CS 241 — Lecture 16

Bartosz Antczak *Instructor: Kevin Lanctot* *March 2, 2017*

Remark

Assignment 7 is out, and according to Prof. Lanctot, it's the hardest assignment to complete in the course. About 50% of students won't finish it. Because of this, Prof. Lanctot will review parsing (which is the topic of assignment 7).

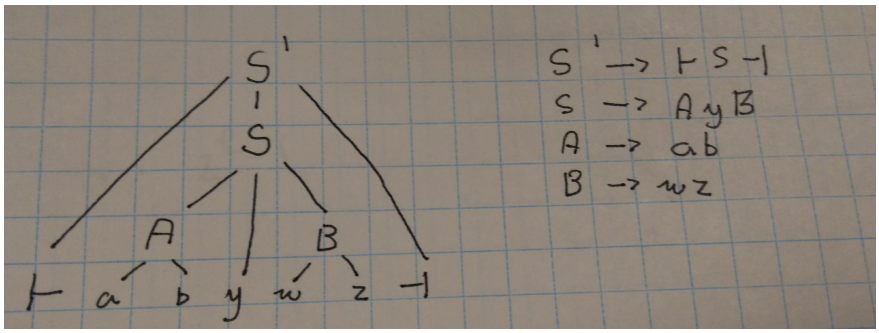
16.0.1 Review of Parsing

Parsing validates whether a given word in accepted for a particular context-free grammar. There's two types of parsing: top-down or bottom-up.

Top-Down Parsing

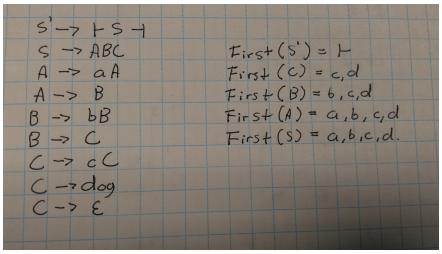
Start with our start state (S) and using the given derivation rules for our CFG, we can check if there exists a derivation from S to the given word.

Example 16.0.1. *Top-down parsing on a particular CFG using the word 'abywz'*



How can we determine which derivation rule to use? We use the implementation for top-down parsing, called LL(1) Parsing. Before we implement LL(1) Parsing, we need to outline a couple of methods:

- `first(`α`):` given a non-terminal α, find all of the possible tokens that can be derived from it. It returns the first character from those tokens:



- `follow(`α`)` returns the terminals that occur immediately following α
- `empty(`α: returns true if the given terminal or non-terminal α can disappear (i.e., $\alpha \to \varepsilon$). If α is a terminal, it returns false immediately, since terminals cannot disappear