COMP2012/G52LAC
Languages and Computation
Lecture 10

Derivation Trees and Ambiguity

Henrik Nilsson

University of Nottingham

Recap: Definition of CFG

A CFG $G = (N, T, P, S)$ where

- N is a finite set of ***nonterminals*** (or ***variables*** or ***syntactic categories***)
- T is a finite set of ***terminals***
- $N \cap T = \emptyset$ (disjoint)
- P is a finite set of ***productions*** of the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup T)^*$
- $S \in N$ is the ***start symbol***

Recap: The Directly Derives Relation (1)

To formally define the language generated by

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

we first define a binary relation $\underset{G}{\Rightarrow}$ on strings over $N \cup T$, read "***directly derives in grammar*** G", being the least relation such that

$$\alpha A \gamma \underset{G}{\Rightarrow} \alpha \beta \gamma$$

whenever $A \to \beta$ is a production in G where $A \in N$ and $\alpha, \beta, \gamma \in (N \cup T)^*$.

Recap: The Directly Derives Relation (2)

When it is clear which grammar G is involved, we use $\Rightarrow$ instead of $\underset{G}{\Rightarrow}$.

Example: Given the grammar

$$\begin{array}{rcl} S & \rightarrow & \epsilon \mid aA \\ A & \rightarrow & bS \end{array}$$

we have

$$\begin{array}{rcl} S & \Rightarrow & \epsilon \\ S & \Rightarrow & aA \end{array} \qquad \begin{array}{rcl} aA & \Rightarrow & abS \\ SaAaa & \Rightarrow & SabSaa \end{array}$$

Recap: The Derives Relation (1)

The relation $\overset{*}{\underset{G}{\Rightarrow}}$, read "***derives in grammar*** G", is the reflexive, transitive closure of $\underset{G}{\Rightarrow}$.

That is, $\overset{*}{\underset{G}{\Rightarrow}}$ is the least relation on strings over $N \cup T$ such that:

Recap: The Derives Relation (1)

The relation $\overset{*}{\underset{G}{\Rightarrow}}$, read "***derives in grammar*** G", is the reflexive, transitive closure of $\underset{G}{\Rightarrow}$.

That is, $\overset{*}{\underset{G}{\Rightarrow}}$ is the least relation on strings over $N \cup T$ such that:

- $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta \quad$ if $\quad \alpha \underset{G}{\Rightarrow} \beta$

Recap: The Derives Relation (1)

The relation $\overset{*}{\underset{G}{\Rightarrow}}$, read "***derives in grammar*** G", is the reflexive, transitive closure of $\underset{G}{\Rightarrow}$.

That is, $\overset{*}{\underset{G}{\Rightarrow}}$ is the least relation on strings over $N \cup T$ such that:

- $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ if $\alpha \underset{G}{\Rightarrow} \beta$
- $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$ (reflexive)

Recap: The Derives Relation (1)

The relation $\overset{*}{\underset{G}{\Rightarrow}}$, read "***derives in grammar*** G", is the reflexive, transitive closure of $\underset{G}{\Rightarrow}$.

That is, $\overset{*}{\underset{G}{\Rightarrow}}$ is the least relation on strings over $N \cup T$ such that:

- $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ if $\alpha \underset{G}{\Rightarrow} \beta$
- $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$ (reflexive)
- $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ if $\alpha \overset{*}{\underset{G}{\Rightarrow}} \gamma \wedge \gamma \overset{*}{\underset{G}{\Rightarrow}} \beta$ (transitive)

Recap: The Derives Relation (2)

Again, we use $\stackrel{*}{\Rightarrow}$ instead of $\underset{G}{\stackrel{*}{\Rightarrow}}$ when G is obvious.

Example: Given the grammar

$$\begin{array}{rcl} S & \rightarrow & \epsilon \mid aA \\ A & \rightarrow & bS \end{array}$$

we have

$$\begin{array}{rcl} S & \stackrel{*}{\Rightarrow} & \epsilon \\ S & \stackrel{*}{\Rightarrow} & aA \\ aA & \stackrel{*}{\Rightarrow} & abS \end{array} \qquad \begin{array}{rcl} S & \stackrel{*}{\Rightarrow} & abS \\ S & \stackrel{*}{\Rightarrow} & ababS \\ S & \stackrel{*}{\Rightarrow} & abab \end{array}$$

Recap: Lang. Generated by a Grammar

The ***language generated*** by a context-free grammar

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

denoted $L(G)$, is defined as follows:

$$L(G) = \{w \mid w \in T^* \ \wedge \ S \underset{G}{\overset{*}{\Rightarrow}} w\}$$

A language L is a ***Context-Free Language*** (CFL) iff $L = L(G)$ for some CFG G.

A string $\alpha \in (N \cup T)^*$ is a ***sentential form*** iff $S \overset{*}{\Rightarrow} \alpha$.

Simple Arithmetic Expressions

$SAE = (N = \{E, I, D\}, T = \{+, *, (,), 0, 1, \ldots, 9\}, P, E)$
where P is given by:

$$
\begin{array}{rcl}
E & \rightarrow & E + E \\
 & | & E * E \\
 & | & (E) \\
 & | & I \\
I & \rightarrow & DI \mid D \\
D & \rightarrow & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{array}
$$

Note: $A \rightarrow \alpha \mid \beta$ shorthand for $A \rightarrow \alpha$, $A \rightarrow \beta$.

Derivation Trees (1)

A tree is a ***derivation tree*** for a CFG $G = (N, T, P, S)$ iff

1. Every node has a label from $N \cup T \cup \{\epsilon\}$.
2. The label of the root node is S.
3. Labels of interior nodes belong to N.
4. If a node n has label A and nodes $n_1, n_2, \ldots, n_k$ are children of n, from left to right, with labels $X_1, X_2, \ldots X_k$, respectively, then $A \to X_1 X_2 \ldots X_k$ is a production in P.
5. If a node n has label ϵ, then n is a leaf and the only child of its parent.

Derivation Trees (2)

- The string of ***leaf labels*** read from left to right, eliding any ϵ, constitute the ***yield*** of the tree.
- For a CFG $G = (N, T, P, S)$, a string $\alpha \in (N \cup T)^*$ is the yield of some derivation tree iff $S \underset{G}{\overset{*}{\Rightarrow}} \alpha$.