Lecture 11 — Context-Free Grammars (2)

CSE205: Theory of Computation

Minseok Jeon

2026 Fall

Parse Trees

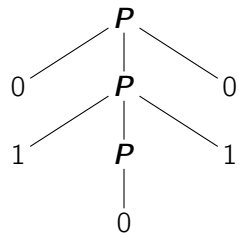
Definition (Parse Tree)

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

1. The root is labeled S, the start variable.
2. Each interior node is labeled by a variable in V.
3. Each leaf is labeled by a variable, a terminal, or ε. However, if a 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 $X_1, X_2, \ldots, X_k$ from the left, then $A \to X_1 X_2 \cdots X_k$ is a production in P.

Example: A Parse Tree

A parse tree for the palindrome 01010 in the grammar $\boldsymbol{P} \to \boldsymbol{\varepsilon} \mid 0 \mid 1 \mid 0\boldsymbol{P}0 \mid 1\boldsymbol{P}1$:



Yields

Definition (Yield)

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

- For a parse tree of a grammar $\boldsymbol{G}$, the yield is always a string in $(\boldsymbol{V} \cup \boldsymbol{T})^*$.
- If the root is the start symbol $\boldsymbol{S}$ and every leaf is a terminal or $\boldsymbol{\epsilon}$, then the yield is a string of $\boldsymbol{L}(\boldsymbol{G})$.

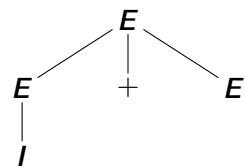
Parse Trees: Expressions

A context-free grammar for 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 derivation $E \Rightarrow E + E \Rightarrow I + E$ has the parse tree:



Relationship between Parse Trees and Derivations

Theorem

Let $G = (V, T, S, P)$ be a context-free grammar and let $w \in T^*$. Then the following are equivalent:

1. $S \Rightarrow^* w$.
2. $S \Rightarrow^*_{lm} w$ *(leftmost derivation).*
3. $S \Rightarrow^*_{rm} w$ *(rightmost derivation).*
4. *There is a parse tree of G whose yield is w.*

A parse tree abstracts away the order in which variables are replaced; a single parse tree corresponds to exactly one leftmost (and one rightmost) derivation.

Ambiguous and Unambiguous Grammars

Definition (Ambiguity)

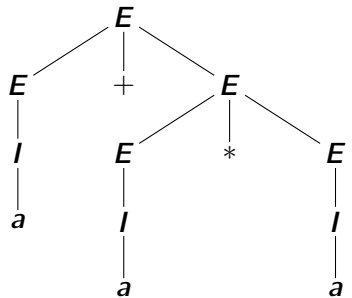
A context-free grammar G is *ambiguous* if there exists some $w \in L(G)$ that has at least two distinct parse trees. If each string has at most one parse tree, G is *unambiguous*.

Theorem

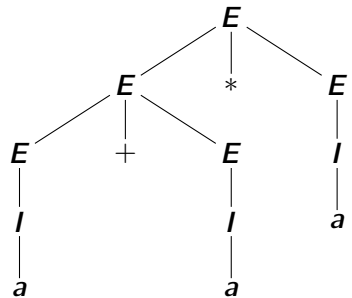
For each grammar $G = (V, T, S, P)$ and string $w \in T^*$, w has two distinct parse trees if and only if w has two distinct leftmost derivations from S.

Example: Two Parse Trees for $a + a * a$

The expression grammar $E \to I \mid E + E \mid E * E \mid (E)$ is ambiguous: $a + a * a$ has two distinct parse trees.



$+$ applied last



$*$ applied last

Example: Two Leftmost Derivations for $a + a * a$

The same ambiguity appears as two distinct leftmost derivations:

First:

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

Second:

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

General Facts about Ambiguity

- There is no algorithm to remove ambiguity from a context-free grammar.
- There is no algorithm that can even tell us whether a given context-free grammar 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 (in practice)

Although ambiguity is undecidable in general, in practice we can often rewrite a grammar to be unambiguous.

An ambiguous grammar:

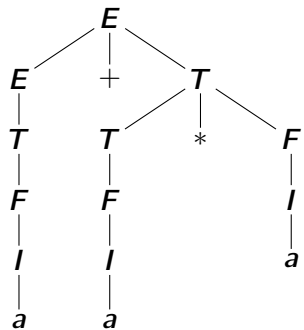
$$
\begin{array}{rcl}
E & \to & I \mid E + E \mid E * E \mid (E) \\
I & \to & a \mid b \mid Ia \mid Ib \mid I0 \mid I1
\end{array}
$$

An equivalent unambiguous grammar (enforcing precedence and associativity):

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

Example: The Only Parse Tree for $a + a * a$

In the unambiguous grammar, $*$ binds tighter than $+$, giving a single parse tree for $a + a * a$:



Inherent Ambiguity

Definition (Inherently Ambiguous Language)

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

Example

The language

$$L = L_1 \cup L_2$$

where

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

is inherently ambiguous. Strings of the form $a^n b^n c^n$ must be generated in two different ways, so no grammar for L can be unambiguous.