Discrete Applied Mathematics 32 (1991) 211–222
North-Holland

Trace monoids with some invertible generators: two decision problems

C. Wrathall

Department of Mathematics, University of California, Santa Barbara, CA 93106, USA

Received 16 October 1988

Abstract

Wrathall, C., Trace monoids with some invertible generators: two decision problems, Discrete Applied Mathematics 32 (1991) 211–222.

Linear-time algorithms are given for the Word and Conjugacy Problems for trace monoids (or, free partially commutative monoids, commutation monoids) in which, in addition, some specified generators have inverses.

1. Introduction

Algebraic structures that extend free monoids by allowing some of the generators to commute have been studied extensively under such names as trace monoids, free partially commutative (or abelian) monoids and commutation monoids (see, e.g., [1, 2, 6, 10, 13]). One motivation for the recent interest is the possibility of using such structures for modeling aspects of parallelism or concurrency: a string may represent a sequence of operations, some pairs of which may be performed in parallel or in either order without changing the outcome.

The topic of this paper is a further extension in which not only do specified pairs of generators commute but also some specified set of generators have inverses. This extension is in keeping with the motivation noted above in that for some (but not necessarily all) operations, there may be another operation that reverses the effect: e.g., "increment (integer) register 1" could have the inverse operation "decrement register 1".

A structure $M(\theta_0, \Gamma_0)$ of this type can be determined by the set θ_0 of pairs of commuting or independent generators and the set Γ_0 of invertible generators. Those two sets give rise to a Thue system, or rewriting system on strings, with rules expressing the commutations and cancellations of letters. The Thue system, called

T_1, used here to present the monoid $M(\theta_0, \Gamma_0)$ completely expresses those relationships; this choice simplifies other calculations about $M(\theta_0, \Gamma_0)$ because T_1 has the preperfect property. The elements of $M(\theta_0, \Gamma_0)$ are then represented by congruence classes of strings under the Thue congruence generated by T_1.

For any rewriting system, an important question is the complexity of the Word Problem, that is, of deciding whether two objects are equivalent under the rules of the system. It is shown here that the Word Problem for any finitely-generated trace monoid with some invertible letters can be solved in linear time (even by a Turing machine): given two strings, it can be decided in time linear in the sum of their lengths whether they are congruent modulo T_1. The route taken by the algorithm is to first convert the strings to their "normal forms", which can be easily compared.

Another question of interest is the Conjugacy Problem. By analogy with groups and free monoids, elements m_1, m_2 of a monoid could be termed "conjugate" if $m_1 \cdot p = p \cdot m_2$ for some conjugator p—how difficult is it to decide whether two elements are conjugate? It is shown here that the Conjugacy Problem for any finitely-generated trace monoid with some invertible letters can also be solved in linear time.

The algorithms presented here are essentially the same as those that can be used when all the generators are invertible [15], and are based on the theory and algorithms that have been developed for trace monoids.

2. Rewriting systems and commutation monoids

The notation used here for strings and free monoids follows that of Lothaire [9], with the exception that e is used to denote the empty string. In particular, $|x|$ denotes the length of string x, and $|x|_a$ denotes the number of occurrences of the letter a in x.

A *rewriting system* is a set U and a binary relation $\Rightarrow$ on U called "reduction"; the reflexive-transitive closure of $\Rightarrow$ is denoted by $\overset{*}{\Rightarrow}$, and the equivalence relation it generates, by $\overset{*}{\Leftrightarrow}$. An element u_1 is *irreducible* if there is no element u_2 such that $u_1 \Rightarrow u_2$. The system is *Church-Rosser* if $u_1 \overset{*}{\Leftrightarrow} u_2$ implies the two elements have a common descendant, that is, an element u_3 such that $u_1 \overset{*}{\Rightarrow} u_3$ and $u_2 \overset{*}{\Rightarrow} u_3$.

A *Thue system* is a rewriting system on a free monoid Σ^* whose basis is a set T of pairs of strings. For such a system, write $x \leftrightarrow y$ if $x = rus$ and $y = rvs$ for some strings r, u, v, s, such that either $(u, v) \in T$ or $(v, u) \in T$. The Thue congruence determined by T is the reflexive-transitive closure of the one-step relation $\leftrightarrow$, and the monoid determined by T is the quotient of Σ^* by its Thue congruence. One rewriting step might or might not change the length of the string: when $x \leftrightarrow y$, write:

(i) $x \to y$, if $|x| > |y|$,
(ii) $x | \to y$, if $|x| \geq |y|$,
(iii) $x | - | y$, if $|x| = |y|$.

The reflexive-transitive closures of these relations are denoted by $\xrightarrow{*}$, $|\xrightarrow{*}$ and $|\overset{*}{-}|$, respectively. A string is *minimal* (with respect to T) if it is a shortest string in its congruence class.

A Thue system is *preperfect* if whenever strings u and v are congruent there is some string w such that $u|\xrightarrow{*}w$ and $v|\xrightarrow{*}w$. An equivalent "local" condition is that if $u \leftarrow x|\overset{*}{-}|y \rightarrow v$ then $u|\xrightarrow{*}w$ and $v|\xrightarrow{*}w$ for some w [11]. For a preperfect system, congruent strings can be "joined" by a mixture of length-reducing and length-preserving rules; if in fact all the reductions can be done first then the system is *almost confluent*: for congruent strings u and v, there is some string w such that $u\xrightarrow{*}|\overset{*}{-}|w$ and $v\xrightarrow{*}|\overset{*}{-}|w$.

A *commutation monoid* $M(\theta)$ is specified by an alphabet Σ and a symmetric and irreflexive "independence" (or "concurrency") relation $\theta \subseteq \Sigma \times \Sigma$. The independence relation determines a congruence relation on Σ^* by allowing pairs of independent letters to commute: if $(a,b)\in\theta$ then $xaby|-|xbay$ for all strings x, y. The monoid $M(\theta)=\Sigma^*/|\overset{*}{-}|$ is then the monoid determined by the Thue system $\{(ab,ba):(a,b)\in\theta\}$. Call strings x and y "independent" if $\mathrm{alph}(x)\times\mathrm{alph}(y)\subseteq\theta$; that is, if every letter in x is independent of every letter in y.

We may add structure to a commutation monoid by allowing some of the generating letters to have inverses. Suppose that Σ_0 is an alphabet, $\Gamma_0\subseteq\Sigma_0$ is the set of invertible letters, and θ_0 is an independence relation on Σ_0. These determine a certain monoid $M(\theta_0,\Gamma_0)$ in which cancellation rules apply to letters with inverses and commutation rules apply as specified by the independence relation.

Let $\Gamma_1=\{\bar{a}:a\in\Gamma_0\}$ be a set of formal inverses for the letters in Γ_0, and let $\Gamma=\Gamma_0\cup\Gamma_1$ and $\Sigma=\Sigma_0\cup\Gamma_1$. For $x=a_1\ldots a_n$, $a_i\in\Gamma$, x^{-1} denotes the string $\bar{a}_n\ldots\bar{a}_1$ (where $\bar{\bar{a}}=a$). Let $\theta\subseteq\Sigma\times\Sigma$ be the extension of θ_0 to Σ:

$$\begin{aligned}\theta=\theta_0 &\cup\{(\bar{a},b):(a,b)\in\theta_0,\ a\in\Gamma_0\}\\ &\cup\{(a,\bar{b}):(a,b)\in\theta_0,\ b\in\Gamma_0\}\\ &\cup\{(\bar{a},\bar{b}):(a,b)\in\theta_0,\ a,b\in\Gamma_0\}.\end{aligned}$$

Let T_1 be the following Thue system on Σ:

$$T_1=\{(a\bar{a},e),(\bar{a}a,e):a\in\Gamma_0\}\cup\{(cd,dc):(c,d)\in\theta\}.$$

Let $\equiv$ denote the congruence generated by the Thue system T_1, and $M(\theta_0,\Gamma_0)=\Sigma^*/\equiv$. Relative to T_1, $x\rightarrow y$ if y is formed from x by cancelling a pair of inverse letters, and $x|-|y$ if y is formed by interchanging a pair of adjacent independent letters in x.

The commutation monoid corresponding $M(\theta_0,\Gamma_0)$ is the monoid $M(\theta)=\Sigma^*/|\overset{*}{-}|$ presented by the Thue system $\{(cd,dc):(c,d)\in\theta\}$. When the alphabet Σ_0 is finite, also θ_0, Γ_0 and Σ will necessarily be finite, so both this Thue system and the system T_1 will be finite.

The system T_1 is redundant in the sense that some of the rules it contains can be

derived from others; however, in general some redundancy is necessary for the preperfect property (shown in Theorem 3.3) to hold.

Let $A_1, \ldots, A_N$ be a collection of subsets of Σ that cover Σ and have the following properties: for all $a, b \in \Sigma$

(i) when $a \in \Gamma_0$, for all i, $a \in A_i$ exactly when $\bar{a} \in A_i$;

(ii) if $(a, b) \notin \theta$, then, for some j, $a \in A_j$ and $b \in A_j$;

(iii) if there is some i such that $a, b \in A_i$, then $(a, b) \notin \theta$.

For each i, let $\pi_i : \Sigma^* \to A_i^*$ be the projection of Σ onto A_i, that is, the homomorphism determined by defining $\pi_i(a) = a$ for $a \in A_i$ and $\pi_i(a) = e$ for $a \notin A_i$. Note that if $\pi_i(x) = e$ whenever $a \in A_i$, then string x is independent of letter a.

Let $\Pi : \Sigma^* \to A_1^* \times \cdots \times A_N^*$ be the function defined by $\Pi(w) = (\pi_1(w), \ldots, \pi_N(w))$. Because $M(\theta)$ is a commutation monoid, we have the following correspondence between $M(\theta)$ and the product $A_1^* \times \cdots \times A_N^*$. Variations of this fact, for particular choices of the projection sets $A_1, \ldots, A_N$, have been proved by Cori and Perrin [4] and by Duboc [5].

Proposition 2.1. *For all $u, v \in \Sigma^*$, $u \mid \overset{*}{-} \mid v$ if and only if $\Pi(u) = \Pi(v)$.*

Two facts about commutation monoids that follow easily from Proposition 2.1 (or by induction on the number of "interchanging" steps) are that they are cancellative, and, for each letter $a \in \Sigma$, if $ar \mid \overset{*}{-} \mid s$, then there exist p and q such that $s = paq$, p is independent of a, and $r \mid \overset{*}{-} \mid pq$.

Although the statement of Proposition 2.1 implicitly assumes that the alphabet Σ is finite, the result also holds when the alphabet is countable, with the obvious extension of the collection of subsets A_i to a (possibly) countable collection.

3. Preperfectness, minimality and the Word Problem

Assume throughout this section that we are dealing with a fixed but arbitrary alphabet Σ_0, independence relation θ_0 on Σ_0 and subset Γ_0 of invertible generators, with the Thue system T_1 and monoid $M(\theta_0, \Gamma_0)$ as derived in Section 2. It is shown that the Thue system T_1 is preperfect (Theorem 3.3), and that when T_1 is finite, there is a linear-time algorithm for the problem of finding a minimal string congruent to a given string (Theorem 3.4). It easily follows that the Word Problem for the monoid $M(\theta_0, \Gamma_0)$ as presented by T_1 can be solved in linear time when the underlying alphabet Σ_0 is finite.

The results are obtained by means of the following reduction relation on tuples of strings that is suggested by Proposition 2.1; the reduction relation will be seen to faithfully represent T_1 and to be Church-Rosser (Theorem 3.2). While a direct proof that T_1 is preperfect is not difficult, the reduction relation is useful for the algorithms, which deal with strings w in their projected forms $\Pi(w)$.

Definition. For tuples $s,t \in A_1^* \times \cdots \times A_N^*$, *$s$ reduces to t* in one step, written $s \Rightarrow t$, if for some $a \in \Gamma$ and some $k \geq 0$, for every i,

(i) if $a \notin A_i$, then $t_i = s_i$; and

(ii) if $a \in A_i$, then $s_i = u_i a \bar{a} v_i$, $t_i = u_i v_i$ and $|u_i|_a + |u_i|_{\bar{a}} = k$.

Call a reduction using invertible letter a and constant k, "cancelling a at sum k".

The following proposition gives the basic connection between the reduction relation $\Rightarrow$ on tuples and the Thue system T_1.

Proposition 3.1.

(a) *For strings u, v the following are equivalent:*

(i) $\Pi(u) \Rightarrow \Pi(v)$;

(ii) $u \mid \overset{*}{-} \mid u_1 \to v_1 \mid \overset{*}{-} \mid v$ *for some strings* u_1, v_1; *and*

(iii) $u = xay\bar{a}z$ *and* $v \mid \overset{*}{-} \mid xyz$ *for some strings* x, y, z *such that* y *is independent of* a.

(b) *The reduction relation* $\Rightarrow$ *preserves* $\Pi(\Sigma^*)$; *that is, if* $\Pi(u) \Rightarrow t$ *then* $t \in \Pi(\Sigma^*)$.

Reduction of tuples may lead into $\Pi(\Sigma^*)$ from outside the class. For example, suppose $a \in \Gamma_0$, and $\{a, b\} \subseteq A_1$, $\{a, c\} \subseteq A_2$ and $\{b, c\} \subseteq A_3$. Then $(a\bar{a}b, ca\bar{a}, bc) \Rightarrow (b, c, bc) = \Pi(bc)$, but $(a\bar{a}b, ca\bar{a}, bc) \notin \Pi(\Sigma^*)$.

Proof of Proposition 3.1. If (iii) holds then, since y is independent of a, $u = xay\bar{a}z \mid \overset{*}{-} \mid xya\bar{a}z$, and so for (ii) we may take $u_1 = xya\bar{a}z$ and $v_1 = xyz$. If (ii) holds then $u_1 = xa\bar{a}y$ and $v_1 = xy$ for some x, y and a, and $\Pi(u) = \Pi(u_1)$ and $\Pi(v) = \Pi(v_1)$. If $a \notin A_i$ then $\pi_i(u) = \pi_i(xy) = \pi_i(v)$; and if $a \in A_i$, then $\pi_i(u) = \pi_i(x)a\bar{a}\pi_i(y)$ and $\pi_i(v) = \pi_i(x)\pi_i(y)$ with $|\pi_i(x)|_b = |x|_b$ for every letter b in A_i, so in particular $|\pi_i(x)|_a + |\pi_i(x)|_{\bar{a}}$ is the constant value $|x|_a + |x|_{\bar{a}}$. Thus, $\Pi(u) \Rightarrow \Pi(v)$ and (i) holds.

To see that (i) implies (iii), suppose $\Pi(u)$ reduces to a tuple t by cancelling $a \in \Gamma$ at sum k, as displayed in the definition above. The letter a belongs to some set A_i, say A_1, so $\pi_1(u) = u_1 a \bar{a} v_1$ with $|u_1|_a + |u_1|_{\bar{a}} = k$. From the definition of the projection homomorphism π_1, $u = xay\bar{a}z$ where $\pi_1(x) = u_1$, $\pi_1(y) = e$, and $\pi_1(z) = v_1$. Because π_1 does not erase a or $\bar{a}$, $|x|_a + |x|_{\bar{a}} = k$. Consider any other index i such that a belongs to A_i; then $\pi_i(u) = u_i a \bar{a} v_i = \pi_i(xay\bar{a}z) = \pi_i(x)a\pi_i(y)\bar{a}\pi_i(z)$. Since $|u_i|_a + |u_i|_{\bar{a}} = k = |\pi_i(x)|_a + |\pi_i(x)|_{\bar{a}}$, it must be the case that $u_i = \pi_i(x)$ and $\bar{a}v_i = \pi_i(y)\bar{a}\pi_i(z)$. Since $\pi_1(y) = e$, y and hence $\pi_i(y)$ can contain no occurrence of $\bar{a}$, so $\pi_i(y) = e$ and $v_i = \pi_i(z)$. Thus, $u = xay\bar{a}z$ with y independent of a (since $\pi_i(y) = e$ whenever $a \in A_i$). Also, for $a \in A_i$, $t_i = u_i v_i = \pi_i(x)\pi_i(z) = \pi_i(xyz)$ and for $a \notin A_i$, $t_i = \pi_i(u) = \pi_i(xyz)$; thus, $t = \Pi(xyz)$ and if $t = \Pi(v)$ then $v \mid \overset{*}{-} \mid xyz$. Notice that this argument also shows (b). □

It follows easily from Proposition 3.1 that T_1 is preperfect if and only if $\Rightarrow$ is confluent on $\Pi(\Sigma^*)$ in the sense that if $s \overset{*}{\Leftarrow} \Pi(u) \overset{*}{\Rightarrow} t$ then s and t have a common

descendant. The relation $\Rightarrow$ has the stronger property that it is Church-Rosser, that is, it is confluent on the whole class of tuples, not just those derived from $\Pi(\Sigma^*)$.

Theorem 3.2. (1) *The reduction relation* $\Rightarrow$ *is Church-Rosser.*
(2) *For all strings* x *and* y, $x \equiv y$ *if and only if* $\Pi(x) \overset{*}{\Leftrightarrow} \Pi(y)$.

Proof. It is shown below that $\Rightarrow$ is locally confluent; since it reduces the total length of a tuple, general principles allow us to conclude that it is Church-Rosser (see, e.g., [7]). For the second statement, a simple induction argument using Proposition 3.1 shows that $x \equiv y$ implies $\Pi(x) \overset{*}{\Leftrightarrow} \Pi(\upsilon)$. The obvious induction proof does not work for the reverse implication, since one-step reduction may lead into $\Pi(\Sigma^*)$ from a tuple not in $\Pi(\Sigma^*)$; however, if $\Pi(x) \overset{*}{\Leftrightarrow} \Pi(y)$ then (since $\Rightarrow$ is Church-Rosser) $\Pi(x)$ and $\Pi(y)$ have a common descendant, which will be of the form $\Pi(z)$, and so x and y will both be congruent to z, and $x \equiv y$.

To see that $\Rightarrow$ is locally confluent, suppose $x \Leftarrow z \Rightarrow y$ for some tuples x, y and z, where z reduces to x by cancelling a at sum k, and to y by cancelling b at sum j. Then either $x = y$ (if the deleted pairs overlap) or there is tuple w such that $x \Rightarrow w \Leftarrow y$.

From the definition, if $a \in A_i$ then $z_i = u_i a \bar{a} \upsilon_i$ and $x_i = u_i \upsilon_i$ with $|u_i|_a + |u_i|_{\bar{a}} = k$, and otherwise $z_i = x_i$; and if $b \in A_i$ then $z_i = r_i b \bar{b} s_i$ and $y_i = r_i s_i$ with $|r_i|_b + |r_i|_{\bar{b}} = j$, and otherwise $z_i = y_i$.

First suppose that a is neither b nor $\bar{b}$. There is a tuple w such that $x \Rightarrow w$ by cancelling b at sum j, and $y \Rightarrow w$ by cancelling a at sum k. The components of w can be defined as follows:

(1) If $a, b \notin A_i$, take $w_i = z_i = x_i = y_i$.
(2) If $a \in A_i$ and $b \notin A_i$, take $w_i = u_i \upsilon_i = x_i$.
(3) If $a \notin A_i$ and $b \in A_i$, take $w_i = r_i s_i = y_i$.
(4) If $a, b \in A_i$, then $z_i = u_i a \bar{a} \upsilon_i = r_i b \bar{b} s_i$, and either $u_i = r_i b \bar{b} t_i$ for some t_i, or $r_i = u_i a \bar{a} t_i$ for some t_i; in the first case, take $w_i = r_i t_i \upsilon_i$, and in the second, take $w_i = u_i t_i s_i$.

It is straightforward to verify that $x \Rightarrow w$ and $y \Rightarrow w$. The only point to note is that if $u_i = r_i b \bar{b} t_i$ then $|r_i t_i|_a + |r_i t_i|_{\bar{a}} = |u_i|_a + |u_i|_{\bar{a}} = k$; and similarly, if $r_i = u_i a \bar{a} t_i$ then $|u_i t_i|_b + |u_i t_i|_{\bar{b}} = |r_i|_b + |r_i|_{\bar{b}} = j$.

Now suppose that a is either b or $\bar{b}$, and, say, $a \in A_1$, so that $z_1 = u_1 a \bar{a} \upsilon_1 = r_1 b \bar{b} s_1$. The constraint of the number of a's and $\bar{a}$'s forces the deleted pairs to overlap in every component (in which they occur) in exactly the same way they overlap in z_1. By symmetry, we may assume that u_1 is no longer than r_1, giving rise to three cases.

Case 1. If $|u_1| = |r_1|$, then $u_1 = r_1$, $a = b$, $\upsilon_1 = s_1$ and $k = j$. When $a \notin A_i$, $x_i = y_i = z_i$. Consider any index i such that $a \in A_i$: $z_i = u_i a \bar{a} \upsilon_i = r_i a \bar{a} s_i$, $x_i = u_i \upsilon_i$, $y_i = r_i s_i$. Since $|u_i|_a + |u_i|_{\bar{a}} = k = j = |r_i|_a + |r_i|_{\bar{a}}$, it follows that $u_i = r_i$, so again $x_i = y_i$; hence $x = y$.

Case 2. If $|u_1| = |r_1| - 1$, then $a = \bar{b}$, $r_1 = u_1 a$, $v_1 = a s_1$ and $k = j - 1$. As in the previous case, $x = y$, since whenever $a \in A_i$, we have $z_i = u_i a \bar{a} v_i = r_i \bar{a} a s_i$ with $|u_i a|_a + |u_i a|_{\bar{a}} = k + 1 = j = |r_i|_a + |r_i|_{\bar{a}}$, and so $r_i = u_i a$.

Case 3. If $|u_1| \leq |r_1| - 2$, then $r_1 = u_1 a \bar{a} t_1$, $v_1 = t_1 b \bar{b} s_1$ and $j \geq k + 2$, and it will follow that x and y have a common one-step descendant w. If $a \notin A_i$, take $w_i = z_i = x_i = y_i$. If $a \in A_i$, then $z_i = u_i a \bar{a} v_i = r_i a \bar{a} s_i$ with $|u_i a \bar{a}|_a + |u_i a \bar{a}|_{\bar{a}} = k + 2 \leq j - |r_i|_a + |r_i|_{\bar{a}} < |r_i a|_a + |r_i a|_{\bar{a}}$, so $r_i = u_i a \bar{a} t_i$ and $v_i = t_i b \bar{b} s_i$ for some t_i; take $w_i = u_i t_i s_i$. Then $x \Rightarrow w$ by cancelling b at sum $j - 2$, and $y \Rightarrow w$ by cancelling a at sum k. □

A reduction relation on tuples that checks only the number of a's before the cancelled pair would be sufficient for Proposition 3.1 and Theorems 3.3 and 3.4. However, that reduction will not in general be Church–Rosser: it will only be Church–Rosser when no letter in Γ belongs to more than one set A_i, and in that case it is equal to the reduction relation used here.

The correspondence given in Proposition 3.1 and the information in the previous theorem allow us to conclude the following.

Theorem 3.3. *For an independence relation θ_0 on Σ_0 and set $\Gamma_0 \subseteq \Sigma_0$ of invertible letters, the Thue system*

$$T_1 = \{(a\bar{a}, e), (\bar{a}a, e) : a \in \Gamma_0\} \cup \{(cd, dc) : (c, d) \in \theta\}$$

on alphabet Σ is preperfect.

As a consequence of the preperfect property, if x and y represent the same element of $M(\theta_0, \Gamma_0)$ and x is minimal, then $y \mid \overset{*}{\rightarrow} x$. Also, if both x and y are minimal and $x \equiv y$, then $y \mid \overset{*}{-} \mid x$.

The Thue system T_1 is almost-confluent exactly when the independence relation and the invertible letters are disjoint, in the sense that if $(a, b) \in \theta_0$ then $a, b \notin \Gamma_0$. If they are disjoint, then whenever $x \mid \overset{*}{\rightarrow} y$, the derivation can be rearranged into the form $x \overset{*}{\rightarrow} z \mid \overset{*}{-} \mid y$ for some z; hence, since the system is preperfect, it is almost-confluent. On the other hand, if there is some pair $(a, b) \in \theta_0$ with (say) $a \in \Gamma_0$ then $ab\bar{a}$ is congruent to b but neither string is reducible, so the system is not almost-confluent.

The Thue system T_1 generates the same congruence on Σ^* as the "nonredundant" system

$$T_0 = \{(a\bar{a}, e), (\bar{a}a, e) : a \in \Gamma_0\} \cup \{(cd, dc) : c, d \in \Sigma_0, (c, d) \in \theta_0\}$$

based on the unextended independence relation θ_0. The system T_0 will not be preperfect unless θ_0 and Γ_0 are disjoint: if there is some $a \in \Gamma_0$ and pair $(a, b) \in \theta_0$, then $\bar{a}b \leftarrow \bar{a}ba\bar{a} \mid - \mid a\bar{a}ba\bar{a} \rightarrow b\bar{a}$ using the rules of T_0, but only length-increasing rules of T_0 apply to $\bar{a}b$ and $b\bar{a}$. However, if θ_0 and Γ_0 have no letter in common, then T_0 will be almost-confluent.

To this point, the results hold whether the alphabet Σ_0 (and hence the system T_1)

is finite or countable. Turning to questions of algorithms for T_1 and $M(\theta_0, \Gamma_0)$, a finiteness condition is necessary. First consider the question of finding a normal form for a given string relative to T_1, i.e., a representative for the string in $M(\theta_0, \Gamma_0)$.

Theorem 3.4. *For each monoid $M(\theta_0, \Gamma_0)$ with a finite set of generators Σ_0, there is a linear-time algorithm to find a minimal string congruent (modulo T_1) to a given string.*

Proof. The construction, given a string w, of a shortest string congruent to w is the same as that for a string in a free partially commutative group [14], but for completeness is briefly sketched here. The first step is to produce the "projected normal form" of w, that is, an irreducible tuple $R(w)$ such that $\Pi(w) \stackrel{*}{\Rightarrow} R(w)$, and the second is to reconstruct $R(w)$ into a minimal string congruent to w.

Each procedure uses N pushdown stores; the ith store "applies to a" if a belongs to A_i. For the reduction procedure the stores are initially empty, and it operates by reading w symbol-by-symbol, processing each as follows:

> If the next symbol a belongs to Γ and the top symbol of each store that applies to a is $\bar{a}$, then erase all those symbols $\bar{a}$; otherwise, print a on each store that applies to it.

After this processing is finished, the stores contain a tuple of words $R(w)$, from which a minimal string congruent to w can be printed right-to-left, as follows:

> Examine the pushdown stores to find a letter b with the property that b is on the top of every store that applies to it. Remove that occurrence of b from each such pushdown store, and print it as the next output letter. Continue until all the pushdown stores are emptied.

(When there is more than one candidate for the letter to be printed, any one can be chosen.)

By construction, $\Pi(w) \stackrel{*}{\Rightarrow} R(w)$ and $R(w)$ is irreducible by $\Rightarrow$, and (using Proposition 3.1) $R(w) = \Pi(w_0)$ for some string w_0. The reconstruction procedure will succeed in emptying the stores because at each step the tuple of their contents is in $\Pi(\Sigma^*)$, and it produces a string $r(w)$ such that $R(w) = \Pi(r(w))$; because $R(w)$ is irreducible, $r(w)$ is a shortest string in its congruence class. Then $\Pi(w) \stackrel{*}{\Rightarrow} \Pi(r(w))$, so that $w \mid \stackrel{*}{\rightarrow} r(w)$ and hence $r(w)$ is a minimal string congruent to w. □

To solve the Word Problem in $M(\theta_0, \Gamma_0)$, the algorithm in Theorem 3.4 can be used, but reconstruction is not necessary: $w_1 \equiv w_2$ if and only if their projected normal forms $R(w_1)$ and $R(w_2)$ are identical. Since those two tuples of strings can be constructed and compared in time linear in $|w_1| + |w_2|$, the Word Problem can be solved in linear time.

Corollary 3.5. *For each monoid $M(\theta_0, \Gamma_0)$ with finite set of generators Σ_0, there is a linear-time algorithm for the Word Problem relative to the Thue system T_1.*

4. The conjugacy problem

Assume as in Section 3 that some fixed commutation monoid with inverses $M(\theta_0, \Gamma_0)$ and associated Thue system T_1 are under consideration. This section deals with conjugacy in $M(\theta_0, \Gamma_0)$, and, in particular, with the Conjugacy Problem: decide whether two given strings are conjugate (modulo T_1). In an arbitrary monoid, conjugacy need not be a symmetric relation, and the Conjugacy Problem may be undecidable even for monoids presented by well-behaved rewriting systems [12]. However, for a monoid $M(\theta_0, \Gamma_0)$, conjugacy is symmetric, and it is closely linked to conjugacy in the corresponding commutation monoid $M(\theta)$. As in a free monoid, conjugacy in a finitely-generated commutation monoid can be tested in linear time by solving a related pattern-matching problem [8]; from the link between conjugacy in $M(\theta_0, \Gamma_0)$ and in $M(\theta)$, it will follow that when T_1 is finite, the Conjugacy Problem for $M(\theta_0, \Gamma_0)$ as presented by T_1 can be solved in linear time (Theorem 4.4).

Definition. Strings x and y are
(i) *conjugate in* $M(\theta_0, \Gamma_0)$ if there is some string w such that $xw \equiv wy$;
(ii) *conjugate in* $M(\theta)$ if there is some string z such that $xz \mid\!\overset{*}{-}\!\mid zy$.

These two conjugacy relations are clearly reflexive and transitive. Duboc [5] has shown that conjugacy in commutation monoids is symmetric, and hence it is an equivalence relation. As shown below (Corollary 4.3), conjugacy in $M(\theta_0, \Gamma_0)$ is also an equivalence relation.

The Conjugacy Problem for $M(\theta_0, \Gamma_0)$ will be reduced to testing instead whether two "cyclically minimal" strings that are derived from the given ones are conjugate in the monoid $M(\theta)$. The situation is analogous to that in a free group, where two elements are conjugate exactly when their cyclic reductions are conjugate in the underlying free monoid.

Definition. A string x is *cyclically minimal* if x^2 is minimal (relative to T_1).

A cyclically minimal string is minimal and cannot be rewritten to begin with a letter and end with the inverse of that letter; this is analogous to cyclically reduced elements of a free group. Each element of $M(\theta_0, \Gamma_0)$ has a central cyclically minimal part, which may be termed its "core".

Definition. For a string $x \in \Sigma^*$, let z be any minimal string congruent to x (modulo T_1), and let $u \in \Gamma^*$, $y \in \Sigma^*$ be any strings such that $z \mid\!\overset{*}{-}\!\mid uyu^{-1}$ and y is cyclically minimal. Then y is a *core* of x.

Although different strings might be obtained by rewriting x under this definition, the notion is well defined in the sense that (as shown by the following lemma) the possible cores form a single element of the monoid $M(\theta)$.

Lemma 4.1. *If $uyu^{-1} \equiv vzv^{-1}$, both strings are minimal, and y and z are cyclically minimal, then $y \mid \overset{*}{-} \mid z$.*

Proof. Since uyu^{-1} and vzv^{-1} are minimal and T_1 is preperfect, $uyu^{-1} \mid \overset{*}{-} \mid vzv^{-1}$. If $u = e$, then $y \mid \overset{*}{-} \mid vzv^{-1}$, so, since y is cyclically minimal, $v = e$ and $y \mid \overset{*}{-} \mid z$. Continuing by induction on $|u|$, suppose $u = aw$, so that $awyw^{-1}\bar{a} \mid \overset{*}{-} \mid vzv^{-1}$. If $|v|_a = 0$ then (using the property noted after Proposition 2.1) v is independent of a and there is some z_1 such that $z \mid \overset{*}{-} \mid az_1\bar{a}$, contradicting the assumption that z is cyclically minimal. Therefore, $v \mid \overset{*}{-} \mid at$ for some string t, and $wyw^{-1} \mid \overset{*}{-} \mid tzt^{-1}$. Both wyw^{-1} and tzt^{-1} are minimal (since they are substrings of minimal strings) and w is shorter than u, so $y \mid \overset{*}{-} \mid z$. □

From the following lemma, we see that cyclically minimal strings that are conjugate in $M(\theta_0, \Gamma_0)$ must be conjugate in $M(\theta)$.

Lemma 4.2. *If x and y are cyclically minimal and $xz \equiv zy$, then there exist u, v such that $xu \mid \overset{*}{-} \mid uy$ and $vx \mid \overset{*}{-} \mid yv$.*

Proof. The proof is by induction on the length of the conjugator z, which we may assume to be a minimal string. If xz and zy are minimal (and, in particular, if $z = e$) then $xz \mid \overset{*}{-} \mid zy$, so we may take $u = z$; and, since conjugacy is an equivalence relation for $M(\theta)$, there is some v such that $vx \mid \overset{*}{-} \mid yv$.

Suppose then that xz is not minimal. Since both x and z are minimal and T_1 is preperfect, there are strings x_1, z_1 and letter $a \in \Gamma$ such that $x \mid \overset{*}{-} \mid x_1a$ and $z \mid \overset{*}{-} \mid \bar{a}z_1$. Then $xz \mid \overset{*}{-} \mid x_1a\bar{a}z_1 \to x_1z_1$ so $(ax_1)z_1 \equiv axz \equiv azy \equiv a\bar{a}z_1y \equiv z_1y$. The string ax_1 is cyclically minimal because x is cyclically minimal and $x \mid \overset{*}{-} \mid ax_1$, so there exist u_1, v_1 such that $(ax_1)u_1 \mid \overset{*}{-} \mid u_1y$ and $v_1(ax_1) \mid \overset{*}{-} \mid yv_1$. Let $u = x_1u_1$ and $v = v_1a$; then $xu \mid \overset{*}{-} \mid x_1ax_1u_1 \mid \overset{*}{-} \mid x_1u_1y = uy$ and $vx \mid \overset{*}{-} \mid v_1ax_1a \mid \overset{*}{-} \mid yv_1a = yv$, as desired. □

Corollary 4.3. *Conjugacy in $M(\theta_0, \Gamma_0)$ is an equivalence relation.*

Proof. It need only be argued that conjugacy in $M(\theta_0, \Gamma_0)$ is symmetric, so suppose $xw \equiv wy$. Let x_1 be the core of x and let $u \in \Gamma^*$ be a string such that $x \equiv ux_1u^{-1}$ with ux_1u^{-1} minimal; similarly, let y_1 be the core of y where $y \equiv vy_1v^{-1}$. Then x_1 and y_1 are conjugate in $M(\theta_0, \Gamma_0)$ (and hence in $M(\theta)$): one conjugator is $u^{-1}wv$ since $x_1u^{-1}wv \equiv u^{-1}xwv \equiv u^{-1}wyv \equiv u^{-1}wvy_1$. From Lemma 4.2, there is some z such that $zx_1 \mid \overset{*}{-} \mid y_1z$. Let $t = vzu^{-1}$; then $yt \equiv vy_1v^{-1}vzu^{-1} \equiv vy_1zu^{-1} \equiv vzx_1u^{-1} \equiv vzu^{-1}x = tx$. □

Theorem 4.4. *For each monoid $M(\theta_0, \Gamma_0)$ with finite set of generators Σ_0, the Conjugacy Problem (relative to the Thue system T_1) can be solved in linear time.*

Proof. As in the proof of Corollary 4.3, two strings are conjugate in $M(\theta_0, \Gamma_0)$ exactly when their cores are conjugate in $M(\theta)$.

The core of a string can be found in linear time by extending the procedure to find a minimal string congruent to a given string. For $x \in \Sigma^*$, let

- $F(x) = \{a \in \Sigma$: for some y, $x \mid \overset{*}{-} \mid ay\}$,
- $L(x) = \{a \in \Sigma$: for some y, $x \mid \overset{*}{-} \mid ya\}$.

It is easy to see that x is cyclically minimal exactly when it is minimal and there is no letter $a \in \Gamma$ such that $a \in F(x)$ and $\bar{a} \in L(x)$.

Given $w \in \Sigma^*$, the core of w can be found as follows. First, as in Theorem 3.3, form the tuple of strings $R(w) = (x_1, \ldots, x_N) = \Pi(x)$ where x is a minimal string congruent to w. With the strings $x_1, \ldots, x_N$ written on separate tapes, position a head at each end of each string. The sets $F(x)$ and $L(x)$ are evident from the letters under the heads, since letter a belongs to $F(x)$ if and only if it is the first letter of $\pi_i(x) = x_i$ for each i such that a belongs to A_i (and analogously for $L(x)$). If there is no letter $a \in F(x)$ such that $\bar{a}$ belongs to $L(x)$, then x is cyclically minimal and is the core of w. If there is such a letter a, then move both the heads on inward one letter on each store that applies to a; in effect, this replaces $\Pi(x)$ with $\Pi(y)$ where $x \mid \overset{*}{-} \mid ay\bar{a}$. This process can be repeated until a cyclically minimal string (in projected form) is obtained.

To test whether two strings are conjugate in $M(\theta_0, \Gamma_0)$, therefore, it is only necessary to compute their cores and test whether the cores are conjugate in $M(\theta)$. As described above, the cores can be computed in linear time, and conjugacy in a commutation monoid can be tested in linear time [8]; hence conjugacy in $M(\theta_0, \Gamma_0)$ can be tested in linear time. □

References

[1] IJ. Aalbersberg and G. Rozenberg, Theory of traces, Theoret. Comput. Sci. 60 (1988) 1–82.

[2] A. Bertoni, G. Mauri and N. Sabadini, Equivalence and membership problems for regular trace languages, in: Lecture Notes in Computer Science 140 (Springer, Berlin, 1984) 115–133.

[3] R. Book and H.-N. Liu, Rewriting systems on a free partially commutative monoid, Inform. Process. Lett. 26 (1987) 29–32.

[4] R. Cori and D. Perrin, Automates et commutations partielles, RAIRO Informat. Théor. Appl. 19 (1985) 21–32.

[5] C. Duboc, On some equations in free partially commutative monoids, Theoret. Comput. Sci. 46 (1986) 159–174.

[6] M. Flé and G. Roucairol, Maximal serializability of iterated transactions, Theoret. Comput. Sci. 38 (1985) 1–16.

[7] G. Huet, Confluent reductions: abstract properties and applications to term rewriting systems, J. ACM 27 (1980) 797–821.

[8] H.-N. Liu, C. Wrathall and K. Zeger, Efficient solution of some problems in free partially commutative monoids, Inform. and Comput. 89 (1990) 180–198.

[9] M. Lothaire, Combinatorics on Words (Addison-Wesley, Reading, MA, 1983).
[10] A. Mazurkiewicz, Traces, histories and graphs, in: Proceedings MFCS '84, Lecture Notes in Computer Science 176 (Springer, Berlin, 1984) 115–133.
[11] P. Narendran and R. McNaughton, The undecidability of the preperfectness of Thue systems, Theoret. Comput. Sci. 31 (1984) 165–174.
[12] P. Narendran and F. Otto, The problems of cyclic equality and conjugacy for finite complete rewriting systems, Theoret. Comput. Sci. 47 (1986) 27–38.
[13] D. Perrin, Words over a partially commutative alphabet, in: A. Apostolico and Z. Galil, eds., Combinatorial Algorithms on Words (Springer, Berlin, 1985) 329–340.
[14] C. Wrathall, The word problem for free partially commutative groups, J. Symbolic Comput. 6 (1988) 99–104.
[15] C. Wrathall, Free partially commutative groups, in: D.-Z. Du and G. Hu, eds., Combinatorics, Complexity, and Computing (Kluwer Academic Publishers, Dordrecht, 1989) 195–216.