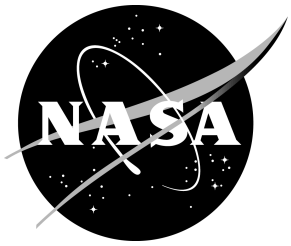
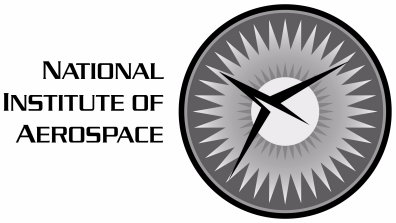
NASA/CR-2004-213032
NIA Report No. 2004-02





Termination Proofs for String Rewriting Systems via Inverse Match-Bounds

Alfons Geser
National Institute of Aerospace, Hampton, Virginia

Dieter Hofbauer
University of Kassel, Kassel, Germany

Johannes Waldmann
Polytechnic University of Leipzig, Leipzig, Germany

July 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

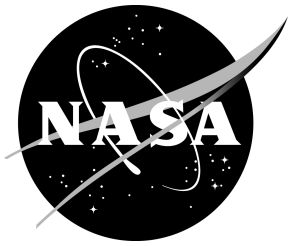
- Email your question via the Internet to help@sti.nasa.gov

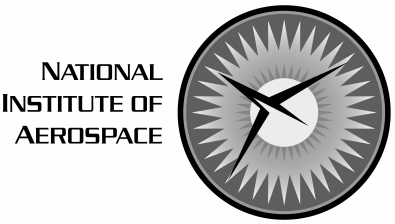
- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2004-213032
NIA Report No. 2004-02





Termination Proofs for String Rewriting Systems via Inverse Match-Bounds

Alfons Geser
National Institute of Aerospace, Hampton, Virginia

Dieter Hofbauer
University of Kassel, Kassel, Germany

Johannes Waldmann
Polytechnic University of Leipzig, Leipzig, Germany

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NCC-1-02043

July 2004

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

TERMINATION PROOFS FOR STRING REWRITING SYSTEMS VIA INVERSE MATCH-BOUNDS*

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

ABSTRACT

Annotating a letter by a number, one can record information about its history during a reduction. A string rewriting system is called match-bounded if there is a global upper bound to these numbers. In earlier papers we established match-boundedness as a strong sufficient criterion for both termination and preservation of regular languages.

We show now that the string rewriting systems whose inverse (left and right hand sides exchanged) is match-bounded, also have exceptional properties, but slightly different ones. Inverse match-bounded systems effectively preserve context-free languages; their sets of normalizable strings and their sets of immortal strings are effectively regular. These sets of strings can be used to decide the normalization, the termination and the uniform termination problems of inverse match-bounded systems.

We also show that the termination problem is decidable in linear time, and that a certain strong reachability problem is decidable, thus solving two open problems of McNaughton's.

Inverse match-boundedness, unlike match-boundedness, does not entail termination. Like match-bounds, inverse match-bounds prove linear derivational complexity in the terminating case.

1 INTRODUCTION

The termination and uniform termination problems are undecidable for string rewriting systems (also called semi-Thue systems).

The two problems amount to the membership and emptiness problems for the set of immortal strings, i.e., strings that initiate an infinite derivation. For any class of string rewriting systems where this set is effectively a regular language, the two problems are decidable. This is the basis of a new automated termination criterion.

By annotating a letter with a number, one can record information about its history during a reduction. A string rewriting system is called *match-bounded* if there is an upper bound to these numbers. In an earlier paper [8, 9], we showed that match-bounded string rewriting systems terminate, have linear derivational complexity, and preserve regular languages. A match-bounded system can be encoded into a *deleting* system [12], and the descendant

*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`.

relation of the latter can be decomposed into components that are known to preserve regular languages.

In the present article we focus on string rewriting systems whose *inverse* (left and right hand sides of the rules exchanged) is match-bounded. Such systems cover classes of systems like McNaughton's inhibitor systems and Ginsburg and Greibach's terminal bounded grammars. We show that the set of normalizing strings is effectively regular for inverse match-bounded systems, whence it is decidable whether such systems are normalizing. We also show that certain strong rechability problems are decidable, solving an open problem of McNaughton's.

Our main achievement is to prove that the set of immortal strings is effectively regular for inverse match-bounded systems. As mentioned above, we thus prove that the termination and the uniform termination problems are decidable for inverse match-bounded systems. Again we use the decomposition result for deleting systems. For inverse deleting systems we get preservation of context-free languages. This is sufficient, as we can show, to construct a finite automaton that recognizes the set of immortal strings. Its membership problem is decidable in linear time, solving another open problem of McNaughton's.

Inverse match-bounded systems, unlike match-bounded systems, need not terminate. If they terminate, their derivational complexity is linear.

The article is organized as follows. After recalling definitions and essential results on deleting and match-bounded rewriting systems in Section 3 and 4, we study inverse match-boundedness in Section 5 (normalization properties, preservation of context-freeness, reachability properties) and Section 6 (termination properties). Derivational complexity is investigated in Section 7. Finally, we shortly describe an implementation of our algorithms in Section 8.

We have presented some of the results reported here at the 28th International Symposium on Mathematical Foundations of Computer Science MFCS 2003 at Bratislava, Slovak Republic [8].

2 PRELIMINARIES

Standard notations for strings and string rewriting can be found, for instance, in [3]. A *string rewriting system* over an alphabet Σ is a relation $R \subseteq \Sigma^* \times \Sigma^*$, inducing 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 (resp. right) 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 $(x_0, x_1, \dots)$ with $x_i \to_R x_{i+1}$ for all i. Define the set of *immortal* strings $\mathrm{Im}(R)$ as the set of all $x_0 \in \Sigma^*$ that initiate an infinite R-derivation. We call R *terminating on* $L \subseteq \Sigma^*$ if $\mathrm{Im}(R) \cap L = \emptyset$. If R is terminating on Σ^*, we say that R is *terminating*. In order to classify lengths of derivations for terminating systems, define the *derivation height* function modulo R on Σ^* by $\mathrm{dh}_R(x) = \max\{n \in \mathbb{N} \mid \exists y \in \Sigma^* : x \to_R^n y\}$. The *derivational complexity* of R is defined as the function $n \mapsto \max\{\mathrm{dh}_R(x) \mid |x| \leq n\}$ on $\mathbb{N}$.

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.

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 $\mathrm{Inf}(\rho) = \emptyset$.

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 [2].

For a relation $\rho \subseteq \Sigma^* \times \Sigma^*$ 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 $(a,c) \notin (R|_\Delta)^*$.

A relation $s \subseteq \Sigma^* \times \Gamma^*$ is a *substitution* if $s(\epsilon) = \{\epsilon\}$ and $s(xy) = s(x)s(y)$ for $x, y \in \Sigma^*$, so 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 $a \in \Sigma$. For instance, if $\mathcal{L}$ is the family of finite (context-free) languages, then s is a *finite* (resp. *context-free*) *substitution*. If $\epsilon \notin s(a)$ for every $a \in \Sigma$, then s is *epsilon-free*. Note that a finite substitution is finitely branching, and the same holds for the inverse of a finite and epsilon-free substitution.

3 DELETING STRING REWRITING SYSTEMS

In this section we shortly recall definitions and results regarding *deleting* string rewriting systems [12], a topic that can be traced back to Hibbard [11]. This class of string rewriting systems enjoys a strong decomposition property. As an immediate consequence, deleting systems preserve regularity of languages, and inverse deleting systems preserve context-freeness. All these results will be frequently used in the sequel. For proofs and for a description of the decomposition algorithm we refer to [12].

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

Furthermore, we have the following effective decomposition result.

Theorem 1 ([12]). *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 ([11, 12]). *Every inverse deleting string rewriting system effectively preserves context-free languages.*

Corollary 2 ([12]). *Every deleting string rewriting system effectively preserves regularity.*

In the present paper, we will frequently refer to a slightly specialized version of the above 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 as a direct consequence we get $R^{-*} = R^{*-} = (s \circ c^-)^- = c \circ s^-$.

Proof. By Theorem 1 we have $R^* = (s \circ C^{-*}) \cap (\Sigma^* \times \Sigma^*)$, where $s \subseteq \Sigma^* \times \Gamma^*$ is a finite substitution, and C is a context-free string rewriting system over Γ. Reviewing the construction in [12], we find that s is epsilon-free, and that no ϵ can occur on either side of C, thus $C^* \cap (\Sigma^* \times \Gamma^*)$ coincides with an epsilon-free context-free substitution $c \subseteq \Sigma^* \times \Gamma^*$. Note that $C^{-*} = C^{*-}$. □

Remark 1. The assumption $\epsilon \notin \mathrm{rhs}(R)$ in Corollary 3 cannot be dropped. Consider the example $R = \{aa \to \epsilon\}$, and assume $R^* = s \circ c^-$ for substitutions s and c. We have $c(\epsilon) = \{\epsilon\}$, since c is a substitution. Now, $(aa, \epsilon) \in R^*$ implies $(aa, \epsilon) \in s$, thus $\epsilon \in s(a)$. But then we have $(a, \epsilon) \in s \circ c^-$, contradicting $(a, \epsilon) \notin R^*$. Note that $R = \{a \to \epsilon\}$ is not a counterexample, as in this case $R^* = s$ for the substitution $s : a \mapsto \{a, \epsilon\}$.

4 MATCH-BOUNDED STRING REWRITING SYSTEMS

The theory of deleting systems can be applied to obtain results for *match-bounded* rewriting. A derivation is match-bounded if dependencies between rule applications are limited. To make this precise, we annotate positions in strings by natural numbers that indicate their *match height*. Positions in a reduct will get height $h+1$ if the minimal height of all positions in the corresponding redex was h. In this section, we summarize essential results from [8, 9].

Given an 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

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

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, for $n \in \mathbb{N}$ we have $R^n = \mathrm{lift}_0 \circ \mathrm{match}(R)^n \circ \mathrm{base}$, thus $R^* = \mathrm{lift}_0 \circ \mathrm{match}(R)^* \circ \mathrm{base}$.

It is convenient to compare the height vectors of strings that have the same base. For $u, v \in (\Sigma \times \mathbb{N})^*$ we write $u \geq v$ if $\mathrm{base}(u) = \mathrm{base}(v)$ and $\mathrm{height}(u) \geq_n \mathrm{height}(v)$, where $\geq_n$ denotes the pointwise greater-or-equal ordering on $\mathbb{N}^n$. The relations $\geq$ and $\to_{\mathrm{match}(R)}$ commute:

Lemma 1 ([9]). $\geq \circ \to_{\mathrm{match}(R)} \subseteq \to_{\mathrm{match}(R)} \circ \geq$.

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, respectively) 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 $\mathrm{match}(R)$. Note that $\max(\mathrm{height}(x)) \leq c$ is equivalent to $x \leq \mathrm{lift}_c(\mathrm{base}(x))$. Obviously, a system that is match-bounded for L is also match-bounded for any subset of L by the same bound. Further, by Lemma 1, 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 $\Sigma \times \{0, 1, \ldots, c\}$.

Lemma 2 ([9]). *If R is match-bounded for L by c, then for $n \in \mathbb{N}$, $R^n|_L = (\mathrm{lift}_0 \circ \mathrm{match}_c(R)^n \circ \mathrm{base})|_L$, thus $R^*|_L = (\mathrm{lift}_0 \circ \mathrm{match}_c(R)^* \circ \mathrm{base})|_L$.*

Lemma 3 ([9]). *For all $c \in \mathbb{N}$, the system $\mathrm{match}_c(R)$ is deleting.*

(Dually we know that if R is deleting, then R 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. Further, Corollary 2 implies that match-bounded systems preserve regularity, therefore match-boundedness by a given bound is decidable. These results are stated here for later reference.

Theorem 2 ([9]). *If R is match-bounded for L, then R is terminating on L.*

Proposition 2 ([9]). *Every match-bounded string rewriting system has linear derivational complexity.*

Theorem 3 ([9]). *If R is match-bounded for a regular language L, then $R^*(L)$ is effectively regular.*

Theorem 4 ([9]). *The following problem is decidable:*

GIVEN: *A string rewriting system R; a regular language L; $c \in \mathbb{N}$.*

QUESTION: *Is R match-bounded for L by c?*

5 INVERSE MATCH-BOUNDED STRING REWRITING SYSTEMS

Our focus in this article is on string rewriting systems that are inverse match-bounded, i.e., systems obtained from match-bounded ones by exchanging left and right hand sides of rules. Inverse match-bounded systems have both interesting applications and nice decidability properties. In this section, we show that the set of normalizing strings can be effectively determined for this class of rewriting systems, therefore normalization becomes decidable. Further, since context-free languages are preserved, we get decidability of a rather strong version of the reachability problem.

Example 1. Peg solitaire is a one-person game. The objective is to remove pegs from a board. A move consists of one peg X hopping over an adjacent peg Y, landing on the empty space on the opposite side of Y. After the hop, Y is removed. Peg solitaire on a one-dimensional board corresponds to the string rewriting system

$$P = \{■■□ \to □□■,\ □■■ \to ■□□\},$$

where ■ stands for "peg", and □ for "empty". One is interested in winning positions, i.e., the language of all positions that can be reduced to one single peg, which is $P^{-*}(□^*■□^*)$. Regularity of $P^{-*}(□^*■□^*)$ is a "folklore theorem", see [15] for its history. In particular, this was shown by Ravikumar [16] who considered *change bounds*, a concept that is closely related to match bounds, but only applicable to length-preserving string rewriting systems. The system P is inverse match-bounded by 2, so we obtain yet another proof of that result. Note that in this example, P and its inverse P^- are isomorphic due to the strong symmetry of the system, which is uncommon.

Example 2. McNaughton [14] introduced a class of string rewriting systems that has good decidability properties. A system R is called an *inhibitor system*, if there is a letter $\iota \notin \Sigma$, the *inhibitor*, such that $\ell \in \Sigma^+$ and $r \in (\Sigma \cup \{\iota\})^* \setminus \Sigma^*$ for every rule $\ell \to r$ in R. Each inhibitor system is inverse deleting for the ordering that makes the inhibitor ι greater than every other letter, hence it is inverse match-bounded by 1. For example, the inhibitor system $R = \{baa \to aa\iota babba\}$ is inverse match-bounded by 1 as height 2 does not occur in $\mathrm{match}_2(R^-)^*(\{a_0, b_0, \iota_0\}^*) = (\{a_0, \iota_0, b_0\} \cup b_1\{a_1, b_1\}^* a_1^2)^*$.

5.1 Normalization Properties

A string is called *normalizing* if it has a descendant that is in normal form. A string rewriting system is called normalizing if every string is.

Theorem 5. *For an inverse match-bounded string rewriting system, the set of normalizing strings is effectively regular.*

Proof. The set of normal forms is $\mathrm{NF}(R) = \Sigma^* \setminus (\Sigma^* \cdot \mathrm{lhs}(R) \cdot \Sigma^*)$, which is therefore regular. Hence by Theorem 3, the set of normalizing strings, $R^{-*}(\mathrm{NF}(R))$, is effectively regular. □

Note that a system R is normalizing if and only if $R^{-*}(\mathrm{NF}(R)) = \Sigma^*$, so normalization of inverse match-bounded systems is decidable.

Corollary 4. *The following problem is decidable:*

GIVEN: *An inverse match-bounded string rewriting system R.*

QUESTION: *Is R normalizing?*

Example 3. The system $R = \{b^2ab^3 \to ab^6a\}$ is normalizing although it admits the loop $b^2ab^6 \to_R ab^6ab^3 \to ab^4ab^6a$ [7]. We get $\mathrm{match}_2(R^-)^*(\{a_0, b_0\}^*) = (\{a_0, b_0\} \cup (b_1^2)^+ a_1 (b_1^3)^+)^*$, hence R is inverse match-bounded by 1. The construction of $R^{-*}(\mathrm{NF}(R)) = \Sigma^*$ by Theorem 5 yields another proof that R is normalizing.

5.2 Preserving Context-freeness

Inverse match-bounded systems preserve context-free languages. This result will be used in Section 5.3 and in Section 6.

Theorem 6. *For a context-free language L, if a string rewriting system R is inverse match-bounded for $R^*(L)$ then $R^*(L)$ is context-free.*

Proof. Let $\bar{L} = R^*(L)$, and let R^- be match-bounded for $\bar{L}$ by c. For $\rho = \mathrm{lift}_0 \circ \mathrm{match}_c(R^-)^* \circ \mathrm{base}$ we get $R^{-*}|_{\bar{L}} = \rho|_{\bar{L}}$ by Lemma 2. Intersection with $\bar{L} \times \bar{L}$ on both sides of this equation yields $R^{-*} \cap (\bar{L} \times \bar{L}) = \rho \cap (\bar{L} \times \bar{L})$, which by $R^{-*} = R^{*-}$ is equivalent to $R^* \cap (\bar{L} \times \bar{L}) = \rho^- \cap (\bar{L} \times \bar{L})$. We have $R^*(\bar{L}) \subseteq \bar{L}$, and by definition of ρ one proves $\rho^-(\bar{L}) \subseteq \bar{L}$, hence $R^*|_{\bar{L}} = \rho^-|_{\bar{L}}$, so $R^*(L) = \rho^-(L)$ from $L \subseteq \bar{L}$.

The system $\mathrm{match}_c(R^-)$ is deleting by Lemma 3, so $\mathrm{match}_c(R^-)^{-*}$ effectively preserves context-freeness by Corollary 1. Also the inverse morphisms base^- and lift_0^- effectively preserve context-freeness, so the same is true for $\rho^- = \mathrm{base}^- \circ \mathrm{match}_c(R^-)^{-*} \circ \mathrm{lift}_0^-$. □

Typically, one would choose a suitable regular language $M \supseteq R^*(L)$ in order to prove that R is inverse match-bounded for M by Theorem 4. The obvious choice $M = \Sigma^*$ leads to the following corollary.

Corollary 5. *Every inverse match-bounded string rewriting system effectively preserves context-free languages.*

Remark 2. Under the weaker assumption that R is inverse match-bounded just for the context-free language L we cannot guarantee that $R^*(L)$ is again context-free. This is shown by the following example. Consider the rewrite system $R = \{\epsilon \to abc, ac \to ca, ab \to ba, bc \to cb\}$ over alphabet $\{a, b, c\}$ and the language $L = \{\epsilon\}$. It is not difficult to see that $R^*(L) \cap c^*b^*a^* = \{c^n b^n a^n \mid n \geq 0\}$, a language that is well-known to be not context-free. The system R, however, is inverse match-bounded for L (by 0, since L contains normal forms modulo R^- only).

Example 4. Ginsburg and Greibach [10] have shown that terminal bounded grammars generate context-free languages. Rules of this type of grammars have a non-empty string of nonterminals as left hand side, and at least one occurrence of a terminal letter in the right hand side. Therefore, a terminal bounded grammar R is inverse match-bounded by 1, since terminal letters always have height 0 in R^--derivations. So Corollary 5 provides another proof of this result from [10].

Example 5. The system $R = \{ab \to da, ac \to acc\}$ is inverse match-bounded by 1. For the context-free language $L = \{ab^n c^n \mid n \geq 1\}$ we get the context-free language $R^*(L) = \{d^{n_1} ab^{n_2} c^n \mid n_1, n_2 \geq 0, n_1 + n_2 = n \geq 1\} \cup \{d^n ac^m \mid m > n \geq 1\}$.

5.3 Reachability Properties

We have seen that inverse match-bounded string rewriting systems effectively preserve context-freeness. As an application, we obtain decidability of a strong version of the reachability problem. The reachability problem for a class $\mathcal{C}$ of string rewriting systems is defined as:

GIVEN: A string rewriting system $R \in \mathcal{C}$; two strings x and y.

QUESTION: Does $x \to_R^* y$ hold?

For the class $\mathcal{C}$ of terminating systems, reachability is easily decided by generating the finite tree of derivations starting from x and checking whether y appears in this tree. By symmetry one solves the problem for the class of inverse terminating systems, which includes the inverse match-bounded systems. This is generalized considerably by the following result.

Theorem 7. *The following problem is decidable:*

GIVEN: *An inverse match-bounded string rewriting system R; a context-free language L; a regular language M.*

QUESTION: *Does $\exists x \in L\ \exists y \in M : x \to_R^* y$ hold?*

Proof. The question is equivalent to $R^*(L) \cap M \neq \emptyset$, so decidability is a consequence of the fact that $R^*(L)$ is effectively context-free by Corollary 5. □

Note that this cannot be extended to context-free languages M, even in the special case where $R = \emptyset$. The reason is that the question whether $L \cap M = \emptyset$ holds for given context-free languages L and M is undecidable.

Corollary 6. *The following problem is decidable:*

GIVEN: *An inverse match-bounded string rewriting system R over Σ; two strings $x, y \in \Sigma^*$.*

QUESTION: *Does $\exists u, v \in \Sigma^* : x \to_R^* uyv$ hold?*

Proof. Employ Theorem 7 for $L = \{x\}$ and $M = \Sigma^*\{y\}\Sigma^*$. □

Example 6. McNaughton [14] shows that the reachability problem is decidable for the class of inhibitor systems, cf. Example 2. An alternative proof for a more general result is by Theorem 7. We can even affirmatively solve the following open problem posed by McNaughton [14] as Open Question 4: Is the following problem decidable: GIVEN: An inhibitor string rewriting system R over Σ; strings $x, y \in \Sigma^*$. QUESTION: Does $\exists u, v \in \Sigma^* : x \to_R^* uyv$ hold?

6 DECIDABLE TERMINATION PROPERTIES

In this section, we prove that termination is decidable for inverse match-bounded systems. This is done by showing that the set of immortal strings is regular for inverse deleting, and so for inverse match-bounded systems. Examples illustrate our approach.

Lemma 4. *Let $c \subseteq \Sigma^* \times \Gamma^*$ be a substitution, and let K be a regular language over Γ. Then $\mathrm{Inf}(c \cap (\Sigma^* \times K))$ is regular.*

Proof. Consider a finite automaton A with state set Q that accepts K. For $p, q \in Q$, denote by $L(A, p, q)$ the set of strings x for which there is a path $p \xrightarrow{x} q$ in A. We define an automaton B over alphabet $\Sigma \times \{F, I\}$ as follows. The sets of states, initial states, and final states of B and A coincide. For $p, q \in Q$ and $a \in \Sigma$, B contains the transition

- $p \xrightarrow{(a,I)} q$ iff the language $c(a) \cap L(A, p, q)$ is infinite,

- $p \xrightarrow{(a,F)} q$ iff the language $c(a) \cap L(A, p, q)$ is finite and non-empty.

We claim that $x = a_1 \dots a_n \in \mathrm{Inf}(c \cap (\Sigma^* \times K))$ for $a_i \in \Sigma$ and $n > 0$ if and only if there is an accepting path in B that is labelled by $(a_1, b_1) \dots (a_n, b_n)$ where at least one b_i equals I. (Note that $c(\epsilon) = \{\epsilon\}$, thus $\epsilon \notin \mathrm{Inf}(c)$.) This can be seen as follows.

By definition, $x \in \mathrm{Inf}(c \cap (\Sigma^* \times K))$ if and only if $c(x) \cap K$ is infinite. Each string $y \in c(x)$ is of the form $y = y_1 \dots y_n$ with $y_i \in c(a_i)$. Thus $y \in c(x) \cap K$ if and only if there is a path $q_0 \xrightarrow{y_1} q_1 \xrightarrow{y_2} \cdots \xrightarrow{y_n} q_n$ in A accepting y, i.e., with q_0 initial and q_n final. Denote by P the set of all such sequences $q_0 \dots q_n$. By construction, we have $y_i \in c(a_i) \cap L(A, q_{i-1}, q_i)$, and

$$c(x) \cap K = \bigcup_{q_0 \dots q_n \in \mathrm{P}} \big(c(a_1) \cap L(A, q_0, q_1)\big) \cdot \ldots \cdot \big(c(a_n) \cap L(A, q_{n-1}, q_n)\big).$$

A finite union of languages is infinite if at least one summand is infinite, and a finite product of languages is infinite if no factor is empty and at least one factor is infinite. Thus $c(x) \cap K$ is infinite if and only if there is a sequence $q_0 \dots q_n \in \mathrm{P}$ such that each $c(a_i) \cap L(A, q_{i-1}, q_i)$ is non-empty and at least one $c(a_i) \cap L(A, q_{i-1}, q_i)$ is infinite. This is equivalent to the existence of a sequence $b_1 \dots b_n \in \{F, I\}^n \setminus F^n$ such that $(a_1, b_1) \dots (a_n, b_n) \in L(B)$. Therefore,

$$\mathrm{Inf}(c \cap (\Sigma^* \times K)) = \pi(L(B) \setminus (\Sigma \times \{F\})^*)$$

where $\pi : (\Sigma \times \{I, F\})^* \to \Sigma^*$ is the morphism induced by $\pi : (a, b) \mapsto a$. □

Lemma 5. *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.*

Proof. We have $\mathrm{Inf}(c \circ T) \subseteq \mathrm{Inf}(c \cap (\Sigma^* \times T^-(\Delta^*)))$ because T is finitely branching, and $\mathrm{Inf}(c \circ T) \supseteq \mathrm{Inf}(c \cap (\Sigma^* \times T^-(\Delta^*)))$ because T^- is finitely branching. Thus $\mathrm{Inf}(c \circ T) = \mathrm{Inf}(c \cap (\Sigma^* \times T^-(\Delta^*)))$, and we conclude by Lemma 4. □

Remark 3. The regularity results in Lemma 4 and Lemma 5 are effective if c is an $\mathcal{L}$-substitution for a family $\mathcal{L}$ of languages that is closed under intersection with regular sets, and for which emptiness and finiteness are decidable. This is the case, e.g., for the family of context-free languages, as used in the proof of Lemma 6 below.

Lemma 6. *For an inverse deleting string rewriting system R, the set $\mathrm{Inf}(R^*)$ is effectively regular.*

Proof. Let R be a system over alphabet Σ such that R^- is deleting. If $\epsilon \in \mathrm{lhs}(R)$ then $\mathrm{Inf}(R^*) = \Sigma^*$, so we may assume $\epsilon \notin \mathrm{lhs}(R)$. Then $R^* = c \circ s^-$ by Corollary 3, where c is a context-free substitution and s is a finite epsilon-free substitution. The claim follows by Lemma 5 and Remark 3. □

Lemma 7. *For an inverse deleting string rewriting system R, we have $\mathrm{Im}(R) = \mathrm{Inf}(R^*)$.*

Proof. A finitely branching binary relation ρ is well-founded if and only if ρ^* is finitely branching and ρ^+ is irreflexive. Because R^- is deleting, R^{-+} is well-founded and hence irreflexive; so $\mathrm{Im}(R) = \mathrm{Inf}(R^*)$. □

Lemma 8. *Let the string rewriting system R be inverse match-bounded by c, and let $S = \mathrm{match}_c(R^-)^-$. Then to every infinite derivation $x_0 \to_R x_1 \to_R \cdots$ there is an infinite derivation $x'_0 \to_S x'_1 \to_S \cdots$ such that $\mathrm{base}(x'_i) = x_i$ for all $i \geq 0$. Therefore, $\mathrm{Im}(R) = \mathrm{base}(\mathrm{Im}(S))$.*

Proof. For every finite initial segment $x_0 \to_R x_1 \to_R \cdots \to_R x_j$ of the given derivation, we construct (by the remark before Definition 2) a derivation $x_{0j} \to_S x_{1j} \to_S \cdots \to_S x_{jj} = \mathrm{lift}_0(x_j)$ such that $\mathrm{base}(x_{ij}) = x_i$. By induction on $j - i$, using the implication

$$x_{ij'} \geq x_{ij} \leftarrow_S x_{i-1,j} \quad \Rightarrow \quad x_{ij'} \leftarrow_S x_{i-1,j'} \geq x_{i-1,j}$$

for the inductive step, we have $x_{ij} \leq x_{ij'}$ for all $0 \leq i \leq j \leq j'$. Define x'_i as the maximum of the finite set $\{x_{ij} \mid j \geq i\} \subseteq \mathrm{base}^{-1}(x_i) \cap \mathrm{height}^{-1}(\{0, \ldots, c\}^*)$. Then for all $i \geq 0$ we have $x'_i \to_S x'_{i+1}$ since there is an index $j > i$ such that $x'_i = x_{ij} \to_S x_{i+1,j} = x'_{i+1}$. □

Theorem 8. *For an inverse match-bounded string rewriting system R, the set $\mathrm{Im}(R)$ is effectively regular.*

Proof. By Lemma 3, the system $S = \mathrm{match}_c(R^-)^-$ is inverse deleting. Therefore $\mathrm{Im}(S)$ is regular by Lemmas 6 and 7, thus also $\mathrm{Im}(R)$ is regular as $\mathrm{Im}(R) = \mathrm{base}(\mathrm{Im}(S))$ by Lemma 8. □

Corollary 7. *Let $\mathcal{L}$ be a family of languages that is closed under intersection with regular sets, and for which emptiness is decidable. Then the following problem is decidable:*

GIVEN: *A language $L \in \mathcal{L}$; an inverse match-bounded string rewriting system R.*

QUESTION: *Is R terminating on L?*

Proof. By Theorem 8, $\mathrm{Im}(R)$ is regular, so emptiness of $\mathrm{Im}(R) \cap L$ is decidable. □

Corollary 8. *Termination and uniform termination are decidable for the class of inverse match-bounded string rewriting systems.*

Proof. Choose $L = \{x\}$ to decide whether there is an infinite derivation starting from string x, and choose $L = \Sigma^*$ to decide uniform termination. □

Example 7. McNaughton [14] shows that termination and uniform termination are decidable for the class of inhibitor systems, see Example 2. We can give an alternative proof of this result by Corollary 8.

As the membership problem for a fixed regular language is decidable in linear time, we obtain:

Corollary 9. *For every inverse match-bounded string rewriting system R, its termination problem is decidable in linear time:*

GIVEN: *A string x.*

QUESTION: *Is R terminating on x?*

Example 8. This affirmatively solves another question posed by McNaughton [14] as Open Problem 2, cf. Example 2: For every inhibitor system, is its termination problem decidable in polynomial time?

Example 9. Consider the following string rewriting systems.

- $\{babaa \to aaababab\}$ is inverse match-bounded by 2.
- $\{aabaaba \to abaabaaab\}$ is inverse match-bounded by 3.
- $\{aaabbab \to abbaaabba\}$ is inverse match-bounded by 3.
- $\{caabca \to aabccaabc\}$ is inverse match-bounded by 2.

Each of these systems R satisfies $\mathrm{Im}(R) = \emptyset$, hence R terminates. — The reader is invited to prove these systems terminating by any other method.

In contrast to match-boundedness, inverse match-boundedness does not entail termination.

Example 10. The inhibitor system $R = \{baa \to aa\imath babba\}$ from Example 2 has a loop of length 3 initiated by $baaa$. Indeed, $baaa \in \mathrm{Im}(R) = \Sigma^*\{ba, baab\}b^*aa\Sigma^*$.

Example 11. The system $R = \{baaba \to aababbaab\} = \{\ell \to r\}$ is inverse match-bounded by 2 and satisfies $\mathrm{Im}(R) = \Sigma^* M \Sigma^*$ for

$$M = \{\ell aa, \ell\ell, b\ell a\} \cup \{\ell a, \ell b, \ell baab, bbaab\}b^*\ell.$$

For instance, ℓaa initiates a loop of length 4.

Example 12. The system $R = \{aabaaba \to abaaabaab\} = \{\ell \to r\}$ is inverse match-bounded by 3; we have $\mathrm{Im}(R) = \Sigma^*\{\ell aa, a\ell a, ara\}\Sigma^*$. The string ℓaa initiates a loop of length 3.

Example 13. The system $R = \{bca(bc)^3 \to a(bc)^3c(bc)^2b\} = \{\ell \to r\}$ is inverse match-bounded by 2 and admits a loop of length 3 (starting from $bc\ell c$). We have $\mathrm{Im}(R) = \Sigma^* M \Sigma^*$ where

$$M = \{bc\ell c, \ell ca(bc)^3, (bc)^2\ell, \ell c\ell, \ell cbc\ell\}.$$

Example 14. The inhibitor system $R = \{ab \to bb\imath aa\}$ is inverse match-bounded by 1; cf. Example 2. It admits a loop $abb \to bb\imath aab \to bb\imath abb\imath aa$. Indeed we get $abb \in \mathrm{Im}(R) = \Sigma^*\{aab, abab, abb\}\Sigma^*$.

Example 15. Consider the system $R = \{ab \to da, ac \to acc\}$ over $\Sigma = \{a, b, c, d\}$ from Example 5, which is inverse match-bounded by 1. Here we obtain $\mathrm{Im}(R) = \Sigma^* ab^* c\Sigma^*$.

Looping one-rule string rewriting systems exist that are not inverse match-bounded:

Example 16. The system $R = \{aabb \to ba\}$ is shown to be not match-bounded in [9]. Hence the system R^-, which admits the loop $bba \to_{R^-} baabb \to_{R^-} aabbabb$, is not inverse match-bounded.

We conclude this section with an example known as *Zantema's Problem*. Proving termination of this system is a "modern classic" in rewriting [4, 6, 13, 17, 18, 19], as it provides a test case where all previous automated methods for termination proofs fail.

Example 17. The one-rule system

$$Z = \{a^2b^2 \to b^3a^3\}$$

is inverse match-bounded by 2 and satisfies $\mathrm{Im}(Z) = \emptyset$, so it terminates. In our implementation (Haskell code compiled with `ghc`), the automaton for $\mathrm{match}(Z^-)^*(\mathrm{lift}_0(\Sigma^*))$, which has 199 states, is found in about 40 CPU seconds on a 2.4-GHz Pentium. (This figure also includes the verification of $\mathrm{Im}(Z) = \emptyset$.) The intermediate constructions according to Theorem 1 involve much larger automata (up to 1576 states with 15999 transitions) over much larger alphabets (up to 283 letters).

We remark that Z can also be proven terminating by verifying the match bound 4 for Z. However, this computation needs considerably more resources, as the corresponding intermediate automata have up to 22241 states. Even the refined method of checking the match bound for right hand sides of forward closures only [9] can only reduce this number to 11307 states.

While we have no evidence that terminating, inverse match-bounded systems exist that are not match-bounded, we do have a few examples ($\{babaa \to aaababab\}$ from Example 9, $\{babaa \to abaabbaba\}$, $\{baaabbaa \to aaabbaaabb\}$) that can be shown inverse match-bounded by our tool (cf. Section 8), but a corresponding proof attempt for match-boundedness fails due to lack of memory.

7 DERIVATION LENGTHS

Inverse match-bounded systems R have linear size growth for strings that are not in $\mathrm{Im}(R) = \mathrm{Inf}(R^*)$. Hence terminating, inverse match-bounded systems have linear derivational complexity.

Theorem 9. *For each inverse match-bounded string rewriting system R there is a constant $k \in \mathbb{N}$ such that $|y| \leq k \cdot |x|$ for all $x \in \Sigma^* \setminus \mathrm{Inf}(R^*)$ and $y \in R^*(x)$.*

Proof. Since the morphisms base and lift preserve lengths of strings, we may assume that R is inverse deleting by Lemma 2 and Lemma 3. As in the proof of Lemma 6 we then 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. Further, $\mathrm{Inf}(R^*) = \mathrm{Inf}(c \circ s^-) = \mathrm{Inf}(c \cap (\Sigma^* \times K))$ as in the proof of Lemma 5 for $K = s(\Sigma^*)$.

We change the automaton B constructed in the proof of Lemma 4 to an automaton B' over the alphabet $\Sigma \times (\mathbb{N} \cup \{I\})$ as follows. In the case where $c(a) \cap L(A, p, q)$ is finite and non-empty, we rather draw an arrow labelled by (a, m) where m is the length of the longest string in $c(a) \cap L(A, p, q)$. Now define $\mu : \Sigma^* \setminus \mathrm{Inf}(R^*) \to \mathbb{N}$ by

$$\mu(a_1 \ldots a_n) = \max\{\sum_{i=1}^{n} m_i \mid (a_1, m_1) \ldots (a_n, m_n) \in L(B') \setminus (\Sigma \times \{I\})^*\}$$

One verifies for all $x \in \Sigma^* \setminus \mathrm{Inf}(R^*)$ that $\mu(x)$ is the maximal length of strings in the set $R^*(x)$. One possible choice of k is therefore obtained as the maximum of all second components of labels in B'. □

Corollary 10. *Every terminating, inverse match-bounded string rewriting system has linear derivational complexity.*

Proof. If R^- is a match-bounded system, then by Proposition 2 there is a constant k' such that $x \to_R^n y$ (i.e., $y \to_{R^-}^n x$) implies $n \leq k' \cdot |y|$ for strings x and y. On the other hand, $|y| \leq k \cdot |x|$ for some constant k by Theorem 9. Hence $n \leq k' \cdot k \cdot |x|$. □

Remark 4. Bounds for both constants k and k' can be determined effectively. We remark that k' can be exponential in the size of the underlying alphabet $\Sigma \times \{0, \ldots, c\}$ in the worst case, assuming that R^- is match-bounded by c, see [12]. For an algorithm to compute constant k see the proof of Theorem 9 above.

All terminating, inverse match-bounded examples in Section 6 have linear derivational complexity. In particular this is true for Example 17 (*Zantema's Problem*), a result due to Tahhan-Bittar [18].

8 IMPLEMENTING MATCH-BOUNDS: MATCHBOX

Our program `Matchbox` can verify that a given string rewriting system R is match-bounded by a given number for a given regular language. So it can verify inverse match-boundedness (for Σ^*) as well, and it provides an "effective version" of Theorem 5. Further, `Matchbox` constructs $\mathrm{Im}(R)$ according to Theorem 8. The program can be accessed via a CGI-interface at

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

its Haskell source is available. In particular, `Matchbox` is able to prove termination for a large number of string rewriting systems for which all standard automated methods (like path orderings, polynomial interpretations (see [5], e.g.), and dependency pairs [1]) fail, and for which only complicated ad-hoc proofs were known, if any.

9 CONCLUSION

We showed that inverse match-bounded string rewriting systems, like match-bounded systems, enjoy nice language preservation and decidability properties. In particular, termination and uniform termination are decidable for the class of inverse match-bounded systems, and terminating systems in this class have linear derivation lengths.

In spite of the formal similarity of match-boundedness and inverse match-boundedness, the line of reasoning for similar properties is quite different. All the same, we have found no terminating examples that are inverse match-bounded but not match-bounded.

Acknowledgements.

This research was initiated while the last two authors were visiting scientists at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center (LaRC), Hampton, VA, in September 2002. The first author is supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046.

REFERENCES

[1] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoret. Comput. Sci.*, 236:133–178, 2000.

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

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

[4] T. Coquand and H. Persson. A proof-theoretical investigation of Zantema's problem. In M. Nielsen and W. Thomas (Eds.), *Proc. 11th Annual Conf. of the EACSL CSL-97*, *Lecture Notes in Comput. Sci.* Vol. 1414, pp. 177–188. Springer-Verlag, 1998.

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

[6] N. Dershowitz and C. Hoot. Topics in termination. In C. Kirchner (Ed.), *Proc. 5th Int. Conf. Rewriting Techniques and Applications RTA-93*, *Lecture Notes in Comput. Sci.* Vol. 690, pp. 198–212. Springer-Verlag, 1993.

[7] A. Geser. Note on normalizing, non-terminating one-rule string rewriting systems. *Theoret. Comput. Sci.*, 243:489–498, 2000.

[8] 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.

[9] 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`.

[10] S. Ginsburg and S. A. Greibach. Mappings which preserve context sensitive languages. *Inform. and Control*, 9(6):563–582, 1966.

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

[12] 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.

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

[14] R. McNaughton. Semi-Thue systems with an inhibitor. *J. Automat. Reason.*, 26:409–431, 2001.

[15] C. Moore and D. Eppstein. One-dimensional peg solitaire, and duotaire. In R. J. Nowakowski (Ed.), *More Games of No Chance*, Cambridge Univ. Press, 2003.

[16] B. Ravikumar. Peg-solitaire, string rewriting systems and finite automata. In H.-W. Leong, H. Imai, and S. Jain (Eds.), *Proc. 8th Int. Symp. Algorithms and Computation ISAAC-97*, *Lecture Notes in Comput. Sci.* Vol. 1350, pp. 233–242. Springer-Verlag, 1997.

[17] 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.

[18] E. Tahhan Bittar. Complexité linéaire du problème de Zantema. *C. R. Acad. Sci. Paris Sér. I Inform. Théor.*, t. 323:1201–1206, 1996.

[19] 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.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01/07/2004	Contractor Report	

4. TITLE AND SUBTITLE

Termination Proofs for String Rewriting Systems via Inverse Match-Bounds

5a. CONTRACT NUMBER

5b. GRANT NUMBER

NCC-1-02043

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Geser, Alfons, Hofbauer, Dieter, and Waldmann, Johannes

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

23-786-10-10

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

National Institute of Aerospace (NIA)
144 Research Drive
Hampton, VA 23666

8. PERFORMING ORGANIZATION REPORT NUMBER

NIA Report No. 2004-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITOR'S ACRONYM(S)

NASA

11. SPONSORING/MONITORING REPORT NUMBER

NASA/CR-2004-213032

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 60, 61
Availability: NASA CASI (301) 621-0390 Distribution: Nonstandard

13. SUPPLEMENTARY NOTES

Langley Technical Monitor: Ricky Butler

14. ABSTRACT

Annotating a letter by a number, one can record information about its history during a reduction. A string rewriting system is called match-bounded if there is a global upper bound to these numbers. In earlier papers we established match-boundedness as a strong sufficient criterion for both termination and preservation of regular languages. We show now that the string rewriting system whose inverse (left and right hand sides exchanged) is match-bounded, also have exceptional properties, but slightly different ones. Inverse match-bounded systems effectively preserve context-free languages; their sets of normalized strings and their sets of immortal strings are effectively regular. These sets of strings can be used to decide the normalization, the termination and the uniform termination problems of inverse match-bounded systems. We also show that the termination problem is decidable in linear time, and that a certain strong reachability problem is deciable, thus solving two open problems of McNaughton's.

15. SUBJECT TERMS

String Rewriting; Termination; Match-Bounded; Inhibitor; Terminal Bounded Grammar; Reachable

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19b. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	20	19b. TELEPHONE NUMBER *(Include area code)* (301) 621-0390