Sentential Forms

Let $G = (V, T, P, S)$ be a CFG, and $\alpha \in (V \cup T)^*$. If

$$S \overset{*}{\Rightarrow} \alpha$$

we say that α is a *sentential form*.

If $S \underset{lm}{\Rightarrow} \alpha$ we say that α is a *left-sentential form*, and if $S \underset{rm}{\Rightarrow} \alpha$ we say that α is a *right-sentential form*

Note: $L(G)$ is those sentential forms that are in T^*.

Example: Take G from slide 138. Then $E * (I + E)$ is a sentential form since

$$E \Rightarrow E*E \Rightarrow E*(E) \Rightarrow E*(E+E) \Rightarrow E*(I+E)$$

This derivation is neither leftmost, nor rightmost

Example: $a * E$ is a left-sentential form, since

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E$$

Example: $E*(E+E)$ is a right-sentential form, since

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E)$$

Parse Trees

- If $w \in L(G)$, for some CFG, then w has a *parse tree*, which tells us the (syntactic) structure of w

- w could be a program, a SQL-query, an XML-document, etc.

- Parse trees are an alternative representation to derivations and recursive inferences.

- There can be several parse trees for the same string

- Ideally there should be only one parse tree (the "true" structure) for each string, i.e. the language should be *unambiguous*.

- Unfortunately, we cannot always remove the ambiguity.

Constructing Parse Trees

Let $G = (V, T, P, S)$ be a CFG. A tree is a *parse tree* for G if:

1. Each interior node is labelled by a variable in V.

2. Each leaf is labelled by a symbol in $V \cup T \cup \{\epsilon\}$. Any ϵ-labelled leaf is the only child of its parent.

3. If an interior node is lablelled A, and its children (from left to right) labelled

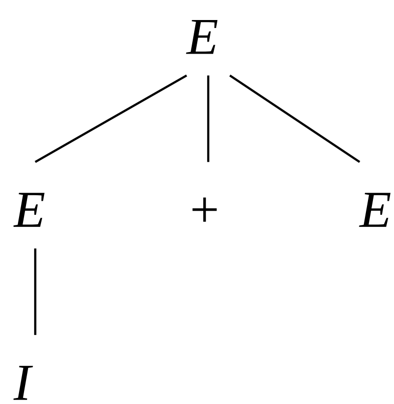
   $$X_1, X_2, \ldots, X_k,$$

   then $A \rightarrow X_1 X_2 \ldots X_k \in P$.

Example: In the grammar

$$
\begin{aligned}
&1.\ E \to I \\
&2.\ E \to E + E \\
&3.\ E \to E * E \\
&4.\ E \to (E) \\
&\quad\quad \vdots
\end{aligned}
$$

the following is a parse tree:

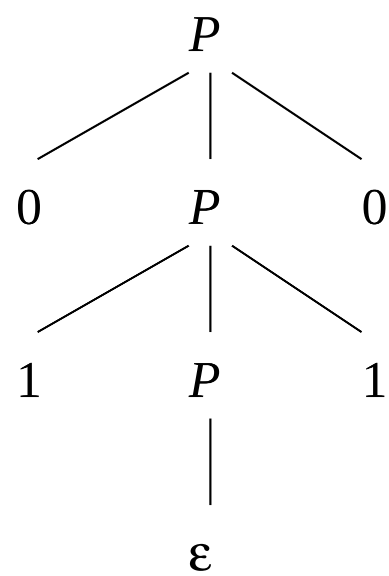


This parse tree shows the derivation $E \overset{*}{\Rightarrow} I + E$

Example: In the grammar

1. $P \rightarrow \epsilon$
2. $P \rightarrow 0$
3. $P \rightarrow 1$
4. $P \rightarrow 0P0$
5. $P \rightarrow 1P1$

the following is a parse tree:



It shows the derivation of $P \stackrel{*}{\Rightarrow} 0110$.

The Yield of a Parse Tree

The *yield* of a parse tree is the string of leaves from left to right.

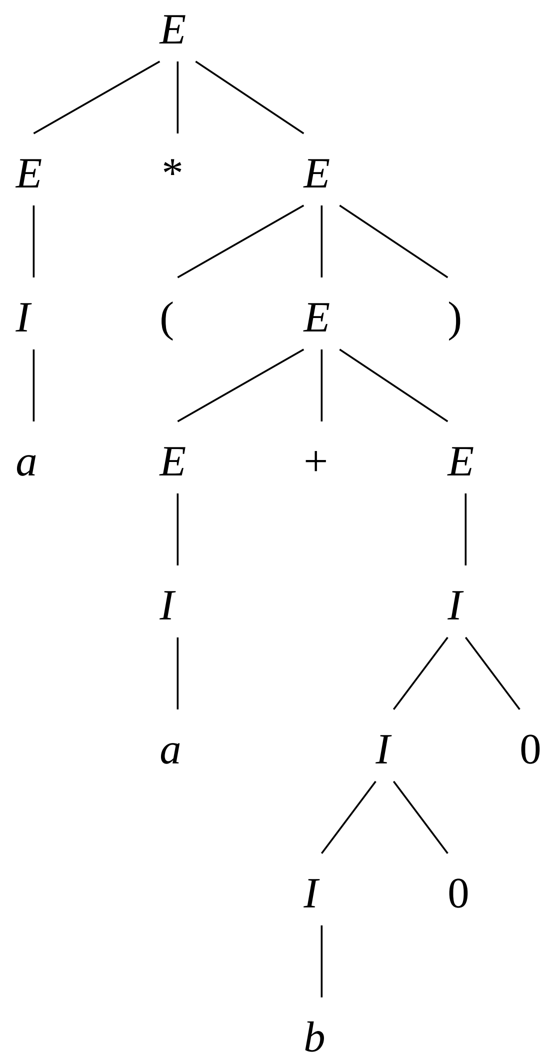
Important are those parse trees where:

1. The yield is a terminal string.

2. The root is labelled by the start symbol

We shall see the the set of yields of these important parse trees is the language of the grammar.

Example: Below is an important parse tree



The yield is $a * (a + b00)$.

Compare the parse tree with the derivation on slide 141.

Let $G = (V, T, P, S)$ be a CFG, and $A \in V$. We are going to show that the following are equivalent:

1. We can determine by recursive inference that w is in the language of A
2. $A \stackrel{*}{\Rightarrow} w$
3. $A \underset{lm}{\stackrel{*}{\Rightarrow}} w$, and $A \underset{rm}{\stackrel{*}{\Rightarrow}} w$
4. There is a parse tree of G with root A and yield w.

To prove the equivalences, we use the following plan.