Acta Cybernetica, Vol. 12, No. 1, Szeged, 1995

Generalized DOL trees[*]

Lila Kari[†] Grzegorz Rozenberg[‡] Arto Salomaa [§]

Abstract

Infinite unlabeled trees having a finite number of different subtrees (also called infinite regular trees) arise in a natural way from a DOL system which also gives a natural labeling for the tree. A much more compact representation for the tree often results from a DOL system with fragmentation.

Keywords: formal languages, DOL system, fragmentation, tree labeling.

1 Introduction

One of the simplest, if not the simplest, models extensively investigated in the theory of computing is the *DOL system*. By definition, a *DOL system* is a triple $G = (\Sigma, h, w)$, where Σ is a finite alphabet, $h : \Sigma^* \longrightarrow \Sigma^*$ is a morphism, and $w \in \Sigma^*$ is a word (usually called the *axiom*). The DOL system G generates the sequence $S(G)$ of words $w_0, w_1, w_2, \ldots$, where

$$w_0 = w \text{ and } w_i = h^i(w) = h(w_{i-1}) \text{ for } i \geq 1.$$

Thus, $S(G)$ is obtained from the axiom by iterating the morphism. (Our exposition is largely self-contained. If need arises, [3] can be consulted. [1] and [4] are some of the recent papers concerning DOL systems.)

As an example, consider the DOL system with the alphabet $\Sigma = \{a, b\}$, axiom $w = a$ and the morphism h defined by the rules

$$a \longrightarrow b, \quad b \longrightarrow ab.$$

This is the well-known "Fibonacci system", where the lengths of the words in the generated sequence

$$a, b, ab, bab, abbab, bababbab, \ldots$$

form the sequence of Fibonacci numbers. The following tree, labeled by the letters of Σ, depicts the generation process:

[*] Research partially supported by the Academy of Finland, project 11281. All correspondence to Lila Kari.

[†] Department of Mathematics, University of Western Ontario, London, Ontario, N6A 5B7 Canada

[‡] Department of Computer Science, Leiden University, 2300 RA Leiden, The Netherlands

[§] Academy of Finland and Department of Mathematics, University of Turku, 20 500 Turku, Finland

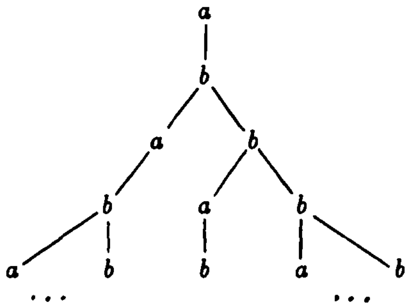

Figure 1.

For the DOL systems G considered in this paper, we assume that $|w| = 1$ (that is, the axiom is a letter) and that h is *nonerasing* (that is, we are dealing with *PDOL systems*). These assumptions guarantee that the sequence $S(G)$ can always be represented as an infinite tree labeled by letters of Σ, where all branches continue ad infinitum.

Remark. If we allow $|w| > 1$, we are dealing with forests instead of trees. An additional letter used only as the axiom brings us back to trees. Moreover, our main result remains valid for general DOL systems as well. Consequently, our assumptions do not exclude any interesting cases.

Infinite (labeled) trees obtained in the way described above are referred to as *DOL trees.* The formal definition of a DOL tree should be clear and is omitted here. It is also clear that if you begin with an infinite unlabeled tree that possesses only finitely many different subtrees (such trees are often referred to as *regular*), then you can label it with finitely many labels and view the result as a DOL tree. The labels constitute the alphabet of the corresponding DOL system.

The arity of each letter is the length of the right side of the rule for the letter.

Regular trees play a central role in the theory of automata, nonrecursive program schemes, etc. Such matters are of no direct concern to us in this paper. For the sake of later reference, we summarize the above discussion in the following lemma. Thus, an infinite unlabeled tree is *regular* if it possesses only finitely many different subtrees. The *unlabeled version* of a DOL tree is obtained from a DOL tree by removing the labels.

Lemma 1.1 *The unlabeled version of a DOL tree is regular. Conversely, every regular tree can be labeled to become a DOL tree.*

If we do not make the convention above (to the effect that all branches of the trees continue ad infinitum), then the DOL systems should containg also erasing rules.

As a further example, consider the tree

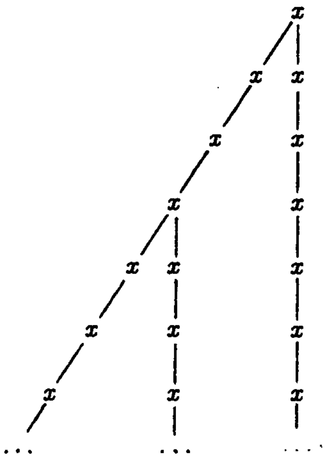

Figure 2.

Thus, a new branch is born at every third node of the stem. Clearly, the DOL system with the axiom a and the rules

$$a \longrightarrow bd, \quad b \longrightarrow c, \quad c \longrightarrow a, \quad d \longrightarrow d$$

provides the labeling.

Let us modify the example in such a way that the new branches are born at nodes whose distance from the root is a prime number. Then it is not possible to label the tree in such a way that it becomes a DOL tree. Indeed, infinitely many different subtrees arise.

2 Fragmentation

Consider the DOL system with the axiom a and rules

$$(1) \quad a \longrightarrow bc, b \longrightarrow bdc, c \longrightarrow bd^2c, d \longrightarrow bd^4c.$$

The beginning of the tree is as in Figure 3. We obviously need four labels for the simple reason that we have nodes of four different degrees: 2, 3, 4 and 6.

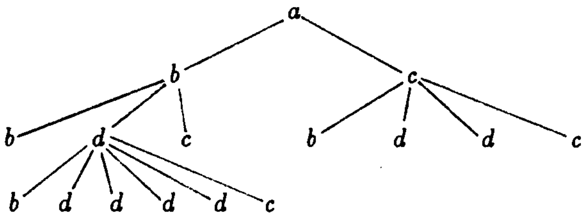

Figure 3.

However, we can represent the tree in the following much more compact way. The idea resembles DOL systems with *fragmentation*, introduced originally in [2].

Assume that the alphabet Σ contains a special letter $\#$, viewed as a marker or separator. Then we speak of *$\#$-guarded subwords* of words y over Σ. They are the maximal parts of y separated by $\#$. For instance, if $y = ab\#a\#bab\#b$, then the $\#$- guarded subwords are ab, a, bab, b. Formally, a word x not containing $\#$ is a *$\#$-guarded subword* of y iff $\#x\#$ is a subword of $\#y\#$.

Consider a *marked DOL system* $G_{\#} = (\Sigma, h, w)$, where the alphabet contains the marker $\#$, for which the rule is $\# \longrightarrow \#$. (Also now we assume that h is nonerasing and $|w| = 1$.) We now associate to $S(G_{\#})$ a tree labeled by words over $(\Sigma - \{\#\})^*$. The labels of the tree will be the $\#$- guarded subwords of the words in $S(G_{\#})$. In this process, several consecutive $\#$'s will be identified with one $\#$. Trees obtained in this fashion will be referred to as *generalized DOL trees*. Let us consider some examples.

If the marker $\#$ does not occur in the sequence, the generalized DOL tree has no branches. The generalized tree associated to the Fibonacci system is:

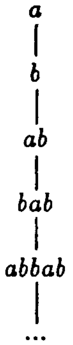

Figure 4.

Consider next the marked DOL system with the axiom a and rule

$$(2) \quad a \longrightarrow a^2\#a^3.$$

The generalized DOL tree begins now as follows:

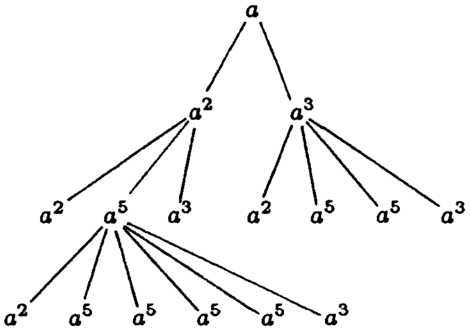

Figure 5.

We observe that the unlabeled tree is exactly the same as the one considered at the beginning of this section. Thus, in place of (1), we have obtained the much more compact representation (2)!

The term "generalised" can be justified as follows. An ordinary DOL system G can be transformed into a marked one $G_{\#}$ by separating all letters on the right sides of the rules with the marker $\#$. Since the axiom is a singleton letter, all labels of the resulting tree are singletons. Then the unlabeled versions of the DOL tree associated to G and the generalized DOL tree associated to $G_{\#}$ coincide. Consequently, the following result holds true.

Lemma 2.1 *The unlabeled version of every DOL tree equals the unlabeled version of a generalized DOL tree.*

Our main purpose is to prove the converse of Lemma 2.1. Thus, the unlabeled versions of DOL trees and generalised DOL trees coincide. However, in general, a marked DOL system provides a much more compact representation for the tree than a DOL system.

By Lemma 1.1, it suffices to prove that, given a marked DOL systems $G_{\#}$, there is a constant k such that all words appearing as labels in the generalized tree are of length less than k. Unfortunately, as such this claim is not true. Any DOL system generating an infinite language and not containing at all the marker $\#$ in its sequence, such as the Fibonacci system, provides a counterexample. Another counterexample is provided by the system with the axiom a and the rules

$$(3) \quad a \longrightarrow b\#ab, \quad b \longrightarrow b^2.$$

The generalized DOL tree is in this case

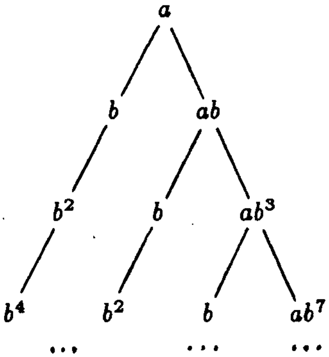

Figure 6.

However, in both cases our claim holds true. The above tree is generated by the DOL system with the axiom a and rules $a \longrightarrow ba$, $b \longrightarrow b$. The generalized tree of the Fibonacci system is generated by the DOL system with the axiom a and the rule $a \longrightarrow a$.

The tool for obtaining a constant k as described above is to eliminate the unbounded growth by transforming the given marked DOL system $G_{\#}$ into a marked system with the same (unlabeled version of the) generalized tree. We say that a letter b of $G_{\#}$ is *useful* if $b \Longrightarrow^{*} u\#v$, for some words u and v (that is, $h^{i}(b) = u\#v$, for some u, v, i). Otherwise, b is *useless*. Thus, the sequence starting from a useless letter does not contain the marker $\#$. Clearly, usefulness is a decidable property.

The *useful variant* $G'_{\#}$ of a marked DOL system $G_{\#}$ is constructed as follows. If all letters appearing in $S(G_{\#})$ are useful, then $G'_{\#} = G_{\#}$. If all letters are useless, then the axiom of $G'_{\#}$ is a and $a \longrightarrow a$ is the only rule. If every #- guarded subword of the right-hand sides of the rules contains a useful letter, then to get $G'_{\#}$ we simply remove from $G_{\#}$ all useless letters and their occurrences in the rules. The case remains, where $S(G_{\#})$ contains useful letters but some #-guarded subword of the right-hand side of some rule consists of useless letters. To get $G'_{\#}$, we also now first remove from $G_{\#}$ all useless letters and their occurrences in the rules. Then we add a new letter c with the rule $c \longrightarrow c$. Finally, all #-guarded subwords that previously consisted of useless letters are replaced by c.

For instance, if $G_{\#}$ is defined by the rules (3), $G'_{\#}$ will be defined by the rules

$$a \longrightarrow c\#a, \quad c \longrightarrow c.$$

If $G_{\#}$ has the axiom a and the rules

$$a \longrightarrow d\#bcc\#d, \ b \longrightarrow a^2d\#ab, \ c \longrightarrow cd, \ d \longrightarrow dcd,$$

then $G'_{\#}$ will be defined by the rules

$$a \longrightarrow c\#b\#c, \ b \longrightarrow a^2\#ab, \ c \longrightarrow c.$$

The following result is immediate by the construction of $G'_{\#}$.

Lemma 2.2 *If $G'_{\#}$ is the useful variant of a marked DOL system $G_{\#}$, then the unlabeled versions of the generalized DOL trees associated to $G_{\#}$ and $G'_{\#}$ coincide.*

3 The main result

We will establish in this section the converse of Lemma 2.1.

Theorem 3.1 *The unlabeled version of every generalized DOL tree equals the unlabeled version of a DOL tree. Moreover, given a marked DOL system producing a generalized tree, the corresponding DOL system can be effectively constructed.*

Thus, every tree possessing a compact representation (2) is a DOL tree, (as far as the unlabeled versions are concerned) and the corresponding DOL representation (1) can be effectively constructed. Let us discuss still a more sophisticated example.

A marked DOL system $G_{\#}$ has the axiom a and rules

$$a \longrightarrow a\#ab\#ab^2, \quad b \longrightarrow a.$$

Observe first that both a and b are useful and, thus, $G'_{\#} = G_{\#}$. Since the generalized tree is quite involved, we give it in parts, continuing the process as long as new labels are born:

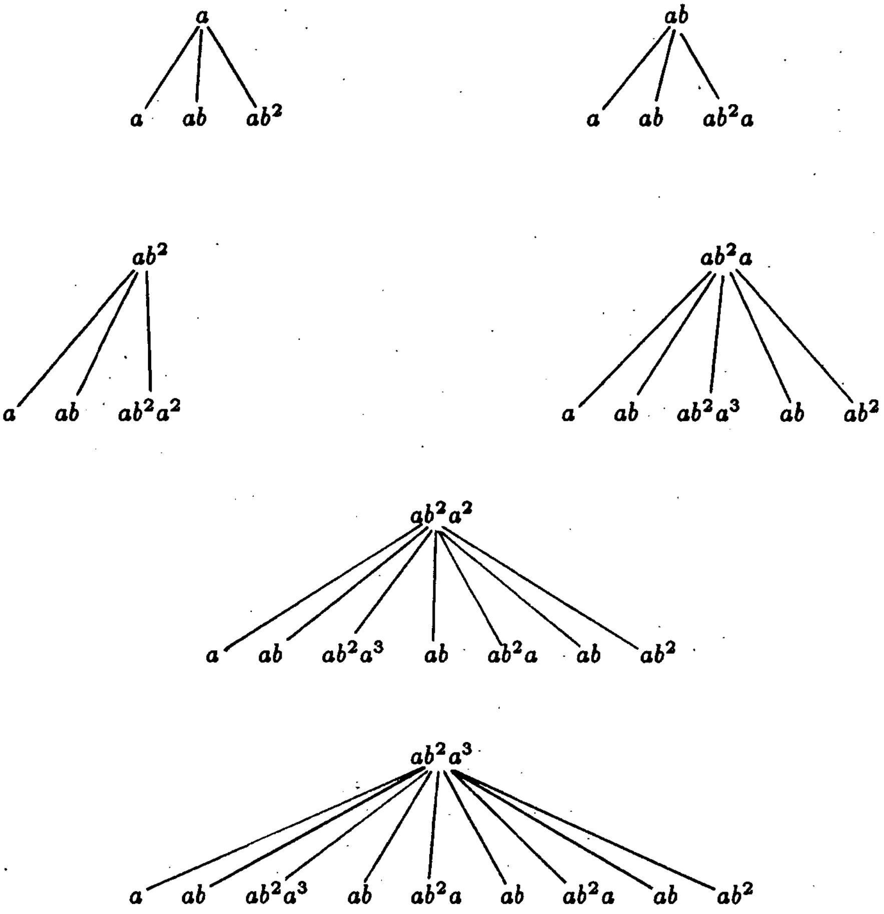

Figure 7.

Thus, if we denote the six labels appearing at the roots by a, b, c, d, e, f, we obtain the rules of the corresponding DOL system:

$$\begin{array}{lll} a \longrightarrow abc & b \longrightarrow abd, & c \longrightarrow abe, \\ d \longrightarrow abfbc, & e \longrightarrow abfbdbc, & d \longrightarrow abfbdbdbc. \end{array}$$

We will now establish our Theorem. By Lemma 2.2 we may restrict the attention to useful variants. We have to show that a constant k can be effectively computed from the system such that all labels in the generalized tree are shorter than k. More specifically, we have to establish the following result.

Lemma 3.1 *Assume that $G_{\#}$ is a marked DOL system coinciding with its useful variant, $G_{\#} = G'_{\#}$. Then a constant k can be effectively computed such that the length of every label in the generalized DOL tree of $G_{\#}$ is at most k.*

Proof. The alphabet Σ contains at most one useless letter, c. Let Σ' be the subalphabet obtained by excluding # and c, and let r be the cardinality of Σ'. Thus, all letters of Σ' are useful. Define the *rank* of a letter $a \in \Sigma'$ to be the smallest integer k such that $h^k(a)$ contains an occurrence of #. Clearly, the rank can be effectively computed and every letter is of rank $\leq r$.

Consider the lengths of #-guarded subwords of the words $h(a)$ when a ranges over letters of rank 1. Let m_1 be twice the maximal length. Define further

$$m_2 = \max\{|h(a)| \mid a \text{ is of rank } > 1\},$$
$$M = \max\{m_1, m_2\}.$$

We claim that we can choose

$$k = M^r + M^{r-1} + \ldots + M = (M^r - 1)M/(M - 1).$$

Let v be a label in the generalized DOL tree. We have to estimate $|v|$ and show that $|v| \leq k$. Clearly, we may assume that v is not the label of the root. Hence, v is a #-guarded subword of $h(\bar{v})$, where $\bar{v}$ is in the sequence $S(G_{\#})$. The situation can be depicted as follows, with $v = u_1 u_3 u_2$:

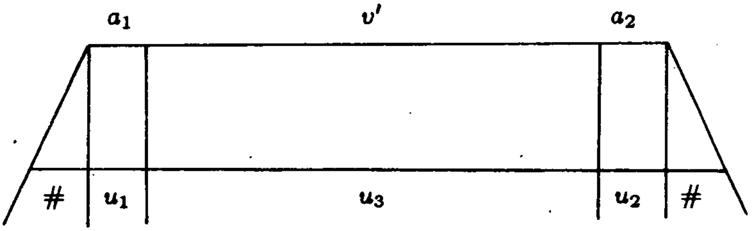

Figure 8.

Here every letter of v', if any, is of rank > 1, and a_1, a_2 are letters of rank 1. Thus, we look how the #-guarded subword v is created. a_1 and a_2 may also produce something else beyond the marker #. One of them (or both) may be missing if we are dealing with a prefix or suffix. We obtain the estimates

$$|u_1| + |u_2| \leq m_1 \text{ and } |u_3| \leq m_2|v'|$$

and, consequently,

$$|v| \leq M \cdot |v'| + M.$$

We now estimate similarly the length $|v'|$. (In fact, we obtain an upper bound for an eventually longer word that contains also a_1, a_2 and maybe still a prefix and suffix.) By considering the preceding word in the sequence, we get an analogous picture and the estimate

$$|v'| \leq M \cdot |v''| + M,$$

where the letters of v'', if any, are of rank > 2. Consequently,

$$|v| \leq M(M \cdot |v''| + M) + M = M^2 \cdot |v''| + M^2 + M.$$

Continuing in the same way, we obtain

$$|v| \leq M^r|v^{(r)}| + M^r + M^{r-1} + \ldots + M,$$

where every letter in $v^{(r)}$, if any, is of rank $> r$. But there are no letters of rank $> r$. Thus, $v^{(r)}$ is the empty word and, consequently,

$$|v| \leq M^r + M^{r-1} + \ldots + M = k.$$

This concludes the proof of Lemma 3.1 and also the proof of our Theorem. □

Consider the example discussed at the beginning of this section. We obtain

$$r = 2, \quad m_1 = 2 \cdot 3 = 6, \quad m_2 = 1, \quad M = 6, \quad k = 42,$$

whereas in the actual construction the maximal word length was 6. Indeed, our bound k can be improved. For instance, in the definition of m_1 it suffices to consider the sum of the lengths of the maximal #-guarded prefix and suffix, rather than twice the maximal word length. This improvement gives $m_1 = M = 4$, $k = 20$.

4 Conclusion

We have introduced a compact way of representing certain infinite trees. The method uses DOL systems with fragmentation and leads to trees whose nodes are labeled by words. Although the lengths of such words may grow beyond all bounds, the unlabeled versions of the trees are still regular and, thus, possess a DOL representation. However, the loss in compactness in the transition to the DOL representation can be enormous.

We do not investigate in this paper the complexity issues involved or for which classes of trees the new representation is especially suitable. We conclude with the following result along these lines. The result is easily established by extending the example of the preceding section, for values of $r > 2$, to contain the rules

$$a \longrightarrow a\#ab\#ab^2\#\ldots\#ab^r, \quad b \longrightarrow a.$$

Lemma 4.1 *For each $r > 2$, there is an infinite unlabeled tree T such that (i) T is the unlabeled version of the generalized DOL tree of a mbrked DOL system with 2 letters, and (ii) T is not the unlabeled version of the tree of any DOL system with $\leq r$ letters.*

References

[1] J.Dassow, G.Paun and A.Salomaa. On the union of OL languages. *Information Processing Letters* 47 (1993) 59-63.

[2] G.Rozenberg, K.Ruohonen and A.Salomaa. Developmental systems with fragmentation. *International Journal of Computer Mathematics* 5 (1976) 177- 191.

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

[4] A.Salomaa. Simple reductions between DOL language and sequence equivalence problems. *Discrete Applied Mathematics* 41 (1993) 271- 274.

Received January, 1995