Parikh's theorem

Håkan Lindqvist

Abstract

This chapter will discuss Parikh's theorem and provide a proof for it. The proof is done by induction over a set of derivation trees, and using the Parikh mappings obtained from the set of terminal derivation trees and the possible enlargement of them during derivation. Moreover, the usefulness of the theorem will be lightly addressed.

1 Introduction

This chapter will explain Parikh's theorem [Par66] and provide a proof for it. Parikh's theorem is important since it shows that a context–free language over a singleton alphabet must be regular [RE97]. The theorem is also practical to prove whether a language is context-free or not [Gol77]. Moreover, it can be used quite easily to prove whether the language contains a string with the same number of two given terminals. An example of how the theorem can be used to show that property will be presented together with the theorem itself. The biggest implication of the theorem though, is that it shows that if the order of the symbols is ignored, then it is impossible to distinguish between a regular set and a context-free language. This in turn implicates that context-free languages can have a richer structure than those obtained from regular sets.

It should be noted that there are other ways in which Parikh's theorem can be proved than using induction over derivation trees. One such technique is presented in [Gol77] by using a strengthened type of the pumping lemma.

The chapter is structured as follows. The following section presents basic definitions needed for the definition of Parikh's theorem. Next, the theorem itself is presented and briefly discussed. The theorem section is followed by a proof divided into three steps. The chapter is concluded with a discussion of the theorem, its proof and usefulness.

2 Basic definitions

Before the discussion of the Parikh's theorem, some definitions are made to facilitate the upcoming discussion.

Definition 2.1 (Linear subset) A *linear subset*, M, of $\mathbb{N}^n$ is given by tuples $t_0, \ldots, t_m \in \mathbb{N}^n$, where $m \in \mathbb{N}$ and $n \in \mathbb{N}^+$.

$$\begin{aligned} M &= \{t_0 + l_1 t_1 + \ldots + l_m t_m \,|\, l_1, \ldots, l_m \in \mathbb{N}\} \\ &= t_0 + \{t_1, \ldots, t_m\}^\star \end{aligned}$$

That is, M is a linear subset, which is constructed by taking a basic tuple and adding tuples to it from a finite set an arbitrary number of times.

The next definition provides a notion for the union of several linear subsets.

Definition 2.2 (Semilinear subset) A *semilinear subset*, M', of $\mathbb{N}^n$, is a union of finitely many linear subsets $M_1, \ldots, M_k$, where $k \in \mathbb{N}^+$. In other words:

$$M' \quad = \quad M_1 \cup \ldots \cup M_k$$

Example 2.3 (Linear and semilinear sets) Consider the following second order linear subsets, and their union:

$$\begin{array}{lcl} M_1 & = & (1,2) + \{(3,5),(7,11)\}^\star \\ M_2 & = & (1,1) + \{(2,3),(5,7)\}^\star \\ M' & = & M_1 \cup M_2 \end{array}$$

By definition, M' is a semilinear subset.

Definition 2.4 (The Parikh mapping) Let $\Sigma = \{a_1, \ldots, a_n\}$, where $n \in \mathbb{N}^+$ and the order of $a_1, \ldots, a_n$ is arbitrary, but fixed (i.e. one order is chosen, but which one does not matter). For $w \in \Sigma^*$, the *Parikh image* is

$$\Psi(w) \quad = \quad (m_1, \ldots, m_n)$$

Each number m_i is the number of occurrences of a_i in w.

When referring to the Parikh mapping for a language, this should be taken to mean the mapping applied to all the words in the language. This idea is expressed in the next definition.

Definition 2.5 (The Parikh mapping on a language) For a language $L \subseteq \Sigma^\star$, let $\Psi(L) = \{\Psi(w) \,|\, w \in L\}$.

Example 2.6 (Parikh mapping of words from a context-free grammar) Consider the context-free grammar $G = (N, \Sigma, R, S)$ with rules:

$$\begin{array}{lcl} S & \rightarrow & ASB \;\mid\; BSA \;\mid\; ab \\ A & \rightarrow & a \\ B & \rightarrow & b \end{array}$$

This means that the language generated by G contains strings where a's and b's can occur intermixed. But, as is clear from the language definition, the same number of a's and b's will always be present. Moreover, there will always be at least one of each letters.

Now, let a_1 and a_2 from Definition 2.4 be a and b respectively. Consider the words $w_1 = aaaabbbb \in L(G)$ and $w_2 = babababa \in L(G)$. Both of these words have the same Parikh mapping; $\Psi(w_i) = (4,4)$, where $1 \leq i \leq 2$.

It is interesting to note that most information embedded in a word generated with a context-free grammar is thrown away by the Parikh mapping.

To clearly illustrate the relationship between the first definitions of this section and the Parikh mapping, the following example shows how the Parikh mapping of a whole language can be specified.

Example 2.7 (Parikh mapping of a context-free language) Assume that the language being considered is generated from the grammar specified in example 2.6.

It is clear from the formulation of the grammar that the smallest word that can be generated using the grammar from example 2.6 is ab. Each time a word is increased in size, during its generation, exactly one a and one b will be added. Hence, the Parikh mapping for the grammar is:

$$\Psi(G) = \{(1,1),(2,2),\ldots,(n,n)\}, \text{where } n \in \mathbb{N}^+$$

3 Parikh's theorem

In this section Parikh's theorem is discussed; the proof of the theorem follows in the next section.

Theorem 3.1 (Parikh's theorem) For every context-free language L, $\Psi(L)$ is effectively semilinear. The tuples specifying $\Psi(L)$ can be constructed effectively from a context-free grammar generating L (i.e. there is an algorithm to perform the work to produce the tuples specifying $\Psi(L)$; Lemma 4.7 can be used directly to produce the tuples).

Since the theorem states that there is an effective way to specify $\Psi(L)$, it is for example possible to decide whether a given language contains a certain type of string. This is illustrated in the following example.

Example 3.2 (Existence of a certain string in a context-free language) For context-free grammars, G, it is decidable whether the languages $L(G)$ contains a string with the same number of a's and b's, by using the following construction, which effectively describes the Parikh image of the language.

By Parikh's theorem, we can build a finite automaton, A, such that $\Psi(L(G)) = \Psi(L(A))$. Let $|w_a|$ refer to the number of a's in a word w in some language, and let G_0 be a context-free grammar generating the language

$$L(G_0) = \{w \in \{a,b\} \mid |w|_a = |w|_b\}$$

Since $L(A) \cap L(G_0)$ is effectively context-free, and a context-free language can be checked for emptiness [Sud98], this can prove whether or not such a string is present. For example, this can be done by checking if the set of reachable symbols is empty for the grammar generating $L(A) \cap L(G_0)$, which has been proven to be exactly $L(G_0)$ [Sud98].

4 Proving Parikh's theorem

The proof of Parikh's theorem is divided into three parts; two special techniques are used to divide the problem of proving the theorem into parts that are more readily provable. Hence, this section will first introduce those techniques, with their associated proofs, and then use the results to prove Parikh's theorem itself.

4.1 Techniques, first part (language)

Given a context-free grammar $G = (N, \Sigma, R, S)$, rather than considering $L(G)$, look at the language $L^{\sim}(G)$ containing only the strings generated by derivation trees in which each nonterminal in N occurs.

Since Parikh's theorem states that the language $L(G)$ is semilinear, the same must hold for the language $L^{\sim}(G)$ for it be useful in any proof that should hold for $L(G)$. Hence, it is proved that $L^{\sim}(G)$ is semilinear.

Lemma 4.1 If $\Psi(L^{\sim}(G))$ is semilinear for all context-free languages, then $\Psi(L(G))$ is semilinear.

Proof Construct all grammars $G_1, \ldots, G_k$, where $k \in \mathbb{N}^+$, obtained from G by deleting nonterminals. Then $L(G) = L^{\sim}(G_1) \cup \ldots \cup L^{\sim}(G_k)$, and thus $\Psi(L(G)) = \Psi(L^{\sim}(G_1)) \cup \ldots \cup \Psi(L^{\sim}(G_k))$ is semilinear by Definition 2.2. ■

In the proof, note that the number of languages, k, is limited by the number of nonterminals in the grammar G to $k = 2^{|N|} - 1$. The subtraction of one is due to the fact that the nonterminal S must be kept.

4.2 Techniques, second part (definitions)

This part of the proof focuses on the derivation trees that are used in generation of the language $L^{\sim}(G)$. Three special kinds of trees, with restrictive properties, are defined.

Definition 4.2 (Set of terminal derivation trees with root S) Let T be the set of all terminal derivation trees with the root S that satisfies:

1. All nonterminals occur in the tree .
2. No nonterminal occurs more than $k = |N|$ times on every path.

The members of the set T corresponds to the derivation trees for the language $L^{\sim}(G)$. The set T is used in the proof to define the Parikh mapping of $L^{\sim}(G)$ as a union of individual mappings.

The next set defined is a variation of the set T, which is important since it allows for expansion of the tree.

Definition 4.3 (The set of all terminal trees) Let $\widetilde{T}$ be the set of all terminal derivation trees satisfying condition 1 in Definition 4.2 (cf. Figure 2).

The final set corresponds to the rules in a context-free grammar that makes a string larger during its derivation, that is a rule on the form $A \to uAv$, where

$u, v \in \Sigma$. Note that it can be used to increase the size of the trees in set $\widetilde{T}$ by replacing a nonterminal symbol A on a path in a tree $\widetilde{t} \in \widetilde{T}$ by a tree $t \in I$, where the tree t has the same nonterminal A as root and as a leaf, as is defined in the following definition.

Definition 4.4 (Set of intermediate trees) Let I be the set of all derivation trees with root $A \in N$, containing exactly one nonterminal leaf also labeled A. In addition, the trees in I are required to satisfy condition 2 from Definition 4.2.

Observation 4.5 Note that the constant k from Definition 4.2 is at most $k^2 = |N| \times c$, where c is the number of times the nonterminals occur on a path. Hence, both the trees in T and I are of finite height since the maximum length of any path has an upper bound.

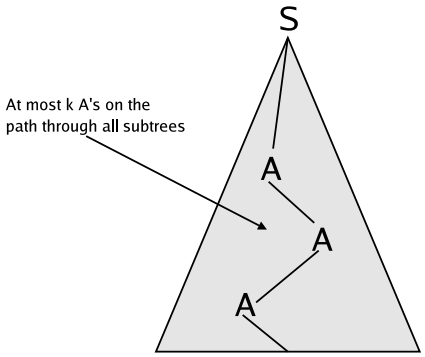

Figure 1: The special tree type T described in Definition 4.2. On any path, a nonterminal occurs at most k times. However, each nonterminal must occur somewhere in the tree

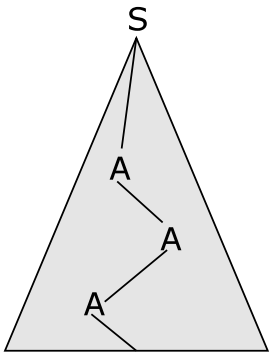

Figure 2: The special tree type $\widetilde{T}$. Contrary to the tree T, the number of any nonterminal is not restricted on the path

4.3 Proof of Parikh's theorem

Using the techniques from the previous sections, it is now possible to prove theorem 3.1.

Definition 4.6 (The set of yielded strings) Let $w_1, \ldots, w_q$, where $q \in \mathbb{N}^+$, be the set of yielded string from the trees in the set T (cf. Definition 4.2), and let W be the set of all string uv such that uAv is the result of some tree in I for some nonterminal $A \in N$.

A closer look at Definition 4.6 makes it apparent that the members of the set W represents possible subtrees that may be used to make a derivation tree larger. In other words, it corresponds to rules in the specification of a context-free grammar that makes a string longer by introducing a nonterminal between two terminals.

Lemma 4.7 The following equivalence holds:

$$\Psi\left(L^{\sim}\left(G\right)\right) = \left.\begin{array}{l} \Psi\left(w_1\right) + \Psi\left(W\right)^{\star} \\ \cup \\ \vdots \\ \cup \\ \Psi\left(w_q\right) + \Psi\left(W\right)^{\star} \end{array}\right\} \Phi$$

Observation 4.8 The empty string can be discarded from the set W in Lemma 4.7 without imposing any restrictions on the Parikh mapping Ψ since it adds nothing to a summation. Moreover, a finite number of additions of the mapping $\Psi\left(W\right)$ to some mapping $\Psi\left(w_q\right)$ corresponds to replacing some nonterminal A in a tree from $\widetilde{T}$ with a tree from the set I (cf. Figure 2), thus making tree larger. This can be seen directly by looking closely at Definition 4.6.

Proof The proof of Lemma 4.7 will be by induction.

First direction: $\Phi \subseteq L^{\sim}\left(G\right)$, where Φ is the union in Lemma 4.7.

Phrased in a another manner, the first direction of the proof handles the case $m = \left(m_1, \ldots, m_n\right) \in \Phi \rightarrow m \in \Psi\left(L^{\sim}\left(G\right)\right)$.

Induction basis: Let $m = \Psi\left(w_i\right)$ for some $i \in \mathbb{N}^+$, then $w_i \in L^{\sim}\left(G\right)$ and thus $\Psi\left(w_i\right) \in \Psi\left(L^{\sim}\left(G\right)\right)$. By the definition of $\widetilde{T}$ (cf. Definition 4.3), w_i corresponds to the strings forming the language $L^{\sim}\left(G\right)$.

Induction hypothesis: The statement holds for m'; $m' \in \Psi\left(L^{\sim}\left(G\right)\right)$. Since m' is a tuple from the union Ψ (cf. Lemma 4.7), it follows that m' is bounded by a finite number of additions of the mapping $\Psi\left(W\right)$ to some $\Psi\left(w_q\right)$.

Inductive step: Let $m = m' + \Psi\left(u\right)$, for some $u \in W$.

There is some tree $t \in \widetilde{T}$ with the result w, such that $\Psi\left(w\right) = m'$. Further, there is a derivation tree $t' \in I$ with the result $u_0 A v_0$ such that $u = u_0 v_0$ (i.e. the u described in definition 4.6).

Now, construct the derivation tree obtained by replacing any A–labeled node p in t with the tree obtained from t' by replacing its A–leaf with the subtree of t

rooted at p (cf. Figure 3 for an illustration of this process). The resulting tree belongs to $\widetilde{T}$ and its result, z, satisfies $\Psi(z) = \Psi(w) + \Psi(u) = m$.

Since the resulting sum exactly corresponds to the sum presented at the beginning of the inductive step, the inclusion $\Phi \subseteq \Psi(L^{\sim}(G))$ is proved.

Second direction: $L^{\sim}(G) \subseteq \Phi$, where Φ is the union in Lemma 4.7.

Phrased differently, the second direction of the proof shows that if $t \in \widetilde{T}$, with result w, then $\Psi(w) \in \Phi$.

Induction basis: If $t \in T$, then $w = w_i$ for some $i \in \mathbb{N}^+$, and thus $\Psi(w) \in \Phi$.

Induction hypothesis: The statement holds for all trees in $\widetilde{T}$ that are smaller than t.

Inductive step: Let $p_1, \ldots, p_n$, where $n \in \mathbb{N}^+$, be nodes on some path and let the index indicates each node's relative position on that path. Furthermore, let all nodes be labeled with the same nonterminal A, such that all *proper* subtrees of the tree rooted at p_1 satisfies condition 2 from Definition 4.2.

Let t_i be the tree obtained by removing nodes in t at node p_i and p_{i+1}. Conversely, the tree $\bar{t}_i$ is the subtree obtained when removing the nodes between p_i and p_{i+1} from t. Then $t_i \in I$ and $\Psi(w) = \Psi(\bar{u}_i) + \Psi(u_i)$, where $\bar{u}_i$ and u_i are the results of $\bar{t}_i$ and t_i respectively (cf. Figure 4 for an illustration of this process).

Now it remains to be shown that $\bar{t}_i$ can be chosen in such a way that it belongs to $\widetilde{T}$.

Let $N \setminus \{A\} = \{B_1, \ldots, B_{n-1}\}$, where $n \in \mathbb{N}^+$. Then $\bar{t}_i \in \widetilde{T}$ if t_i contains all B_j's in (the whole tree) t, for some $j \notin \{1, \ldots, n-1\}$. (Again, cf. Figure 4; the nodes that should be present in t_i corresponds to the A that is removed). But since there are n choices for i, there must be at least one $i \in \{0, \ldots, n-1\}$ for which this does not happen. Hence, $\bar{t}_i \in \widetilde{T}$, and the inductive hypothesis applies. ■

The proof shows that the Parikh mapping of a context-free language is equal to that which is obtained from the terminal derivation trees, which have the shortest possible paths (i.e. all subtrees that can be pumped occur at most one time), and their results and adding the contribution of subtrees that can be pumped an arbitrary number of times to the corresponding linear subset. This is exactly what Lemma 4.7 states.

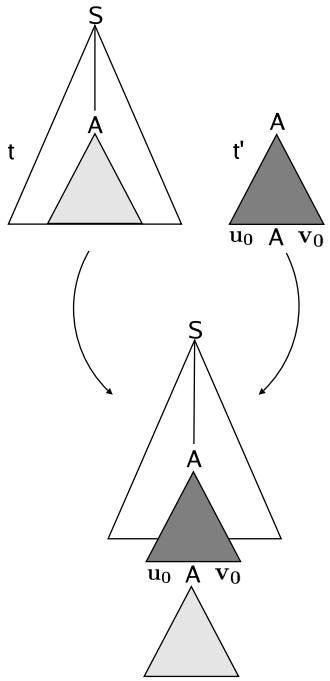

Figure 3: A tree from the set I is used to increase the size of a tree from the set $\widetilde{T}$. Hence, the result of the tree is increased with the strings $u_0, v_0 \in \Sigma^*$

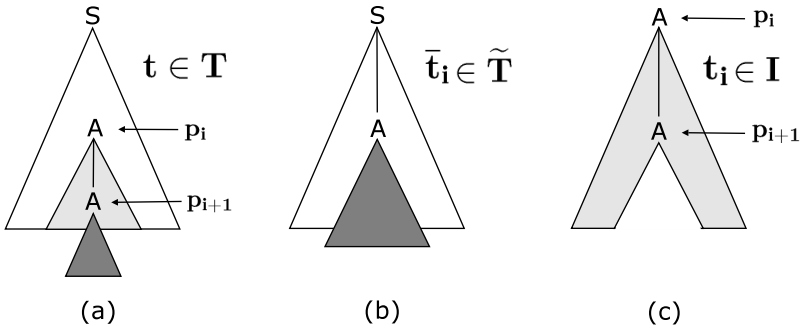

Figure 4: The fundamental parts of a tree are shown (a): trees from the set T, I and $\widetilde{T}$ are illustrated. In (b), the subtree belonging to the set I is removed from (a). In (c), the removed tree is illustrated on its own

4.4 Discussion

In this chapter, Parikh's theorem [Par66] was presented. As is hinted in example 3.2, the theorem is excellent for showing whether or not a string with the same number of some terminals is present in a given language. This is so simple since all information preserved by the Parikh mapping 2.4 is the count of each symbol in a given word. Moreover, it is also very useful for proving whether a given language is context-free or not. However, it is remarkable that Parikh came up with the idea of the proof, since the exact conditions controlling the structure of the trees presented in definition 4.2 through 4.4 are nontrivial, in the sense that it is not obvious that those conditions must hold.

References

[Gol77] J. Goldstine. A simplified proof of parikh's theorem. *Discrete Mathematics*, 19:235–239, 1977.

[Par66] Rohit J. Parikh. On context–free languages. *Journal of the Association for Computing Machinery*, 13(4):570–581, 1966.

[RE97] G. Rozenberg and A. Salomaa (Eds.). *Handbook of Formal Languages, vol. 1.* Springer, 1997.

[Sud98] Thomas A. Sudkamp. *Languages and Machines, 2nd Ed.* Addison Wesley, 1998.