Ogden's Lemma for Regular Tree Languages

Marco Kuhlmann*
Dept. of Linguistics and Philology
Uppsala University, Sweden

16th December 2013

1 Introduction

Pumping lemmata are elementary tools for the analysis of formal languages. While they usually cannot be made strong enough to fully capture a class of languages, it is generally desirable to have as strong pumping lemmata as possible. However, this is counterbalanced by the experience that strong pumping lemmata may be hard to prove, or, worse, hard to use—this experience has been made, for example, in the study of the output languages of tree transducers, where other proof techniques, so-called *bridge theorems*, make better tools (Engelfriet and Maneth, 2002). The purpose of this squib is to strengthen the standard pumping lemma for the class of *regular tree languages* (Gécseg and Steinby, 1997), without sacrificing its usability, in the same way as Ogden strengthened the pumping lemma for context-free string languages (Ogden, 1968).

The paper is structured as follows. Section 2 introduces our notation. Section 3 presents the main lemma and motivates it using a small, formal example. Finally, Section 4 contains the proof of the main lemma.

2 Preliminaries

We assume the reader to be familiar with the standard concepts from the theory of tree languages. The notation that we use in this paper is mostly

*I wish to thank Mathias Möhl and an anonymous reviewer for pointing out errors, and for comments that helped to improve the quality of the presentation. The work reported in this paper was partially funded by the German Research Foundation.

identical to the one used in the survey by Gécseg and Steinby (1997); the major difference is our way of denoting substitution into contexts.

We write $\mathbb{N}$ for the set of non-negative natural numbers, and $[n]$ as an abbreviation for the set $\{\, i \in \mathbb{N} \mid 1 \leq i \leq n \,\}$. Given a set A, we write $|A|$ for the cardinality of A, and A^* for the set of all strings over A.

Let Σ be a ranked alphabet. For a tree $t \in T_\Sigma$, we write $|t|$ to denote the *size of* t, defined as the number of nodes of t. A *path in* t is a sequence of nodes of t in which each node but the first one is a child of the node preceding it. Let $\circ$ be a symbol with rank zero that does not occur in Σ. Recall that a *context over* Σ is a tree c over $\Sigma \cup \{\, \circ \,\}$ in which $\circ$ occurs exactly once. We call the (leaf) node at which the symbol $\circ$ occurs the *hole* of the context. We write $|c|$ to denote the *size of the context* c, defined as the number of non-hole nodes of c. Finally, given a context c and a tree t, we write $c \cdot t$ for the tree obtained by substituting t into c at its hole. Note that Gécseg and Steinby denote this tree by $t \cdot c$ or $c(t)$.

A subset $L \subseteq T_\Sigma$ is a *tree language over* Σ. A tree language is *regular*, if there is a finite-state tree automaton that accepts L.

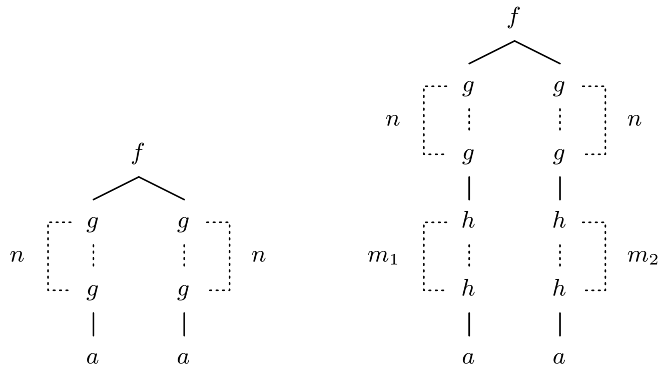
3 Motivation

To motivate the need for a strong pumping lemma for regular tree languages, we start with a look at the standard one (Gécseg and Steinby, 1997)[1]:

Lemma 1 *For every regular tree language $L \subseteq T_\Sigma$, there is a number $p \geq 1$ such that any tree $t \in L$ of size at least p can be written as $t = c' \cdot c \cdot t'$ in such a way that $|c| \geq 1$, $|c \cdot t'| \leq p$, and $c' \cdot c^n \cdot t' \in L$, for every $n \in \mathbb{N}$.* □

Just as the pumping lemma for context-free string languages, Lemma 1 is most often used in its contrapositive formulation, which specifies a strategy for proofs that a language $L \subseteq T_\Sigma$ is *not* regular: show that, for all $p \geq 1$, there exists a tree $t \in L$ of size at least p such that for any decomposition $c' \cdot c \cdot t'$ of t in which $|c| \geq 1$ and $|c \cdot t'| \leq p$, there is a number $n \in \mathbb{N}$ such that $c' \cdot c^n \cdot t' \notin L$. It is helpful to think of a proof according to this strategy as a game against an imagined ADVERSARY, where our objective is to prove that L is non-regular, and ADVERSARY's objective is to foil this proof. The game consists of four alternating turns: In the first turn, ADVERSARY must choose a number $p \geq 1$. In the second turn, we must respond to this choice

[1] The lemma given here is in fact slightly stronger than the one given by Gécseg and Steinby (1997) (Proposition 5.2), and makes pumpability dependent on the size of a tree, rather than on its height.



(a) L_1 (b) L_2

Figure 1: Two tree languages that are not regular

by providing a tree $t \in L$ of size at least p. In the third turn, ADVERSARY must choose a decomposition of t into fragments $c' \cdot c \cdot t'$ such that $|c| \geq 1$ and $|c \cdot t'| \leq p$. In the fourth and final turn, we must provide a number $n \in \mathbb{N}$ such that $c' \cdot c^n \cdot t' \notin L$. If we are able to do so, we win the game; otherwise, ADVERSARY wins. We can prove that L is non-regular, if we have a winning strategy for the game.

Consider the language $L_1 = \{\, f(g^n \cdot a, g^n \cdot a) \mid n \geq 1 \,\}$, shown schematically in Figure 1a. Using Lemma 1, it is easy to show that this language is non-regular: we always win by presenting ADVERSARY with the tree $t = f(g^p \cdot a, g^p \cdot a)$. To see this, notice that in whatever way ADVERSARY decomposes t into fragments $c' \cdot c \cdot t'$ such that $|c| \geq 1$, the pumped tree $c' \cdot c^2 \cdot t'$ does not belong to L_1. In particular, if c is rooted at a node that is labelled with g, then the pumped tree violates the constraint that the two branches have the same length.

Unfortunately, Lemma 1 sometimes is too blunt a tool to show the non-regularity of a tree language. Consider the language

$$L_2 \;=\; \{\, f(g^n \cdot h^{m_1} \cdot a, g^n \cdot h^{m_2} \cdot a) \mid n, m_1, m_2 \geq 1 \,\}$$

(see Figure 1b). It is not unreasonable to believe that L_2, like L_1, is non-regular, but it is impossible to prove this using Lemma 1. To see this, notice that ADVERSARY has a winning strategy for $p \geq 2$: for every tree $t \in L_2$ that we can provide in the second turn of the game, ADVERSARY can choose any decomposition $c' \cdot c \cdot t'$ in which $c = h(\circ)$ and $t' = a$. In this case, $|c| \geq 1$, $|c \cdot t'| \leq p$, and both deleting and pumping c yield only valid trees in L_2. Intuitively, we would like to force ADVERSARY to choose a decomposition that contains a g-labelled node, thus transferring our winning strategy for L_1—but this is not warranted by Lemma 1, which merely asserts that a pumpable context does exist *somewhere* in the tree, but does not allow us to delimit the exact region. The pumping lemma that we prove in this paper makes a stronger assertion:

Lemma 2 *For every regular tree language $L \subseteq T_\Sigma$, there is a number $p \geq 1$ such that every tree $t \in L$ in which at least p nodes are marked as distinguished can be written as $t = c' \cdot c \cdot t'$ such that at least one node in c is marked, at most p nodes in $c \cdot t'$ are marked, and $c' \cdot c^n \cdot t' \in L$, for all $n \in \mathbb{N}$.* □

Note that, in the special case where all nodes are marked, Lemma 2 reduces to Lemma 1.

Lemma 2 can be seen as the natural correspondent of Ogden's Lemma for context-free string languages (Ogden, 1968). Its contrapositive corresponds to the following modified game for tree languages L: In the first turn, ADVERSARY has to choose a number $p \geq 1$. In the second turn, we have to choose a tree $t \in L$ and mark at least p nodes in t. In the third turn, ADVERSARY has to choose a decomposition $c' \cdot c \cdot t'$ of t *in such a way that at least one node in c and at most p nodes in $c \cdot t'$ are marked.* In the fourth and final turn, we have to choose a number $n \in \mathbb{N}$ such that $c' \cdot c^n \cdot t' \notin L$. In this modified game, we can implement our idea from above to prove that the language L_2 is non-regular: we can always win the game by presenting ADVERSARY with the tree $t = f(g^p \cdot h(a), g^p \cdot h(a))$ and marking all nodes that are labelled with g as distinguished. Then, in whatever way ADVERSARY decomposes t into segments $c' \cdot c \cdot t'$, the context c contains at least one node labelled with g, and the tree $c' \cdot c^2 \cdot t'$ does not belong to L_2.

4 Proof

Our proof of Lemma 2 builds on the following technical lemma:

Lemma 3 *Let Σ be a ranked alphabet. For every tree language $L \subseteq T_\Sigma$ and every $k \geq 1$, there exists a number $p \geq 1$ such that every tree $t \in L$ in which at least p nodes have been marked as distinguished can be written as $t = c' \cdot c_1 \cdots c_k \cdot t'$ in such a way that for each $i \in [k]$, the context c_i contains at least one marked node, and the tree $c_1 \cdots c_k \cdot t'$ contains at most p marked nodes.* □

Proof Let m be the maximal rank of any symbol in Σ. Note that if m is zero, then each tree over Σ has size one, and the lemma trivially holds with $p = 2$. For the remainder of the proof, assume that $m \geq 1$. Put $g_\Sigma(n) = \sum_{i=0}^{n} m^i$, and note that $g_\Sigma(n) < g_\Sigma(n+1)$, for all $n \in \mathbb{N}$. We will show that we can choose $p = g_\Sigma(k)$.

Let $t \in L$ be a tree in which at least one node has been marked as distinguished. We call a node u of t *interesting*, if it either is marked, or has at least two children from which there is a path to an interesting node. It is easy to see from this definition that from every interesting node, there is a path to a marked node. Let $d(u)$ denote the number of interesting nodes on the path from the root node of t to u, excluding u itself. We make two observations about the function $d(u)$:

First, there is exactly one interesting node u with $d(u) = 0$. To see that there is at most one such node, let u_1 and u_2 be distinct interesting nodes with $d(u_1) = d(u_2) = n$; then the least common ancestor u of u_1 and u_2 is an interesting node with $d(u) = n - 1$. To see that there is at least one such node, recall that every marked node is interesting.

For the second observation, let u be an interesting node with $d(u) = n$. The number of interesting descendants v of u with $d(v) = n+1$ is at most m. To see this, notice that each path from u to v starts with u, continues with some child u' of u, and then visits only non-interesting nodes w until reaching v. From each of these non-interesting nodes w, there is at most one path that leads to v. Therefore, the path from u to v is uniquely determined except for the choice of the child u', which is a choice among at most m alternatives.

Taken together, these observations imply that the number of interesting nodes u with $d(u) \leq k - 1$ is bounded by the value $g_\Sigma(k - 1)$.

Now, let t be a tree in which at least $g_\Sigma(k)$ nodes have been marked as distinguished. Then there is at least one interesting node u with $d(u) = k$, and hence, at least one path that visits at least $k + 1$ interesting nodes. Choose any path that visits the maximal number of interesting nodes, and let $\vec{u}$ be a suffix of that path that visits exactly $k + 1$ interesting nodes, call them $v_1, \ldots, v_{k+1}$. We use $\vec{u}$ to identify a decomposition $c_1 \cdots c_k \cdot t'$ of t as follows: for each $i \in [k]$, choose v_i as the root node of c_i, choose v_{i+1} as the hole of c_i, and choose v_{k+1} as the root node of t'. This decomposition satisfies the required properties: To see that the tree $c_1 \cdots c_k \cdot t'$ contains at most p marked nodes, notice that, by the choice of $\vec{u}$, no path in t that starts at v_1 contains more than $k+1$ interesting nodes, and hence the total number of interesting nodes in the subtree rooted at v_1 is bounded by $g_\Sigma(k) = p$. To see that every context c_i, $i \in [k]$, contains at least one marked node, let v be one of the interesting nodes in c_i, and assume that v is not itself marked. Then v has at least two children from which there is a path to an interesting, and, ultimatively, to a marked node. At most one of these paths visits v_{i+1}; the marked node at the end of the other path is a node of c_i. ■

With Lemma 3 at hand, the proof of Lemma 2 is straightforward, and essentially identical to the proof given for the standard pumping lemma (Gécseg and Steinby, 1997):

PROOF (OF LEMMA 2) Let $L \subseteq T_\Sigma$ be a regular tree language, and let M be a tree automaton with state set Q that recognizes L. We will apply Lemma 3 with $k = |Q|$. Let $t \in L$ be a tree in which at least p nodes are marked as distinguished, where p is the number from Lemma 3. Then t can be written as $c' \cdot c_1 \cdots c_k \cdot t'$ such that for each index $i \in [k]$, the context c_i contains at least one marked node, and the tree $c_1 \cdots c_k \cdot t'$ contains at most p marked nodes. Note that each context c_i, $i \in [k]$, is necessarily non-empty. Since M has only k states, it must arrive in the same state at the root nodes of at least two contexts c_i, $i \in [k]$, or at the root node of some context c_i, $i \in [k]$, and the root node of t'. A decomposition of t of the required kind is then obtained by cutting t at these two nodes. ■

Note that by choosing $k = m \cdot |Q|$ in this proof, where $m \geq 1$, it is easy to generalize Lemma 2 as follows:

Lemma 4 *For every regular tree language $L \subseteq T_\Sigma$ and every $m \geq 1$, there is a number $p \geq 1$ such that every tree $t \in L$ in which at least p nodes are marked as distinguished can be written as $t = c' \cdot c_1 \cdots c_m \cdot t'$ such that for each $i \in [m]$, at least one node in c_i is marked, at most p nodes in $c_1 \cdots c_m \cdot t'$ are marked, and $c' \cdot c_1^n \cdots c_m^n \cdot t' \in L$, for all $n \in \mathbb{N}$.* □

References

Joost Engelfriet and Sebastian Maneth. Output string languages of compositions of deterministic macro tree transducers. *Journal of Computer and System Sciences*, 64(2):350–395, 2002. doi: 10.1006/jcss.2001.1816.

Ferenc Gécseg and Magnus Steinby. Tree languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 1–68. Springer, 1997.

William Ogden. A helpful result for proving inherent ambiguity. *Mathematical Systems Theory*, 2(3):191–194, 1968. doi: 10.1007/BF01694004.