# Parikh's Theorem

Giuliano Losa

November 15, 2010

# Words and Languages

- A word $w$ is a sequence of symbols in some alphabet $\Sigma$.
- A language is a set of words.
- For example, $\{abb, accbba, aa, bab\}$ is a language over $\Sigma = \{a, b, c\}$.
- A language can have infinitely many words.

# Parikh Image

- If $w$ is a word over some $\Sigma$, we denote by $\Pi_\Sigma(w)$ the Parikh image of $w$ over alphabet $\Sigma$.
- $\Pi_\Sigma(w)$ maps a character in $\Sigma$ to its number of occurences in $w$.
- The Parikh image of a language $L$ over $\Sigma$ is $\{\Pi_\Sigma(w) | w \in L\}$. It is denoted by $\Pi_\Sigma(L)$.

## Examples

- $\Pi_{\{a,b,c\}}(bccba) = (1,2,2)$
  where $(1,2,2)$ stands for $\{(a,1),(b,2),(c,2)\}$
- $\Pi_{\{a,b,c\}}(cabaaabb) = (4,3,1)$

# Letter-equivalence

- Two words $w_1$ and $w_2$ over $\Sigma$ are letter-equivalent iff $\Pi_\Sigma(w_1) = \Pi_\Sigma(w_2)$.
- Two languages $L_1$ and $L_2$ over $\Sigma$ are letter-equivalent iff $\Pi_\Sigma(L_1) = \Pi_\Sigma(L_2)$. It is denoted by $L_1 =_{\Pi_\Sigma} L_2$.
- We also define $L_1 \subseteq_{\Pi_\Sigma} L_2$ with the obvious meaning.

## Examples

- *abbcaa* is letter-equivalent to *baacab*.
- *cbaccba* is letter-equivalent to *ccbaabc*.
- $\{a^n b^n | n \in \mathbb{N}\}$ is letter-equivalent to $\{(ab)^n | n \in \mathbb{N}\}$.

## Remark

The alphabet $\Sigma$ will be implicit in the rest of the talk.

# Regular Languages

A language is regular if it is accepted by some finite automaton.
I assume you are familiar with regular languages. . .

# Context-free Grammars

$G = (V, T, P, S)$

- $V$ is a set of variables, denoted $A_1, A_2, A_3 \ldots$
- $T$ is a set of terminal symbols, denoted $a, b, c \ldots$
- $S$ is the start variable.
- $P$ is a set of productions of the form $A_i \to \alpha$ where $\alpha \in (V \cup T)^*$

We suppose that $S \to A_1$ is the only production involving $S$.

### Example

Let $V = \{A_1, A_2\}$, $T = \{a, b, c\}$, $P = \{A_1 \to A_1 A_2 | a, A_2 \to A_2 A_2 | b\}$

### Remark

We will now always call our grammar G.

# Steps and Derivations

### Steps

Given $\alpha, \beta \in (V \cup T)^*$, $\beta$ is derivable in one step from $\alpha$, denoted $\alpha \Rightarrow \beta$, if there exists a production $A \to \gamma$ and $\alpha_1, \alpha_2 \in (V \cup T)^*$ such that:

$$\alpha = \alpha_1 A \alpha_2 \qquad \text{and} \qquad \beta = \alpha_1 \gamma \alpha_2$$

### Example

Let $V = \{A_1, A_2\}$, $T = \{a, b, c\}$, $P = \{A_1 \to A_1A_2|a, A_2 \to A_2A_2|b\}$.
$A_1aA_2b \Rightarrow A_1aA_2A_2b$ because $A_2 \to A_2A_2$ is a production.
$A_1aA_2b \Rightarrow A_1abb$ because $A_2 \to b$ is a production.

# Derivations and Language

- A derivation is a sequence of steps.
- We denote by $\Rightarrow^*$ the reflexive transitive closure of $\Rightarrow$.

### Example

Let $V = \{A_1, A_2\}$, $T = \{a, b, c\}$, $P = \{A_1 \rightarrow A_1A_2 | a, A_2 \rightarrow A_2A_2 | b\}$.
$A_1 \Rightarrow A_1A_2 \Rightarrow aA_2 \Rightarrow aA_2A_2 \Rightarrow aA_2b \Rightarrow abb$ is a derivation.
$S \Rightarrow^* A_1A_2bbA_2b$
$S \Rightarrow^* abbbb$

### Language of grammar G

The language of G, denoted $L(G)$, is the set $\{x \in T^* | S \Rightarrow^* x\}$

### Example

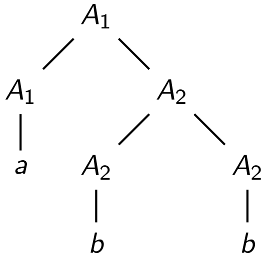
$abbbb$ is in $L(G)$ because $S \Rightarrow^* abbbb$ and $abbbb \in \{a, b, c\}^*$.

# Context-free Language

A language $L$ is context-free if there is a context-free grammar G such that $L = L(G)$.

# Parse Trees: Example

Consider derivation $A_1 \Rightarrow A_1 A_2 \Rightarrow aA_2 \Rightarrow aA_2 A_2 \Rightarrow aA_2 b \Rightarrow abb$.
It has the following parse tree $t$:




## Yield of a parse tree

The yield of $t$, denoted $Y(t)$, is such that $Y(t) = abb$.
The yield of a set of trees $T$, denoted $Y(T)$, is the set of yields of trees in $T$.

# Parse Trees are Nice

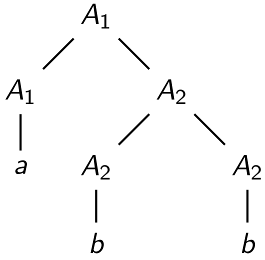
I think trees are way easier to manipulate and reason about than derivations.

In our proofs, we'll try to reduce our goals to goals about trees. . .

For example, we can define the language of G in term of trees:
Let $T_G$ be the set of parse trees of G, then $L(G) = Y(T_G)$.

# Parse Trees and Derivations

The following parse tree produces derivation
$A_1 \Rightarrow A_1A_2 \Rightarrow aA_2 \Rightarrow aA_2A_2 \Rightarrow aA_2b \Rightarrow abb$.
Can it produce others?




Yes! For example:

$$A_1 \Rightarrow A_1A_2 \Rightarrow A_1A_2A_2 \Rightarrow A_1A_2b \Rightarrow aA_2b \Rightarrow abb$$

# Semilinear Sets

## Linear Sets

A subset $S$ of $\mathbb{N}^k$ is called linear if it is of the form

$$\{u_0 + k_1 u_1 + k_2 u_2 + \cdots + k_n u_n\}$$

Where $u_i$ is a vector of $\mathbb{N}^k$ and $k_i \in \mathbb{N}$.

We write $S = u_0 + < u_1, u_2, \ldots, u_n >$

## Example

$$(0,0,1) + < (1,0,0), (1,1,1), (0,1,1) >$$
$$=$$
$$\{(0,0,1) + k_1(1,0,0) + k_2(1,1,1) + k_3(0,1,1) | k_1, k_2, k_3 \in \mathbb{N}\}$$

# Parikh's theorem

## Two formulations

- Any context-free language is letter-equivalent to a regular language.
- The Parikh image of any context-free language is a semilinear set.

## They are equivalent

That's because given a semilinear set $S$, we can easily build a regular language $L$ such that $\Pi(L) = S$.

# Constructive Proof of Formulation 1

## Formulation 1:

Any context-free language is letter-equivalent to a regular language.

## Proof

Given grammar $G$, we will build a finite automaton $M$ such that $L(G) =_{\Pi} L(M)$.

Presentation adapted from J.Esparza, P.Ganty, S.Kiefer, M.Luttenberger: Parikh's Theorem: A simple and direct construction.

# Chomsky Normal Form

## The construction needs a grammar in Chomsky Normal Form

All productions are of the form $A \rightarrow BC$ or $A \rightarrow a$, where $a \neq \epsilon$

### Example

Grammar for balanced parens:

$$A_1 \rightarrow A_1 A_1 | (A_1) | \epsilon$$

In Chomsky Normal Form (without $\epsilon$):

$$A_1 \rightarrow A_2 A_3 | A_1 A_1 | A_4 A_3$$
$$A_4 \rightarrow A_2 A_1$$
$$A_2 \rightarrow ($$
$$A_3 \rightarrow )$$

CNF is nice because of the shape of its parse trees

# The Construction: Preliminary Definitions

## Projections and Parikh image

We consider a grammar $G = (V, T, P, S)$ with axiom $S = A_1$.
For $\alpha \in (V \cup T)^*$, let $\alpha_{/V}$ (resp $\alpha_{/T}$) denote the projection of $\alpha$ onto $V$ (resp. $T$).
Let $\Pi_V(\alpha) = \Pi(\alpha_{/V})$ and $\Pi_T(\alpha) = \Pi(\alpha_{/T})$.

## Examples

Let $V = \{A_1, A_2\}$, $T = \{a_1, a_2, a_3\}$, $P = \{A_1 \to A_1 A_2 | a, A_2 \to A_2 A_2 | b\}$
Let $\alpha = a_1 A_2 a_2 A_1 A_1$
Then,

$\alpha_{/T} = a_1 a_2$ and $\alpha_{/V} = A_2 A_1 A_1$
$\Pi_V(\alpha) = (2, 1)$ and $\Pi_T(\alpha) = (1, 1, 0)$

# Transitions

Recall that steps are such that:
$\alpha = \alpha_1 A \alpha_2 \qquad \Rightarrow \qquad \beta = \alpha_1 \gamma \alpha_2$ if $A \to \gamma$ is a production.
The transition associated to a step is the triple
$t(\alpha \Rightarrow \beta) = (\Pi_V(\alpha), \gamma_{/T}, \Pi_V(\beta))$.

## Example

$t(A_2 a A_1 \Rightarrow A_2 A_2 a A_1) = ((1,1), \epsilon, (1,2))$
$t(A_2 A_1 b A_1 \Rightarrow b A_1 b A_1) = ((2,1), b, (2,0))$

# The Construction: k-Parikh automaton

For $k \in \mathbb{N}$, the k-Parikh automaton of $G$ is the NFA $M_G^k = (Q, T, \delta, q_0, \{q_f\})$ defined as follows:

- $Q = \{(x_1, \ldots, x_n) \in \mathbb{N}^n | \sum_{i=1}^n x_i \leq k\}$
- $\delta = \{t(\alpha \Rightarrow \beta) | \Pi_V(\alpha), \Pi_V(\beta) \in Q\}$
- $q_0 = \Pi_V(S) = (1, 0, \ldots, 0)$
- $q_f = \Pi_V(\epsilon) = (0, 0, \ldots, 0)$.

Recall:
$t(A_2 a A_1 \Rightarrow A_2 a_1 A_2 A_3) = ((1, 1, 0), \epsilon, (0, 2, 1))$

Example on the board
$V = \{A_1, A_2\}$, $T = \{a_1, a_2, a_3\}$, $P = \{A_1 \to A_1 A_2 | a, A_2 \to A_2 A_2 | b\}$

# Theorem

Let $n = |V|$, the number of variables in the grammar $G$.

$L(G)$ and $L(M_G^{n+1})$ have the same Parikh image.
In other words, $L(G) =_\Pi L(M_G^{n+1})$

$|Q| = O(4^n)$. Is the construction space efficient?

# Where Are We?

We want to prove formulation 1 of Parikh's theorem by building a finite automaton M such that $L(G) =_{\Pi} L(M)$.

- From $G$ in Chomsky Normal Form, we define $M_G^k$.
- We choose $M$ to be the (n+1)-Parikh automaton $M_G^{n+1}$.
- We'd like to prove that $L(M_G^{n+1}) =_{\Pi} L(G)$.

# Observation

- $Q = \{(x_1, \ldots, x_n) \in \mathbb{N}^n | \sum_{i=1}^n x_i \leq k\}$
- $\delta = \{t(\alpha \Rightarrow \beta) | \Pi_V(\alpha), \Pi_V(\beta) \in Q\}$
- $q_0 = \Pi_V(S) = (1, 0, \ldots, 0)$
- $q_f = \Pi_V(\epsilon) = (0, 0, \ldots, 0)$.

- $\delta$ suggests that states of the automaton be interpreted as set of words over $(V \cup T)^*$.
- By definition of $Q$, any word $w$ associated with a state of the k-Parikh automaton is such that $\Pi_V(w)$ is of length at most $k$.
- By definition of $\delta$, a run of the k-Parikh automaton corresponds to a derivation such that at each step the word obtained $w$ is such that $\Pi_V(w)$ has length at most $k$.

# More Formally: Index of a derivation

### Definitions

A derivation $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m$ has index $k$ if for every $\alpha_i$, $\alpha_{i/V}$ has length at most $k$.

### Example

$A_1 \Rightarrow A_1 A_2 b \Rightarrow A_1 bb \Rightarrow A_1 A_2 bb \Rightarrow aA_2 bb \Rightarrow abbb$ has index 2.

### $L_k(G)$

Set set of words derivable through derivations of index at most $k$ is denoted $L_k(G)$.

# Let Us Restate Our Observation

$$L(M_G^k) =_\Pi L_k(G)$$

Since $L_k(G) \subseteq L(G)$ (why?), we already have

$$\forall k \geq 1. L(M_G^k) \subseteq_\Pi L(G)$$

Hence one inclusion holds:

$$L(M_G^k) \subseteq_\Pi L(G)$$

Moreover the other inclusion, $L(G) \subseteq_\Pi L(M_G^{n+1})$, reduces to

$$L(G) \subseteq_\Pi L_{n+1}(G)$$

# Where Are We?

We want to prove formulation 1 of Parikh's theorem by building a finite automaton M such that $L(G) =_\Pi L(M)$.

- We choose $M$ to be the (n+1)-Parikh automaton $M_G^{n+1}$.
- We'd like to prove that $L(M_G^{n+1}) =_\Pi L(G)$.
- We observed that $M_G^k$'s runs correspond exactly to all derivations of index up to $k$: $L(M_G^k) =_\Pi L_k(G)$
- Our observation implies that $L(M_G^k) \subseteq_\Pi L(G)$.
- Hence it remains to prove $L(G) \subseteq_\Pi L(M_G^{n+1})$, which by our observation reduces to $L(G) \subseteq_\Pi L_{n+1}(G)$.

# Current Goal

We would like to prove:

$$L(G) \subseteq_{\Pi} L_{n+1}(G)$$

$L_{n+1}(G)$ is the language of words generated by derivations of index at most $n+1$. $n$ is the number of variables in $G$.

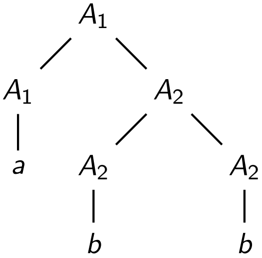
Let us work with parse trees and generalize a bit: how can we characterize parse trees of words in $L_k(G)$ ?

# Parse Trees and Index of Derivations

The following parse tree $t$ produces (among others) the following two derivations:

$A_1 \Rightarrow A_1A_2 \Rightarrow aA_2 \Rightarrow aA_2A_2 \Rightarrow aA_2b \Rightarrow abb$ of index 2 and

$A_1 \Rightarrow A_1A_2 \Rightarrow A_1A_2A_2 \Rightarrow A_1A_2b \Rightarrow aA_2b \Rightarrow abb$ of index 3.




Even though derivation 2 has index 3, $Y(t) = abb$ is in $L_2(G)$ because derivation 1 has index 2.

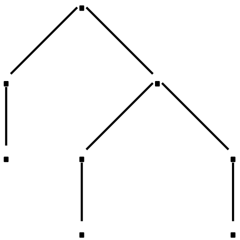
Hence to determine the minimum $k$ such that $Y(t) \in L_k(G)$, we need to find the derivation in $t$ of minimal index.

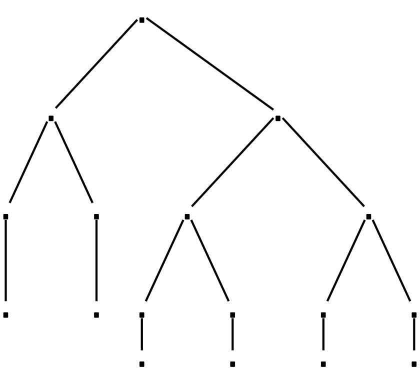
# Parse trees and $L_k(G)$

A word $w = Y(t)$ is in $L_k(G)$ if we can find in tree $t$ a derivation of index smaller or equal to $k$.

### Examples

Find the derivation with minimal index among the possible derivations. Remember that $G$ is in Chomsky Normal Form.





# Parse Trees and Dimension

## Dimension of a parse tree

The dimension of a parse tree $t$ is inductively defined as follows:
If $t$ is of the form $A \to a$, then $d(t) = 0$.
Otherwise,

$$d(t) = \begin{cases} d(t_1) + 1 & \text{if } d(t_1) = d(t_2) \\ max(d(t_1), d(t_2)) & \text{if } d(t_1) \neq d(t_2) \end{cases}$$

Where $t_1$ and $t_2$ are the left and right subtrees of $t$.

## Property

A parse tree of dimension $k$ contains a derivation of index $k+1$.

## Sets of parse trees

We denote by $T_G^k$ the set of parse trees of $G$ of dimension $k$.
We denote by $T_G$ the set of parse trees of $G$.

# Parse Trees of Restricted Dimension and $L_{n+1}(G)$

We crafted the definition of dimension so that parse trees of dimension $k$ contain at least one derivation of index $k+1$.

Hence we have:

$$\forall k \geq 0. Y(T_G^k) \subseteq L_{k+1}(G)$$

## Lemma

$$L_{k+1}(G) = \bigcup_{i=0}^{k} Y(T_G^i)$$

This will allow use to reduce our current goal to something about parse trees...

# Consequence for our goal

Remember: we want to prove that $L(G) \subseteq_{\Pi} L(M_G^{n+1})$.
By the lemma, it is equivalent to $L(G) \subseteq_{\Pi} \bigcup_{i=0}^{n} Y(T_G^i)$.
Remember: by definition of $T_G$, $L(G) = Y(T_G)$.
Hence our new goal is

$$Y(T_G) \subseteq_{\Pi} \bigcup_{i=0}^{n} Y(T_G^i)$$

## How to prove it?

We need to show that for any tree $t \in T_G$ there is a tree $t' \in T_G$ with maximum dimension $n$ such that $Y(t) =_{\Pi} Y(t')$

# Where Are We?

We want to prove formulation 1 of Parikh's theorem by building a finite automaton M such that $L(G) =_\Pi L(M)$.

- We choose $M$ to be the (n+1)-Parikh automaton $M_G^{n+1}$.
- We'd like to prove that $L(M_G^{n+1}) =_\Pi L(G)$.
- We showed that $L(M_G^k) \subseteq_\Pi L(G)$.
- We reduced $L(G) \subseteq_\Pi L(M_G^{n+1})$ to $L(G) \subseteq_\Pi L_{n+1}(G)$.
- We related parse trees of $G$ and $L_{n+1}(G)$ by $L_{n+1} = \bigcup_{i=0}^{n} Y(T_G^i)$
- Hence we need to show that $Y(T_G) \subseteq_\Pi \bigcup_{i=0}^{n} Y(T_G^i)$: for any tree $t \in T_G$ there is a tree $t' \in T_G$ with maximum dimension $n$ such that $Y(t) =_\Pi Y(t')$.

# Final Proof Step

### Goal

We need to show that $Y(T_G) \subseteq_\Pi \bigcup_{i=0}^{n} Y(T_G^i)$, i.e. that for any tree $t \in T_G$ there is a tree $t' \in T_G$ with maximum dimension $n$ such that $Y(t) =_\Pi Y(t')$

### Proof by induction on the board

# Recap

We want to prove formulation 1 of Parikh's theorem by building a finite automaton M such that $L(G) =_\Pi L(M)$.

- We choose $M$ to be the (n+1)-Parikh automaton $M_G^{n+1}$.
- We'd like to prove that $L(M_G^{n+1}) =_\Pi L(G)$.
- We observed that $M_G^k$'s runs correspond exactly to all derivations of index up to $k$: $L(M_G^k) =_\Pi L_k(G)$
- Our observation implies that $L(M_G^k) \subseteq_\Pi L(G)$.
- Hence it remains to prove $L(G) \subseteq_\Pi L(M_G^{n+1})$, which by our observation reduces to $L(G) \subseteq_\Pi L_{n+1}(G)$.
- We related parse trees of $G$ and $L_{n+1}(G)$ by $L_{n+1} = \bigcup_{i=0}^{n} Y(T_G^i)$
- We have shown that for any tree $t \in T_G$ there is a tree $t' \in T_G$ with maximum dimension $n$ such that $Y(t) =_\Pi Y(t')$. Hence $Y(T_G) \subseteq_\Pi \bigcup_{i=0}^{n} Y(T_G^i)$
- QED

# What About Semilinear Sets?

Sorry, not enough time left. . .
A nice proof appears in:

Dexter C. Kozen, "Automata and Computability", chapter H.

Thanks for listening!