ON A FAMILY OF L LANGUAGES
RESULTING FROM SYSTOLIC TREE AUTOMATA[*]

K. Culik II, J. Gruska[1], A. Salomaa[2]

Research Report CS-81-36

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

January 1982

[1]   On a leave of absence from the Computer Research Centre in Bratislava,
      Czechoslovakia

[2]   On a leave of absence from the University of Turku, Finland

A B S T R A C T

Recent work on systolic tree automata has given rise to a rather natural subfamily of EOL languages, referred to as systolic EOL languages in this paper. Systolic EOL languages possess some remarkable properties. While their family contains (because of its closure under Boolean operations) intuitively quite complicated languages, it still has decidable equivalence problem. Especially interesting is the fact that similar decision problems for slightly more general families lead to the celebrated open problems concerning Z-rational power series.

1.   Introduction

Various types of systolic automata were recently introduced,
[2] - [4], basically as a model for VLSI. However, it was also observed
that these new models gave rise to a number of new problems and problem
areas which are interesting also from the point of view of classical
language theory.

The purpose of this paper is to study a family of L languages,
[6], arising quite naturally from the consideration of systolic tree auto-
mata, [2], [4]. More specifically, this family is a subfamily of the
family of EOL languages, [6]. In this paper we refer to the languages in
this subfamily as <u>systolic</u> EOL languages. The family of systolic EOL
languages is large enough to contain all regular languages and, in addition,
also quite complicated languages. Because the family is closed under
Boolean operations, we obtain a subfamily of EOL languages such that the
complements of the languages in this subfamily are still EOL. (Observe
that none of the most common L families is closed under complementation.)
On the other hand, the family of systolic EOL languages is small enough to
possess a number of important decidability properties.

The paper is divided as follows. After a discussion on pre-
liminaries in Section 2, the above mentioned facts about systolic EOL
languages, as well as some modifications, will be discussed in Section 3.
We also obtain a natural characterization of systolic EOL languages.
Finally, Section 4 deals with the problem of whether or not the decid-
ability results can be extended to concern a somewhat larger class.

While the problem itself remains open, we obtain the surprising result that if indeed the decidability can be established for the larger family then this solves also two celebrated open problems from the theory of formal power series, [7]: The decidability of the existence of 0 and of the existence of negative numbers in a given Z-rational sequence. Because no reduction results are known for the latter two problems, this shows that the decidability of the problem considered in Section 4 might be even harder to establish.

While this paper is largely self-contained, we refer the reader to [6] (resp. [7]) for definitions and facts concerning L systems (resp. formal power series). Reference [2] may be consulted for more detailed definitions and background material as regards systolic tree automata.

2. Preliminaries

A binary systolic tree automaton works basically as follows. Consider an infinite binary tree without leaves. We may define the "levels" of the tree in the natural way. Consider then an input word w over Σ with length t . We choose the first level in the tree with $n \geq t$ vertices. The word $w \#^{n-t}$ (where # is a special symbol not in Σ) is now "fed", letter by letter, to the level in question. This is formalized as follows. We consider also another alphabet Σ_0 (referred to as the operating alphabet) and a function

$$g : \Sigma \cup \{\#\} \rightarrow \Sigma_0 .$$

The nodes in the level in question are labeled (in the correct order!) with the g-values of the letters in the word $w \#^{n-t}$.

Information now flows bottom-up and in parallel. We consider also another function $h : \Sigma_0^2 \rightarrow \Sigma_0$. If the sons of a node have already been assigned (from left to right) the values a and b , their father gets the value $h(a,b)$. The word w is accepted by our binary systolic tree automaton (in short, BT-VLSI) if the root of the tree gets in this way a value from Σ_0' , where Σ_0' is a designated subset of Σ_0 . Thus, we may speak of the accepted language, as well as of BT-VLSI acceptable languages.

Formally, a BT-VLSI is determined by the three alphabets Σ, Σ_0 and Σ_0' , and the two functions g and h. The reader is referred to [2] for further details and examples.

A (general) <u>systolic tree automaton</u> (in short, T-VLSI) differs from the binary one in that, instead of the infinite binary tree, we consider an arbitrary infinite tree without leaves, having only finitely many infinite subtrees. (The latter condition assures that the tree is reasonably regular. Of course, the tree is given in some effective manner.) The definition of acceptance remains the same as before except that now, instead of one binary function h , we have several functions h_i where i stands for the arity (i.e., number of sons) of the node in question.

In fact, the definition of T-VLSI given in [2] is somewhat more general: the tree is labeled (still preserving the regularity condition of finitely many labeled subtrees), and the functions h_i and g depend also on the label of the node in question. However, according to a normal form result established in [2], every language acceptable by this more general model is also acceptable by the restricted model considered above. This holds true for BT-VLSI's as well.

The T-VLSI's and BT-VLSI's considered above are <u>deterministic</u>. Formally this means that the range of the functions g and h is Σ_0. The corresponding <u>nondeterministic</u> automata are defined in the natural way. Formally, in connection with g and h we consider subsets of Σ_0. For details, the reader is referred to [4].

We now summarize some results from [2] and [4] that will be needed in the next section.

Theorem 1.   The family of BT-VLSI acceptable languages contains all regular languages and is closed under Boolean operations. It is decidable whether or not the language accepted by a given BT-VLSI is empty and also whether or not the languages accepted by two given BT-VLSI's are equal. Every language accepted by a nondeterministic BT-VLSI is accepted by a deterministic one.

3.   Systolic EOL Languages

We begin with a result establishing the interconnection between systolic tree automata and EOL systems, [6]. We make use of the different characterizations of the family of EOL languages, in particular, the finite macro OL systems (FMOL systems) due originally to [5]. (See also Theorem II.2.2 in [6].)

Theorem 2.   Every language acceptable by a binary systolic tree automaton is also an EOL language.

Proof.   Consider a language L , accepted by a BT-VLSI

$$G = (\Sigma, \Sigma_0, \Sigma_0', g, h)$$

where the different items are the ones defined in the previous section. For each letter A in Σ_0' , we define a OL system M(A) as follows.

The alphabet of M(A) consists of the letters B^t , B^m , B^e , where B ranges over the alphabet Σ_0 , as well as of the letter A . (Intuitively, t , m , e stand for „terminal", „mixed", „ending" , respectively.) The axiom of M(A) is A .

Consider now the function h . Whenever h(B, C) = D , the system M(A) has the productions

$$D^t \rightarrow B^t C^t \ , \qquad D^m \rightarrow B^t C^m \ , \qquad D^m \rightarrow B^t C^e \ ,$$
$$D^m \rightarrow B^m C^e \ , \qquad D^e \rightarrow B^e C^e \ .$$

Moreover, whenever $h(B, C) = A$, then $M(A)$ has the productions

$$A \rightarrow B^t C^t \quad , \qquad\qquad\qquad A \rightarrow B^t C^m \quad .$$

$M(A)$ has no further productions and, thus, we have completed the definition of the system $M(A)$.

We define, finally, a homomorphism α mapping a subalphabet of $M(A)$ into $\Sigma \cup \{\varepsilon\}$ where ε denotes the empty word. Assume that $g(\#) = B$. Then $\alpha(B^e) = \varepsilon$. Assume, further, that a is in Σ and that $g(a) = C$. Then $\alpha(C^t) = a$.

It is now easy to verify that the language L equals the union of the languages $\alpha(L(M(A)))$ where A ranges over Σ_0' . Indeed, the productions of $M(A)$ simulate in a top-down manner the bottom-up behaviour of our BT-VLSI G. It is also taken care of, by the definition of α and by the three types t, m, e of the letters, that the role of the end marker $\#$ is correct: it appears always at the end only and always at the right half of the tree. Moreover, there is at least one terminal symbol in the right half of the tree.

Because L equals the union of the languages $\alpha(L(M(A)))$, it is clear that L is EOL. Instead of the union, we can also consider a single basic OL system, by introducing a new initial letter. Observe also that it causes no difficulties that α is not defined in the whole alphabet of $M(A)$: we can let the remaining letters go into a garbage symbol and intersect the entire morphic image with a regular set. $\qquad\Box$

In view of Theorem 2, we refer in this paper the languages accepted by binary systolic tree automata as <u>binary systolic EOL languages</u> or, briefly, <u>systolic EOL languages</u>. It is clear that systolic EOL languages form a proper subfamily of the whole family of EOL languages. Indeed, systolic EOL languages are FMOL languages of a special kind. The underlying OL system is binary and backward deterministic (invertible) in the sense that the right side BC of a production uniquely determines its left side. (Instead of binary trees, we could consider any balanced trees.) Moreover, for each letter B of the OL system, its macro substitution either consists of a single letter or of the empty word, or else is empty. Invertibility holds also for the macro substitutions. An explicit characterization for systolic EOL languages is given in Theorem 5.

As an example, consider the BT-VLSI $G = (\{a,b\}, \{A,B,C,D\}, \{C\}, g, h)$ where g and h are defined by

$$g(a) = A, \quad g(b) = B, \quad g(\#) = D ;$$
$$h(A,A) = A ,$$
$$h(B,B) = B ,$$
$$h(A,B) = h(A,C) = h(C,B) = C ,$$
$$h(x,y) = D \text{ in all other cases } .$$

It is easy to verify that the language accepted by G equals

$$L = \{w \in a^{+}b^{+} \mid w \text{ is of length } 2^{n}, \text{ for some } n \geq 1\} .$$

The same language L can also be expressed in the form $\alpha(L(M))$ where the homomorphism α and the OL system M are defined as follows.

The axiom of M is C , and the productions are:

$$C \rightarrow A^t B^t \; , \qquad C \rightarrow A^t B^m \; , \qquad C \rightarrow A^t C^t \; , \qquad C \rightarrow A^t C^m \; ,$$

$$C \rightarrow C^t B^t \; , \qquad\qquad\qquad C \rightarrow C^t B^m \; ,$$

$$C^t \rightarrow A^t B^t \; , \qquad\qquad C^m \rightarrow A^t B^m \; , \qquad\qquad C^m \rightarrow A^t B^e \; ,$$

$$C^m \rightarrow A^m B^e \; , \qquad\qquad C^e \rightarrow A^e B^e \; ,$$

$$C^t \rightarrow A^t C^t \; , \qquad\qquad C^m \rightarrow A^t C^m \; , \qquad\qquad C^m \rightarrow A^t C^e \; ,$$

$$C^m \rightarrow A^m C^e \; , \qquad\qquad C^e \rightarrow A^e C^e \; ,$$

$$C^t \rightarrow C^t B^t \; , \qquad\qquad C^m \rightarrow C^t B^m \; , \qquad\qquad C^m \rightarrow C^t B^e \; ,$$

$$C^m \rightarrow C^m B^e \; , \qquad\qquad C^e \rightarrow C^e B^e \; ,$$

$$B^t \rightarrow B^t B^t \; , \qquad\qquad B^m \rightarrow B^t B^m \; , \qquad\qquad B^m \rightarrow B^t B^e \; ,$$

$$B^m \rightarrow B^m B^e \; , \qquad\qquad B^e \rightarrow B^e B^e \; ,$$

$$A^t \rightarrow A^t A^t \; , \qquad\qquad A^m \rightarrow A^t A^m \; , \qquad\qquad A^m \rightarrow A^t A^e \; ,$$

$$A^m \rightarrow A^m A^e \; , \qquad\qquad A^e \rightarrow A^e A^e \; .$$

The homomorphism α is defined by

$$\alpha(A^t) = a \; , \qquad\qquad \alpha(B^t) = b \; ,$$
$$\alpha(x) = \$ \qquad \text{for} \quad x \neq A^t, B^t \; .$$

Then the systolic EOL language L can be expressed as

$$L = \alpha(L(M)) \cap \{a,b\}^* \; .$$

We have presented the construction along the lines of the proof of Theorem 2 (except that we did not introduce the unnecessary symbol D at all). It is obvious that in this example the productions can be simplified considerably.

Below is depicted the acceptance of the word $a^3 b^5$ by the BT-VLSI G:
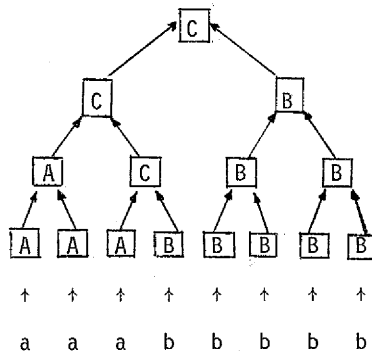
The OL system M generates the word $(A^t)^3 (B^t)^5$ as follows:
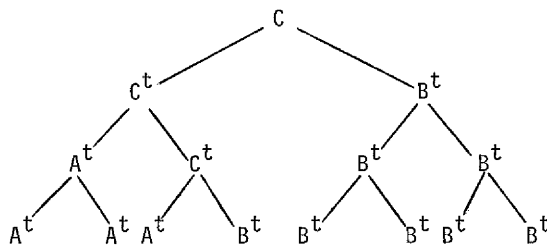
From this word the word $a^3 b^5$ is immediately obtained by the morphism α .

12.

Theorem 1 now yields the following two corollaries.

Theorem 3.   The family of systolic EOL languages contains all regular languages and is closed under Boolean operations.

Theorem 4.   The equivalence problem is decidable for systolic EOL languages.

Theorem 3 shows that, for instance, the complements of the languages

$$\{a^{2^n} \mid n \geqq 1\} , \quad \{a^{2^n-1} \mid n \geqq 1\} ,$$
$$\{a^{2^n}b^{2^n}a^{2^{n-1}}b^{2^{n-1}} \ldots a^2b^2 \, ab \mid n \geqq 0\} ,$$

as well as all languages obtainable from those and regular languages by Boolean operations are, in fact, systolic EOL languages. The EOL nature of these languages seems far from being clear, for instance, in view of the results presented in [6]. This indicates that systolic tree automata might give some really new insight also as regards the theory of L systems.

Apart from Theorem 4, a number of other decidability results can be obtained for the family of systolic EOL languages. For instance, the emptiness is decidable, either by Theorem 1 or by Theorem 2 and the fact that it is decidable for EOL languages. Intuitively, the decidability results mean that systolic EOL languages constitute a fairly „small" family whereas the other facts (closure properties, the regular languages as well as complicated exponential languages being included) seem to indicate that we are dealing with a „large" family!

We now give a characterization of systolic EOL languages. The characterization is especially pleasing because an analogous character-ization can be obtained for the whole family of EOL languages. We begin with two definitions.

A language $L \subseteq \Sigma^*$ is called a <u>semibinary EOL language</u> if L can be written as

$$L = H(L(G)) \ ,$$

where (i) $G = (\Sigma \cup \{\#\}, P, S)$ is a OL system such that the axiom S is a letter and the right side of every production in P is of length 2 and (ii) H is the homomorphism defined by

$$H(a) = a \text{ for } a \in \Sigma \ , \quad H(\#) = \varepsilon \ .$$

If, in addition, $L(G)$ consists of words $w\#^i$ where $w \in \Sigma^*$ and $i \geqq 0$ we refer to L as a <u>semibinary suffix EOL language</u>.

Clearly, every semibinary EOL language is an EOL language. We are now ready for the characterization results.

<u>Theorem 5.</u>   A language is an EOL language if and only if it is a coding (i.e., a homomorphic image under a letter-to-letter homomorphism) of a semibinary EOL language. A language is a systolic EOL language if and only if it is a coding of a semibinary suffix EOL language.

<u>Proof.</u>   Consider the first sentence. The „if" - part follows by the closure property of EOL languages. The „only if" - part is a reformu-lation of Lemma 2.3 in [1].

Consider then the second sentence. The „only if" - part is a direct consequence of the proof of Theorem 2. (Observe, in particular, that L being the union of the languages $\alpha(L(M(A)))$ implies that L is a coding of a semibinary suffix EOL language.)

To prove the „if" - part, consider a semibinary suffix EOL language $L \subseteq \Sigma^*$. Thus, $L = H(L(G))$, where the OL system $G = (\Sigma \cup \{\#\}, P, S)$ and the homomorphism H satisfy the conditions of the definition above. Consider, further, a language $L_1 \subseteq \Delta^*$ such that $L_1 = c(L)$, where $c : \Sigma^* \to \Delta^*$ is a coding.

Consider now a nondeterministic BT-VLSI K, defined as follows. The terminal alphabet of K is Δ. The operating alphabet consists of all letters a in $\Sigma \cup \{\#\}$ and their „barred versions" $\overline{a}$, as well as of an additional letter $\hat{S}$. The letter $\hat{S}$ is the only designated one (i.e., $\Sigma_0' = \{\hat{S}\}$). The input and transition functions g and h are defined as follows.

Whenever $c(x) = a$, for some $a \in \Delta$, then x is in $g(a)$. Moreover, $g(\#) = \{\overline{\#}\}$.

Whenever $A \to BC$ is a production in P, then
(i) A is in $h(B,C)$, (ii) $\overline{A}$ is in $h(B,\overline{C})$, and (iii) $\overline{A}$ is in $h(\overline{B},\overline{C})$. Moreover, whenever $A \to BC$ is in P and the axiom S of G derives according to G a word $AD_1 \ldots D_n$ such that a word in $\#^*$ is derivable according to G from $D_1 \ldots D_n$, then $\hat{S}$ is both in $H(B,C)$ and in $h(B,\overline{C})$. (A special case of our conditions is that the production $S \to BC$ is in P.)

The reader should have no difficulties to verify that
$L(K) = L_1$. Observe, in particular, that if $\hat{S}$ is introduced too early
in a computation according to K then this computation cannot continue
because h is undefined.

Consequently, by Theorem 1 (last sentence), a deterministic
BT-VLSI K' can be found such that $L_1 = L(K')$. Hence, L_1 is a
systolic EOL language, which completes the proof of Theorem 5. $\square$

4.   Interconnections with Decidability in Z-rational Formal Power Series

The two most celebrated open decision problems in the theory of formal power series can be stated as follows.

(1)   Is it decidable whether or not a negative number appears in a given Z-rational sequence of integers?

(2)   Is it decidable whether or not the number 0 appears in a given Z-rational sequence of integers?

Since every Z-rational sequence a_i , $i = 1, 2, \ldots,$ can be obtained from the upper right-hand corners of the powers M^i , where M is a square matrix with integer entries, problems (1) and (2) are at the moment perhaps the most simply stated open decision problems. Moreover, a number of important language-theoretic problems have been reduced to problems (1) and (2). The reader is referred to [6] and [7] for further details.

We consider now the problem of whether or not the decidability results expressed in Theorems 1 and 4 can be extended to concern arbitrary T-VLSI acceptable languages (instead of BT-VLSI acceptable languages or, equivalently, systolic EOL languages.) More explicitly, we consider the following problem:

(3)   Is it decidable whether or not the language accepted by a given T-VLSI is empty?

We observe first that the decidability of (3) immediately implies the decidability of the equivalence problem for T-VLSI's because, by [2], also languages acceptable by T-VLSI's are closed under Boolean operations.

Hereby, we always consider families obtainable from the <u>same</u> underlying infinite tree, i.e., we do not compare languages resulting from different trees. (So the equivalence problem concerns only languages resulting from the same infinite tree.)

While we conjecture that (3) is decidable, the proof will be extremely difficult because of the following result.

<u>Theorem 6</u>.   The decidability of (3) implies the decidability of both (1) and (2).

Before starting the proof of Theorem 6, we would like to point out that, inspite of numerous efforts, no reduction results are known between the problems (1) and (2) (i.e., that the decidability of one of them would imply the decidability of the other). Intuitively, this indicates that the proof of the decidability of (3) will be more difficult than that of (1) and (2).

We need the following auxiliary result.

<u>Lemma 7</u>.   (i) To establish the decidability of (1) (resp. (2)), it suffices to construct an algorithm for deciding of two given PDOL length sequences a_n and b_n ($n = 0, 1, 2, \ldots$) whether or not $a_n \geq b_n$ holds for all n (resp. whether or not there exists an n such that $a_n = b_n$). (ii) The PDOL systems defining the length sequences a_n and b_n in (i) may be assumed without loss of generality to have the property that, for any fixed integer k, the right side of every production is of length $\geq k$. (iii) As regards the decidability of (2), we may assume without loss of generality that we consider only PDOL length sequences a_n and b_n of which it is known that $a_n \geq b_n$ holds for every n .

<u>Proof.</u>   (i) This is a direct consequence of Corollary III.7.3 in [7].
(ii) Assume that a_n' and b_n' are arbitrary PDOL length sequences,
whereas the decidability for the length sequences satisfying the special
property stated in (ii) has been established. We now modify the PDOL
systems generating a_n' and b_n' as follows. On the right side of every
production every occurrence of every letter c is replaced by c^k . This
implies that the length sequences a_n' and b_n' are transformed into the
length sequences $a_n = k^n a_n'$ and $b_n = k^n b_n'$, where a_n and b_n sat-
isfy the special property stated in (ii). The assertion now follows by
the fact that $a_n' - b_n' > 0$ (resp. = 0) if and only if $a_n - b_n > 0$
(resp. = 0).

Finally, to establish (iii) we assume that the decidability of
the existence of an n such that $a_n = b_n$ has been established for PDOL
length sequences a_n and b_n of which it is known that $a_n \geq b_n$ holds
for every n . Let z_n be an arbitrary Z-rational sequence and let
y_n be the Hadamard square of z_n . By Corollary III.7.3 in [7], there
are PDOL length sequences a_n and b_n such that $y_n = a_n - b_n$ holds
for every n . Clearly, we must have $a_n \geq b_n$ for every n and, con-
sequently, we can decide whether or not $y_n = 0$ holds for some n .
But it is obvious that $y_n = 0$ holds for some n if and only if
$z_n = 0$ holds for some n .   □

We are now in the position to establish Theorem 6. Assume that
we know an algorithm for deciding the emptiness of the language accepted
by a given T-VLSI. We show first the decidability of (1). By Lemma 7.(ii),

it suffices to consider PDOL length sequences a_n and b_n where the right side of every production in the defining systems is of length ≥ 2. We have to decide whether or not $a_n \geq b_n$ holds for all n . Let A and B be the axioms, and α and β the morphisms defining the PDOL systems in question.

A T-VLSI G is now constructed as follows. We first define the underlying labeled tree T . The root is labeled by S and the nodes in the first level are labeled by the letters of BA . (Thus, there are $a_1 + b_1$ nodes in the first level.) The morphisms β and α are now simulated on the left and right part of the tree as follows. Assume that $\beta(u) = v_1 \ldots v_i$ (resp. $\alpha(\overline{u}) = \overline{v}_1 \ldots \overline{v}_i$), for some letters u and v . By our assumption, we know that $i \geq 2$. The beginning of the subtree starting from the node labeled by u (resp. $\overline{u}$) in the left (resp. right) part of T looks as follows:
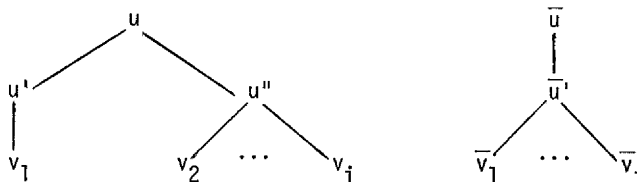
Clearly, the tree T thus obtained satisfies the regularity condition. From the length point of view it looks as follows:
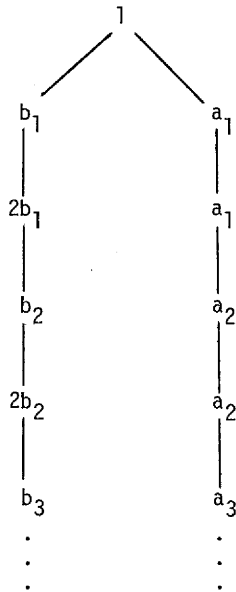
The terminal alphabet of our T-VLSI G consists of only one letter c . A word is accepted if and only if it comes from an even level (i.e., $2b_i - a_i$ - level) and does not reach the right part of the tree. This acceptance condition is easy to define formally in terms of the g and h functions: we keep track of the leftmost node N_ℓ of the right part of the tree. In even levels, the input # to N_ℓ sends an accepting signal (letter of the operating alphabet). This is the only possibility for acceptance.

We now claim that the language L(G) is empty if and only if $a_n \geqq b_n$ holds for every n . Indeed, assume that there is an m such that $b_m > a_m$. Consider the word $w = c^{2b_m}$. Clearly, there is not

enough space for w in the $b_m - a_m$ - level, whereas on the next level (i.e., $2b_m - a_m$ - level) w becomes acceptable: it does not reach the right part of the tree. Hence $L(G)$ is not empty.

Conversely, assume that $a_n \geqq b_n$ holds for all n . We show that $L(G)$ is empty. Consider an arbitrary terminal word c^i . If $i \leqq b_1 + a_1$ then c^i is clearly not accepted. Otherwise, there is a unique j such that $b_j + a_j < i \leqq b_{j+1} + a_{j+1}$. Now c^i is fed in either on the $2b_j - a_j$ - level or on the $b_{j+1} - a_{j+1}$ - level. In both cases it will be rejected because in the former case it reaches the right part of the tree.

Having shown that (1) is decidable, we now prove that also (2) is decidable (under our assumption that (3) is decidable).

By Lemma 7.(ii)-(iii), we assume that we are given the PDOL length sequences a_n and b_n of which we know that $a_n \geqq b_n$ holds for all n and that the right side of every production of the PDOL system generating b_n is of length $\geqq 3$. We have to decide whether or not $a_n = b_n$ holds for some n .

We now construct a T-VLSI G' slightly different from G considered above. From the length point of view, the underlying tree is now

$$1$$

Left branch: b_1 — $2b_1+1$ — b_2 — $2b_2+1$ — b_3 — $\vdots$

Right branch: a_1 — a_1 — a_2 — a_2 — a_3 — $\vdots$
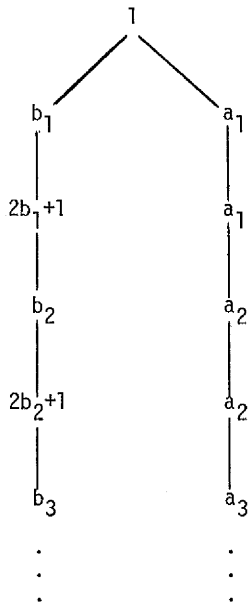
(Observe that we can realize the left part because we applied Lemma 7.(ii) for $k = 3$.) The acceptance condition is now: a word is accepted if and only if it comes from the $(2b_i + 1) - a_i$ - level, for some i , and exactly fills the left part of the tree on this level. (Thus, we have to keep track also of the rightmost node in the left part.)

The reader should have no difficulties in verifying that $L(G')$ is nonempty if and only if there is an n such that $a_n = b_n$. This completes the proof of Theorem 6.

We mention, finally, that it is an open problem whether or not every T-VLSI acceptable language is an EOL language. To establish effectively a positive answer to this problem seems extremely difficult. The

construction of Theorem 2 does not work because of reasons similar to the ones exhibited in the proof of Theorem 6.   Indeed, if we can effectively construct an EOL system accepting a given T-VLSI language, we can also decide problems (1) and (2).

REFERENCES

[1]   K. Culik II, On some families of languages related to developmental systems. Internat. J. Comput. Math. 4 (1974), pp. 31-42.

[2]   K. Culik II, A. Salomaa and D. Wood, VLSI systolic trees as acceptors. Res. Report CS-81-32, Dept. of Computer Science, University of Waterloo, Waterloo, Ontario (1981).

[3]   K. Culik II, J. Gruska and A. Salomaa, Systolic trellis automata (for VLSI). Ibid., CS-81-34.

[4]   K. Culik II, J. Gruska and A. Salomaa, Systolic automata for VLSI on balanced trees. Ibid., CS-82-01.

[5]   K. Culik II and J. Opatrny, Macro OL systems. Internat. J. Comput. Math. 4 (1975), pp. 327-342.

[6]   G. Rozenberg and A. Salomaa, The Mathematical Theory of L Systems. Academic Press (1980).

[7]   A. Salomaa and M. Soittola, Automata - Theoretic Aspects of Formal Power Series. Springer-Verlag (1978).