University of Zagreb
Faculty of Electrical Engineering and Computing
Department of Electronics, Microelectronics, Computer and Intelligent Systems

INTRODUCTION TO COMPUTER THEORY

Ak. year 2022/2023

3. laboratory exercise

In the third laboratory exercise, the task is to be realized programmatically**deterministic push automaton simulator**(**DPA**) which accepts strings with an acceptable state. The input to the automaton simulator is a textual record of its definition and an input string, and the output is a textual record of the sets of states in which the automaton was for each loaded character of the input string of the stack character string and information about the acceptability of the input string. The work of the program that needs to be done as part of the exercise is shown in Figure 1.

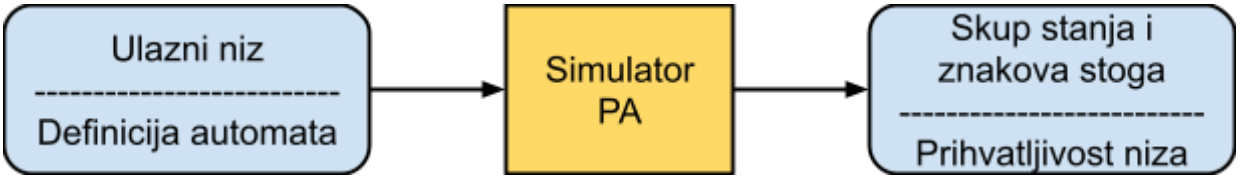

Figure 1 - Principle operation of the PA simulator

The format for writing the input string and PA definition is:

- 1 line: Input strings separated by |. The symbols of each individual string are separated by a comma.
- 2nd line: Set of states separated by a comma
- Line 3: Comma-separated set of input characters
- Line 4: Comma-separated set of stack characters
- Line 5: Comma-separated set of acceptable states
- 6th line: Initial state
- Line 7: Start stack character
- 8th line and all other lines: Format transition function

currentState, inputCharacter, characterTherefore->newState, string of CharactersTherefore

According to the definition in the textbook, the stack characters that are placed on the stack in the transition are placed on the stack from right to left, i.e. the character at the top of the stack after the transition is the leftmost character in the sequencestring of characters Therefore.States, input characters, stack characters and acceptable states will be separated by a comma (if there are more than one). An empty string character**ε**is replaced by a sign **$.**

The automaton described by the input file will satisfy the following constraints:

1) The names of all states are strings of lowercase and uppercase letters of the English alphabet and decimal digits. The length of the state name will not exceed 20 characters. The number of machine states will not exceed 20. Each state of the machine will be specified exactly once.
2) The set of input characters will be a subset of the set of lowercase letters of the English alphabet and decimal digits. Each input character will appear exactly once. Each input character is represented **exactly one**a character belonging to lowercase letters of the English alphabet or decimal digits. The first line of the file, which contains the input strings, will not contain more than 1500 characters.

3) The character set will therefore be a subset of the upper case set of the English alphabet. Each character will therefore appear exactly once. Each character is therefore represented**exactly one**capital letter of the English alphabet.
4) The set of acceptable states will contain zero or more states from the set of states. Each acceptable state will be specified exactly once.
5) Each line of the transition function will have a unique triple on the left side of the character string ->: currentState, inputSign, signTherefore.

Example of a PA definition input file that accepts the language {w2w_R| w(0+1)*}is shown in Figure 2.

```
01      0|0,2,0|1,2,0
02      q1,q2,q3
03      0,1,2
04      J, N, K
05      q3
06      q1
07      K
08      q1,0,K->q1,NK
09      q1,1,K->q1,JK
10      q1,0,N->q1,NN
11      q1,1,N->q1,JN
12      q1,0,J->q1,NJ
13      q1,1,J->q1,JJ
14      q1,2,K->q2,K
15      q1,2,N->q2,N
16      q1,2,J->q2,J
17      q2,0,N->q2,$
18      q2,1,J->q2,$
19      q2,$,K->q3,$
```

Figure 2 - Example of PA definition

The completed PA simulator should be printed in which**condition**found the automaton for each input character of a particular string,**the contents of the stack**you**string acceptability**(0 or 1) after the input string has been processed. The characters on the stack are located next to the corresponding status character, separated by the sign #, while they are pairs state#signseparated by |. If there is nothing on the stack (the stack is empty), then a character is printed**$**as an empty stack marker. If the input string is not fully processed and does not exist

push automaton transition for the current input character, the current state and the character at the top of the stack, the automaton should printfailinstead of moneystate#sign,after which the machine ends its operation. String acceptance is at the end of the printout and is separated by |. The results for each input string are printed on a separate line, that is, each output line is separated by a newline character (\n). Each record starts with a set containing the initial state of the automaton and the initial stack character. An example of the output for the PA and input strings defined in Figure 2 is shown in Figure 3. The numbers on the left side of the figure indicate lines and are not part of the output.

```
01      q1#K|q1#NK|0
02      q1#K|q1#NK|q2#NK|q2#K|q3#$|1
03      q1#K|q1#JK|q2#JK|fail|0
```

Figure 3 - Example of the automaton simulator output

Important: In the case of acceptance of the sequence (the automaton has read the entire input sequence and is in an acceptable state), the last configuration that is printed is the configuration in which the acceptance condition is met, i.e. .....|qn#K1...Km|1.If epsilon transitions from the acceptable state qn are defined, possible further configurations of the automaton are not printed.

Notes:

1) It is not necessary to check the correctness of the formatting of the input file or the correctness of the automaton. There will be no overlap between the state set and the alphabet symbol set.

2) The time limit on program execution for any input definition of the automaton is 10 seconds

3) The entry point for Java solutions should be in the SimPa class, and the entry point for Python solutions should be in the SimPa.py file.