Research report 100

OBSERVATIONS ON THE DISJOINTNESS PROBLEM FOR RATIONAL SUBSETS OF FREE PARTIALLY COMMUTATIVE MONOIDS

Alan Gibbons & Wojciech Rytter*

(RR100)

Abstract

Let I be a partially commutative alphabet of size three. Let M denote the free partially commutative monoid generated by I. The disjointness problem for rational subsets of M is:

for two given rational (described by regular expressions) subsets, X, Y of M decide if $X \cap Y = \emptyset$.

In this paper we show that the problem is decidable for every commutativity region over the alphabet I. It is known (see (3)) that the problem is undecidable in the case of the four letters alphabet. Hence we give a sharp bound on the number of letters for which the problem is decidable. A similar situation occurs for the unique decipherability problem with partially commutative alphabets. It was shown in (4) that this problem is decidable for alphabets of size three and that it is undecidable for alphabets of size four. We show that the unique decipherability problem with partially commutative alphabet I is a special case of the disjointness problem of rational subsets of the monoid generated by I. This and our algorithm for the disjointness problem give alternative and much simpler proof of the decidability of the unique decipherability problem with partially commutative alphabets of size three. Let $I=\{a,b,c\}$. It was proved in (5) using multicounter machines that if a commutes with c and b, and b does not commute with c then the disjointness problem is decidable. We give here a simpler proof for this case and prove the decidability for all other possible commutativity relations for three letters alphabet.

* on leave from Institute of Informatics
Warsaw University

Department of Computer Science
University of Warwick
Coventry
CV4 7AL, UK

May 1987

OBSERVATIONS ON THE DISJOINTNESS PROBLEM FOR RATIONAL SUBSETS OF FREE PARTIALLY COMMUTATIVE MONOIDS

Alan Gibbons and Wojciech Rytter #

Dept. of Computer Science, University of Warwick, Coventry CV4 7AL, UK.

(All correspondence to the first author)

Let I be a partially commutative alphabet of size three. Let M denote the free partially commutative monoid generated by I. The disjointness problem for rational subsets of M is:

for two given rational (described by regular expressions) subsets X, Y of M decide if $X \cap Y = \emptyset$.

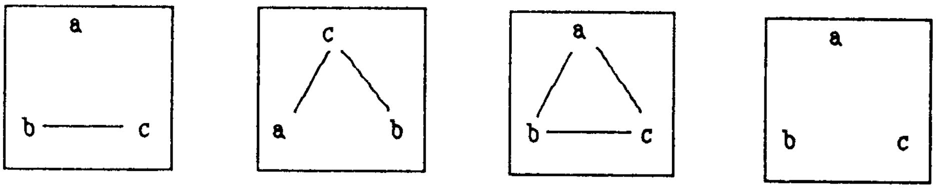
In this paper we show that the problem is decidable for every commutativity relation over the alphabet I. It is known (see [3]) that the problem is undecidable in the case of the four letters alphabet. Hence we give a sharp bound on the number of letters for which the problem is decidable. A similar situation occurs for the unique decipherability problem with partially commutative alphabets. It was shown in [4] that this problem is decidable for alphabets of size three and that it is undecidable for alphabets of size four. We show that the unique decipherability problem with partially commutative alphabet I is a special case of the disjointness problem of rational subsets of the monoid generated by I. This and our algorithm for the disjointness problem give alternative and much simpler proof of the decidability of the unique decipherability problem with partially commutative alphabets of size three. Let $I=\{a,b,c\}$. It was proved in [5] using multicounter machines that if a commutes with c and b, and b does not commute with c then the disjointness problem is decidable. We give here a simpler proof for this case and prove the decidability for all other possible commutativity relations for three letters alphabet.

Keywords: rational subsets, partially commutative monoids, disjointness, decidability

---

On leave from Institute of Informatics, Warsaw University, Poland

Languages over partially commutative alphabets are generalizations of classical formal languages and the same classical decision problems can be considered for such languages. Many classical decision problems for rational subsets over alphabets of size 3 were shown in [5] to be undecidable (using the results of [7]). Suprisingly the disjointness problem turns out to be decidable.
A partially commutative alphabet (called also a concurrent alphabet) is a pair (I, C), where I is a finite set of symbols and C is a symmetric irreflexive relation on I.
The symbols of I can represent processes (see [9,8]) and the relation C then represents which of these processes can be executed independently (C is also called the concurrency relation).
Two strings v and w are said to be equivalent (with respect to C) if v can be obtained from w by several applications of the operation of commuting certain two adjacent symbols a,b such that $(a,b) \in C$. We write in this case $v \approx_C w$ (later we shall omit the subscript C).
In this paper we consider only alphabets of size three. We fix the alphabet I={a,b,c}. There are (up to isomorphism) four possible commutativity relations C: R1, R2, R3 or R4. They are presented in Fig.1 as undirected graphs, with an edge between two letters if they commute.



R1 R2 R3 R4

Fig.1. Possible types of commutativity relations over I={a,b,c}.

For example if C=R1 then bcacbabbc ≈ cbabcabcb.
The free partially commutative monoid (fpcm, for short) over I is the set M of equivalence classes of the relation $\approx_C$. These equivalence classes were called traces in [2,10,8,9] and subsets of a fpcm M were called trace languages in [2]. Let [x] denote the equivalence class containing x.
Classical formal languages are languages over alphabets in which no symbols commute. Any classical language L over the alphabet I has a corresponding trace language, by taking all traces containing at least one element of L. In this sense rational subsets of M (rational trace languages) correspond to classical regular languages L.

The subset L' of M is a rational set iff L'={[x]: x is in L} for some (classical) regular language L over the alphabet I. It is technically simpler to deal with sets of strings instead of sets X, Y of equivalence classes of strings. Hence instead of considering subsets of fpcm (trace languages) we consider in this paper their classical language versions. To this end we introduce the operation CL.

Let L be a classical language over the alphabet I, by CL(L) we denote the set

$$\{w : w \approx v \text{ for some } v \in L\}.$$

CL is called the closure operation.
The disjointness problem can be formulated now as follows:
for two given regular languages L1, L2 (described by regular expressions) decide whether

$CL(L1) \cap CL(L2) = \emptyset$.
In the case of relations R1 and R2 we reduce this problem to the emptiness problem for context-free languages (which is decidable).

Throughout the paper $I=\{a,b,c\}$. Let ε denote the empty word.

Let h be the homomorphism: $h(a)=a$, $h(b)=b$, $h(c)=\varepsilon$.

Lemma 1.
If the commutativity relation C is R1 then for every two regular languages L1, L2 over I the language $L = h(CL(L1) \cap CL(L2))$ is a context free language. If the regular expressions describing L1, L2 are given then the context-free grammar generating L can be effectively constructed.

Proof.
Let A1 (A2) be the finite automaton accepting L1 (L2). Let Q1 (Q2) be the set of states of A1 (A2). We construct a nondeterministic counter automaton A accepting L. The counter can take arbitrary integer value, it is initially zero and in one step it can decrease or increase by one. The set of states of A is $Q1 \times Q2$. The initial state of A is (the initial state of A1, the initial state of A2) and accepting states of A are pairs of the form (an accepting state of A1, an accepting state of A2). We can imagine A as a composition of A1 and A2. Whenever A reads the symbol a or b then A1 and A2 go to the next states and their next states form a pair which is a next state of A.

At any time A can nondeterministically assume in an ε-move the input symbol c for one of the machines A1, A2 (without advancing its input head and disregarding the input text), then the simulated machine makes a move as if it was reading the symbol c. If the machine A1 is chosen then the counter is incremented by one, if it is A2 then the counter is decremented by one. The machine A accepts iff there is a computation which ends in an accepting state and during the computation the counter is empty whenever we read the symbol a. The counter should also equal zero at the end of the computation.

The input to A is of the form bb..babb...bab...bab... . The machine is guessing two sequences of c's to be interleaved with the symbols b between two consecutive occurences of the symbol a. One sequence goes to the machine A1 and the other to the machine A2. Using the counter A checks whether the lengths of these sequences (between two consecutive symbols a) are the same.

In other words for a given string w A guesses two strings w1, w2 from $h^{-1}(w)$, such that w1 is in L1 and w2 is in L2 and $w1 \approx w2$.

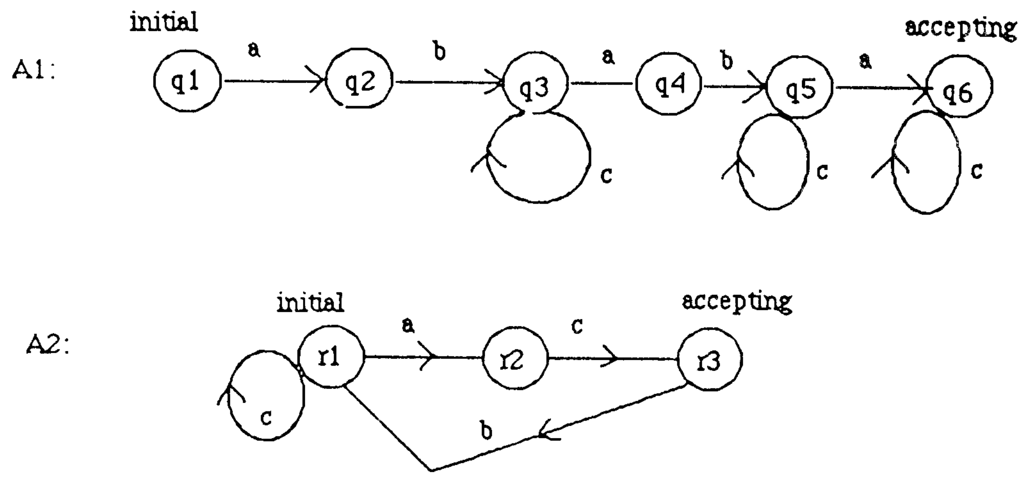


Fig.2. The automata A1, A2 accepting the languages L1, L2.

For example let A1, A2 be as as in Fig.2, and L1, L2 be the languages accepted by, respectively, A1, A2. $L1 \cap L2 = \emptyset$, however $CL(L1) \cap CL(L2) \neq \emptyset$. The automaton A (the composition of A1 and A2) accepts the word w=ababa $\in$ h($CL(L1) \cap CL(L2)$). A can guess a sequence of three symbols c to be interleaved with w in two ways resulting in words w1=abcabcac and w2=acbacbac. w1$\in$L1 and w2$\in$L2, moreover $w1 \approx w2$.

The power of A lies in its nondeterminism. We show an accepting computation path of A for the input word w. A starts in the state (q1,r1) with counter equal zero. A reads the symbol a and goes to the state (q2,r2). Then A guesses the symbol c for the automaton A2, (the state of A2 is changed to q3), A goes to the state (r2,q3) and increments the counter by one. Next A reads the symbol b and goes to the state (r3,q1). Then A guesses one symbol c for the automaton A1, A goes to the state (r3,q1) and decreases the counter by one. Now A reads the symbol a and goes to the state (q4,r2), however it checks first whether the counter is zero. The counter is zero, hence A continues its computation. Finally A arrives at the accepting state (q6,r3) with the counter equal zero after reading the whole word w. The automaton A accepts w.

The counter automaton is a special case of the pushdown automaton. The absolute value of the counter can be represented by the number of symbols on the stack, and the sign of the counter can be stored in the finite memory. The context free grammar generated the language accepted by A can be constructed using the description of the automaton A. Then the problem is reduced to checking the emptiness of a context free language. This completes the proof.

Remark

It can be proved, using a more complicated construction, that $L=h(CL(L1)\cap CL(L2))$ is a regular language, where L1,L2 are as in the lemma.

The next lemma concerns the case when the commutativity relation is R3. If all letters commute then the cardinality of the alphabet does not matter, we can also speak about context free languages instead of regular languages. This is so because in this case for each context free language L there is a regular language L' such that CL(L')=CL(L). The sets CL(L) are nicely characterized by semilinear sets. We refer the reader to [6] for the definition and properties of semilinear sets.

Lemma 2.
If all the letters commute then the disjointness problem for rational subsets is decidable for any finite alphabet.

Proof.
Let L1, L2 be two regular languages over the same finite alphabet. Assume that all the letters commute. Then CL(L1) and CL(L2) are semilinear sets, see [6]. It follows now from Theorem 5.6.1 in [6] that the set $CL(L1)\cap CL(L2)$ is also a semilinear set (effectively constructed) and the disjointness prblem for semilinear sets is decidable. This completes the proof.

The proof of decidability given in [5] for the commutativity relation R2 was using the results about reversal-bounded multicounter machines. We provide here a much simpler proof similar to the proof of Lemma 1 by reducing the problem to the emptiness problem for context free languages.

Lemma 3.
If the commutativity relation C is R2 then for every two regular languages L1, L2 over I the language $L=h1(CL(L1)\cap CL(L2)$ is a context free language. If the regular expressions describing L1, L2 are given then the context-free grammar generating L can be effectively constructed.

Proof.
We proceed ij a similar manner to the proof of Lemma 1. Let A1 (A2) be the finite automaton accepting L1 (L2). Let Q1 (Q2) be the set of states of A1 (A2). We construct a nondeterministic counter automaton A accepting L.

The set of states of A is $Q1\times Q2$. The initial state of A is the pair (the initial state of A1, the initial state of A2).
Accepting states of A are pairs of the form (an accepting state of A1, an accepting state of A2). A is a composition of A1 and A2.

Whenever A reads the symbol a or b then A1 and A2 go to the next states and their next states form a pair which is a next state of A.

At any time A can nondeterministically assume in an ε-move the input symbol c for one of the machines A1, A2 (without advancing its input head and disregarding the input text), then the simulated machine makes a move as if it was reading the symbol c. If the machine A1 is chosen then the counter is incremented by one, if it is A2 then the counter is decremented by one. The machine A accepts iff there is a computation which ends in an accepting state and the final value of the counter is zero. Using the desciption of A a corresponding context-free grammar can be effectively constructed. This completes the proof.

Theorem 4.
The disjointness problem for rational subsets over partially commutative alphabets of size three is decidable.

Proof.
The thesis follows from Lemmas 1,2 and 3 and the decidability of the emptiness problem for context free languages.

We show that the unique decipherability problem is a special case of the disjointness problem of rational subsets.
Let $Q = \{w_1,...,w_k\}$ be a set of nonempty words over I. The words w_i are called codewords and W - a code-set. We say that W is uniquely decipherable over I iff the equality

$$w_{i1}w_{i2} \cdots w_{ip} \approx w_{j1}w_{j2} \cdots w_{jq}$$

implies that $(i_1,i_2,...,i_p) = (j_1,j_2,...,j_q)$.
The unique decipherability problem is the following: Is W uniquely decipherable over I?

The generalized unique decipherability problem was introduced in [4]. Let W be a code set and $Z = (u_0,v_0,u_1,v_1)$ be a four-tuple of words over I. We say that (W,Z) is not uniquely decipherable over I if there exist sequences $(i_1,i_2,...,i_p)$, $(j_1,j_2,...,j_q)$ such that

$u_0w_{i1}w_{i2} \cdots w_{ip}u_1 \approx v_0w_{j1}w_{j2} \cdots w_{jq}v_1$.

The generalized unique decipherability problem is:
decide if (W,Z) is uniquely decipherable .

The unique decipherability problem for a code-set W can be easily reduced to the finite set of generalized unique decipharability problems (W,Z), where Z are all four-tuples (u_0,v_0,u_1,v_1) of code-words from W such that $u_0 \neq v_0$. Then the problem can be seen as a special case of the disjointness problem of rational subsets, because of the following obvious fact.

Lemma 5.

Let $Z=(u_0,v_0,u_1,v_1)$ and let L_1,L_2 be the following rational subsets over the alphabet I:
$L_1= u_0W^*u_1$, $L_2= v_0W^*v_0$.

Then (W,Z) is not uniquely decipherable if and only if $L_1 \cap L_2 \neq \emptyset$.

Hence the decidability of the unique decipherability problem with partially commutative three letter alphabet follows directly from our theorem about disjointness problem.
In fact the decidability results presented in [3] are much stronger and are related to the structure of the graph corresponding to the commutativity relation (all graphs with three nodes have good structure in this sense). However their proofs are more complicated than the proofs presented here. The shortness of the proof of Theorem 4.2 in [4] can be misleading, because it is incorrect (it can be corrected but it needs a more complicated argument). If one wants to prove the decidability of the unique decipharability problem restricted only to partially commutative alphabets of size three then the proof via theorem 4 and lemma 5 is simpler than via graph theoretic constructions in [4].

The following theorem (showing that the size three of the alphabet is sharp) was proved in [3]:

Theorem 6.

The disjointness problem for rational subsets over partially commutative alphabets of size four is undecidable.

The alternative proof follows from Lemma 5 and the fact that the unique decipherability problem for partially commutative alphabets of size four is undecidable [4].

Let NL denote the class of problems solvable in nondeterministic logarithmic space. It is easy to prove that the disjointness problem for regular languages (over noncommutative alphabets) is in NL. It was shown in [12] that the unique decipherability problem (over noncommutative alphabet) is log-space complete in NL. Hence the disjointness problem for regular languages is also complete in NL, because the unique decipherability problem is. This implies the following natural question: is the disjointness problem for rational subsets over three letter partially commutative alphabets also log-space complete in NL? In fact it remains only to show that it is in NL.

Acknowledgment

The proof of Lemma 3 was suggested to us by M.S.Paterson.

References.

[1] J. Berstel. Transductions and context-free languages. Teubner (1979)

[2] A.Bertoni,G.Mauri,N.Sabadini. Equivalence and membership problem for regular trace languages. in ICALP 82, Lect.Notes in Comp.Science 140,(1982),pp.61-71

[3] C.Choffrut. Free partially commutative monoids. Techn.Report, Laboratoire Informatique Theorique et Programmation 86.20, March (1986)

[4] M.Chrobak,W.Rytter. The unique decipherability problem with partially commutative alphabet. in the proceedings of Math.Found.of Comp.Science, Lect.Notes in Comp.Science 233 (1986) 256-263

[5] A.Gibbons,W.Rytter. On the decidability of some problems about rational subsets of free partially commutative monoids. To appear in Theor.Computer Science (1987)

[6] S.Ginsburg.Mathematical theory of context free languages. New York,McGraw-Hill Book Co. (1966)

[7] O.H.Ibarra. Reversal-bounded multicounter machines and their decision problems. Journal of ACM 25:1 (1978) pp.106-133

[8] O.H.Ibarra. The unsolvability of the equivalence problem for e-free ngsm's with unary input (output) alphabet and applications. SIAM Journal on Computing 7:4 (1978) pp.524-532

[9] A.Mazurkiewicz.Concurrent program schemes and their interpretations,DAIMI PB 78,Aarhus University (1977)

[10] A.Mazurkiewicz. Traces, histories, graphs: instances of processes monoid. Lecture Notes in Comp.Science 176

[11] W.Rytter. Some properties of trace languages. Fundamenta Informaticae VII.1 (1984) pp.107-127

[12] W.Rytter. The space complexity of the unique decipherability problem. Inf.Proc.Letters 23 (1986) pp.1-3

[13] M.Szijarto. A classification and closure properties of languages for describing concurrent systems behaviours. Fundamenta Informaticae IV.3 (1981) pp.531-550

[14] A.Tarlecki. Notes on the implementability of formal languages by concurrent systems. ICS PAS Reports 481 (1982) Warsaw