

# Generalized DOL trees\*

Lila Kari<sup>1</sup>, Grzegorz Rozenberg<sup>2</sup>, Arto Salomaa<sup>3</sup>

<sup>1</sup> Department of Mathematics, University of Western Ontario  
London, Ontario, N6A 5B7 Canada

<sup>2</sup> Department of Computer Science, Leiden University,  
2300 RA Leiden, The Netherlands

<sup>3</sup> Academy of Finland and Department of Mathematics,  
University of Turku, 20 500 Turku, Finland

## 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  $\Sigma$  is a finite alphabet,  $h : \Sigma^* \rightarrow \Sigma^*$  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  $w_0, w_1, w_2, \dots$ , 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

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

---

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



**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 contain also erasing rules.

As a further example, consider the tree

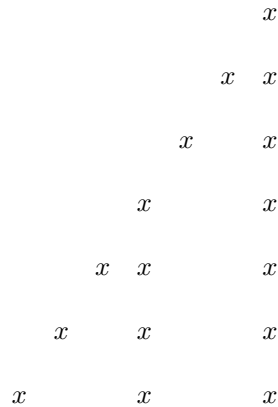


Figure 2.

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

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

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

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

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.

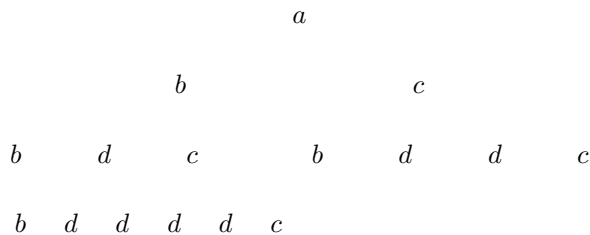


Figure 3.

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  $\Sigma$  contains a special letter  $\#$ , viewed as a marker or separator. Then we speak of  *$\#$ -guarded subwords* of words  $y$  over  $\Sigma$ . 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  $\# \rightarrow \#$ . (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:

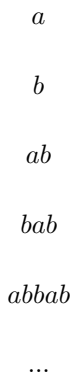


Figure 4.



The generalized DOL tree is in this case

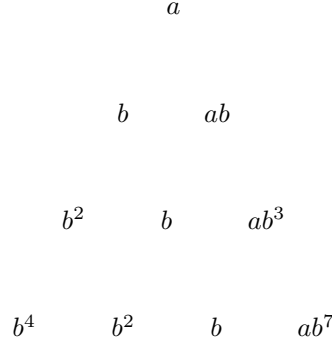


Figure 6.

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

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  $b \xrightarrow{*} u\#v$ , 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  $a \rightarrow a$  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  $c \rightarrow c$ . 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

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

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

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

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

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

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:

$a$	$ab$
$a \quad ab \quad ab^2$	$a \quad ab \quad ab^2a$
$ab^2$	$ab^2a$
$a \quad ab \quad ab^2a^2$	$a \quad ab \quad ab^2a^3 \quad ab \quad ab^2$

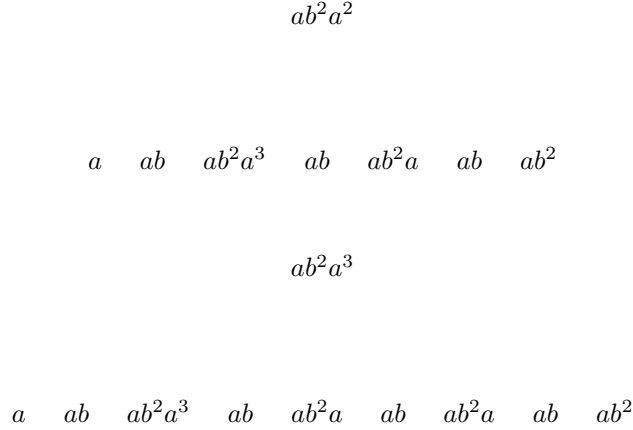


Figure 7.

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:

$$\begin{array}{lll}
a \rightarrow abc & b \rightarrow abd, & c \rightarrow abc, \\
d \rightarrow abfbc, & e \rightarrow abfbbbc, & f \rightarrow abfbbdbc.
\end{array}$$

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  $\Sigma$  contains at most one useless letter,  $c$ . Let  $\Sigma'$  be the subalphabet obtained by excluding  $\#$  and  $c$ , and let  $r$  be the cardinality of  $\Sigma'$ . Thus, all letters of  $\Sigma'$  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

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





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

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

**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 marked 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.