COSE215: Theory of Computation

Lecture 11 — Context-Free Grammars (2)

Hakjoo Oh
2017 Spring

Mid-term Exam

- 4/24 (Mon), 09:00–10:15 (in class)
- Do not be late.
- Coverage: finite automata, regular expressions, regular languages, context-free grammars
- Based on lectures and homework.
- No classes on 4/26 (Wed).

Parse Trees

Definition (Parse Trees)

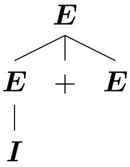
Let $G = (V, T, S, P)$ be a grammar. The *parse trees* for G are trees with the following conditions:

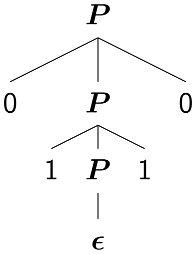
1. The root is S, the start variable.
2. Each interior node is labeled by a variable in V.
3. Each leaf is labeled by either a variable, a terminal, or ϵ. However, if the leaf is labeled ϵ, it must be the only child of its parent.
4. If an interior node is labeled A, and its children are labeled

$$X_1, X_2, \ldots, X_k$$

respectively, from the left, then $A \to X_1, X_2, \ldots, X_k$ is a production in P.

Example





Yields

Definition (Yields)

The string obtained by concatenating the leaves of a parse tree from the left is called the *yield* of the tree.

Relationship between Parse Trees and Sentential Forms

Theorem

Let $G = (V, T, S, P)$ be a context-free grammar. Then, the following are equivalent:

1. $S \Rightarrow^* w$.
2. $S \Rightarrow^*_{lm} w$.
3. $S \Rightarrow^*_{rm} w$.
4. *There is a parse tree whose yield is w.*

Parse Trees

A context-free grammar for expressions:

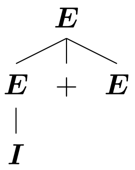
$$G = (\{E, I\}, \{+, *, (,), a, b, 0, 1\}, E, P)$$

$$\begin{array}{rcl} E & \rightarrow & I \mid E + E \mid E * E \mid (E) \\ I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

A derivation:

$$E \Rightarrow E + E \Rightarrow I + E.$$

The parse tree of the derivation:



Formal Definition

Definition (Parse Trees)

Let $G = (V, T, S, P)$ be a grammar. The *parse trees* for G are trees with the following conditions:

1. The root is S.
2. Each interior node is labeled by a variable in V.
3. Each leaf is labeled by either a variable, a terminal, or ϵ. However, if the leaf is labeled ϵ, it must be the only child of its parent.
4. If an interior node is labeled A, and its children are labeled

   $$X_1, X_2, \ldots, X_k$$

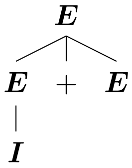
   respectively, from the left, then $A \to X_1, X_2, \ldots, X_k$ is a production in P.

Example 1: Expressions

$$G = (\{E, I\}, \{+, *, (,), a, b, 0, 1\}, E, P)$$

$$\begin{aligned} E &\rightarrow I \mid E + E \mid E * E \mid (E) \\ I &\rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{aligned}$$

A parse tree:

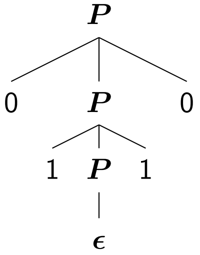


Example 2: Palindromes

$$G = (\{P\}, \{0, 1\}, P, A)$$

$$P \rightarrow \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$$

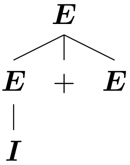
A parse tree:

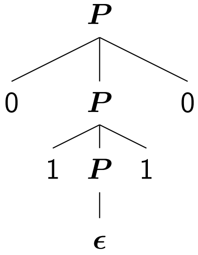


Yields

Definition (Yields)

The string obtained by concatenating the leaves of a parse tree from the left is called the *yield* of the tree.





Relationship between Parse Trees and Derivations

Theorem

Let $G = (V, T, S, P)$ be a context-free grammar. Then, the following are equivalent:

1. $S \Rightarrow^* w$.
2. $S \Rightarrow^*_{lm} w$.
3. $S \Rightarrow^*_{rm} w$.
4. *There is a parse tree whose yield is w.*

Ambiguous and Unambiguous Grammars

Definition

A context-free grammar is *ambiguous* if there exists some $w \in L(G)$ that has at least two distinct parse trees. If each string has at most one parse tree, the grammar is *unambiguous*.

Theorem

For each grammar $G = (V, T, S, P)$ *and string* $w \in T^*$, w *has two distinct parse trees if and only if* w *has two distinct leftmost derivations from* S.

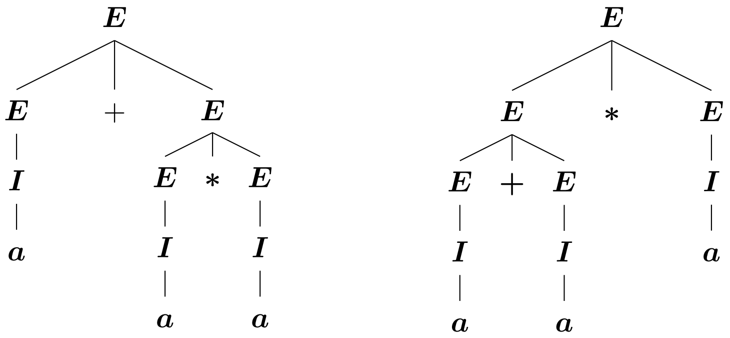
Example

The grammar of expressions:

$$G = (\{E, I\}, \{+, *, (,), a, b, 0, 1\}, E, P)$$

$$\begin{array}{rcl} E & \rightarrow & I \mid E + E \mid E * E \mid (E) \\ I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

Two distinct parse trees for $a + a * a$:



Example

The grammar of expressions:

$$G = (\{E, I\}, \{+, *, (,), a, b, 0, 1\}, E, P)$$

$$\begin{array}{rcl} E & \rightarrow & I \mid E + E \mid E * E \mid (E) \\ I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

Two distinct leftmost derivations for $a + a * a$:

- $E \Rightarrow E + E \Rightarrow I + E \Rightarrow a + E \Rightarrow a + E * E \Rightarrow a + I * E \Rightarrow a + a * E \Rightarrow a + a * I \Rightarrow a + a * a$
- $E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow I + E * E \Rightarrow a + E * E \Rightarrow a + I * E \Rightarrow a + a * E \Rightarrow a + a * I \Rightarrow a + a * a$

General Facts

- There is no algorithm to remove ambiguity from a CFG.
- There is no algorithm that can even tell us whether a CFG is ambiguous or not.
- There are context-free languages that are inherently ambiguous; for these languages, removing the ambiguity is impossible.

Finding an unambiguous grammar is possible in practice

An ambiguous grammar:

$$\begin{array}{rcl} E & \rightarrow & I \mid E + E \mid E * E \mid (E) \\ I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

Finding an unambiguous grammar is possible in practice

An ambiguous grammar:

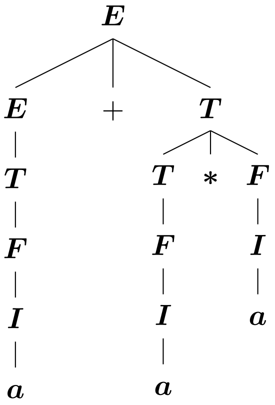
$$
\begin{array}{rcl}
E & \rightarrow & I \mid E + E \mid E * E \mid (E) \\
I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1
\end{array}
$$

An unambiguous grammar:

$$
\begin{array}{rcl}
I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\
F & \rightarrow & I \mid (E) \\
T & \rightarrow & F \mid T * F \\
E & \rightarrow & T \mid E + T
\end{array}
$$

Example

The only parse tree for $a + a * a$:



Inherent Ambiguity

Definition

A language L is *inherently ambiguous* if every grammar that generates L is ambiguous.

Example

$$L = L_1 \cup L_2$$

where

$$L_1 = \{a^n b^n c^m \mid n, m \geq 0\}, \quad L_2 = \{a^n b^m c^m \mid n, m \geq 0\}$$