Theory of Computation Slides based on Michael Sipser's Textbook

Moses A. Boudourides[1]

Visiting Associate Professor of Computer Science
Haverford College

[1] Moses.Boudourides@cs.haverford.edu

Section 2.1

Context–Free Grammars

February 15, 2022

Context–Free Grammars, I

Definition: Context-Free Grammars

A **context–free grammar** (**CFG**) is a 4–tuple $G = (V, \Sigma, S, P)$, where

- V is a finite set of **variables**, $S \in V$ is the **start variable**,
- Σ is a finite set of **terminal symbols** or **terminals** such that $V \cap \Sigma = \varnothing$, and
- P is a finite set, the elements of which are called **grammar rules** or **productions** and they are of the form

$$A \to \alpha,$$

where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$.

Context–Free Grammars, II

Notation

Given a CFG $G = (V, \Sigma, S, P)$ and two strings $\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$ and $\beta \in (V \cup \Sigma)^*$, a **derivation** from α to β, denoted as

$$\alpha \implies \beta,$$

is an one–step process to obtain β from α by using a rule in P in order to replace the single occurrence of S in α by the right–hand side of that rule. Furthermore, the notations

$$\alpha \implies^n \beta \text{ or } \alpha \implies^n_G \beta$$

refer to a sequence of (exactly) n steps of derivations and the notations

$$\alpha \implies^* \beta \text{ or } \alpha \implies^*_G \beta$$

refer to a sequence of 0 or more steps of derivations.

Context–Free Grammars, III

Definition: The Language Generated by a CFG

If $G = (V, \Sigma, S, P)$ is a CFG, the **language generated by G** is

$$L(G) = \{x \in \Sigma^* \mid S \Longrightarrow_G^* x\}.$$

A language L is a **context–free language (CFL)** if there is a CFG G with $L = L(G)$.

Notation

The symbol "$|$" inside the set of productions denotes a disjunction (or). For instance, $P = \{S \to aSb \mid ab\}$ includes

$$S \to aSb,$$
$$S \to ab.$$

Example 1

If $G = (\{S\}, \{a, b\}, S, \{S \to aSb \mid ab\})$, then $L(G) = \{a^n b^n \mid n \in \mathbb{Z}, n \geq 1\}$.

Context–Free Grammars, IV

Example 2

If G is a CFG with productions $S \to aSa \mid aBa, B \to bB \mid b$, then $L(G) = \{a^i b^j a^i \mid i, j \in \mathbb{Z}, i \geq 0, j > 0\}$.

For every nonnegative integers n, m, k,

$$S \Longrightarrow^n a^n S a^n \Longrightarrow^m a^n \left(a^m B a^m\right) a^n \Longrightarrow^k a^{n+m} b^k B a^{n+m}$$
$$\Longrightarrow a^{n+m} b^{k+1} a^{n+m}.$$

Example 3

Find the CFG generating $L = \{a^i b^{2i} \mid i \in \mathbb{Z}, i \geq 0\}$.

$$P = \{S \to aSbb \mid \varepsilon\}$$

Example 4

Find the CFG generating $L = \{x \in (a+b)^* \mid n_a(x) = n_b(x) \geq 0\}$.

$$P = \{S \to SS \mid aSb \mid bSa \mid \varepsilon\}$$

Derivation Trees, I

Definition

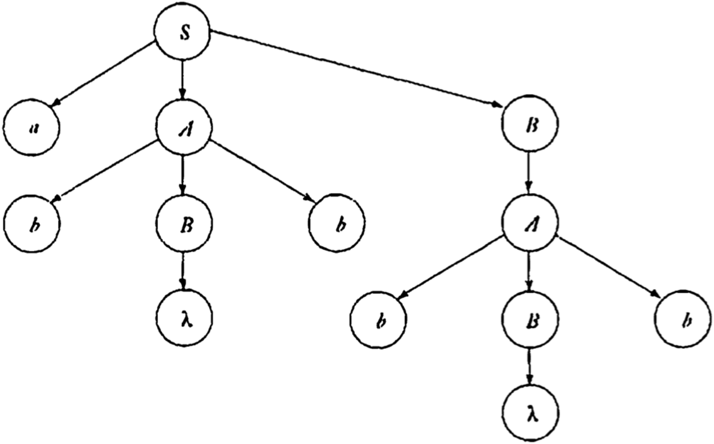
Let $G = (V, \Sigma, S, P)$ a CFG. A **derivation tree** (or **parse tree**) for G is an ordered tree such that:

- the root is labeled S;
- every leaf has a label from $\Sigma \cup \{\varepsilon\}$;
- every interior vertex has a label from V;
- if a vertex has label A, and its children are labeled (from left to right) $a_1, a_2, \ldots, a_n$, where $a_j \in V \cup \Sigma \cup \{\varepsilon\}$, for $j = 1, 2, \ldots, n$, then P contains a production of the form $P \to a_1 a_2 \cdots a_n$.

Derivation Trees, II

Example of Derivation Tree

The CFG G with productions $S \to aAB$, $A \to bBb$, $B \to A \mid \varepsilon$ has the following derivation tree (among other derivation trees):



Derivation Trees, III

Definition

The **yield** of a tree is the string of terminals (symbols) obtained by reading the leaves of the tree from left to right, omitting any ε's encountered. The precise meaning of the ordering "from left to right" is that terminals are yielded in the order they are encountered when the tree is traversed in a depth–first manner, always taking the leftmost ununexplored branch.

Example of a Yield

The yield of the derivation tree of the previous example is the string *abbbb*. Notice that the corresponding CFL is the set of all strings over $\{a, b\}$ starting with a and followed by an even positive number of b's.

Theorem

If G is a CFG, then, for every $x \in L(G)$, there exists a derivation tree of G whose yield is x. Conversely, the yield of any derivation tree is in $L(G)$.

Ambiguity

Definition

A derivation in a CFG is a **leftmost derivation** (**LMD**) if, at each step, a production is applied to the leftmost variable-occurrence in the current string. A **rightmost derivation** (**RMD**) is defined similarly.

Theorem

If G is a CFG, then, for every $x \in L(G)$, the following three statements are quivalent:

1. x has more than one derivation tree;
2. x has more than one leftmost derivation;
3. x has more than one rightmost derivation.

Definition

A CFG G is **ambiguous** if, for at least one $x \in L(G)$, x has more than one derivation tree (or, equivalently, more than one leftmost derivation).

Equivalent Context–Free Grammars, I

Definition

Let $G_1 = (V_1, \Sigma, S_1, P_1)$ and $G_2 = (V_2, \Sigma, S_2, P_2)$ be two CFGs over the same set of terminals (alphabet) Σ. The grammars G_1 and G_2 are called **equivalent CFGs** if they are generating the same language, i.e., if $L(G_1) = L(G_2)$.

Notation

Let $L_{a=b}$ denote the (nonregular) language of strings with equal number of a's and b's, i.e.,

$$L_{\sharp a=\sharp b} = L\{x \in (a+b)^* \mid n_a(x) = n_b(x) \geq 0\}.$$

Proposition

The following two CFGs $G_1 = \{\{S_1\}, \Sigma, S_1, P_1\}$ and $G_2 = \{\{S_2, A, B\}, \Sigma, S_2, P_2\}$ are equivalent, both generating the language $L_{\sharp a=\sharp b}$, when

$P_1 = \{S_1 \to S_1S_1 \mid aS_1b \mid bS_1a \mid \varepsilon\}$,

$P_2 = \{S_2 \to aB \mid bA \mid \varepsilon, A \to aS_2 \mid bAA, B \to bS_2 \mid aBB\}$.

Equivalent Context–Free Grammars, II

Proposition

Prove that $L(G_1) = L_{\sharp a=\sharp b}$.

Proposition

Prove that $L(G_1) \subseteq L_{\sharp a=\sharp b}$.

Using **induction on the number of derivations** n, we are going to show that: If, for every integer $n \geq 1$, and $x \in L(G_1)$, $S_1 \Longrightarrow^n x$, then $n_a(x) = n_b(x)$, i.e., $x \in L_{\sharp a=\sharp b}$.

Base case: True, for $n = 1$, because $S_1 \Longrightarrow \varepsilon \in L_{\sharp a=\sharp b}$.
Inductive hypothesis: Suppose that there exists integer $k \geq 1$ such that, for all nonnegative integers $i \leq k$, if $x \in L(G_1)$ such that $S_1 \Longrightarrow^i x$, then $x \in L_{\sharp a=\sharp b}$.
Inductive step: We need to show that, if $y \in L(G_1), S_1 \Longrightarrow^{k+1} y$, then $y \in L_{\sharp a=\sharp b}$. Notice that, in this case, there exists $z \in (V_1 \cup \Sigma)^*$ such that $S_1 \in z$ and $S_1 \Longrightarrow^{k+1-j} z \Longrightarrow^j y$, where $j \geq 1$ (because we need at least one rule of P_1 in order to replace a variable by a terminal). However, since $k+1-j < k$, the inductive hypothesis implies that $n_a(z) = n_b(z)$ and the subsequent application of (any) rules of P_1 does not change the equality in the number of a's and b's, meaning that $y \in L_{\sharp a=\sharp b}$.

Equivalent Context–Free Grammars, III

Proposition

Prove that $L_{\sharp a = \sharp b} \subseteq L(G_1)$.

Using **induction on the length of strings** n, we are going to show that: For every integer $n \geq 0$, if $x \in L_{\sharp a = \sharp b}$ with $|x| = n$, then $S_1 \implies^* x$, i.e., $x \in L(G_1)$.

Base case: True, for $n = 0$, because $n_a(\varepsilon) = n_b(\varepsilon) = 0$ and $S_1 \implies \varepsilon$.
Inductive hypothesis: Suppose that there exists integer $k \geq 1$ such that, for all i, $0 \leq i \leq k$, if $x \in L_{\sharp a = \sharp b}$ and $|x| = i$, then $S_1 \implies^* x$.
Inductive step: We need to show that, if $y \in L_{\sharp a = \sharp b}$ with $|y| = k + 1$, then $S_1 \implies^* y$. Depending on how it starts and ends, y can be one of the following four forms: (i) $y = azb$, (ii) $y = bza$, (iii) $y = aza$, (iv) $y = bzb$, where in all cases $z \in \Sigma^*$ has $|z| = k + 1 - 2 = k - 1$. Actually, it suffices to do cases (i) and (iii) (because (ii) can be treated similarly to (i) and (iv) similarly to (iii)).

(i) If $y = azb$, since $n_a(z) = n_b(z)$ and $|z| = k - 1$, the inductive hypothesis implies that $S_1 \implies^* z$. Hence, $S_1 \implies aS_1b \implies^* azb = y$, which was what we needed to show.

(iii) If $y = aza$, since $n_b(z) = n_a(z) + 2$ and $|z| = k - 1$, z should be represented as $z = u_1bu_2bu_3$, for three strings $u_p \in L_{\sharp a = \sharp b}$, for $p = 1, 2, 3$, such that $0 \leq |u_p| \leq |u_1| + |u_2| + |u_3| = |z| - 2 = k - 3$. Then, by the inductive hypothesis, $S_1 \implies^* u_1|u_2|u_3$ and, moreover, $S_1 \implies S_1S_1 \implies S_1S_1S_1 \implies^2 aS_1bS_1bS_1a \implies^* aS_1bu_2bS_1a \implies^* au_1bu_2bS_1a \implies^* au_1bu_2bu_3a = aza = y$ and that was what we wanted to show.

Equivalent Context–Free Grammars, IV

Proposition

Prove that $L(G_2) = L_{\sharp a = \sharp b}$.

Proposition

Prove that $L(G_2) \subseteq L_{\sharp a = \sharp b}$.

Using **induction on the number of derivations** n, we are going to show that: If, for every integer $n \geq 1$, $S_2 \implies^n x \in \Sigma^*$, i.e., $x \in L(G_2)$, then $n_a(x) = n_b(x)$, i.e., $x \in L_{\sharp a = \sharp b}$.

Base case: True, for $n = 1$, because $S_2 \implies \varepsilon \in L_{\sharp a = \sharp b}$.
Inductive hypothesis: Suppose that there exists integer $k \geq 1$ such that, for all nonnegative integers $i \leq k$, if $S_2 \implies^i x \in \Sigma^*$, then $x \in L_{\sharp a = \sharp b}$.
Inductive step: We need to show that, if $S_2 \implies^{k+1} y \in \Sigma^*$, then $y \in L_{\sharp a = \sharp b}$. Notice that, in this case, there exists $z \in (V_2 \cup \Sigma)^*$ such that $S_2, A, B \in z$ and $S_2 \implies^{k+1-j} z \implies^j y$, where $j \geq 1$ (because we need at least one rule of P_2 in order to replace a variable by a terminal). However, since $k + 1 - j < k$, the inductive hypothesis implies that $n_a(z) = n_b(z)$ and the subsequent application of (any) rules of P_2 does not change the equality in the number of a's and b's, meaning that $y \in L_{\sharp a = \sharp b}$.

Equivalent Context–Free Grammars, V

Proposition

Prove that $L_{\sharp a=\sharp b} \subseteq L(G_2)$.

Using **induction on the length of strings** n, we are going to show that: For every integer $n \geq 0$, if $x \in L_{\sharp a=\sharp b}$ with $|x| = n$, then $S_2 \Longrightarrow^* x$, i.e., $x \in L(G_2)$.

Base case: True, for $n = 0$, because $n_a(\varepsilon) = n_b(\varepsilon) = 0$ and $S_2 \Longrightarrow \varepsilon$.
Inductive hypothesis: Suppose that there exists integer $k \geq 1$ such that, for all i, $0 \leq i \leq k$, if $x \in L_{\sharp a=\sharp b}$ and $|x| = i$, then $S_2 \Longrightarrow^* x$. Actually, k should be taken even, because any string with equal number of a's and b's has even length.
Inductive step: We need to show that, if $y \in L_{\sharp a=\sharp b}$ with $|y| = k+2$, then $S_2 \Longrightarrow^* y$. Depending on how it starts and ends, y can be one of the following three forms: (i) $y = abz$, (ii) $y = baz$, (iii) y does not start either with ab or ba, where in the first two cases $|z| = k+1-2 = k-1$. Actually, it suffices to do cases (i) and (iii) (since (ii) can be treated similarly to (i)).

(i) If $y = abz$, since $n_a(z) = n_b(z)$ and $|z| = k-1$, the inductive hypothesis implies that $S_2 \Longrightarrow^* z$. Hence, $S_2 \Longrightarrow aB \Longrightarrow abS_2 \Longrightarrow^* abz = y$, which was what we needed to show.

Equivalent Context–Free Grammars, V (cont.)

Proof of Proposition (cont.)

(iii) If y does not start either with ab or ba, then necessarily y is represented as $y = u_1u_2$, where $u_1, u_2 \in L_{\sharp a=\sharp b}$ and $2 \leq |u_1|, |u_2| < |u_1| + |u_2| = |y| = k+2$, i.e., $0 \leq |u_1|, |u_2| \leq k$. Then the inductive hypothesis implies that $S_2 \Longrightarrow^* u_1 \mid u_2$. Moreover, since all the grammar rules in the CFG G_2 have a variable at the rightmost place and the only variable terminating (to a terminal) is S_2, accoding to the Theorem in the previous section about ambiguity, every $u \in L_{\sharp a=\sharp b}$ with $|u| \leq k$ can be produced as $S_2 \Longrightarrow^* uS_2$. Therefore, $S_2 \Longrightarrow^* u_1S_2 \Longrightarrow^* u_1u_2 = y$, which was what we needed to show.

Remark

In the previous Example, both $G1$ and G_2 are ambiguous.
Exampes:

$S_1 \Longrightarrow S_1S_1 \Longrightarrow aS_1bS_1 \Longrightarrow abS_1abS_1 \Longrightarrow abab$

$S_1 \Longrightarrow aS_1b \Longrightarrow abS_1ab \Longrightarrow abab$

$S_2 \Longrightarrow bA \Longrightarrow bbAAA \Longrightarrow bbaS_2A \Longrightarrow bbabAA \Longrightarrow bbabbAAA \Longrightarrow bbabbaaa$

$S_2 \Longrightarrow bA \Longrightarrow bbAAA \Longrightarrow bbaS_2A \Longrightarrow bbaA \Longrightarrow bbabAA \Longrightarrow bbabbAAA \Longrightarrow bbabbaaa$

Can you find a nonambiguous CFG for $L_{\sharp a=\sharp b}$?