INFORMATIQUE THÉORIQUE ET APPLICATIONS

KAREL CULIK II
JUHANI KARHUMÄKI

Loops in automata and HDTOL relations

Informatique théorique et applications, tome 24, n° 4 (1990), p. 327-337

<http://www.numdam.org/item?id=ITA_1990__24_4_327_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 24, n° 4, 1990, p. 327 à 337)

LOOPS IN AUTOMATA AND HDTOL RELATIONS (*)

by Karel CULIK II (1) and Juhani KARHUMÄKI (2)

Communicated by J. E. PIN

Abstract. – *We show that n-tape automata containing only simple loops*, i. e. *no state is involved in two loops, have several properties which general n-tape automata, or even automata with parallel loops only, do not have. In particular, the intersection of relations defined by two simple n-tape automata is so called HDTOL relation. This implies several old and new decidability results for simple n-tape automata.*

Résumé. – *Nous montrons que les automates à n bandes contenant seulement des boucles simples, c'est-à-dire pour lesquels aucun état n'appartient à deux boucles, possèdent différentes propriétés que les automates généraux à n bandes ou même les automates possèdant seulement des boucles parallèles ne partagent pas. En particulier l'intersection de relations définies par deux automates à n bandes ayant seulement des boucles simples est ce que l'on appelle une relation HDTOL. Ceci implique plusieurs résultats de décidabilité anciens et nouveaux concernant les automates simples à n bandes.*

1. INTRODUCTION

It is well known, *see* [13], that the loop structure of an automaton or of a program can have an enormous effect on the complexity of the bahavior of the automaton or the program. The purpose of this note is to emphasize this phenomenon.

We shall be dealing with n-tape finite automata introduced in [14]. We call such an automaton simple if all of its nodes are involved in at most one loop, and show that for two simple automata the intersection of the relations defined by them can be described effectively by an HDTOL language, *cf.* [15], or more precisely the intersection is an HDTOL relation. This certainly does

(*) Receveid May 1988, revised in March 1989.
Research supported by the National Science Foundation Grant, No. CCR-8702752.
(1) Department of Computer Science, University of South Carolina, Columbia, S. C. 29208.
(2) Department of Mathematics, University of Turku, Finland.

Informatique théorique et Applications/Theoretical Informatics and Applications
0988-3754/90/04 327 11/$3.10/

not hold for n-tape automata in general. Indeed, even for automata with only parallel, but not nested loops the emptiness of the intersection is undecidable.

An important notion for this paper is that of an HDTOL relation, introduced implicitly in [4] and later used in [6]. A relation $R \subseteq \Sigma^* \times \Delta^*$ is an HDTOL relation iff it can be expressed in the form $R = \{(h(x), g(x)) \mid x \in L\}$ for some HDTOL language L and two morphisms h and g. For our current purposes we generalize this notion to n dimensions. This notion is strongly motivated by Nivat Theorem, *cf.* [1], characterizing rational relations.

From the point of view of decision problems HDTOL relations became a particularly powerful tool when it was shown that morphisms can be tested on HDTOL languages, *cf.* [5]. An example of the use of this tool is shown here. As an application of our "intersection theorem" we generalize the known result that the equivalence of two simple deterministic n-tape automata is decidable, *cf.* [13] or [11], to the following: It is decidable whether two n-tape finite transducers based on simple deterministic n-tape automata are equivalent on their common domain. The proof uses an idea of "HDTOL matching" of computations of two n-tape automata developed further in [7].

As pointed out by the referee an alternate way to prove our results is to use the theory of the SLB family of languages, *cf.* [9] and [12]. We, however, want to emphasize here our techniques based na HDTOL matching of the equivalent computations.

2. PRELIMINARIES

We assume that the reader is familiar with basics of automata theory and the theory of semilinear sets of vectors, *cf.* [10] and [8], respectively. In addition we list [15] as the reference dealing with HDTOL languages.

An *n-tape automaton* over disjoint alphabets Σ_i, for $i = 1, \ldots, n$, is a quadruple

$$\mathscr{A} = (Q, T, q_0, F), \tag{1}$$

where Q is a finite set of states, $T \subseteq Q \times \bigcup_{i=1}^{n} (\Sigma_i \cup \#_i) \times Q$ is a finite transition relation, q_0 is the initial state, and $F \subseteq Q$ is the set of final states. (Here $\#_i$ denotes the endmarker of a word in Σ_i^*.) The automaton $\mathscr{A}$ accepts an n-tuple $(w_1, \ldots, w_n)$ of words in $\Sigma_1^* \times \ldots \times \Sigma_n^*$ iff $\mathscr{A}$ as a one-tape automaton accepts at least one word from the set $w_1 \#_1 \sqcup w_2 \#_2 \sqcup \ldots \sqcup w_n \#_n$ where

denotes the shuffle operation. The set of all n-tuples accepted by $\mathscr{A}$ is denoted by $R(\mathscr{A})$. An n-tuple automaton $\mathscr{A}$ of (1) is *deterministic* iff Q and T can be partitioned as follows:

$$Q=\bigcup_{i=1}^{n} Q_i \qquad \text{with} \quad Q_i \cap Q_j=\varnothing \quad \text{for } i\neq j \tag{2}$$

and

$$T=\bigcup_{i=1}^{n} T_i, \qquad \text{where} \quad T_i=T\cap Q_i\times(\Sigma_i\cup\{\#_i\})\times Q, \tag{3}$$

and $\operatorname{card}(T_i\cap\{q\}\times\{a\}\times Q)\leqq 1$ for $i=1, \ldots, n$, $q\in T_i$ and $a\in\Sigma_i\cup\{\#_i\}$. Conditions (2) and (3) mean that the state of $\mathscr{A}$ uniquely determines on which tape automaton can next read. In addition, if not only the current state but also the symbol to be read is fixed, then the next state is uniquely determined. It follows that if $\mathscr{A}$ is deterministic, then for each n-tuple $(w_1, \ldots, w_n)\in\Sigma_1^*\times\Sigma_2^*\times\ldots\times\Sigma_n^*$ there exists at most one accepting computation of $\mathscr{A}$. It also follows that the requirement that Σ_i's are disjoint becomes superfluous in the case of deterministic n-tape automata.

An n-tape automaton $\mathscr{A}$ is *simple* if each of its states is involved in at most one loop of $\mathscr{A}$ (viewed as a one-tape automaton). Obviously, final states of a simple automaton can be specified as states having no exists. A *simple branch* is a simple n-state automaton having only one final state from which there are no transitions. We say that an automaton contains *only parallel* loops if in any loop there exists at most one state which is involved in another loop.

We state an easy fact:

LEMMA 1: *Each relation defined by a simple n-tape automaton is a finite union of relations defined by simple branches.*

By associating outputs to transitions of an n-tape automaton we can talk about *n-tape transducers* (of certain type). Here the outputs are over a single tape, but this is not an important restriction.

Next we define our central notion of an HDTOL relations, *cf.* [4] and [6]. An *HDTOL language* L over Σ is a language of the form

$$L=g\left(\bigcup_{i=1}^{\infty} L_i\right)$$

where

$$L_0 = \{w\}, \qquad L_{i+1} = \bigcup_{j=1}^{t} h_j(L_i) \quad (i \geqq 1)$$

for a word $w \in \Delta^*$, morphisms $h_j : \Delta^* \to \Delta^*$ $(j = 1, \ldots, t)$, $g : \Delta^* \to \Sigma^*$ and an integer $t \geqq 1$. Let $\Sigma_1, \ldots, \Sigma_n$ be finite (not necassarily disjoint) alphabets. A relation $\rho \subseteq \Sigma_1^* \times \ldots \times \Sigma_n^*$ is called an *HDTOL relation* iff there exists an alphabet Γ, an HDTOL language L over Γ and morphisms $g_i : \Gamma^* \to \Sigma_i^*$ such that

$$\rho = \{(g_1(w), \ldots, g_n(w)) \mid w \in L\}. \tag{4}$$

The definition of an HDTOL relations is motivated by Nivat representation of rational relations, *cf.* [1]. They also provide a generalization of rational relations since. as it is easy to see, each regular language is an HDTOL language. HDTOL relations have a couple of interesting properties which we believe make them important and useful family of relations. First of all, they are purely morphically defined. Secondly, the underlying family of languages has desirable properties, such as that the emptiness is decidable and that morphisms can be tested on these languages, *cf.* [15] and [6], respectively.

Finally, we recall the definition of a semilinear set. Let $\mathbb{N}$ denote the set of nonnegative integers. A set K is *linear* iff it is a subset of $\mathbb{N}^k$, for some $k \geqq 1$, and of the form

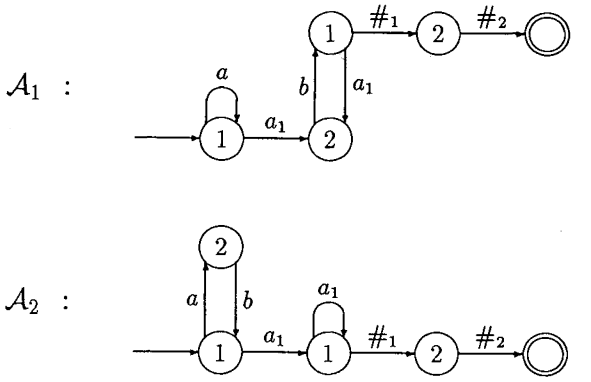
$$K = \left\{ x_0 + \sum_{i=1}^{N} n_i x_i \mid n_i \geqq 0 \right\} \quad \text{for} \quad N \geqq 0 \quad \text{and} \quad x_0, x_1, \ldots, x_N \text{ in } \mathbb{N}^k.$$

Further, K is *semilinear* iff it is a finite union of linear sets.

3. RESULTS

In this section we consider simple n-tape automata and show that they possess some desirable properties general n-tape automata do not have. First we recall that the intersection of two relations defined by deterministic n-tape automata need not to be realized by such an automaton. This is seen,

for example, by considering the following automata:



Here a and a_1 are read from the first tape and b from the second. Clearly,

$$R = R(\mathcal{A}_1) \cap R(\mathcal{A}_2) = \{ (a^n a_1^n, b^n) \mid n \geqq 1 \}$$

and hence it cannot be accepted by a 2-tape automaton.

In the above example the automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are simple, and still the intersection is not a rational relation. However, we are going to show that such an intersection is always an HDTOL relation.

THEOREM 1: *For two simple n-tape automata $\mathcal{A}_1$ and $\mathcal{A}_2$ the intersection $R(\mathcal{A}_1) \cap R(\mathcal{A}_2)$ is an HDTOL relation. Moreover, its HDTOL representation can effectively be found.*

Proof: Since HDTOL languages are closed under union so are HDTOL relations. Consequently, by Lemma 1, we can assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are simple branches. Hence their loops are linearly ordered, say from 1 to k_1 in $\mathcal{A}_1$ and from 1 to k_2 in $\mathcal{A}_2$. Assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are over common alphabets Σ_i, $i = 1, \ldots, n$.

For an n-tuple $w = (w_1, \ldots, w_n) \in \Sigma_1^* \times \ldots \times \Sigma_n^*$ accepted by $\mathcal{A}_i$ let $\text{comp}_i(w)$ denote the set of computations of w in $\mathcal{A}_i$. Now, since $\mathcal{A}_i$'s are simple branches each computation $x \in \text{comp}(\mathcal{A}_i)$ is *uniquely determined* by a vector in $\mathbb{N}^{k_i}$ telling how many times each loop in $\mathcal{A}_i$ is performed when computing x. Let us denote this vector by $\psi_i(x)$. We define a subset of

$\mathbb{N}^{k_1+k_2}$ by

$$N(\mathscr{A}_1, \mathscr{A}_2) = \{(\psi_1(x_1), \psi_2(x_2)) \mid x_i \in \mathrm{comp}_i(w) \text{ for some } w \in R(\mathscr{A}_1) \cap R(\mathscr{A}_2)\} \quad (5)$$

and claim that $N(\mathscr{A}_1, \mathscr{A}_2)$ is a semilinear set.

In order to prove our claim we define "projections" of $\mathscr{A}_i$'s as follows. For $i=1, 2$ and $j=1, \ldots, n$ let $\mathscr{A}_i(j)$ denote a one-tape automaton obtained from $\mathscr{A}_i$ by replacing all transitions labeled by letters in

$$\bigcup_{\substack{m=1 \\ m \neq j}}^{k_i} (\Sigma_m \cup \{\#_m\})$$

by transitions labeled by the empty word 1. Consequently, the automata $\mathscr{A}_i(j)$ have the same loop structure as $\mathscr{A}_i$, but since they allow to read the empty word they might be nondeterministic even if $\mathscr{A}_i$ were not. Obviously, we can modify the above notation like comp_i and ψ_i for automata $\mathscr{A}_i(j)$ as well, let us denote these by $\mathrm{comp}_{i,j}$ and $\psi_{i,j}$. Then since automata $\mathscr{A}_i(j)$ are ordinary one-tape automata (and only simple branches as such) it is straightforward to see, for example by induction on the number of loops, that, for $j=1, \ldots, n$, the sets

$$N(\mathscr{A}_1(j), \mathscr{A}_2(j)) = \{(\psi_{1,j}(x_1), \psi_{2,j}(x_2)) \mid x_i \in \mathrm{comp}_{i,j}(w) \text{ for some } w \text{ in } L(\mathscr{A}_1(j)) \cap L(\mathscr{A}_2(j))\},$$

are semilinear subsets of $\mathbb{N}^{k_1+k_2}$. Thus the semilinearity of $N(\mathscr{A}_1, \mathscr{A}_2)$ follows from the identity

$$N(\mathscr{A}_1, \mathscr{A}_2) = \bigcap_{j=1}^{n} N(\mathscr{A}_1(j), \mathscr{A}_2(j))$$

and from the fact that semilinear sets are closed under intersection.

Now, we are ready to construct an HDTOL representation for $R(\mathscr{A}_1) \cap R(\mathscr{A}_2)$. First of all recall that the computations of $\mathscr{A}_i$, for $i=1, 2$, are uniquely determined by vectors in $\mathbb{N}^{k_i}$, respectively. Secondly, by construction of $N(\mathscr{A}_1, \mathscr{A}_2)$ we can conclude the following. If $z \in N(\mathscr{A}_1, \mathscr{A}_2)$ and z_1 (resp. z_2) is a vector in $\mathbb{N}^{k_1}$ (resp. $\mathbb{N}^{k_2}$) obtained from k_1 first (resp. k_2 last) components of z, then (z_1 and z_2) describe computations of the same word in $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively. Conversely, if $Z_1 \in \mathbb{N}^{k_1}$ and $Z_2 \in \mathbb{N}^{k_2}$ describe computations of the same word in $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively, then Z_1, Z_2 is

a vector in $N(\mathscr{A}_1, \mathscr{A}_2)$. Intuitively, elements in $N(\mathscr{A}_1, \mathscr{A}_2)$ tell simultaneously how computations of fixed n-tuples can be carried out in both of the automata $\mathscr{A}_1$ and $\mathscr{A}_2$, and thus gives a guidance which computations have to be matched.

As we noted HDTOL relations are closed under union. Hence we may assume that $N(\mathscr{A}_1, \mathscr{A}_2)$ is a linear subset of $\mathbb{N}^{k_1+k_2}$, say

$$N(\mathscr{A}_1, \mathscr{A}_2) = \{ x_0 + \sum_{t=1}^{N} n_t x_t \mid n_t \geqq 0 \}$$

for some vectors $x_0, x_1, \ldots, x_N \in \mathbb{N}^{k_1+k_2}$. Let $\pi_j : \mathbb{N}^{k_1+k_2} \to \mathbb{N}$ be the j-th projection. Since automata $\mathscr{A}_i$, for $i=1, 2$, are simple branches we can describe them by words

$$\alpha_i(0)\,\beta_i(1)\,\alpha_i(1)\ldots\beta_i(k_i)\,\alpha_i(k_i), \tag{6}$$

where each $\beta_i(r)$ is the label of n-th loop in $\mathscr{A}_i$ and $\alpha_i(r)$ is the label of that path of the computation in $\mathscr{A}_i$ which is in between $(r-1)$st and r-th loops. Now an HDTOL system G is defined as follows: S is the start symbol and it has tables:

$$T_0: \quad S \to \alpha_1(0)\,\beta_1(1)^{\pi_1(x_0)} X_1\,\alpha_2(1)\,\beta_1(2)^{\pi_2(x_0)} X_2 \ldots \beta_1(k_1)^{\pi_{k_1}(x_0)} X_{k_0}\,\alpha_1(k_1)$$

$$\#\,\bar{\alpha}_2(0)\,\bar{\beta}_2(1)^{\pi_{k_1+1}(x_0)} X_1 \ldots \bar{\beta}_2(k_1+k_2)^{\pi_{k_1+k_2}(x_0)} X_{k_1+k_2}\,\bar{\alpha}_2(k_1+k_2),$$

$$T_{N+1}: \quad \begin{cases} X_j \to 1 & \text{for } j=1, \ldots, k_1+k_2, \\ \# \to 1, \end{cases}$$

$$T_s: \quad \begin{cases} X_j \to \beta_1(j)^{\pi_j(x_j)} X_j & \text{for } j=1, \ldots, k_1, \\ X_j \to \bar{\beta}_2(j)^{\pi_j(x_j)} X_j & \text{for } j=k_1+1, \ldots, k_1+k_2, \end{cases}$$

for $s=1, \ldots, N$. If productions in a table are not specified for some letters, then they are assumed to be identities. Here bars mean that the corresponding words are over the barred copy of the original alphabet. Let $L(G)$ be the language generated by G, and $L = L(G) \cap (\Sigma \cup \bar{\Sigma})^*$. By the closure properties of HDTOL languages L is an HDTOL language. Now, it follows from the construction that if we define for $i=1, \ldots, n$ morphisms $h_i : (\Sigma \cup \bar{\Sigma})^* \to \Sigma_i^*$ by

$$h_i(a) = a \qquad \text{for } a \in \Sigma_i,$$

$$h_i(a) = 1 \qquad \text{otherwise,}$$

then

$$R(\mathscr{A}_1) \cap R(\mathscr{A}_2) = \{(h_1(w), \ldots, h_n(w)) \mid w \in L\}.$$

Hence $R(\mathscr{A}_1) \cap R(\mathscr{A}_2)$ is an HDTOL relation. The second sentence of the theorem follows from the proof. □

Actually, the barred copies of letters in construction of G are not needed in the proof of Theorem 1. They are introduced in order to obtain easily one of the corollaries of the theorem. However, the use of HDTOL systems is essential also for the proof of Theorem 1, since even if G would generate only computations of $\mathscr{A}_1$ the parallel applications of loops is necessary to select only those computations which have a corresponding one in $\mathscr{A}_2$.

Theorem 1 and its proof techniques provide a few interesting corollaries. First we obtain new proofs for the equivalence and inclusion problems for simple not necessarily deterministic n-tape automata, *cf.* [13] or [11].

COROLLARY 1: *The equivalence and inclusion problems for simple n-tape automata are decidable.*

Proof: Clearly, it suffices to consider only the inclusion problem. Let $\mathscr{A}_1$ and $\mathscr{A}_2$ be simple n-tape automata. Then, by Lemma 1, there exist simple branches $\beta_i(j)$, $i=1, 2$ and $j=1, \ldots, k_i$, such that

$$R(\mathscr{A}_i) = \bigcup_{j=1}^{k_i} R(\beta_i(j)).$$

Hence, to test the inclusion

$$R(\mathscr{A}_i) \subseteq R(\mathscr{A}_2)$$

it is enough to test the inclusions

$$R(\beta_1(j)) \subseteq R(\mathscr{A}_2) \qquad \text{for} \quad j=1, \ldots, k_1. \tag{7}$$

So let us consider (7) for a fixed j, say $j=1$. Clearly, (7) holds for $j=1$ iff

$$R(\beta_1(1)) = \bigcup_{j=1}^{k_2} (R(\beta_1(1)) \cap R(\beta_2(j))).$$

But, by the proof of Theorem 1, we can find, for each $j=1, \ldots, k_2$, a semilinear set describing all those computations of $\beta_1(1)$, that compute n-tuples which are also in $R(\beta_2(j))$. Hence, testing (7) is reduced to testing whether a union of semilinear set is universal, *i. e.* contains all vectors of the

appropriate dimension. This is a well known decidable property of semilinear sets. □

We note that the equivalence problem for all 2-tape deterministic automata is decidable, *cf.* [2], while the inclusion problem is undecidable, *cf.* [1].

Our second Corollary is restricted to deterministic automata. However, the result seems to be new, and shows the power of our techniques.

COROLLARY 2: *It is decidable whether two simple deterministic n-tape transducers are equivalent on their common domain.*

Proof: Now we need the whole power of the proof of Theorem 1. In fact, we have to extend the construction of G slightly. Namely, instead of generating labels of computations, *cf.* (6), we can as well generate transitions, *i. e.* real computations. So it follows that we can construct an HDTOL language L which matches the corresponding computations of the underlying automata $\mathscr{A}_1$ and $\mathscr{A}_2$. That is to say a word of the form $w \# \bar{w}$ is in L iff w and $\bar{w}$ are computations of the same n-tuple in $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively. Since $\mathscr{A}_1$ and $\mathscr{A}_2$ are assumed to be deterministic, the $\bar{w}$ above is unique for w, and vice versa. Consequently, to test the equivalence of the transducers it is enough to test two morphisms h and g defined below on HDTOL language L, which is shown decidable in [6]. In order to define h and g let E_i, for $i=1, 2$, denote the sets of transitions of the transducers and let Δ be their common output alphabet. Of course, we may assume that $E_1 \cap E_2 = \varnothing$. Then $h, g : (E_1 \cup E_2)^* \to \Delta^*$ are defined by

$$h(t) = o(t) \quad \text{for} \quad t \in E_1,$$

$$h(t) = 1 \quad \text{otherwise,}$$

and

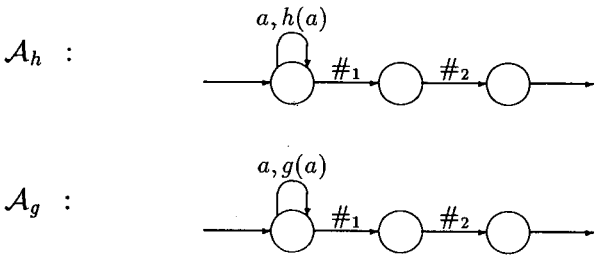
$$g(t) = o(t) \quad \text{for} \quad t \in E_2,$$

$$g(t) = 1 \quad \text{otherwise,}$$

where $o(t)$ denotes the output associated to a transition $t \in E_1 \cup E_2$. □

We conclude this section with a remark showing that Theorem 1 cannot be extended to arbitrary n-tape automata. In fact, our next example shows

that it cannot be extended even to the case when parallel, but not nested loops are allowed. Let $h, g: \Sigma^* \to \Delta^*$ be morphisms. We define 2-tape automata



where edges labeled by a, $h(a)$ and a, $g(a)$ exist for all $a \in \Sigma$. Then, clearly

$$I = R(\mathcal{A}(h)) \cap R(\mathcal{A}(g)) = \{(x, h(x)) \mid x \in E(h, g)\}$$

where $E(h, g)$ is the equality language of (h, g), *i. e.*

$$E(h, g) = \{x \in \Sigma^* \mid h(x) = g(x)\}.$$

It is well known that there exists a nonrecursive equality language, *cf.* [3], so that not only is the representation of Theorem 1 impossible for I, but also the emptiness of I is undecidable.

4. CONCLUDING REMARKS

We have demonstrated that the simple n-tape automata are considerably easier to handle than the general, or even the parallel loop n-tape automata. In particular, we have shown that although the intersection of relations of two simple n-tape automata need not be rational it has a similar Nivat representation: Instead of regular languages we have to use HDTOL languages which constitute a generalization of regular languages. In our terminology the above intersections are HDTOL relations. It follows from our proof techniques and from the decidability properties of HDTOL languages that we can reprove several decidability results for simple n-tape automata, and establish some new ones as well.

We have also pointed out that our main result (Theorem 1) cannot be generalized to cover all n-tape automata, or even n-tape automata with parallel loops only. This supports the view that parallel loop n-tape automata are already much more complicated than simple ones. For example, their

equivalence problem, which is – according to our knowledge – open, seems to be much more difficult than that of simple n-tape automata.

REFERENCES

[1] J. BERSTEL, *Transductions and Context-Free Languages*, Teubner, Stuttgart, 1979.

[2] M. BIRD, *The Equivalence Problem for Deterministic two-tape Automata*, J. Comput. Sci., Vol. 7, 1973, pp. 218-236.

[3] K. CULIK II, *A Purely Homomorphic Characterization of Recursively Enumerate Sets*, J. Assoc. Comput. Mach., Vol. 6, 1979, pp. 345-350.

[4] K. CULIK II and J. KARHUMÄKI, *Systems of equations over a free Monoid and Ehrenfeucht's Conjecture, Discrete Math.*, Vol. 43, 1983, pp. 139-153.

[5] K. CULIK II and J. KARHUMÄKI, *The Equivalence of Finite Valued Ttansducers (on HDTOL Languages) is Decidable*, Theoret. Comput. Sci., Vol. 47, 1986, pp. 71-84.

[6] K. CULIK II and J. KARHUMÄKI, *Systems of Equations Over a Finitely Generated free Monoid Having an Effectively Equivalent Finite Subsystem*, Lecture Notes in Mathematics (to appear).

[7] K. CULIK II and J. KARHUMÄKI, *HDTOL Matching of Computations of Multiple Automata*, Acta Informatica (to appear).

[8] S. GINSBURG, *The Mathematical Theory of Context-Free Languages*, McGraw Hill, New York, 1966.

[9] S. GINSBURG and E. H. SPANIER, *AFL with the Semilinear Property*, J of Comput. and System Sciences, Vol. 5, 1971, pp. 365-396.

[10] M. HARRISON, *Introduction to Formal Language Theory*, Addison-Wesley, Reading, 1978.

[11] E. KINBER, *The Inclusion Problem for Some Classes of Deterministic Multitape Automata*, Theoret. Comput. Sci., Vol. 26, 1983, pp. 1-24.

[12] M. LATTEAUX, *Intersections de languages algébriques bornés*, Acta Informatica, Vol. 11, 1979, pp. 233-240.

[13] H. R. LEWIS, *A New Decidability Problem with Applications*, Proceedings of 18th FOCS Conference, 1979, pp. 62-73.

[14] M. RABIN and D. SCOTT, *Finite Automata and their Decision Problems*, I.B.M. J. Res. Develop., Vol. 3, 1959, pp. 114-125.

[15] G. ROZENBERG and A. SALOMAA, *The Mathematical Theory of L Systems*, Academic Press, New York, 1980.