Context-Free Grammars

Notation for recursive description of languages.
Example:

$Roll \rightarrow < ROLL > \ Class \ Studs \ < /ROLL >$
$Class \rightarrow < CLASS > \ Text \ < /CLASS >$
$Text \rightarrow Char \ Text$
$Text \rightarrow Char$
$Char \rightarrow a \cdots$ (other chars)
$Studs \rightarrow Stud \ Studs$
$Studs \rightarrow \epsilon$
$Stud \rightarrow < STUD > \ Text \ < /STUD >$

- Generates "documents" such as:

```
<ROLL><CLASS>cs154</CLASS>
    <STUD>Sally</STUD>
    <STUD>Fred</STUD>
        ...
</ROLL>
```

- *Variables* (e.g., *Studs*) represent sets of strings (i.e., languages).
  - In sensible grammars, these strings share some common characteristic or roll.
- *Terminals* (e.g., a or $< ROLL >$) = symbols of which strings are composed.
  - "Tags" like $< \ ROLL \ >$ could be considered either a single terminal or the concatenation of 6 terminals.
- *Productions* = rules of the form $Head \rightarrow$ *Body*.
  - $Head$ is a variable.
  - *Body* is a string of zero or more variables and/or terminals.
- *Start Symbol* = variable that represents "the language."
- Notation: $G = (V, \Sigma, P, S)$ = (variables, terminals, productions, start symbol).

Example

A simpler example generates strings of 0's and 1's such that each block of 0's is followed by at least as many 1's.

$S \rightarrow AS \mid \epsilon$
$A \rightarrow 0A1 \mid A1 \mid 01$

- Note vertical bar separates different bodies for the same head.

Derivations

- $\alpha A\beta \Rightarrow \alpha\gamma\beta$ whenever there is a production $A \rightarrow \gamma$.
  - Subscript with name of grammar, e.g., $\underset{G}{\Rightarrow}$, if necessary.
  - Example: $011AS \Rightarrow 0110A1S$.
- $\alpha \overset{*}{\Rightarrow} \beta$ means string α can become β in zero or more derivation steps.
  - Examples: $011AS \overset{*}{\Rightarrow} 011AS$ (zero steps); $011AS \overset{*}{\Rightarrow} 0110A1S$ (one step); $011AS \overset{*}{\Rightarrow} 0110011$ (three steps).

Language of a CFG

$L(G)$ = set of terminal strings w such that $S \overset{*}{\underset{G}{\Rightarrow}} w$, where S is the start symbol.

Aside: Notation

- $a, b, \ldots$ are terminals; $\ldots, y, z$ are strings of terminals.
- Greek letters are strings of variables and/or terminals, often called *sentential forms*.
- $A, B, \ldots$ are variables.
- $\ldots, Y, Z$ are variables or terminals.
- S is typically the start symbol.

Leftmost/Rightmost Derivations

- We have a choice of variable to replace at each step.
  - Derivations may appear different only because we make the *same* replacements in a different order.
  - To avoid such differences, we may restrict the choice.
- A *leftmost* derivation always replaces the leftmost variable in a sentential form.
  - Yields *left-sentential forms*.
- *Rightmost* defined analogously.
- $\underset{lm}{\Rightarrow}$, $\underset{rm}{\Rightarrow}$, etc., used to indicate derivations are leftmost or rightmost.

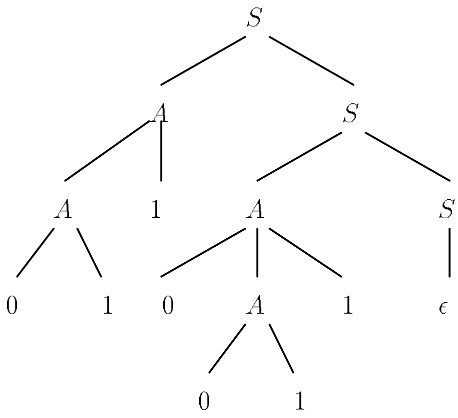
Example

- $S \underset{lm}{\Rightarrow} AS \underset{lm}{\Rightarrow} A1S \underset{lm}{\Rightarrow} 011S \underset{lm}{\Rightarrow} 011AS \underset{lm}{\Rightarrow} 0110A1S \underset{lm}{\Rightarrow} 0110011S \underset{lm}{\Rightarrow} 0110011$
- $S \underset{rm}{\Rightarrow} AS \underset{rm}{\Rightarrow} AAS \underset{rm}{\Rightarrow} AA \underset{rm}{\Rightarrow} A0A1 \underset{rm}{\Rightarrow} A0011 \underset{rm}{\Rightarrow} A10011 \underset{rm}{\Rightarrow} 0110011$

Derivation Trees

- Nodes = variables, terminals, or ϵ.
  - ❖ Variables at interior nodes; terminals and ϵ at leaves.
  - ❖ A leaf can be ϵ only if it is the only child of its parent.
- A node and its children from the left must form the head and body of a production.

Example



Equivalence of Parse Trees, Leftmost, and Rightmost Derivations

The following about a grammar $G = (V, \Sigma, P, S)$ and a terminal string w are all equivalent:

1. $S \overset{*}{\Rightarrow} w$ (i.e., w is in $L(G)$).
2. $S \overset{*}{\underset{lm}{\Rightarrow}} w$
3. $S \overset{*}{\underset{rm}{\Rightarrow}} w$
4. There is a parse tree for G with root S and *yield* (labels of leaves, from the left) w.

- Obviously (2) and (3) each imply (1).

Parse Tree Implies LM/RM Derivations

- Generalize all statements to talk about an arbitrary variable A in place of S.
  - Except now (1) no longer means w is in $L(G)$.
- Induction on the height of the parse tree.

Basis: Height 1: Tree is root A and leaves $w = a_1, a_2, \ldots, a_k$.

- $A \rightarrow w$ must be a production, so $A \underset{lm}{\Rightarrow} w$ and $A \underset{rm}{\Rightarrow} w$.

Induction: Height > 1. Tree is root A with children $X_1, X_2, \ldots, X_k$.

- Those X_i's that are variables are roots of shorter trees.
  - Thus, the IH says that they have LM derivations of their yields.
- Construct a LM derivation of w from A by starting with $A \underset{lm}{\Rightarrow} X_1 X_2 \cdots X_k$, then using LM derivations from each X_i that is a variable, in order from the left.
- RM derivation analogous.

Derivations to Parse Trees

Induction on length of the derivation.

Basis: One step. There is an obvious parse tree.

Induction: More than one step.

- Let the first step be $A \Rightarrow X_1 X_2 \cdots X_k$.
- Subsequent changes can be reordered so that all changes to X_1 and the sentential forms that replace it are done first, then those for X_2, and so on (i.e., we can rewrite the deriviation as a LM derivation).
- The derivations from those X_i's that are variables are all shorter than the given deriviation, so the IH applies.
- By the IH, there are parse trees for each of these derivations.
- Make the roots of these trees be children of a new root labeled A.

Example

Consider derivation $S \Rightarrow AS \Rightarrow AAS \Rightarrow AA \Rightarrow A1A \Rightarrow A10A1 \Rightarrow 0110A1 \Rightarrow 0110011$

- Subderivation from A is: $A \Rightarrow A1 \Rightarrow 011$
- Subderivation from S is: $S \Rightarrow AS \Rightarrow A \Rightarrow 0A1 \Rightarrow 0011$
- Each has a parse tree; put them together with new root S.