UNIT 5: TOP-DOWN PARSING TECHNIQUES

We want to incorporate a repetitive sentence into a high-level language. The sentence can be represented by the following regular expression:

repeat (identifier | number) >> sentence^{+} <<

A program consists of at least one statement, where statements can be assignments, conditionals, and loops.

NOTE: The symbols "|" and "+" are part of the regular expressions, the others are part of the language.

It is required:

1. Define the grammar G that would generate valid programs of this programming language. Consider the assignment and conditional statements as terminal symbols of the grammar.

2. Obtain the LL(1) analysis table for the previous G grammar or an equivalent one that allows an LL(1) analysis.

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

SOLUTION:

A grammar to generate the language defined:

G = {assigment, condition, id, n, repeat, (,), <<, >>},{**S**, **S'**, **B**, **E**, **R**}, {**S**}
(1) **S**::= **E S'**
(2) **S'**::= **S**
(3) **S'**::= λ
(4) **E**::= assigment
(5) **E**::= condition
(6) **E**::= **B**
(7) **B**::= repeat (**R**
(8) **R**::= id) >> **S** <<
(9) **R**::= n) >> **S** <<

Σ_N	FIRST				FOLLOW				
S	assigment	repeat	condition		<<	$			
S'	λ	assigment	repeat	condition	<<	$			
E	assigment	repeat	condition		assigment	repeat	condition	<<	$
B	repeat				assigment	repeat	condition	<<	$
R	id	n			assigment	repeat	condition	<<	$

Table LL(1)		Σ_T									
		$	**(**	**)**	**<<**	**>>**	**assigment**	**condition**	**repeat**	**id**	**n**
Σ_N	**S**						1	1	1		
	S'	3			3		2	2	2		
	E						4	5	6		
	B								7		
	R									8	9