Computer Science Department

TECHNICAL REPORT

CANCELLATIVITY IN FINITELY PRESENTED SEMIGROUPS

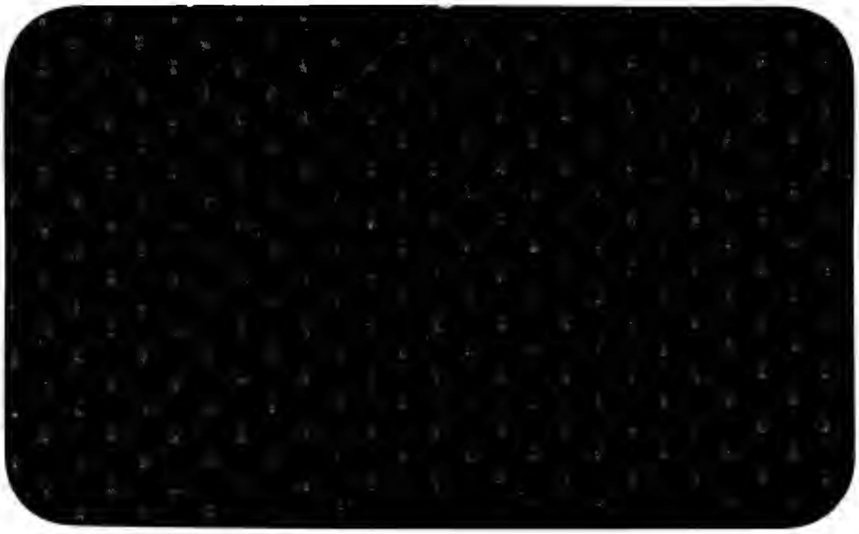
by

Paliath Narendran

Colm Ó' Dúnlaing

June 26, 1986

Report #228



CANCELLATIVITY IN FINITELY PRESENTED SEMIGROUPS

by

Paliath Narendran

Colm Ó' Dúnlaing

June 26, 1986

Report #228

CANCELLATIVITY IN FINITELY PRESENTED SEMIGROUPS[1]

Paliath Narendran
G.E. Corporate Research and Development Center
Schenectady, NY 12345

and

Colm Ó'Dúnlaing
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, NY 10012

Abstract

The question of whether a monoid presented by a finite Thue system is cancellative is shown to be undecidable (its negation is semidecidable), even when the Thue system is Church-Rosser. A decision procedure is described for the case of monadic Church-Rosser Thue systems and general commutative Thue systems.

[1] The results in this paper were first presented at the Workshop for Combinatorial Algorithms in Algebraic Structures, held in October 1985 at the Europäische Akademie, Otzenhausen, West Germany, under the auspices of Universität Kaiserslautern. Work by the first author was supported in part by the National Science Foundation under grant DCR 84-08461. Work by the second author was supported in part by the National Science Foundation under grant DCR 84-01898.

CANCELLATIVITY IN FINITELY PRESENTED SEMIGROUPS

1. Introduction

Term-rewriting systems have been of considerable interest in recent years due to their applications in such diverse areas as abstract data types, theorem proving, database schemes and computer algebra. Most of these applications are centered around their *word problems*, or, in other words, equivalences of terms in their equational theories. Though in their full generality they are computationally infeasible (because the word problem is undecidable in general) some large decidable subclasses of them have been found. For instance, term-rewriting systems with the *Church-Rosser property* (also called *canonical* or *complete* term-rewriting systems) are specially of interest, since the Church-Rosser property enables us to compute normal forms for equivalence classes of terms.

In this paper, we focus our attention on string-rewriting systems or *Thue systems*, which one can regard as presentations of semigroups. These have been studied in great detail in [2,3,11,15,16,17]. The Church-Rosser property (though defined in a more restricted way than for general term-rewriting systems) again plays an important role in these studies. The algebraic properties of semigroups presented by Church-Rosser Thue systems have received a lot of attention. Their algorithmic properties have been investigated too; here the effort has been to identify decision problems, undecidable in general, which turn out to be decidable once the Church-Rosser property is assumed.

In this paper we consider an algorithmic problem - that of deciding whether a semigroup is cancellative - for Church-Rosser Thue systems and for general commutative Thue systems. We show that the Church-Rosser property is not of much help here; the problem is undecidable for arbitrary Church-Rosser systems. But if they are also assumed to be *monadic* then cancellativity is decidable. For commutative systems which are canonical (i.e., Church-Rosser relative to the lexicographic ordering of vectors), we show that cancellativity is in co-**NP**, and deduce that cancellativity is decidable for general commutative Thue systems.

2. Basic Definitions

2.1. Strings Over an Alphabet

Let Σ be any finite alphabet and Σ^* the set of all possible strings over Σ, including the null string λ. Given a string w in Σ^*, $|w|$ denotes its length. Given strings u and v, their product uv is obtained by concatenating v onto u. A string x is said to be a *prefix* (respectively, *suffix*) of y if there exists z such that $y = xz$ (respectively, zx); x is a *proper* prefix (respectively, suffix) of y if x is a prefix (suffix) of y and $|x| < |y|$.

2.2. Thue Systems

A Thue system T is a set of pairs of strings over Σ^*:

$$T = \{L_i \longleftrightarrow R_i : i = 1, \ldots, k\}.$$

Formally, the elements of T, which we call *rules* (or relations) of the Thue system, are merely ordered pairs of strings, but the notation '$L \longleftrightarrow R$' is more suggestive than '(L,R).' The Thue congruence $\overset{*}{\longleftrightarrow}$ defined by T is the reflexive transitive closure of the relation $\longleftrightarrow$ defined as follows: if $u \longleftrightarrow v$ is an element of T, then for all x, y, $xuy \longleftrightarrow xvy$ and $xvy \longleftrightarrow xuy$. When $x \overset{*}{\longleftrightarrow} y$, we say that x and y are *congruent modulo T*. (Sometimes we write $\longleftrightarrow_T$, and so on, where the defining Thue system is not obvious from the context. Also, the phrase 'modulo T' will be omitted whenever it is obvious from the context.) For any string x, $[x]_T$ denotes the congruence class containing x, i.e.,

$$[x]_T = \{y : y \overset{*}{\longleftrightarrow}_T x\}.$$

We write $x \longrightarrow y$ if $x \longleftrightarrow y$ and $|x| > |y|$. Let $\overset{*}{\longrightarrow}$ denote the reflexive, transitive closure of $\longrightarrow$. The relation $\overset{*}{\longrightarrow}$ is referred to as *reduction* (modulo T). When a rule $L \longleftrightarrow R$ has sides of differing length, we call the longer the *redex* and the shorter the *reduct* in the rule; more generally, if $\alpha \longleftrightarrow \beta$ or $\beta \longleftrightarrow \alpha$ is such a rule and α is the redex, then for any strings u and w we write $u\alpha w \longrightarrow u\beta w$ and speak of the designated occurrences of α and β in the respective strings as the redex and reduct, respectively. Inverse to the reduction relation $\overset{*}{\longrightarrow}_T$ is the *expansion* relation $\overset{*}{\longleftarrow}_T$: we write also $\longleftarrow_T$ to indicate the application of a single length-increasing rewrite. Thus we can speak of 'reducing a redex' or 'expanding a reduct' in a string. If $x \overset{*}{\longrightarrow} y$ then x is an *ancestor* of y and y is a *descendant* of x. For a set X of strings, $\Delta^*(X)$ denotes the set of all strings which are descendants of strings in X; i.e.,

$$\Delta^*(X) = \{x : w \overset{*}{\longrightarrow} x \text{ for some } w \text{ in } X\}.$$

Two strings x and y are said to be *joinable* if they have a common descendant. Clearly two joinable strings are congruent. A string w is *irreducible* (modulo T) if there is no y such that $w \longrightarrow y$. $IRR(T)$ denotes the set of all strings that are irreducible modulo T. If $x \overset{*}{\longrightarrow} y$ and y is in $IRR(T)$, then y is called a *normal form* for x.

A Thue system T is *Church-Rosser* if and only if every two congruent strings are joinable. In other words, for every choice of x and y, $x \overset{*}{\longleftrightarrow} y$ implies that for some z, $x \overset{*}{\longrightarrow} z$ and $y \overset{*}{\longrightarrow} z$. It can be shown that in a Church-Rosser system every string has a unique normal form.

A Church-Rosser system T is *reduced* if, for every rule $L \longleftrightarrow R$ in T, neither L nor R is reducible modulo $T - \{L \longleftrightarrow R\}$. Two Thue systems T and U are *equivalent* if they are defined over the same alphabet and for any two strings x and y, $x \overset{*}{\longleftrightarrow}_T y$ if and only if $x \overset{*}{\longleftrightarrow}_U y$. In view of the following result, we shall assume in the rest of the paper that every Church-Rosser system is reduced:

Proposition 2.1 [13]: For every Church-Rosser Thue system T, there is a unique reduced Church-Rosser Thue system T' equivalent to T. If T is finite then T' is also effectively computable from T. ●

All the systems T constructed in this paper will be both Church-Rosser and reduced based on the following condition, whose sufficiency is not difficult to show:

> If T is a Thue system in which (i) in every rule the two sides have different lengths and the redex determines the reduct uniquely, (ii) no two distinct redexes of T overlap at all, and (iii) all reducts have the same length, then T is Church-Rosser and reduced.

A Thue system is *monadic* if, in every rule, one side has length 1 or 0, and the other side is longer.

3. Cancellative Semigroups

In this paper we shall investigate the property of cancellativity in finitely presented monoids. Throughout the discussion, a Thue system T is considered to *present* a monoid M_T (i.e., a semigroup with identity) in the following sense:

The elements of M_T are the congruence classes $[x]_T$,

Given two elements $[x]_T$ and $[y]_T$, their product in the semigroup is defined as $[xy]_T$ (clearly, this is independent of choice of representatives of the two congruence classes), and

$[\lambda]_T$ is the identity in the monoid.

A semigroup G satisfies the *left cancellation law* (or is *left-cancellative*) if for all members α, x, and y of G, $\alpha x = \alpha y$ implies $x = y$. The right cancellation law is defined similarly in the obvious way. A semigroup is *cancellative* if it is both left- and right-cancellative.

In what follows, of course, we shall be thinking only of semigroups which are monoids presented by appropriate Thue systems. Throughout the rest of the paper we say 'T is cancellative' to abbreviate 'the semigroup presented by T is cancellative.' The following should be noted first:

Lemma 3.0: Given a finite Thue system T, it is undecidable whether T is (i) left-cancellative, (ii) right-cancellative, or (iii) both.

Proof. The proofs for all three are identical, so we shall consider (i) only. It is enough to show that left-cancellativity is a Markov property [11,12,16]. In other words, it is enough to prove (a) the property is a property of the monoid M_T independent of its presentation: immediate from the definition; (b) there exists a system T with this property: any presentation of the trivial semigroup is cancellative; and (c) there exists a monoid M_S which cannot be embedded in any left-cancellative monoid M_T: true, since, first, the system $S = \{c \longleftrightarrow ca\}$ is clearly not (left-) cancellative, and, second, no cancellative monoid contains a non-cancellative submonoid. **Q.E.D.**

Indeed, it is possible to extend the above result to Thue systems which are monadic. We shall derive this stronger result from the analogue to Markov's theorem in finitely presented groups; then we can exploit the fact that every finite presentation of a group, in which the inverses of symbols are implicit, may be used to generate effectively a Thue system which presents an isomorphic *monoid*; the latter Thue system is obtained by explicitly adding new symbols to represent the inverse and explicitly adding rules of the form $aa^{-1} \longleftrightarrow \lambda$ and $a^{-1}a \longleftrightarrow \lambda$. The resulting Thue system is special, i.e., in every rule the reduct is the empty string λ.

Proposition 3.0.1. (See [10]). Let S be a special (and therefore monadic) Thue system presenting a group G. Given any word w over S one can construct effectively a monadic Thue system $S(w)$ (also presenting a group) over a larger alphabet which contains the alphabet of S, and a homomorphism h from M_S to $M_{S(w)}$ such that if w is congruent to λ then $S(w)$ presents the trivial group, and if w is not congruent to λ then h is an embedding. ●

We can assume that the group G presented by S has undecidable word problem, and also that we can identify a symbol a in the alphabet for the presentation such that a is not equal to the identity in G, so it is not congruent to λ (modulo S). If necessary, choose a to be a new symbol not mentioned in the rules of S and add it to the alphabet for G: in the resulting group the word problem remains undecidable. Let c be a symbol not in the alphabet of $S(w)$, and define $T(w) = S(w) \cup \{ca \longleftrightarrow c\}$. Clearly $T(w)$ is monadic, and it is easy to show that $M_{T(w)}$ is (left-) cancellative if and only if $w \overset{*}{\longleftrightarrow}_S \lambda$. Since the word-problem for G reduces effectively to deciding whether certain monoids $M_{T(w)}$ are (left-) cancellative, and we can choose G to have undecidable word-problem, the following generalization of Lemma 3.0 is immediate.

Theorem 3.0.2 Cancellativity, or left-cancellativity, is an undecidable property of Thue systems S even when the systems S are required to be monadic. ●

We next consider the technically more challenging problem of determining whether cancellativity is decidable for *Church-Rosser* (and, by implication, reduced) Thue systems T.

Lemma 3.1: A Thue system T presents a left-cancellative semigroup M_T if and only if the following condition holds:

For all a in Σ and x, y in Σ^*,

$$ax \overset{*}{\longleftrightarrow}_T ay \text{ implies } x \overset{*}{\longleftrightarrow}_T y.$$

Proof: Clearly the 'only if' part is true. To prove the converse, suppose that there exist strings α, x', and y' such that x' and y' are not congruent (modulo T) but $\alpha x'$ and $\alpha y'$ are. Let us suppose that α has minimum length among such strings, and write $\alpha = a\beta$ for some alphabet symbol a. Then by minimality of $|\alpha|$, $\beta x'$ and $\beta y'$ are not congruent, and so we can choose $x = \beta x'$ and $y = \beta y'$. ●

Corollary 3.1.1: A Church-Rosser Thue system T presents a left-cancellative semigroup M_T if and only if the following condition holds:

For all a in Σ and x, y in $IRR(T)$,

$$ax \overset{*}{\longleftrightarrow}_T ay \text{ implies } x = y.$$

Proof: An easy consequence of Lemma 3.1 and the definitions. ●

The following further corollary is easily shown:

Lemma 3.2: Let T be a reduced Church-Rosser system such that M_T is left-cancellative. Then T cannot contain rules of the form $ax \longleftrightarrow ay$ where $a \in \Sigma$. ●

The following theorem is stronger than Lemma 3.1 in the case where T is Church-Rosser and reduced.

Theorem 3.3: Let T be a reduced Church-Rosser system. Then T is *not* left-cancellative if and only if there exist a in Σ and y_1, y_2 in $IRR(T)$ such that

(a) $y_1 \neq y_2$,

(b) $ay_1 \longleftrightarrow_T^* ay_2$, and

(c) the redexes in ay_1 and ay_2 are incompatible (i.e., neither is a prefix of the other, so the strings cannot be reduced by the same rule of T).

Proof: The 'if' part is trivial.

Only if: Assume M_T is not left-cancellative and let S be the set of all triples (a,y_1,y_2) satisfying (a) and (b). By Corollary 3.1.1, S is nonempty. Let (a,y_1,y_2) be a triple from S such that

(d) $|y_1|+|y_2|$ is minimal.

Now claim that (c) holds; otherwise, there exists a common prefix β of y_1 and y_2 and a rule $a\beta \longleftrightarrow \gamma$ in T which can be applied to both strings. Write $y_i = \beta z_i$ ($i = 1,2$), so γz_i are congruent but z_i are not. Choose the longest (necessarily nonempty) suffix of γ, $a'\beta'$, say, where a' is a single symbol, such that $a'\beta' z_i$ are congruent but $\beta' z_i$, which we write as y_i', are not. Then the triple (a',y_1',y_2') is also in S, contradicting (d): this proves the claim. **Q.E.D.**

The criteria in Theorem 3.3 can be rephrased as follows. Suppose that $au \longrightarrow v$ is a rule which applies to ay_1, so, by (c), u is not a prefix of y_2. Let us write

$$R_1(au) = \{\ aw : w \in IRR(T) \text{ and } u \text{ is a prefix of } w\ \},$$

and

$$R_2(au) = \{\ ax : x \in IRR(T) \text{ and } u \text{ is not a prefix of } x\}.$$

So ay_i is in $R_i(au)$ ($i = 1,2$), and they have a common descendant in

$$\Delta^*(R_1(au)) \cap \Delta^*(R_2(au))$$

(the intersection of the sets of descendants of strings in $R_1(au)$ and $R_2(au)$); therefore this intersection is nonempty. Conversely, suppose that the intersection is nonempty. Then there must exist strings ay_i in $R_i(au)$ ($i = 1,2$), which have a common descendant. These strings are clearly different by definition of the two sets $R_i(au)$; thus ay_i are congruent but y_i are not, and the system T is not left-cancellative. Thus we can conclude the following:

Theorem 3.4. A reduced Church-Rosser Thue system T presents a left-cancellative monoid if and only if for all redexes au of rules of T, $\Delta^*(R_1(au)) \cap \Delta^*(R_2(au)) = \varnothing$. ●

Note that the sets $R_i(au)$ are always regular. Thus the above theorem gives us a decision procedure for finite monadic Church-Rosser Thue systems, since for every (effective description of a) regular set R, $\Delta^*(R)$ is also regular and effectively computable [2], and the same holds, obviously, for $IRR(T)$. Thus,

Theorem 3.5: Given a finite monadic Church-Rosser Thue system T, it is decidable whether M_T is left-cancellative. ●

Clearly this also implies the existence of a decision procedure to test whether a finite monadic Church-Rosser Thue system is cancellative, since right-cancellativity can be shown to be decidable by a symmetric argument.

4. The Case of Arbitrary Church-Rosser Systems

In this section we show that the problem of cancellativity is undecidable for arbitrary Church-Rosser systems. The construction is a modification of that given in [14]. We start with a Turing Machine Z with an undecidable halting problem and construct a Church-Rosser Thue system T_1 over an alphabet Σ_1 and a regular set $FINAL \subset \Sigma_1^*$ with the following property:

> T_1 is cancellative, and, given a string x, the question of whether $[x]_{T_1} \cap FINAL$ is empty is undecidable.

We then construct another Thue system T_2 (by adding more rules to T_1) and a symbol a from an extended alphabet Σ_2 such that

> T_2 is cancellative and Church-Rosser, $a \in \Sigma_2 - \Sigma_1$, and given a string w, the question of whether there exists a string y such that $ay \xrightarrow{*} w$ is undecidable.

Finally, we show how, for a given w, T_2 can be extended to T_3 such that T_3 is cancellative if and only if there does not exist any string y such that $ay \xrightarrow{*} w$.

We shall now outline the construction of T_1. Since the construction to be discussed here is only a trivial modification of that in [14], we shall present it with a minimum of detail. Let $Z = (K, \Sigma, \Pi, \mu, q_0, q_f, \beta)$ be a Turing machine where K is the (finite) set of states, Σ and Π the input and tape alphabets respectively ($\Sigma \subseteq \Pi$), μ the transition function (expressed as quintuples: see below), q_0 the initial state, q_f the final state, and β the blank symbol. Assume also that the halting problem for Z is undecidable. We now construct T_1 which is cancellative and Church-Rosser, over an alphabet Γ, such that the action of the Turing Machine Z is simulated by *expansion* modulo T_1. The alphabet Γ is described below.

One begins with K (regarded as a finite alphabet) and Π, which are (without loss of generality) disjoint sets. 'Mirror-image sets' $\overline{\mathrm{K}}$ and $\bar{\Pi}$ are constructed, where $\bar{\Pi} = \{ \bar{a} : a \in \Pi\}$, $\mathrm{K} = \{q_0, \cdots, q_n\}$, and $\overline{\mathrm{K}} = \{p_0, \cdots, p_n\}$. Symbols in Π and $\bar{\Pi}$ we call respectively right and left tape symbols, and the other symbols are called right and left state symbols. As in [14], we classify the symbols of Γ as follows:

Left and right *'endmarkers'* \$ and ¢.

State symbols, from K and $\overline{\mathrm{K}}$; the distinction is that the symbols in K will be called 'right state-symbols,' and the symbol being 'scanned' will lie to their right, and the symbols in $\overline{\mathrm{K}}$ will be called 'left state-symbols,' and the symbol being scanned will be to their left.

Left and right *tape symbols*. The right tape symbols occur to the right of the state-symbol, and are identical to the symbols of Π; $\bar{\Pi}$ is the set of left tape-symbols.

Left *dummy symbols*. These are dummy symbols to the left of the state symbol. There is one symbol L_z for every pair z in the following two sets:

$$\mathrm{K} \times (\{\text{¢}\} \cup \Pi)$$

and

$$(\{\$\} \cup \bar{\Pi}) \times \overline{\mathrm{K}} .$$

There is also a dummy symbol $L_\$$ which can appear only to the left of the symbol \$; D_L denotes the set of left dummy symbols.

Right dummy symbols R_z for the same sets of pairs z, and an additional symbol $R_{\text{¢}}$, which can only appear after the symbol ¢; D_R denotes the set of right dummy symbols.

The alphabet Γ is the set of all symbols in the above five classes. The set $\{\$\} \cup D_L \cup \bar{\Pi}$ comprises the *left symbols* in Γ; similarly the set $\{\text{¢}\} \cup D_R \cup \Pi$ comprises the *right symbols* of Γ. Finally, we define regular sets *CONFIG*, which encodes all possible configurations of the machine Z, and *FINAL*, which encodes all the final configurations of Z:

$$CONFIG = L_\$^* \cdot \{\$\} \cdot (\bar{\Pi} \cup D_L)^* \cdot (\mathrm{K} \cup \overline{\mathrm{K}}) \cdot (\Pi \cup D_R)^* \cdot \{\text{¢}\} \cdot R_{\text{¢}}^* ,$$

$$FINAL = L_\$^* \cdot \{\$\} \cdot (\bar{\Pi} \cup D_L)^* \cdot \{q_f, p_f\} \cdot (\Pi \cup D_R)^* \cdot \{\text{¢}\} \cdot R_{\text{¢}}^* .$$

Clearly *FINAL* ⊆ *CONFIG*.

The construction of T_1 is illustrated by the following table (in it, $\bar{a}$ represents the symbol in $\bar{\Pi}$ corresponding to the symbol a in Π and R_z and L_z are assumed to be pairs of corresponding dummy symbols):

Definition of T_1	
Quintuple of Z	Rules of T_1
$q_i abRq_j$	$q_i a \longleftrightarrow L_{q_i a}\bar{b}q_j$ $\bar{a}p_i \longleftrightarrow L_{\bar{a}p_i}\bar{b}q_j$
$q_i \beta bRq_j$	$q_i\text{¢} \longleftrightarrow L_{q_i\text{¢}}\bar{b}q_j\text{¢}R_{\text{¢}}$ $\$p_i \longleftrightarrow L_\$\$L_{\$p_i}\bar{b}q_j$
$q_i abLq_j$	$q_i a \longleftrightarrow p_j bR_{q_i a}$ $\bar{a}p_i \longleftrightarrow p_j bR_{\bar{a}p_i}$
$q_i \beta bLq_j$	$q_i\text{¢} \longleftrightarrow p_j bR_{q_i\text{¢}}\text{¢}R_{\text{¢}}$ $\$p_i \longleftrightarrow L_\$\$p_j bR_{\$p_i}$
n/a ($i \neq f$)	$q_i R_z \longleftrightarrow L_z L_z q_i$ $L_z p_i \longleftrightarrow p_i R_z R_z$

It can be seen by inspection that T_1 is Church-Rosser and reduced, since no two redexes overlap, and no two rules have the same redex. Note also that the construction of T_1 is basically the same as the construction given in [14] except for the usage of the new dummy symbols $L_\$$ and $R_{\text{¢}}$: they are added to ensure cancellativity. For technical reasons it is useful that p_f and q_f be excluded from the rules of category 5. As in [14], we can prove

Theorem 4.1 : Z halts on input x if and only if there is some string in *FINAL* that is an ancestor of $\$q_0x\text{¢}$ modulo T_1. ●

Consequently it is undecidable for input strings x whether $[\$q_0x\text{¢}]_{T_1} \cap FINAL$ is empty.

Theorem 4.2 : T_1 is cancellative.

Proof. Assume T_1 is not left-cancellative. Then by Theorem 3.3 there exist $a \in \Gamma$, and $x, y \in IRR(T_1)$ such that $x \neq y$, ax and ay are joinable and ax and ay do not get reduced by the same rule.

Let $CRUX1$ be the set of all possible substrings of strings in $CONFIG$; i.e.,

$$CRUX1 = \{\ w : xwy \in CONFIG \text{ for some } x \text{ and } y\ \}.$$

Every string s in Γ^* can be factored as a sequence $s_1 s_2 \cdots s_k$ where the segments $s_1, \ldots, s_k$ are in $CRUX1$, and maximal in the sense that no string s_i is contained in a longer string also belonging to $CRUX1$. (Informally speaking, given s, the sequence $s_1 s_2 \cdots s_k$ can be got by deleting from s the longest prefix belonging to $CRUX1$ and repeating until the string is exhausted.) By arguments similar to those in [14] if $s \overset{*}{\longleftrightarrow} t \pmod{T_1}$ then t has the form $t_1 t_2 \cdots t_k$ where $t_j \in CRUX1$ and $s_j \overset{*}{\longleftrightarrow} t_j \pmod{T_1}$ for $1 \leq j \leq k$. Hence in the proof that follows we may assume $k = 1$ and we only need to consider a, x and y such that ax and ay belong to $CRUX1$. Furthermore, as noted in [14],

> two strings in $CRUX1$ are joinable if and only if one can be reduced to the other.

By inspecting the redexes of T_1 we can conclude that a is either a left dummy-symbol or a left state-symbol. Inspecting the reducts of T_1 we conclude that (since a must recur as a reduct) it is a left dummy-symbol and it recurs in a rule of the form $L_i p_i \longleftrightarrow p_i R_i R_i$. However, expanding the string ay according to this rule yields a string beginning with p_i, and since this is the only occurrence of a state-symbol in the string (since $k = 1$), the new string cannot be expanded further: i.e., $ax \xrightarrow{*} ay$ implies $x = y$. This concludes the proof. **Q.E.D.**

Now we construct a system T_2 extending T_1. First consider the set

$$\{C\} \cdot FINAL \cdot \{D\}$$

where C and D are symbols not in Γ. Let $s_0, \cdots, s_3$ be four new symbols, let Ω denote $\Gamma \cup \{C, D\}$, and let $\bar{\Omega}$ be a disjoint copy of Ω, and which does not contain the symbols s_i (Note: Ω contains both Π and $\bar{\Pi}$; $\bar{\Omega}$ contains yet another copy of $\bar{\Pi}$). We define a Thue system S with the following rules:

$$C\$s_1 \longleftrightarrow s_0 \bar{C}\bar{C}\$\bar{\$},$$
$$as_1 \longleftrightarrow s_1 \bar{a}\bar{a},$$

where a is any left tape- or dummy-symbol,

$$p_f s_2 \longleftrightarrow s_1 \bar{p}_f \bar{p}_f,$$
$$q_f s_2 \longleftrightarrow s_1 \bar{q}_f \bar{q}_f,$$
$$a s_2 \longleftrightarrow s_2 \bar{a}\,\bar{a},$$

where a is any right tape- or dummy-symbol, and

$$¢ D s_3 \longleftrightarrow s_2 \bar{¢}\,\bar{¢}\bar{D}\bar{D}.$$

The rules of S imitate a finite automaton accepting the regular set $\{C\}\cdot FINAL\cdot\{D\}$, if we think of s_3 as being the 'initial' state and the automaton reads its input from right to left. The following lemma contains some straightforward properties of the systems S and T_1, and the simple but tedious proofs will be omitted.

Lemma 4.3. (i) Given a string ω in Ω^*, the string ωs_3 has an ancestor beginning with s_0 (modulo S) if and only if $\omega \in \{C\}\cdot FINAL\cdot\{D\}$, and given a string η in $(\Omega\cup\bar{\Omega})^*$, the string $s_0\eta$ has a descendant ending in s_3 if and only if η is the homomorphic image $h(\theta)$ of a string θ in $\{C\}\cdot FINAL\cdot\{D\}$ under the homomorphism h: $a \rightarrow \bar{a}\bar{a}$ on Ω; (ii) if α is a string over the alphabet $\Omega\cup\bar{\Omega}\cup\{s_0, \cdots s_3\}$, containing just one state-symbol (q_i or p_i) and $\alpha \overset{*}{\longleftrightarrow}_{T_1} \beta$, then either $\alpha \overset{*}{\rightarrow}_{T_1} \beta$ or $\beta \overset{*}{\rightarrow}_{T_1} \alpha$; (iii) if α is a string over the alphabet given in (ii) containing exactly one symbol from $\{s_0, \cdots, s_3\}$, and $\alpha \overset{*}{\longleftrightarrow}_S \beta$, then either $\alpha \overset{*}{\rightarrow}_S \beta$ or $\beta \overset{*}{\rightarrow}_S \alpha$. ●

The system T_2 is defined as follows:

$$T_2 = S \cup T_1.$$

Theorem 4.4. (i) Both S and T_2 are Church-Rosser and reduced; (ii) a string x in Σ^* is accepted by the Turing machine Z if and only if the string $C\$q_0x¢Ds_3$ has an ancestor $s_0\eta$ (modulo T_2), where η is in $\bar{\Omega}^*$; and (iii) T_2 is cancellative.

Proof. (i) This can be argued in the usual way: the redexes do not overlap and no two rules have the same redex. (ii) The 'only if' part is straightforward, so we concentrate on the 'if' part. Therefore, suppose that the string $X = C\$q_0x¢Ds_3$ has an ancestor $s_0\eta$ where η is in $\bar{\Omega}^*$. Let ξ be the unique irreducible descendant of $s_0\eta$ (modulo S) (note: *not* modulo T_2). Notice that every descendant of ξ (modulo T_1) is also irreducible (modulo S): this implies that X is the unique irreducible descendant of ξ (modulo T_2). But no rule of T_1 can affect any symbol s_i, so, since X ends with s_3, so does ξ. By Lemma 4.3, ξ is in $\{C\}\cdot FINAL\cdot\{Ds_3\}$, so Z accepts x by Theorem 4.1.

(iiia) T_2 is left-cancellative: we argue by contradiction. Suppose otherwise, so by Lemma 3.3 there exist strings $x \neq y$ in $IRR(T_2)$, and a symbol σ such that σx and σy are congruent (modulo T_2) and the redexes in these strings are incompatible in the sense that neither is a prefix of the other. Clearly, T_1 is also cancellative over the (larger) alphabet of T_2, so σx and σy are not congruent (modulo T_1). Let ξ and η be the irreducible descendants of σx and σy (modulo S). Therefore we can write

$$\sigma x \xrightarrow{*}_S \xi \xrightarrow{*}_{T_2} \rho \xleftarrow{*}_{T_2} \eta \xleftarrow{*}_S \sigma y, \tag{4.1}$$

and observe, as before, that no string between ξ and η in (4.1) can be reduced by any rule of S. Also we can assume that ρ is in $IRR(T_2)$. Therefore the transformations carrying ξ to η use only the rules of T_1.

Next we observe that the strings ξ and η each contain at least one of the symbols s_i. This is because, since T_1 is cancellative, at least some rules of S must be involved in the above chain of transformations, and therefore at least one string in the sequence, and hence all, must contain at least one symbol from $\{s_0, \cdots, s_3\}$. Let αs be the shortest prefix of ξ containing one of these symbols, and let βt be the shortest such prefix of η (so both s and t are among the symbols s_i). Since no rule of T_1 involves any symbol s_i, it follows that $s = t$ and there exists a prefix π of ρ which is a common descendant of α and β (modulo T_1). Furthermore, by definition of the system S, α, β, and π have the property that they have at most one occurrence of a state-symbol each, all the symbols to the left of it are left-symbols (or C), and all the symbols to the right of it are right-symbols (or D). Thus, these strings are substrings of strings in $\{C\} \cdot CONFIG \cdot \{D\}$, and just as for strings in $CRUX1$ either α is an ancestor of β (modulo T_1) or vice-versa. But neither string can be expanded by any rule of T_1[2] so $\alpha s = \beta t$. However, by definition of S, σx and σy begin with the same string (an ancestor of αs) and therefore begin with compatible redexes, contradicting the assumption. Therefore T_2 is left-cancellative.

(iiib) T_2 is right-cancellative. We can duplicate the reasoning of (iiia) to establish that if it is not right-cancellative then there exists a sequence of transformations[3]

$$x\sigma \longrightarrow_S^- \xi \overset{*}{\longleftrightarrow}_{T_1} \eta \longleftarrow_S^- y\sigma,$$

where x and y are in $IRR(T_2)$, ξ and η are in $IRR(S)$, σ is a single symbol from $\bar{\Omega}$, and the redexes in $x\sigma$ and $y\sigma$ are incompatible. But it is clear from the definition of S that σ determines the redex uniquely, and the contradiction is immediate. This concludes the proof. **Q.E.D.**

[2] This is why rules involving the accepting state-symbols were excluded from category 5 of the table for T_1.

[3] Here $\longrightarrow^-$ denotes the transitive closure of $\longrightarrow$ rather than the reflexive transitive closure $\xrightarrow{*}$.

Given an input string x for Z, we form T_3 from T_2 by adding the rule

$$C\$q_0x¢Ds_3 \rightarrow s_0\, D. \qquad (4.2)$$

Theorem 4.5: (i) T_3 is Church-Rosser and reduced, (ii) it is right-cancellative, and (iii) it is left-cancellative if and only if Z fails to halt on input x.

Proof. (i) This follows from the same considerations as applied to T_1 and T_2.

(ii) Supposing that T_3 is not right-cancellative, by Lemma 3.1 there exist irreducible strings u and v which are incongruent (modulo T_3), and a symbol σ in $\Omega \cup \bar{\Omega} \cup \{s_0, \cdots, s_3\}$ such that

$$u\sigma \xrightarrow{*}_{T_3} \rho \xleftarrow{*}_{T_3} v\sigma. \qquad (4.3)$$

We can assume that (4.3) is realized by a minimal-length sequence of transformations. Since T_2 is cancellative, the sequence (4.3) must involve at least one application of the rule (4.2), whose reduct, s_0D, overlaps no other redex nor reduct in T_3.

Consider all occurrences of s_0D in ρ. Among all the reducts and redexes of the rules in T_3 this string overlaps only one, where it occurs as a reduct of rule (4.2). Thus some occurrences will be expanded in producing $u\sigma$ and others will be expanded in producing $v\sigma$. We may assume that no occurrence persists (without being rewritten) throughout the sequence (4.3), since otherwise we could factorize the strings and work with suffixes to the right of the rightmost 'persistent' occurrence of s_0D. Moreover, we can suppose that not occurrence is expanded both in producing $u\sigma$ and $v\sigma$, since that would be a redundant operation and the sequence (4.3) could be shortened.

If the rightmost occurrence of s_0D in $u\sigma$ is as a suffix then $\sigma = D$. Then u and v are irreducible but uD and vD are not; this is impossible since no redex of T_3 ends in D. Hence we may assume that neither s_0D occurs as a suffix in neither u nor v.

Construct new strings $U\sigma$ and $V\sigma$ by uniformly replacing occurrences of s_0D in $u\sigma$ (respectively, $v\sigma$) by the redex in (4.2). Claim that these strings are congruent (modulo T_2), because if we follow the reductions applied in (4.3) but avoiding applications of the rule (4.2), we obtain a common descendant ρ' which corresponds to replacing every occurrence of s_0D in ρ by the redex $C\$q_0x¢Ds_3$. Since T_2 is cancellative, U and V are congruent (modulo T_2) and therefore u and v are congruent (modulo T_3), a contradiction.

(iii) It is easy to prove the 'only if' part, so we concentrate on the 'if.' Thus suppose there exist strings u and v which are incongruent (modulo T_3) and a symbol σ such that

$$\sigma u \xrightarrow{*}_{T_3} \rho \xleftarrow{*}_{T_3} \sigma v. \tag{4.4}$$

By reasoning as in (ii) we can rule out the possibility that nonempty prefixes of both σu and σv are unaffected by the new rule $X \longleftrightarrow s_0 D$ (X denotes the string $C\$q_0x\text{¢}Ds_3$). Therefore we can assume that $s_0 D$ is a prefix of σv and this is replaced by X somewhere in the sequence of transformations (4.4). We can therefore write (4.4) as

$$s_0 u \xrightarrow{*}_{T_3} Xu' \longrightarrow s_0 Du' \xleftrightarrow{*}_{T_3} s_0 v. \tag{4.5}$$

Furthermore, since the string X has only one occurrence of D, we can write

$$s_0 u \xrightarrow{*}_{T_2} Xu',$$

by, if necessary, 'postponing' reductions involving the rule (4.2) which cannot apply to the designated part of the strings. Note that in Xu', no further reductions (modulo T_2) can be applied except in u'. We then consider another canonical sequence of reductions beginning with $s_0 u$:

$$s_0 u \xrightarrow{*}_{S} u_1 t u_2 \xrightarrow{*}_{T_2} Ytu_2, \tag{4.6}$$

where we first apply rules of S as long as possible, to the leftmost s_i-symbol in the reduced strings, so that in the sequence (4.6) t is an s_i-symbol, leftmost, and no reduction (modulo S) applied to tu_2 can involve the designated occurrence of t, and then reduce u_1 to an irreducible string Y (modulo T_1). Therefore we have

$$Xu' \xleftrightarrow{*}_{T_2} Ytu_2,$$

so these strings possess a common irreducible descendant (modulo T_2). But no reduction (modulo T_2) can further affect either X or Yt, so we conclude that $X = Yt$, so by Theorem 4.4, applied to those prefixes of the strings affected by the transformations (4.6), Z accepts the input string x. This concludes the proof. **Q.E.D.**

It immediately follows that the question: Is T_3 (implicitly parametrized by input strings x for Z) cancellative? is undecidable, and hence cancellativity is undecidable for Church-Rosser systems. Finally, we should note that the property of a finite Church-Rosser system T *not* being cancellative is semidecidable. This follows from the definition of cancellativity since the word problem is decidable for T: there exist strings x, y, and z, where x and y are not congruent but either xz and yz, or zx and zy, are congruent. Combining with Theorem 4.5 it follows that cancellativity is complete for Π_1 in the arithmetic hierarchy.

5. Cancellativity of commutative Thue systems is decidable

In this section, we show that the cancellativity problem is decidable for finitely presented commutative Thue systems. A *commutative* Thue system is a rewriting system over permutable strings, i.e., strings in which the order of symbols is immaterial.

Let T be a commutative Thue system over some alphabet Σ. (Permutable) strings over Σ can be viewed as Parikh vectors or, in other words, n-tuples of non-negative integers where n is the size of the alphabet: the value of the i^{th} element denotes the number of occurrences of the i^{th} symbol in the string.

Strings u and v are said to be *conjugate* if and only if there exists a string w such that $uw \overset{*}{\longleftrightarrow}_T vw$. Let G_T be the abelian group obtained from T by explicitly adding the inverses of symbols in Σ. For a string w over $(\Sigma \cup \Sigma^{-1})$ let $RED(w)$ be its reduced form. (Thus, for instance, $RED(aba^{-1}b^{-1}) = \lambda$.) Let

$$AG(T) = \{RED(u^{-1}v) \longleftrightarrow \lambda, RED(v^{-1}u) \longleftrightarrow \lambda : u \longleftrightarrow v \in T\} \cup \{aa^{-1} \longleftrightarrow \lambda : a \in \Sigma\}.$$

Thus $AG(T)$ is a commutative Thue system such that $M_{AG(T)}$ is isomorphic to G_T. Now it can be shown that

Theorem 5.1: For all $u, v \in \Sigma^*$, u and v are congruent modulo $AG(T)$ if and only if they are conjugate modulo T.

Proof: The 'if' part is trivial.

'Only if': let S abbreviate $AG(T)$. It is enough to show that if u and v are Parikh vectors (with $2n$ co-ordinates, since the inverses are considered explicitly in defining S) such that $u \longleftrightarrow_S v$, then there exists w in Σ^* such that $RED(uw) \longleftrightarrow_T RED(vw)$. For by induction it follows that if $u \overset{*}{\longleftrightarrow}_S v$ then there exists a string w in Σ^* such that $RED(uw) \overset{*}{\longleftrightarrow}_T RED(vw)$, which certainly implies the result when u and v are strings over Σ^* (note that if $x \in \Sigma^*$ then $RED(x) = x$). There are two cases:

(i) $u \longleftrightarrow_S v$ by application of the rule $aa^{-1} \longleftrightarrow \lambda$. Then we can take $w = \lambda$ since $RED(u) = RED(v)$.

(ii) $u \longleftrightarrow_S v$ by application of a rule $xy^{-1} \longleftrightarrow \lambda$, where for some p in Σ^* $px \longleftrightarrow py$ is a rule of T. Write $u = u_1 u_2^{-1} xy^{-1}$ and $v = u_1 u_2^{-1}$, where u_i are vectors in Σ^*. If $w = pu_2 y$, then $RED(wu) = pu_1 x$ and $RED(wv) = pu_1 y$, so $RED(wu) \longleftrightarrow_T RED(wv)$. **Q.E.D.**

Corollary 5.2: Let T be a commutative Thue system that is not cancellative. Then there exist u, v in Σ^* that are congruent modulo $AG(T)$ but not congruent modulo T.

Proof: Follows from the definition of cancellativity. ●

Definition: A semi-Thue system, i.e., a set of oriented rewrite rules over strings, is called *complete* (or *canonical*) if (i) it is noetherian, i.e., no string has infinitely many descendants, and (ii) it is confluent, so all descendants of the same string have a (unique) common descendant.

Proposition 5.3 [8]: For every commutative Thue system T there exists an equivalent finite canonical commutative semi-Thue system which can be effectively constructed from T. ●

Corollary 5.2 and Proposition 5.3 immediately yield the following criterion for a commutative Thue system not to be cancellative.

Theorem 5.4: Let T be a commutative Thue system that is not cancellative and let T' be an equivalent finite canonical semi-Thue system. Then there exist distinct strings $u, v \in IRR(T')$ which are congruent modulo $AG(T)$. ●

We now proceed to examine the word problem for abelian groups in an entirely different light – as a special case of the membership problem for polynomial ideals over **Z**. This has been developed in great detail in [6]. We merely elucidate some of the important points here.

Let $A = \{a_1, a_2, \cdots, a_n\}$ be the generators of an abelian group and let + stand for the group operator. Clearly, every term over the group can be viewed as a linear polynomial with no constant term over **Z** with the elements of A as indeterminates. If F is a finite presentation of an abelian group over the alphabet A, then F can be viewed as a set of equations of the form $c_1 * a_1 + \cdots + c_n * a_n = 0$ where the c_i's are integers. (The notation $c * a_i$, where c is an integer, denotes $a_i + \cdots + a_i$ ($|c|$ times) if c is positive and $-a_i + \cdots + -a_i$ ($|c|$ times) if c is negative.) In other words, each relation in F is a linear polynomial from $\mathbf{Z}[a_1, \cdots, a_n]$ with a constant term of 0. Let (F) stand for the ideal generated by F and $=_F$ denote the congruence generated by F. We can show

Theorem 5.5: Let e_1 and e_2 be two expressions and let p be the polynomial corresponding to $e_1 - e_2$. Then $e_1 =_F e_2$ if and only if $p \in (F)$.

We can also show a stronger version of the above theorem:

Theorem 5.6: Let e_1, e_2 and p be as above and let $F = \{p_1, \cdots, p_m\}$. Then $e_1 =_F e_2$ if and only if there exist integers $c_1, \ldots, c_m$ such that

$$p = \sum_{i=1}^{m} (c_i * p_i).$$

Sketch of Proof: The main point to be noted here is that the p_i's are linear integral polynomials with no constant terms. The theorem now follows from the proof of Claim 1 in the proof of Lemma 3 in [6]. ●

An immediate consequence of Theorem 5.6 is that the word problem for $AG(T)$ can be effectively reduced to linear programming and therefore is in **NP** (see [18]). It can now be shown that testing whether a canonical commutative Thue system T is not cancellative is in **NP**. (Thus cancellativity is in **co−NP**.) Let T be over an alphabet $A = \{ a_1, \cdots , a_n \}$. As before we treat words as Parikh vectors. For ease in exposition, let us establish the following (array-like) notation: for a word x, let $x[i]$ $(1 \le i \le n)$ denote the number of occurrences of the letter a_i in x. Let $T = \{ (L_i \longrightarrow R_i) \mid 1 \le i \le k \}$. We show how to non-deterministically find words u and v such that $u \ne v$, $u, v \in IRR(T)$ and u and v are congruent modulo $AG(T)$.

We treat the $u[i]$'s and $v[i]$'s as unknowns. Note that for two words to be distinct, it is enough that they differ in the number of occurrences of some letter. In other words, $u \ne v$ if and only if $u[j] \ne v[j]$ for some j such that $1 \le j \le n$. In our non-deterministic algorithm, we assume that j has been chosen correctly. Similarly, for a word w to be irreducible by a rule $(L_i \longrightarrow R_i)$, it is necessary and sufficient that $w[j] < L_i[j]$ for some j. Thus u is in $IRR(T)$ if and only if there exist integers $j_1, \cdots , j_k$ such that $u[j_i] < L_i[j_i]$ for all i, $1 \le i \le k$. Hence the j_i's are also choices that we have to make in our non-deterministic algorithm. The condition that u and v be congruent modulo $AG(T)$ can be expressed using a system of linear integral equations as shown in Theorem 5.6.

The outline of the non-deterministic algorithm should be clear by now: choose the $2k + 1$ integer values (each of which is bounded by n, the number of letters in the alphabet) which reflect the choice of inequalities showing that u and v are distinct and irreducible, form the system consisting of the appropriate inequalities and the equations denoting that u and v are congruent (modulo $AG(T)$) as in Theorem 5.6, and check whether this system has an integral solution. The last step mentioned, namely that of checking for solvability, is in **NP** because integer programming is in **NP**. (See [18].) Thus the entire algorithm can be performed in nondeterministic polynomial time. Summarizing:

Theorem 5.7. If T is a canonical commutative Thue system, the question of whether T is *not* cancellative is in **NP**. ●

In view of Proposition 5.3, the following corollary is immediate.

Theorem 5.8. If T is a commutative Thue system, the question of whether T is cancellative is decidable. ●

A last point to note is that the results in this section could be rephrased as follows: Cancellativity in commutative Thue systems is expressible as a formula in Presburger Arithmetic (see [4]) and hence is decidable.

6. Acknowledgements. We are grateful to Friedrich Otto for painstaking and valuable comments, and to Jacques Sakarovitch for some useful criticisms.

7. Bibliography

[1] Adjan, S.I., "Defining Relations and Algorithmic Problems for Groups and Semigroups," *Proc. Steklov Inst. Math.* 85, 1966 (English version published by the American Mathematical Society, 1967).

[2] Book, R., "Confluent and Other Types of Thue Systems," *J. Assoc. Comput. Mach.* 29, Jan. 1982, pp. 171-182.

[3] Book, R., "Decidable Questions of Church-Rosser Congruences," *Theoretical Computer Science* 24 (1983), 301-312.

[4] Fischer, M.J., and Rabin, M.O., "Super-exponential Complexity of Presburger Arithmetic," in R.M. Karp (ed.), *Complexity of Computation*, American Mathematical Society, Providence, RI, 27-41.

[5] Kandri-Rody, A., and Kapur, D. "Computing the Gröbner Basis of a Polynomial Ideal over Integers" In *Proc. Third MACSYMA Users' Conference*, Schenectady, NY, July 1984, pp. 436-451.

[6] Kandri-Rody, A., Kapur, D. and Narendran, P., "An Ideal-Theoretic Approach to Word Problems and Unification Problems over Finitely Presented Commutative Algebras," presented at the *First International Conference on Rewriting Techniques and Applications*, Dijon, France, May 1985. (Published in *Lecture Notes in Computer Science 202*, ed. J.-P. Jouannaud, Springer-Verlag, Berlin, pp. 345-364.)

[7] Kapur, D., and Narendran, P., "The Knuth-Bendix Completion Procedure and Thue Systems," *SIAM Journal on Computing*. **14:4** (November 1985), pp. 1052-1072.

[8] Lankford, D.S., and Ballantyne, A.M., "Decision Procedures for Simple Equational Theories with Commutative-Associative Axioms: Complete Sets of Commutative-

Associative Reductions,'' Automatic Theorem Proving Project, Dept. of Math. and Computer Science, University of Texas, Austin, TX 78712, Report ATP-39, August 1977.

[9] Lothaire, M., *Combinatorics on Words* (Vol. 17, Encyclopaedia of Mathematics and its Applications, G.-C. Rota, Ed.), Addison-Wesley, Reading, Massachusetts, 1983.

[10] Lyndon, R., and Schupp, P. *Combinatorial Group Theory*. Springer-Verlag (1977).

[11] Markov, A. Impossibility of algorithms for recognizing some properties of associative systems, *Dokl. Akad. Nauk SSSR* **77** (1951) 953-956.

[12] Mostowski, A. Review of Markov's paper, *J. Symbolic Logic* **17** (1952) 151-152.

[13] Narendran, P., "Church-Rosser and Related Thue Systems", Doctoral Dissertation, Rensselaer Polytechnic Institute, Troy, N.Y. 12181 (1984).

[14] Narendran, P., Ó'Dúnlaing, C. and Rolletschek, H., "Complexity of Certain Decision Problems about Congruential Languages," *Journal of Computer and System Sciences* **30:3** (June 1985), pp. 343-358.

[15] Nivat, M. (with M. Benois), "Congruences parfaites et quasi- parfaites", *Séminaire Dubreil*, 25e Année, 1971-1972 7-01-09.

[16] Ó'Dúnlaing, C. Undecidable questions related to Church-Rosser Thue systems. *Theoretical Computer Science* **23** (1983) 339-345.

[17] Otto, F., "Deciding Algebraic Properties of Monoids Presented by Finite Church-Rosser Thue Systems," presented at the *First International Conference on Rewriting Techniques and Applications*, Dijon, France, May 1985. (Published in *Lecture Notes in Computer Science 202*, ed. J.-P. Jouannaud, Springer-Verlag, Berlin, pp. 95-106.)

[18] Papadimitriou, C.H., and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.

This book may be kept

FOURTEEN DAYS

A fine will be charged for each day the book is kept overtime.

NYU COMPSCI TR-228
Narendran, Paliath
Cancellativity in finitely
presented semigroups c.1