' = p < |x|$ or both $|x\ell| < p \leq |x\ell y|$ and $p' = p - |\ell| + |r|$. Note that $p \notin \{|x|, |x\ell|\}$. The residual relation is extended to an equivalence relation. For every j and (p, i), there is at most one residual (p', j) of (p, i).

Definition 6 ([11, Definition 4.25]). The *chain graph* (V, E) of an infinite derivation $x_0 \to_R x_1 \to_R \cdots$ is defined as follows. The node set V is the set of all residuals of redex positions of the derivation. The edge set $E = E_0 \cup E_1$ is the least set such that

- if $(p', i-1)$ is the residual of $(p, i) \in V$, then $((p', i-1), (p, i)) \in E_0$;
- if $(p', i-1)$ is the redex position of the step $x_{i-1} \to_R x_i$, and $(p, i) \in V$ has no residual in x_{i-1}, then $((p', i-1), (p, i)) \in E_1$.

By definition, at each node we have a copy of the residual redex.

Lemma 10 ([11, Hilfssatz 4.26]). *The chain graph of an infinite derivation contains a path with infinitely many edges in E_1.*

Proof. One observes that the chain graph is a finitely branching forest of finitely many trees. By the pigeonhole principle, one of the trees must be infinite. By König's Lemma it has an infinite path. If this path has only finitely many edges in E_1 then there is an infinite E_0-path, a contradiction to the definition of V. So it has infinitely many edges in E_1. □

We will call a path in the chain graph a *chain path*, and a path in the ancestor graph an *ancestor path* for short.

Lemma 11 ([11, Hilfssatz 4.27]). *Let $w_0 \to_R w_1 \to_R \cdots$ be an infinite derivation. Then to every chain path that has $k+1$ edges in E_1 there is an ancestor path of length greater than, or equal to, k from a factor of w_0 to a left hand side of R.*

Proof. Let $(p_0, 0), (p_1, 1), \ldots, (p_{n+1}, n+1)$ denote the starting segment of the given chain path, where $n \geq k+1$ is chosen such that $((p_n, n), (p_{n+1}, n+1))$ is the $k+1$-th edge in E_1.

We are going to show by induction on $n - j$ that for every $0 \leq j \leq n$ there is an ancestor path from a factor of w_j that encompasses the residual redex at (p_j, j). The lemma then follows for $j = 0$.

The claim holds trivially for $j = n$, so assume $j < n$. By induction hypothesis, there is an ancestor path from a factor w of w_{j+1} that encompasses the residual redex at $(p_{j+1}, j+1)$.

Case 1: $((p_j, j), (p_{j+1}, j+1)) \in E_0$.

Case 1.1: The contractum of step $w_j \to_R w_{j+1}$ and the factor w neither overlap, nor is one a factor of the other. Then the factor w stays unchanged by the step $w_j \to_R w_{j+1}$, and we keep the ancestor path constructed so far.

Case 1.2: The contractum of step $w_j \to_R w_{j+1}$ and the factor w either overlap, or one is a factor of the other. This induces a factor $w' \in V_R$ in w_j and an edge (w', w) that we can add at the front of the ancestor path.

Case 2: $((p_j, j), (p_{j+1}, j+1)) \in E_1$. Then the contractum of step $w_j \to_R w_{j+1}$ and the factor w either overlap, or one is a factor of the other. This induces a factor $w' \in V_R$ in w_j and an edge (w', w) that we can add at the front of the ancestor path. □

The proof of Theorem 3 is now finished as follows: Suppose there is an infinite derivation $w_0 \to_R w_1 \to_R \cdots$. By Lemma 10, there is a chain path that has infinitely many E_1-edges. So this chain path satisfies the premises of Lemma 11 for every $k \geq 0$ whence there is an infinite forest of ancestor paths starting from factors of w_0. Since there are only finitely many factors in w_0, this is a forest of finitely many trees, so one of the trees must be infinite by the pigeonhole principle. Hence the length of ancestor paths starting from one factor of w_0 is unbounded.