PS2 Theory

Erlend Paulsen Skaaden

Task 1: LL(1) parser

Task 1.1: Adapting the grammar

We are given the following grammar:

$$\begin{aligned} S &\rightarrow sCxLDy \\ C &\rightarrow c \\ L &\rightarrow L;I|I \\ I &\rightarrow i \\ D &\rightarrow d|\epsilon \end{aligned}$$

The problem with this grammar is that it is left-recursive. More specifically, it is the rule $L \rightarrow L;I|I$ that is creating problems. The problem is that when expanding L with i as the lookahead-symbol, we can't possibly know which production to choose. We can fix it by adding a new rule, which removes the recursion:

$$\begin{aligned} S &\rightarrow sCxLDy \\ C &\rightarrow c \\ L &\rightarrow IL' \\ L' &\rightarrow ;IL'|\epsilon \\ I &\rightarrow i \\ D &\rightarrow d|\epsilon \end{aligned}$$

This grammar is LL(1)-parsable.

Task 1.2: FIRST and FOLLOW sets

Non-Terminal	FIRST	FOLLOW	null?
S	s	$	no
C	c	x	no
L	i	d, y	no
L’	;	d, y	yes
I	i	d, y, ;	no
D	d	y	yes

The FIRST and FOLLOW set for the grammar

Task 1.3: LL(1) parsing table

Non-terminal	s	c	x	i	d	y	;
S	$S \rightarrow sCxLDy$						
C		$C \rightarrow c$					
L				$L \rightarrow IL'$			
L’					$L' \rightarrow \epsilon$	$L' \rightarrow \epsilon$	$L' \rightarrow ; IL'$
I				$I \rightarrow i$			
D					$D \rightarrow d \vert \epsilon$	$D \rightarrow \epsilon$	

LL(1) parsing table for the grammar

Task 2: Implementation

The rest of the assignment can be reviewed in the attached code repository.