

CS241 – Week 6 Tutorial Solutions

LL(1) Parsing

Spring 2018

1 Top-down Parsing Solutions

Consider the following context-free grammar G :

$$S' \rightarrow \vdash S \dashv \quad (0)$$

$$S \rightarrow aXYb \quad (1)$$

$$S \rightarrow XY \quad (2)$$

$$X \rightarrow pX \quad (3)$$

$$X \rightarrow \epsilon \quad (4)$$

$$Y \rightarrow q \quad (5)$$

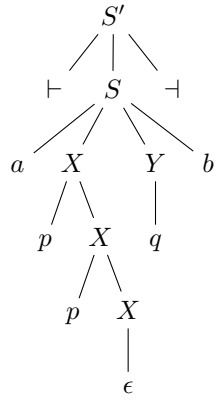
$$Y \rightarrow \epsilon \quad (6)$$

1. Algorithm:

- If there is a terminal on top of the stack and the same terminal is in the input, we consume it.
- If there is a nonterminal on top of the stack and the predictor table has an entry at the position of the nonterminal and the letter on top of the stack, we expand the nonterminal using that rule.
- We accept upon reading EOF.
- In any other case, we reject.

Action	Consumed Input	Stack	Remaining Input
initialize		S'	$\vdash appqb \dashv$
expand $S' \rightarrow \vdash S \dashv$		$\vdash S \dashv$	$\vdash appqb \dashv$
match \vdash	\vdash	$S \dashv$	$appqb \dashv$
expand $S \rightarrow aXYb$	\vdash	$aXYb \dashv$	$appqb \dashv$
match a	$\vdash a$	$XYb \dashv$	$ppqb \dashv$
expand $X \rightarrow pX$	$\vdash a$	$pXYb \dashv$	$ppqb \dashv$
match p	$\vdash ap$	$XYb \dashv$	$pqb \dashv$
expand $X \rightarrow pX$	$\vdash ap$	$pXYb \dashv$	$pqb \dashv$
match p	$\vdash app$	$XYb \dashv$	$qb \dashv$
expand $X \rightarrow$	$\vdash app$	$Yb \dashv$	$qb \dashv$
expand $Y \rightarrow q$	$\vdash app$	$qb \dashv$	$qb \dashv$
match q	$\vdash appq$	$b \dashv$	$b \dashv$
match b	$\vdash appqb$	\dashv	\dashv
match \dashv	$\vdash appqb \dashv$		

Reading the production rules that were applied from top to bottom gives a leftmost derivation. We can use this derivation to obtain the parse tree.



2. If we modify the predictor table accordingly for the row for S , notice that several of the cells have two entries each. Since we require that each cell in the predictor table has at most one entry, this grammar is not LL(1). In this case, the issue arises from the fact that left-recursive rules (i.e. $S \rightarrow Sab$) are not LL(1).