UNITS 7 AND 8: SEMANTIC ANALYSIS and ERROR HANDLING

We want to construct a compiler for a language of definition and application of sequential machines. In the language you can define as many machines and process as many strings as you want. The language format is as follows:

- To declare a sequential machine, the MS instruction is used:

```
MS name_of_the_sequential_machine
{
inputs { symbol₁, symbol₂,…,symbolₙ}
outputs { symbol₁, symbol₂,…,symbolₖ}
states { state₁, state₂, …}
transitions {
(state₁, symbol_input₁, state₂, symbol_output₁)
(state₁, symbol_input₁, state₂, symbol_output₁)
…
(state₁, symbol_input₁, state₂, symbol_output₁)
}
}
```

- For the sequential machine to process a string, it is used:

```
process ( name_of_automaton, string, initial_state)
```

The name of the sequential machine is a string of alphabetic characters. The statement of the set of states, the set of input and output symbols, and the set of transitions can be in any order, but they must always be included in every statement. The set of states, transitions, input and output symbols must have at least one value. An example of a sequential machine definition in this language would be:

```
MS Afirst
{
outputs {1,0}
states {a, c}
inputs {L,M,N}
transitions { (a,L,a,1)
(a,M,a,1)
(a,N,c,0)
(c,L,a,0)
(c,M,a,0)
(c,N,c,1)
}
}
process (Afirst, LLM, a)
process (Afirst, LLM, c)
```

To use the process function, the sequential machine used must be previously declared. The function displays, for the previous example, the following information:

```
MS: Afirst     Input: LLM  Output: 111
MS: Afirst     Input: LLM  Output: 011
```

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

It is required:

1. Define the grammar G that would generate valid sentences of this programming language.

2. Can the grammar G of the first exercise be used to perform an LL(1) analysis? If not, modify it so that it can. Generate the LL(1) table. Does this language require semantic verifications?

3. Does this language require semantic verifications?

SOLUTION:

A grammar that generates the language of the problem is defined as follows:

G = {**A**, **B**, **D**, **E**, **R**, **S**, **V**, **W**, **Z** }, {( ) , { } automatonFD string states final initial name recognize t transitions}, {**S**}

(1) **A**::= λ
(2) **A**::= **S**
(3) **B**::= states { **V** }
(4) **B**::= final { **U** }
(5) **B**::= initial { t }
(6) **B**::= transitions { **W** }
(7) **D**::= automatonFD name { **B B B B** }
(8) **E**::= **D**
(9) **E**::= **R**
(10) **R**::= recognize (name, string)
(11) **S**::= **E A**
(12) **U**::= λ
(13) **U**::= **V**
(14) **V**::= t **Z**
(15) **W**::= λ
(16) **W**::= (t , t , t) **W**
(17) **Z**::= λ
(18) **Z**::= , **V**

Σ_N	FIRST				FOLLOW				
S	automatonFD	recognize			$				
A	λ	automatonFD	recognize		$				
B	states	final	initial	transitions	}	states	final	initial	transitions
D	automatonFD				automatonFD	recognize	$		
E	automatonFD	recognize			automatonFD	recognize	$		
R	recognize				automatonFD	recognize	$		
U	λ	t			}				
V	t				}				
W	λ	(			}				
Z	λ	,			}				

Table LL(1)		Σ_T														
		$	**(**	**)**	**,**	**{**	**}**	**automatonFD**	**string**	**states**	**final**	**initial**	**name**	**recognize**	**t**	**transitions**
Σ_N	**S**							11						11		
	A	1						2						2		
	B									3	4	5				6
	D							7								
	E							8						9		
	R													10		
	U						12								13	
	V														14	
	W		16				15									
	Z				18		17									

It is necessary to do semantic checks. In order to use the recognize function, the automaton must first be defined. This control is simple to perform, it is of the same type as that necessary to control that a variable must be declared before it is used. The other type of control is the one referring to the complete declaration of an automaton. That is, it must meet the following conditions:

- It must be defined the states, axiom, final states final and production rules.
- The set of states must be not empty.
- The axiom must be not empty.
- The axiom and the final states must be declared as states.
- The transitions must be among states in the set of states.

All these controls must be performed generating a data structure that stores the definition of the automaton and then invoking a function that checks its integrity.