Confluent and Other Types of Thue Systems

RONALD V. BOOK

University of California at Santa Barbara, Santa Barbara, California

ABSTRACT Confluent and other types of finite Thue systems are studied. Sufficient conditions are developed for every congruence class and every finite union of congruence classes defined by such a system to be a deterministic context-free language. It is shown that the word problem for Church–Rosser systems is decidable in linear time

Categories and Subject Descriptors F 4 2 [**Mathematical Logic and Formal Languages**] Grammars and Other Rewriting Systems—*Thue systems*, F 4 3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*decision problems*

General Terms Algorithms, Languages, Theory

Additional Key Words and Phrases Thue congruence, confluent systems, Church–Rosser system, preperfect system, almost-confluent system, deterministic context-free language, word problems

Introduction

There are some operations on data structures that can be described in very general terms, for example, "delete," "insert," or "replace." It would be desirable to develop a formal calculus which would enable one to decide questions such as whether two sequences of operations are equivalent. It is known that no such calculus can be developed universally.

Formal systems used to describe such operations are often called "replacement systems." In the study of formula-manipulation systems such as theorem provers, program optimizers, or algebraic simplifiers, replacement systems take the form of term rewriting systems, tree manipulating systems, graph grammars, etc. Often one attempts to show that the system is "confluent" or "Church–Rosser" so that there is a way of describing canonical representatives or unique normal forms. (See [2, 11, 12, 15, 16, 18].)

When dealing with strings, the appropriate notion of replacement system is that of Thue system, that is, a set S of pairs of strings and the relation defined by $xuy \leftrightarrow xvy$ for all x, y and all (u, v) or (v, u) in S. The reflexive transitive closure $\overset{*}{\leftrightarrow}$ of $\leftrightarrow$ is the *Thue congruence generated by S*. As well as being replacement systems, Thue systems and Thue congruences have been used to specify formal languages [3, 10, 11, 16, 17, 19].

In this paper finite Thue systems and their congruences are studied. In Section 2 certain restricted types of Thue systems are defined and their properties investigated; specifically, systems that are "confluent," "Church–Rosser," "almost-confluent," or "preperfect" are studied. If a Thue system S has the property that $(u, v) \in S$ implies

This research was supported in part by the National Science Foundation under Grants MCS 77-11360 and MCS 80-11979.

Author's address Department of Mathematics, University of California at Santa Barbara, Santa Barbara, CA 93106

$|u| > |v|$ and $1 \geq |v|$ and if S is Church–Rosser, then every congruence class of S and every finite union of congruence classes of S is a deterministic context-free language, as are certain infinite unions of congruence classes of S; these and other similar results are established in Section 3.

Deterministic context-free languages are recognized by deterministic pushdown store acceptors, and deterministic pushdown store acceptors can be made to run in linear time. Thus, one is led to examine certain problems about finite Thue systems and to search for good algorithms for their solution. In Section 4 it is shown that for any finite Thue system S, there is a linear-time algorithm to compute from a given string x an irreducible string y such that y is congruent to x. This result is then used to show that for any finite Thue system S that is Church–Rosser, there is a linear-time algorithm to solve the word problem for S.

It is hoped that the results presented here will suggest further attacks on similar problems concerning other types of replacement systems.

1. *Preliminaries*

It is assumed that the reader is familiar with the basic concepts of formal language theory, computability theory, and complexity theory. Some definitions and notation for Thue systems and their congruences are established here.

If Σ is a finite alphabet, then Σ^* is the free semigroup with identity e generated by Σ. If w is a string, then the *length* of w is denoted by $|w|$: $|e| = 0$, $|a| = 1$ for $a \in \Sigma$, and $|wa| = |w| + 1$ for $w \in \Sigma^*$, $a \in \Sigma$.

A *Thue system* S on a finite alphabet Σ is a subset of $\Sigma^* \times \Sigma^*$. Each pair in S is a *relation*. The *Thue congruence generated by* S is the reflexive transitive closure $\overset{*}{\leftrightarrow}_S$ of the relation $\leftrightarrow_S$ defined as follows: For any u, v such that $(u, v) \in S$ or $(v, u) \in S$ and any $x, y \in \Sigma^*$, $xuy \leftrightarrow_S xvy$. Two strings w, z are *congruent* (*mod* S) if $w \overset{*}{\leftrightarrow}_S z$; the *congruence class of* z (*mod* S) is $[z]_S = \{w \mid w \overset{*}{\leftrightarrow}_S z\}$. (Whenever possible the subscript S will be omitted.)

If S_1 and S_2 are Thue systems such that for all x, y, $x \overset{*}{\leftrightarrow}_{S_1} y$ implies $x \overset{*}{\leftrightarrow}_{S_2} y$, then S_1 *refines* S_2; if S_1 refines S_2 and S_2 refines S_1, then S_1 and S_2 are *equivalent*. Clearly S_1 refines S_2 if and only if for every $(u, v) \in S_1$, $u \overset{*}{\leftrightarrow}_{S_2} v$.

Let S be a system of relations.

(a) Write $x \rightarrow y$ provided $x \leftrightarrow y$ and $|x| > |y|$.
(b) Write $x \vdash\!\dashv y$ provided $x \leftrightarrow y$ and $|x| = |y|$.
(c) Write $x \mapsto y$ provided $x \rightarrow y$ or $x \vdash\!\dashv y$.

The reflexive transitive closure of $\rightarrow$ ($\vdash\!\dashv$, $\mapsto$) is denoted $\overset{*}{\rightarrow}$ (respectively, $\overset{*}{\vdash\!\dashv}$, $\overset{*}{\mapsto}$).

If S is a Thue system, then a string x is *irreducible* (mod S) if for all y, $x \leftrightarrow y$ implies $|x| \leq |y|$, and is *minimal* if $x \overset{*}{\leftrightarrow} y$ implies $|x| \leq |y|$.

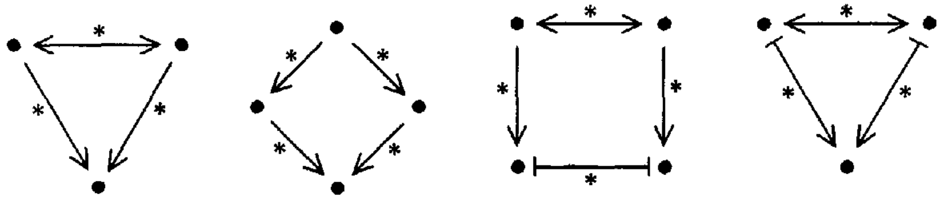
A transformation $x \overset{*}{\rightarrow} y$ is a *reduction*.

In this paper certain restricted Thue systems are considered.

A Thue system S on a finite alphabet Σ is *monadic* if $(u, v) \in S$ implies $|u| > |v|$ and $v \in \Sigma \cup \{e\}$, and is *special* if $(u, v) \in S$ implies $v = e$.

2. *Types of Thue Systems*

A number of different combinatorial replacement systems have been studied in the context of automatic theorem-proving, abstract data types, program optimization, combinatory logic, etc. In an abstract replacement system a binary relation $\Rightarrow$ on a space $\mathbf{B}$ is defined; usually the reflexive transitive closure $\overset{*}{\Rightarrow}$ of this relation is not symmetric and is considered to be a "reduction." An irreducible element of $\mathbf{B}$ is



(a) (b) (c) (d)

FIG. 1. (a) Church–Rosser. (b) Confluent (c) Almost-confluent (d) Preperfect.

considered to be a *normal form* of $(\mathbf{B}, \Rightarrow)$. If the relation $\overset{*}{\Rightarrow}$ is Noetherian, that is, has no infinite descending chains, then for each $x \in \mathbf{B}$ there is a normal form y such that $x \overset{*}{\Rightarrow} y$. Given $\mathbf{B}$ and $\Rightarrow$, define $\Leftrightarrow$ to be $(\Rightarrow \cup \Rightarrow^{-1})$ and $\overset{*}{\Leftrightarrow}$ to be the reflexive transitive closure of $\Leftrightarrow$. Inspired by the result of Church and Rosser [9] on the calculus of λ-conversion, we call a system $(\mathbf{B}, \Rightarrow)$ *Church–Rosser* if for all $x, y \in \mathbf{B}$, $x \overset{*}{\Leftrightarrow} y$ implies that for some $z \in \mathbf{B}$, $x \overset{*}{\Rightarrow} z$ and $y \overset{*}{\Rightarrow} z$. If a system $(\mathbf{B}, \Rightarrow)$ is Church–Rosser and $\overset{*}{\Rightarrow}$ is Noetherian, then for every element $x \in \mathbf{B}$ there is a unique normal form y such that $x \overset{*}{\Rightarrow} y$. Newman [15] called a system *confluent* if it had this property, and he showed that the Church–Rosser property is equivalent to the following: For all $w, x, y \in \mathbf{B}$, if $w \overset{*}{\Rightarrow} x$ and $w \overset{*}{\Rightarrow} y$, then for some z, $x \overset{*}{\Rightarrow} z$ and $y \overset{*}{\Rightarrow} z$. In recent work (e.g., [12]), the property established by Newman has been called "confluent," and systems with this property have been called "confluent systems."

The situation for Thue systems is more complicated. If S is a Thue system on a finite alphabet Σ, then the relation $\xrightarrow{*}_S$ is Noetherian, but the relation $\overset{*}{\mapsto}_S$ is not Noetherian if there is some $(u, v) \in S$ such that $|u| = |v|$. Further, there are a variety of terms that have been used in conjunction with the replacement systems defined by $(\Sigma^*, \rightarrow_S)$ and $(\Sigma^*, \mapsto_S)$: perfect, quasi-perfect, preperfect, finite Church–Rosser [2, 3, 10–12, 16–19]. Here the relation $\rightarrow$ is emphasized, and this leads to the definitions below which follow the historical precedent of "Church–Rosser" for the "triangle" property, using "confluent" for the "diamond" property.

Let S be a Thue system.

(a) S is *Church–Rosser* if for all x, y, if $x \overset{*}{\leftrightarrow} y$, then there exists a z such that $x \xrightarrow{*} z$ and $y \xrightarrow{*} z$ (see Figure 1a).
(b) S is *confluent* if for all w, x, y, if $w \xrightarrow{*} x$ and $w \xrightarrow{*} y$, then there exists a z such that $x \xrightarrow{*} z$ and $y \xrightarrow{*} z$ (see Figure 1b).
(c) S is *almost-confluent* if for all x, y, if $x \overset{*}{\leftrightarrow} y$, then there exist g, h such that $x \xrightarrow{*} g$, $y \xrightarrow{*} h$, and $g \overset{*}{\vdash\!\dashv} h$ (see Figure 1c).
(d) S is *quasi-perfect* if for all w, x, y, if $w \overset{*}{\mapsto} x$ and $w \overset{*}{\mapsto} y$, then there exists a z such that $x \overset{*}{\mapsto} z$ and $y \overset{*}{\mapsto} z$.
(e) S is *preperfect* if for all x, y, if $x \overset{*}{\leftrightarrow} y$, then there exists a z such that $x \overset{*}{\mapsto} z$ and $y \overset{*}{\mapsto} z$ (see Figure 1d).

Abstract replacement systems with properties similar to those of almost-confluent Thue systems were studied by Huet [12] and called "confluent modulo an equivalence relation."

It is clear that for Thue systems, any Church–Rosser system is confluent. The possibility of length-preserving rules allows for confluent systems that are not Church–Rosser: if $S = \{(u, v)\}$ where $u \neq v$ and $|u| = |v|$, then the relation $\rightarrow$ is void so that S is confluent (vacuously), but S is not Church–Rosser since there is no z such that $u \xrightarrow{*} z$ and $v \xrightarrow{*} z$. However, we do have the following fact.

PROPOSITION 2.1. *Let S be a Thue system.*

(*a*) *If $(u, v) \in S$ implies $|u| \neq |v|$, then S is confluent if and only if S is Church–Rosser. Thus, if S is monadic, then S is confluent if and only if S is Church–Rosser.*
(*b*) *Suppose that for every pair $(u, v) \in S$ such that $|u| = |v|$, there exists a z such that $u \xrightarrow{*} z$ and $v \xrightarrow{*} z$. Then S is confluent if and only if S is Church–Rosser.*

PROOF. As noted above, any Church–Rosser system is confluent. For part (a) the fact that any confluent system is Church–Rosser was established by Cochet and Nivat [11], using induction on the number of steps in a transformation $x \overset{*}{\leftrightarrow} y$. The same method can be used to show that for part (b) any confluent system is Church–Rosser. □

Similarly, we have another fact.

PROPOSITION 2.2 [11]. *Let S be a Thue system. Then S is quasi-perfect if and only if S is preperfect.*

Only the terms "confluent," "Church–Rosser," "almost-confluent," and "preperfect" will be used in the remainder of this paper. The four parts of Figure 1 serve to illustrate the differences.

Notice that for any Thue system S_1, the system $S_2 = \{(u, v) \mid u \overset{*}{\leftrightarrow} v \text{ and } |u| \geq |v|\}$ is an infinite almost-confluent Thue system that is equivalent to S_1. Thus every Thue congruence is an infinitely generated almost-confluent congruence, and so an infinite Thue system must be further restricted to be of interest. Only finite Thue systems are considered here.

It is clear that if S is a finite Thue system, then for any string x one can effectively find an irreducible string congruent to x. The set of all irreducible strings of S is a regular set [3]. Every string has at least one minimal string congruent to it, but the question, "Given S and x, is x minimal for $\overset{*}{\leftrightarrow}_S$?" is undecidable [8]. A minimal string can be considered to be a "normal form," and the question of uniqueness of normal forms will be useful here.

PROPOSITION 2.3. *Let S be a Thue system.*

(*a*) *If S is Church–Rosser, then S is almost-confluent, and if S is almost-confluent, then S is preperfect.*
(*b*) *The system S is Church–Rosser if and only if x, y irreducible and $x \overset{*}{\leftrightarrow} y$ imply $x = y$* [12, 15, 17].
(*c*) *The system S is almost-confluent if and only if x, y irreducible (mod S) and $x \overset{*}{\leftrightarrow} y$ imply $x \overset{*}{\vdash\!\dashv} y$* [17].
(*d*) *If S is Church–Rosser or almost-confluent, then a string is irreducible (mod S) if and only if it is minimal* [3].
(*e*) *If $(u, v) \in S$ implies $|u| \neq |v|$, then S is preperfect if and only if S is almost-confluent if and only if S is confluent if and only if S is Church–Rosser.*

Nivat [17] has developed necessary and sufficient conditions for a Thue system to be almost-confluent, and these conditions are decidable. An almost-confluent system with no length-preserving relations is confluent.

PROPOSITION 2.4 [17]. *It is decidable whether a finite Thue system is almost-confluent, and it is decidable whether a finite Thue system is confluent.*

SKETCH OF PROOF. A Thue system S is *locally confluent* if for all w, x, y, $w \to x$ and $w \to y$ imply that for some z, $x \xrightarrow{*} z$ and $y \xrightarrow{*} z$. Since the relation $\xrightarrow{*}$ is

Noetherian, a Thue system is confluent if and only if it is locally confluent (Huet [12] provides a simple proof of this fact). It is clear that a Thue system S is locally confluent if and only if for every pair (u_1, v_1), (u_2, v_2) or (v_1, u_1), (v_2, u_2) of not necessarily distinct relations in S with $|u_1| > |v_1|$ and $|u_2| > |v_2|$, the following conditions hold:

(a) If there are strings x, y such that $u_1x = yu_2$ with $|x| < |u_2|$, then there exists z such that $v_1x \xrightarrow{*} z$ and $yv_2 \xrightarrow{*} z$.
(b) If there are strings x, y such that $u_1 = xu_2y$, then there exists z such that $v_1 \xrightarrow{*} z$ and $xv_2y \xrightarrow{*} z$.

For any pair of relations, one can determine whether (a) and (b) hold. Thus, if S is finite, one can decide whether S is locally confluent.

A similar argument shows that it is decidable whether a finite Thue system is almost-confluent (see [12] or [17]). □

Nivat's results are extended in [7], where it is shown that the question, "Is a finite Thue system Church–Rosser?" is not only decidable but also is tractable, that is, there is a polynomial-time (in the size of the system) algorithm to decide whether a system is Church–Rosser. At this time no algorithm is known for determining whether a finite Thue system is preperfect.

For a Thue system S the *word problem* is the question, "For strings x, y, are x and y congruent (mod S)?" In general, the word problem for Thue systems is undecidable. If S is a preperfect system, then x and y are congruent if and only if there is some z such that $x \overset{*}{\models\!\!\!>} z$ and $y \overset{*}{\models\!\!\!>} z$. Since the number of strings no longer than x is finite, it is clear that the word problem for preperfect systems is decidable. From the fact that the word problem for finitely presented groups is undecidable, it follows that the word problem for finite special Thue systems is undecidable.

From the standpoint of formal language theory it is of interest to determine whether the congruence classes of certain types of congruences have a particular structure. Berstel [3] has surveyed a number of results regarding Thue systems whose congruence classes are (or are not) context-free languages.

A language L is *congruential* if there is a finite Thue system S such that L is the union of finitely many of S's congruence classes. Berstel [3] has shown that the linear context-free language $\{ww^R \mid w \in \{a, b\}^*\}$ is not congruential.

If one knows that every congruence class of a certain type of system is in some class of recursive sets (e.g., the class of context-free languages) and from any such system one can effectively specify algorithms for membership in those sets (e.g., specify context-free grammars and use standard parsing techniques), then one can obtain upper bounds on the complexity of the word problem for such systems. This is part of the motivation for attempting to classify Thue systems as to whether their congruence classes or congruential languages are context-free languages or deterministic context-free languages or . . . .

If the word problem for a class of finite systems of relations is decidable, then so are the refinement and equivalence problems. Thus studying the structure of the congruence classes and congruential languages specified by such a system may yield information about the inclusion and equivalence problems for certain classes of languages. For example, since finite preperfect systems have decidable word problems, the class of congruential languages specified by preperfect systems has a decidable inclusion problem and a decidable equivalence problem.

If S is a Thue system on an alphabet Σ, then the collection of S's congruence classes forms a semigroup with identity, that is, a monoid. The system S is a *presentation* of $\Sigma^*/\overset{*}{\leftrightarrow}_S$. If S is finite, then $\Sigma^*/\overset{*}{\leftrightarrow}_S$ is a *finitely presented* monoid. Most of the results presented here can be interpreted in terms of monoids but will not be restated in that form.

3. *Syntactic Complexity*

The results in this section are about the syntactic complexity of congruence classes; specifically, certain Thue systems are shown to have congruence classes that are deterministic context-free languages.

Assumption. Throughout the remainder of this paper it is assumed that each Thue system S is presented in such a way that $(u, v) \in S$ implies $|u| \geq |v|$. No generality is lost by making this assumption.

A reduction $\alpha \to \beta$ is *leftmost* if $\alpha = xu_1y$, $\beta = xv_1y$, $(u_1, v_1) \in S$, and if $\alpha = x'u_2y'$ with $(u_2, v_2) \in S$ for some v_2, $u_1 \neq u_2$, then xu_1 is a proper prefix of $x'u_2$ or $xu_1 = x'u_2$ and x is a proper prefix of x'. Write $\alpha \to^{L} \beta$ if $\alpha \to \beta$ is leftmost. Let $\overset{*}{\to}^{L}$ denote the reflexive transitive closure of $\to^{L}$.

LEMMA 3.1. *Let S be a Thue system.*

(*a*) *Let $xuy \to^{L} xvy$, where $(u, v) \in S$ is a leftmost reduction. Then x is irreducible (mod S).*
(*b*) *Let $w_0 \to^{L} w_1 \to^{L} \cdots \to^{L} w_n$ be a leftmost reduction. For each $i = 1, \ldots, n$, let $w_{i-1} = x_iu_iy_i$ and $w_i = x_iv_iy_i$, where $(u_i, v_i) \in S$ is the relation used to obtain $w_{i-1} \to^{L} w_i$. Then for each $i = 1, \ldots, n$, the string x_i is irreducible (mod S).*

LEMMA 3.2. *Let S be a Thue system.*

(*a*) *For every x there exists an irreducible y such that $x \overset{*}{\to}^{L} y$.*
(*b*) *Suppose S is Church–Rosser. For all x and all irreducible y, $x \overset{*}{\leftrightarrow} y$ if and only if $x \overset{*}{\to}^{L} y$. Thus, for all x there is a unique irreducible y such that $x \overset{*}{\to}^{L} y$.*
(*c*) *Suppose S is almost-confluent. For all x and all irreducible y, $x \overset{*}{\leftrightarrow} y$ implies that for some irreducible z, $x \overset{*}{\to}^{L} z$ and $z \overset{*}{\vdash\dashv} y$.*

PROOF

(a) Given S, let S' be any subsystem of S with the following properties:

   (i) For each string u such that for some v, $(u, v) \in S$ and $|u| > |v|$, there is exactly one v' such that $(u, v') \in S'$ and $|u| > |v'|$.
   (ii) If $(u, v) \in S'$, then $|u| > |v|$.

   From property (i) it is clear that a string is irreducible (mod S) if and only if it is irreducible (mod S'). From property (ii) and Lemma 3.1(b) it is clear that for each string x there is a y such that y is irreducible (mod S') and there is a leftmost reduction of y from x using at most $|x|$ steps.
(b) If S is Church–Rosser, then for each x there is a unique irreducible y such that $x \overset{*}{\leftrightarrow} y$. From part (a) the uniqueness of y implies $x \overset{*}{\to}^{L} y$.
(c) For any x there is an irreducible z such that $x \overset{*}{\to}^{L} z$ (part (a)). If $y \overset{*}{\leftrightarrow} x$, then $y \overset{*}{\leftrightarrow} z$. Since S is almost-confluent, there exist g and h such that $y \overset{*}{\to} g$, $z \overset{*}{\to} h$, and $g \overset{*}{\vdash\dashv} h$. Since z is irreducible, $z \overset{*}{\to} h$ implies $h = z$. If y is irreducible, $y \overset{*}{\to} g$ implies $g = y$. Thus $y \overset{*}{\vdash\dashv} z$. □

Now the first result can be established.

THEOREM 3.3. *Let S be a finite monadic Thue system with the property that $(u, v) \in S$ and $(u, v') \in S$ imply $v' = v$. If R is a regular set of strings, then the set $\{x \mid$ for some irreducible $y \in R, x \overset{*}{\to}^{L} y\}$* is a deterministic context-free language.

PROOF. Let $t = \max\{|u| \mid \text{for some } v, (u, v) \in S\}$. Construct a deterministic pushdown store acceptor D that operates as follows.

Initially the pushdown store is empty. When the store is empty, attempt to read a new input symbol and push it onto the store. When the store is not empty, read at most the top t symbols on the store and determine whether there exists a string u stored on the top $|u| \leq t$ squares of the store (with the rightmost symbol of u to the top) such that for some v, $(u, v) \in S$. If there is such a u, replace u by v, and once again determine whether there exists a string on the top of the store that can be replaced (this will not be the case if $v = e$ but may be the case if $|v| = 1$); if there is no such u on the top of the store, attempt to read a new input symbol from the input string and push it onto the top of the store. If there are no more input symbols to be read, empty the store symbol by symbol and determine whether the string y remaining on the store (with the rightmost symbol of y on the top) is accepted by a finite-state acceptor A which recognizes all and only strings in the regular set $\{\tilde{w} \mid w \in R\}$, where for any string w, $\tilde{w}$ is the reversal of w; the acceptor A can be considered to be a portion of D's finite-state control. Accept the input string if and only if the string y is in R.

The description of D's operation makes it clear that if $x \xrightarrow{*}{}^{L} y$ and y is irreducible, then D accepts x if and only if $y \in R$.

Suppose that in D's computation on an input string w there are n stages where a string is popped from the top of the pushdown store. For each $i = 1, \ldots, n$, let $x_i u_i$ be the contents of the pushdown store before u_i is popped, let $x_i v_i$ be the contents of the pushdown store after u_i is replaced by v_i, and let y_i be the input string that remains to be processed. Then for each $i = 1, \ldots, n$, $x_i u_i y_i \rightarrow x_i v_i y_i$ is a leftmost reduction. To prove this, proceed by induction.

If u_1 is replaced by v_1, then from the description of D's operation it is clear that $x_1 u_1 \rightarrow x_1 v_1$ is leftmost, and so $x_1 u_1 y_1 \rightarrow x_1 v_1 y_1$ is leftmost. Assume that for all $i < k$, $x_i u_i y_i \rightarrow x_i v_i y_i$ is leftmost, and consider $x_k u_k y_k \rightarrow x_k v_k y_k$. By the induction hypothesis $x_{k-1} u_{k-1} y_{k-1} \rightarrow x_{k-1} v_{k-1} y_{k-1}$ is leftmost, so that x_{k-1} is irreducible by Lemma 3.1. If $v_{k-1} = e$, then $x_{k-1} v_{k-1} = x_{k-1}$ and $x_k u_k y_k \rightarrow x_k v_k y_k$ is such that x_{k-1} is a proper prefix of $x_k u_k$, and so $x_k u_k y_k \rightarrow x_k v_k y_k$ is leftmost since u_k is the first string in $x_{k-1} y_{k-1}$ which is the left-hand side of a relation in S. If $v_{k-1} \neq e$, then $|v_{k-1}| = 1$, so $v_{k-1} \in \Sigma$, and after u_{k-1} is replaced by v_{k-1}, v_{k-1} is the symbol on the top of the store. If $x_{k-1} v_{k-1}$ is irreducible, then as above $x_k u_k y_k \rightarrow x_k v_k y_k$ is leftmost; if $x_{k-1} v_{k-1}$ is not irreducible, then $x_k u_k = x_{k-1} v_{k-1}$ since $|v_{k-1}| = 1$ (otherwise there would be x', y' such that $x' u_k y' = x_k$ and x_{k-1} is not irreducible). Thus $x_k u_k \rightarrow x_k v_k$ is leftmost, and so $x_k u_k y_k \rightarrow x_k v_k y_k$ is leftmost.

By choice of n the top of the pushdown store is not popped while D processes the string y_n with $x_n v_n$ on the pushdown store. Thus from the description of D's operation, $x_n v_n y_n$ is irreducible. Now $x_n v_n y_n \in R$ if and only if $\overline{(x_n v_n y_n)}$ is in $\{\bar{z} \mid z \in R\}$, which is a regular set since R is regular. Thus, for any input string w, if D accepts w, then for some irreducible $z \in R$, $w \xrightarrow{*}{}^{L} z$. □

Now consider Thue systems that are both monadic and Church–Rosser.

THEOREM 3.4. *Let S be a finite monadic Thue system. If S is Church–Rosser, then for any regular set R, the set $\bigcup\{[y] \mid y$ is irreducible and $y \in R\}$ is a deterministic context-free language.*

PROOF. Given S, let S' be any subsystem of S with the property that for each string u such that for some v, $(u, v) \in S$, there is exactly one v' such that $(u, v') \in S'$. Since S' is a subsystem of S and S is monadic, S' is monadic. Clearly a string y is irreducible (mod S) if and only if it is irreducible (mod S'). The system S' satisfies the hypothesis of Theorem 3.3, so that for any regular set R, the language $L(S') = \{x \mid \text{for some irreducible } y \in R,\ x \xrightarrow{*}{}^{\mathrm{L}}_{S'} y\}$ is deterministic context-free. Since S is Church–Rosser, for all x there is a unique irreducible y such that $x \xrightarrow{*}{}^{\mathrm{L}}_{S} y$ (Lemma 3.2(b)). But for all x there exists an irreducible z such that $x \xrightarrow{*}{}^{\mathrm{L}}_{S'} z$ (Lemma 3.2(a)), and since S' is a subsystem of S, this means that $z = y$, so $x \xrightarrow{*}{}^{\mathrm{L}}_{S'} y$. Thus $L(S') = \{x \mid \text{for some irreducible } y \in R,\ x \xrightarrow{*}{}^{\mathrm{L}}_{S} y\}$. Since S is Church–Rosser, $x \overset{*}{\leftrightarrow} y$ and y irreducible implies $x \xrightarrow{*}{}^{\mathrm{L}}_{S} y$, so that $L(S') = \{x \mid \text{for some irreducible } y \in R,\ x \xrightarrow{*}{}^{\mathrm{L}}_{S} y\} = \bigcup \{[y] \mid y \text{ is irreducible } (\text{mod } S) \text{ and } y \in R\}$. As noted above, $L(S')$ is deterministic context-free. □

COROLLARY 3.5. *Let S be a finite monadic Thue system. If S is Church–Rosser, then every congruence class and every finite union of congruence classes of S is a deterministic context-free language.*

A finite Thue system that is special and Church–Rosser is a *Dyck* system.

Cochet and Nivat [10, 11, 16] have shown that if S is a Dyck system, then every congruence class of S is an unambiguous context-free language. Since every deterministic context-free language is unambiguous, the complement of a deterministic context-free language is also deterministic context-free, and there exists an unambiguous context-free language whose complement is unambiguous context-free but not deterministic, Theorem 3.4 considerably strengthens the result of Cochet and Nivat.

Theorem 3.4 does not hold for a confluent system that is not monadic. Berstel [3] attributes the following example to Nivat [16] and Cochet [10]. Let $S = \{(abc, ab), (bbc, cb)\}$. Then S is confluent but not monadic, since $(u, v) \in S$ implies $|v| = 2$. The string abb is irreducible and $[abb] \cap \{a\}^*\{b\}^*\{c\}^* = \{ab^{2^n+1}c^n \mid n \geq 0\}$, which is not a context-free language.

Now consider almost-confluent systems.

THEOREM 3.6. *Let S be a finite Thue system such that $(u, v) \in S$ implies either $|u| = |v|$, or $|u| > |v|$ and $1 \geq |v|$. If S is almost-confluent, then for any regular set R with the property that $x \in R$, x irreducible, and $y \overset{*}{\vdash\!\dashv} x$ imply $y \in R$, the set $\bigcup\{[x] \mid x$ is irreducible and $x \in R\}$ is a deterministic context-free language.*

PROOF. By Lemma 3.2(c), if $w \xrightarrow{*}{}^{\mathrm{L}} z$ and z is irreducible, then $z \overset{*}{\vdash\!\dashv} z'$ for all irreducible z' such that $w \xrightarrow{*} z'$. Thus one can use the conditions on R to modify the construction given in the proof of Theorem 3.3 to obtain a deterministic pushdown store acceptor to recognize $\bigcup\{[x] \mid x \text{ is irreducible and } x \in R\}$. □

COROLLARY 3.7. *Let S be a finite Thue system such that $(u, v) \in S$ implies either $|u| = |v|$, or $|u| > |v|$ and $1 \geq |v|$. If S is almost-confluent, then every congruence class and every finite union of congruence classes of S is a deterministic context-free language.*

Sakarovitch [19] has independently established Corollary 3.7 using entirely different methods.

There exist Thue systems that are not almost-confluent. For example, let $S = \{(abc, e), (ab, ba)\}$. The system S is preperfect but is not almost-confluent, since $abbc$ is irreducible and $abbc \vdash\!\dashv babc \rightarrow b$ so that $abbc$ is not minimal. The congruence class of e (mod S) is not context-free since $[e] \cap \{a\}^*\{b\}^*\{c\}^* = \{a^nb^nc^n \mid n \geq 0\}$, which is not context-free.

Let $S = \{(abc, e), (cba, e)\}$. The system S is special but not preperfect: $ab \leftrightarrow abcba \leftrightarrow ba$, so $ab \overset{*}{\leftrightarrow} ba$, but ab and ba are minimal and it is not the case that $ab \overset{*}{\vdash\!\dashv} ba$. The congruence class of e is not context-free, since $[e] \cap \{a\}^*\{b\}^*\{c\}^* =$

$\{a^n b^n c^n \mid n \geq 0\}$ is not context-free. The system $S' = \{(abc, e), (ab, ba), (ac, ca), (bc, cb)\}$ is preperfect and is equivalent to S.

Now consider special systems with a single defining relation.

The system $S_1 = \{(aba, e)\}$ is not preperfect, since $ab \leftrightarrow ababa \leftrightarrow ba$ so that $ab \overset{*}{\leftrightarrow} ba$, but ab and ba are minimal and it is not the case that $ab \overset{*}{\vdash\!\dashv} ba$. However, $S_2 = \{(aba, e), (ab, ba)\}$ is preperfect and S_2 is equivalent to S_1. It is easy to see that no finite system equivalent to S_1 is almost-confluent.

Let $S = \{((ab)^k a, e)\}$ for some symbols a, b and integer $k \geq 1$. Then $((ab)^k a)ba = ab((ab)^k a)$, so that $ab \overset{*}{\leftrightarrow} ba$. Thus, for every $w \in \{a, b\}^*$ there are integers $p, q \geq 0$ such that $w \overset{*}{\leftrightarrow} a^p b^q$ and $|w| = p + q$. In this case, $S' = \{(a^{k+1}b^k, e), (ba, ab)\}$ is a preperfect system with the property that for every $w \in \{a, b\}^*$ there are unique integers $p, q \geq 0$ such that $w \mapsto a^p b^q$ and either $p \geq 0$ and $q < k$ or $p < k + 1$ and $q \geq 0$. It is easy to see that every congruence class and every finite union of congruence classes of S is a deterministic context-free language.

If w is a nonempty string such that there are no u, v with $|u| < |w|$ and $uw = wv$, then it is easy to see that the system $\{(w, e)\}$ is confluent. If w is a nonempty string such that $w = y^k$ for some y and some $k > 1$ and there are no u, v with $|u| < |y|$ and $uy = yv$, then the system $\{(w, e)\}$ is confluent. If w is a string such that $uw = wv$ for some u, v with $|u| < |w|$ and $u \neq v$, then there exist nonempty strings x, y and $k \geq 1$ with $x \neq y$ and $w = (xy)^k x$ [14]. In this case the system $\{(w, e)\}$ is not confluent, since $xy \leftrightarrow xyw = wyx \leftrightarrow yx$ so that xy and yx are irreducible, congruent, and unequal. For such systems it may still be the case that every congruence class is a context-free language as seen by the above examples. In fact, if x, y are strings such that $xy \neq yx$, then there is a finite-state machine which gives the unique factorization of each string $w \in \{x, y\}^*$ as a concatenation of x's and y's [6]. This leads to the following fact.

THEOREM 3.8. *Let u, v be two strings such that $uv \neq vu$. Let w be a string in $\{u, v\}^*$, $w \neq e$. The system $S = \{(w, e), (uv, vu)\}$ has the property that the intersection of $\{u, v\}^*$ with any congruence class or any finite union of congruence classes of S is a deterministic context-free language.*

At one point the author conjectured that for every string w, every congruence class of the Thue system $\{(w, e)\}$ is a deterministic context-free language. However, Jantzen [13] has shown that if $w = abbaab$, then the Thue system $S = \{(w, e)\}$ generates a congruence that cannot be generated by any finite preperfect system and *no* congruence class of S is context-free.

4. *The Word Problem*

It is known [1] that the word problem for special Thue systems with a single relation is decidable; however, the question of decidability of the word problem for nonspecial Thue systems with a single defining relation is open. In this section it is shown that for finite Thue systems that are Church–Rosser, the word problem is solvable in linear time.

Consider the deterministic pushdown store acceptor D described in the proof of Theorem 3.3. The first phase of D's computation on an input string x produces an irreducible string y on its pushdown store with the property that $x \overset{*}{\rightarrow}{}^{L} y$. If D is altered so as to output the final contents of the pushdown store, then one can run D on x_1 producing irreducible y_1 such that $x_1 \overset{*}{\rightarrow}{}^{L} y_1$, run D on x_2 producing irreducible y_2 such that $x_2 \overset{*}{\rightarrow}{}^{L} y_2$, and then compare y_1 and y_2. If S is Church–Rosser, then x_1

and x_2 are congruent if and only if y_1 is equal to y_2. Since deterministic pushdown store acceptors can be made to run in linear time, this shows that if S is a finite monadic Thue system that is Church–Rosser, then the word problem for S is solvable in linear time.

Consider a finite Thue system that is Church–Rosser but not monadic. It still is the case that two strings are congruent if and only if they are both leftmost reducible to the same irreducible string. But if the system is not monadic, then it may not be possible to implement every leftmost reduction with a pushdown store machine, for it may be the case that $x_i u_i y_i \to x_i v_i y_i$ and $x_{i+1} u_{i+1} y_{i+1} \to x_{i+1} v_{i+1} y_{i+1}$ are leftmost but $|v_i| \geq 2$ and $x_{i+1} u_{i+1}$ is a proper prefix of $x_i v_i$. However, it is still the case that the word problem is solvable in linear time. To prove this, it is useful to first establish a property of all finite Thue systems.

THEOREM 4.1. *If S is a finite Thue system, then there is an algorithm that on input x produces output y such that y is irreducible and $x \xrightarrow{*}{}^{L} y$. This algorithm operates in time linear in the length of the input string.*

PROOF. Recall from Section 3 that it is assumed that if $(u, v) \in S$, then $|u| \geq |v|$. As in the proof of Lemma 3.2(a), let S' be a subsystem of S with the following properties:

(i) For every string u such that for some v, $(u, v) \in S$ and $|u| > |v|$, there is exactly one v' such that $(u, v') \in S'$ and $|u| > |v'|$.
(ii) If $(u, v) \in S'$, then $|u| > |v|$.

As in the proof of Lemma 3.1(a), for every x there is a unique y such that $x \xrightarrow{*}{}^{L}_{S'} y$ and y is both irreducible (mod S') and also irreducible (mod S). Thus it is sufficient to construct any such subsystem S' from S and construct the algorithm from S'.

Let $t_1 = \max\{|u| \mid \text{for some } v, (u, v) \in S'\}$, and let $t_2 = \max\{|v| \mid \text{for some } u, (u, v) \in S'\}$. By construction of S', $t_1 > t_2$.

Construct a deterministic Turing machine D with two pushdown stores, store 1 and store 2, that operates as follows. Initially, store 1 is empty and store 2 contains a string x with the leftmost symbol of x on the top of store 2. The step-by-step computation of D is described in terms of two operations.

(i) *READ.* D attempts to read a new symbol from store 2, popping that symbol from store 2, and pushing that symbol onto the top of store 1. If D is able to read such a symbol, then it performs the SEARCH operation. If D is not able to read such a symbol, then its computation on x is complete, and the result is the contents of store 1.

(ii) *SEARCH.* D reads at most the top t_1 symbols from store 1 and determines whether there exists a string u stored on the top $|u|$ squares of store 1 such that for some v, $(u, v) \in S'$. When there is such a string, D finds the longest such u and pops the top $|u|$ squares of store 1 while writing the corresponding v on the top $|v|$ squares of store 2 such that the leftmost symbol of v is on the top of the store. When there is no such string, D simply restores the top t_1 symbols of store 1. In the former case, perform the SEARCH operation again; in the latter case, perform the READ operation.

By definition of S', if $(u, v) \in S'$, then there is no $v' \neq v$ such that $(u, v') \in S'$; thus D operates deterministically. By construction, if $(u, v) \in S'$, then $|u| > |v|$, and so each time a string u is popped from store 1 and the corresponding string v is pushed onto store 2, the sum of the length of the contents of store 1 and the length of the contents of store 2 is decreased. Thus D's computation on x must halt after at most

$|x|$ "pop u, push v" steps. The running time of D on x is bounded by the product of the number of times D performs READ and the cost of performing SEARCH. The number of times D performs READ is $|x|$ plus the total number of symbols written on store 2 in all of the applications of SEARCH, and this sum is bounded by $|x| + t_2|x|$, since there are at most $|x|$ "pop u, push v" steps. The cost of performing SEARCH is $2t_1$. Thus there is a constant k depending only on S' such that the running time of D on input string x is bounded by $k|x|$.

If y is the string remaining on store 1 after D has completed its computation on x, where the leftmost symbol of y is on the bottom of store 1, then y is irreducible and there is a left-to-right reduction $x \xrightarrow{*}{}^{L} y$. □

Recall from Lemma 3.2(b) that if S is Church–Rosser, then for every x there is a unique irreducible y such that $x \xrightarrow{*}{}^{L} y$. Thus, if S is finite and Church–Rosser, given x_1 and x_2 use the algorithm of Theorem 4.1 to produce irreducible y_1 and y_2 such that $x_1 \xrightarrow{*}{}^{L} y_1$ and $x_2 \xrightarrow{*}{}^{L} y_2$; compare y_1 and y_2 and decide that x_1 is or is not congruent to x_2, depending on whether y_1 is or is not equal to y_2.

THEOREM 4.2. *If S is a finite Thue system that is Church–Rosser, then there is a linear-time algorithm to decide the word problem for S.*

If S is a finite Thue system that is preperfect, then the word problem for S is decidable nondeterministically using at most linear space, and there is an algorithm to solve the uniform word problem for all finite preperfect systems that uses polynomial space. Jantzen and Monien (reported in [4]) have shown that there exists a finite almost-confluent system whose word problem is PSPACE-complete.

In [5] it is shown that certain infinite monadic Thue systems that are confluent have word problems that are solvable deterministically in polynomial time.

ACKNOWLEDGMENTS. I wish to thank Matthias Jantzen and Celia Wrathall for many helpful discussions regarding this work. I am especially grateful to Ann Yasuhara for her very constructive criticism of several drafts of this paper.

REFERENCES

1 ADJAN, S Defining relations and algorithmic problems for groups and semigroups *Proc Steklov Inst Math 85,* 1966 [English version published by American Mathematical Society, 1967]

2 AHO, A, SETHI, R, AND ULLMAN, J Code optimization and finite Church-Rosser systems In *Design and Optimization of Compilers,* R Rustin, Ed, Prentice Hall, 1972, pp 89–105

3 BERSTEL, J Congruences plus que parfaites et languages algébrique *Seminair d'Informatique Théorique,* Institut de Programmation, 1976–77, pp 123–147

4 BOOK, R, JANTZEN, M, MONIEN, B, AND WRATHALL, C Results on Thue systems and semigroups Research announcement, Univ of California, Santa Barbara, Calif, May 1980

5 BOOK, R, JANTZEN, M, AND WRATHALL, C Monadic Thue systems To appear in *Theor Comput Sci*

6 BOOK, R, AND KWAN, C On uniquely decipherable codes with two codewords *IEEE Trans Comput 29* (1980), 324–325

7 BOOK, R, AND O'DUNLAING, C Testing for the Church–Rosser property *Theor Comput Sci 16* (1981), 223–229

8 BOOK, R., AND O'DUNLAING, C. Thue congruences and the Church–Rosser property *Semigroup Forum 22* (1981), 367–379

9 CHURCH, A, AND ROSSER, B Some properties of conversion *Trans Am Math Soc 39* (1939), 472–482

10 COCHET, Y Sur l'algébricité des classes de certaines congruences définies sur le monoide libre Thèse 3e cycle, Rennes, 1971

11 COCHET, Y, AND NIVAT, M Une generalization des ensembles de Dyck *Israel J. Math. 9* (1971), 389–395

12 HUET, G Confluent reductions Abstract properties and applications to term rewriting systems *J ACM 27,* 4 (Oct 1980), 797–821

13 Jantzen, M. On a special monoid with a single defining relation. *Theor. Comput Sci 16* (1981), 61–73
14 Lyndon, R , and Schutzenberger, M The equation $a^M = b^N c^P$ in a free group *Michigan Math. J 9* (1962), 289–298
15. Newman, M H A On theories with a combinatorial definition of "equivalence." *Ann Math. 43* (1943), 223–243
16 Nivat, M On some families of languages related to the Dyck languages. *2nd ACM Symp on Theory of Computing*, Northampton, Mass , 1970, pp 221–225
17 Nivat, M (with Benois, M) Congruences parfaites et quasi-parfaites. Seminaire Dubreil, 25[e] Année, 1971–72, 7-01-09
18 Rosen, B K Tree-manipulating systems and Church–Rosser theorems *J ACM 20*, 1 (Jan 1973), 160–187
19 Sakarovitch, J Syntaxe des languages de Chomsky—Essai sur le déterminisme Thèse de Doctoral d'État, Université Paris VII, 1979

RECEIVED JUNE 1979, REVISED DECEMBER 1980, ACCEPTED JANUARY 1981