CS241 – Week 6 Tutorial

LL(1) Parsing

Spring 2018

1 Top-down Parsing (LL(1))

1.1 Definitions

- LL(1) stands for *Left-to-right* scan of input, *Leftmost* derivations, 1 symbol of lookahead.
- An LL(1) parser is a *top-down* parser; it begins from the start symbol and finds a derivation for the input string.
- A grammar that can be parsed using the LL(1) algorithm is called an LL(1) grammar.
- The LL(1) prediction function for a grammar $G = (N, \Sigma, P, S)$ is defined as:

$$\text{Predict}(A, a) = \{(A \to \gamma) \in P : a \in \text{First}(\gamma) \vee (\text{Nullable}(\gamma) \wedge a \in \text{Follow}(A))\}$$

  Where:

$$\begin{aligned} \text{Nullable}(\gamma) &= \top \text{ iff } \gamma \stackrel{*}{\Rightarrow} \epsilon \\ \text{First}(\gamma) &= \{b \in \Sigma : \gamma \stackrel{*}{\Rightarrow} b\beta \text{ for some } \beta\} \\ \text{Follow}(A) &= \{c \in \Sigma : S' \stackrel{*}{\Rightarrow} \alpha A c \beta \text{ for some } \alpha, \beta\} \end{aligned}$$

- A grammar is LL(1) iff $|\text{Predict}(A, a)| \leq 1 \quad \forall A \in N, a \in \Sigma$.

1.2 Exercises

Consider the following context-free grammar G and its associated predictor table:

$$S' \to \vdash S \dashv \tag{0}$$
$$S \to aXYb \tag{1}$$
$$S \to XY \tag{2}$$
$$X \to pX \tag{3}$$
$$X \to \epsilon \tag{4}$$
$$Y \to q \tag{5}$$
$$Y \to \epsilon \tag{6}$$

	$\vdash$	$\dashv$	a	b	p	q
S'	0					
S		2	1		2	2
X		4		4	3	4
Y		6		6		5

1. Perform a top-down parse of the string $\vdash appqb \dashv$ and draw the parse tree.

2. Consider the grammar G', which is identical to G except with rule 1 ($S \to aXYb$) replaced by the rule $S \to Sab$. Why is it impossible to parse this grammar with an LL(1) parser? That is, what aspect of the grammar makes it not LL(1)?