CS 3813: Introduction to Formal Languages and Automata

Context-free grammars (5.1)

Grammar

- A grammar consists of the following:
  - a set Σ of terminals (same as an alphabet)
  - a set NT of nonterminal symbols, including a starting symbol $S \in NT$
  - a set R of rules
- Example

  $S \rightarrow aS \mid A$

  $A \rightarrow bA \mid \lambda$

Derivation

- Strings are "derived" from a grammar
- Example of a derivation

  $S \Rightarrow aS \Rightarrow aaS \Rightarrow aabA \Rightarrow aab$
- At each step, a nonterminal is replaced by the **sentential form** on the right-hand side of a rule (a sentential form can contain nonterminals and/or terminals)
- Automata recognize languages … grammars generate languages

Context-free grammar

- A grammar is said to be context-free if every rule has a single nonterminal on the left-hand side
- This means you can apply the rule in any context. More complicated languages (such as English) have context-dependent rules. But we only consider context-free grammars in this course.
- A language generated from a context-free grammar is called a context-free language

Exercise

What language is generated by the following context-free grammar:

$S \rightarrow aSa|bSb|a|b|\lambda$

Is this a regular language? Why or why not?

Programming languages

- Programming languages are context-free, but not regular
- Programming languages have the following features that require infinite "stack memory"
  - matching parentheses in algebraic expressions
  - nested if .. then .. else statements, and nested loops
  - block structure

Exercise

- Use set notation to define the language generated by the following grammars

1) $S \to aaSB \mid \lambda$
   $B \to bB \mid b$

2) $S \to aSbb \mid A$
   $A \to cA \mid c$

Derivation

Given the grammar,

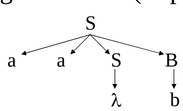
$S \to aaSB \mid \lambda$
$B \to bB \mid b$

the string aab can be derived in different ways.

$S \Rightarrow aaSB \Rightarrow aaB \Rightarrow aab$
$S \Rightarrow aaSB \Rightarrow aaSb \Rightarrow aab$

Derivation (parse) tree

Both derivations on the previous slide correspond to the following derivation (or parse) tree.



The tree structure shows the rule that is applied to each nonterminal, without showing the order of rule applications. Each internal node of the tree corresponds to a nonterminal, and the leaves of the derivation tree represent the string of terminals.

Exercise

Let G be the grammar

$S \to abSc \mid A$
$A \to cAd \mid cd$

1) Give a derivation of ababccddcc.

2) Build the parse tree for the derivation of (1).

3) Use set notation to define L(G).

Leftmost (rightmost) derivation

In a *leftmost derivation*, the leftmost nonterminal is replaced at each step.

In a *rightmost derivation*, the rightmost nonterminal is replaced at each step.

Many derivations are neither leftmost nor rightmost.

If there is a single parse tree, there is also a single leftmost derivation.