

Automaten en Formele Talen Assignment 3

Execution

Deadline: 23:59, Wednesday, May 25, 2018

Inge Bethke

Created by Daan de Graaf

Last changes on: May 5, 2018 (Bas van den Heuvel)

Background

After parsing the source code, we end up with a postfix notation of the source. To solve the expressions, we will use a Turing Machine (TM) with unary notation on the tape. After executing the (postfix) expression with the TM, the tape should contain the solution to the expression.

Assignment 3 (100 points)

The basics of the TM are provided in the framework. The tape class as well as the TM class both feature different methods which can be used. Take your time to carefully read the code. You are going to need to use the methods provided in order to run the TM.

Assignment 3.1: Running the TM (20 points)

After looking through the framework and understanding the implementation of the TM, you will need to implement the run method of the TM. It should take an expression from the register and check each tuple containing a symbol and type. There are four situations:

- If the symbol is a variable, check the *ram* (random access memory) for an occurrence of the variable and print it to the tape. If there is no occurrence, save the variable as a pending variable such that we can write the result back to it after we are done.
- If the symbol is a value, write it to the tape.
- If the symbol is an operator, perform an execution (see Assignment 3.2).
- If the symbol is an assignment, write the result on the tape to the pending variable. Now the variable can be used in other expressions.

Assignment 3.2: Execution (80 points)

There are four possible tape operations: *add*, *subtract*, *multiply* and *divide*. In the TM code you should already have found the `executables` method. Given is the TM transition table for addition. After executing this operator the two numbers on the tape will be added together. Go ahead and try it yourself.

Your goal is to write the same operations for the remaining operations. Take a look at Figure 1 in Appendix A to find a different representation of this operation. It is wise to work out the workings of the other operations in this way, before trying to implement it in the code.

If all done correctly, the result of the expression can be found in the ram of the TM and can be used by the next expression in the source code. Try this yourself to see if it works.

Submission

Put (*only*) your version of TM.py in a zip or tar with your name and student number and upload it to Canvas **before 23:59, Wednesday, May 25, 2018**. If your code does not work with Python 3, your work will not be graded and you will receive 0 points.

A Addition Automaton

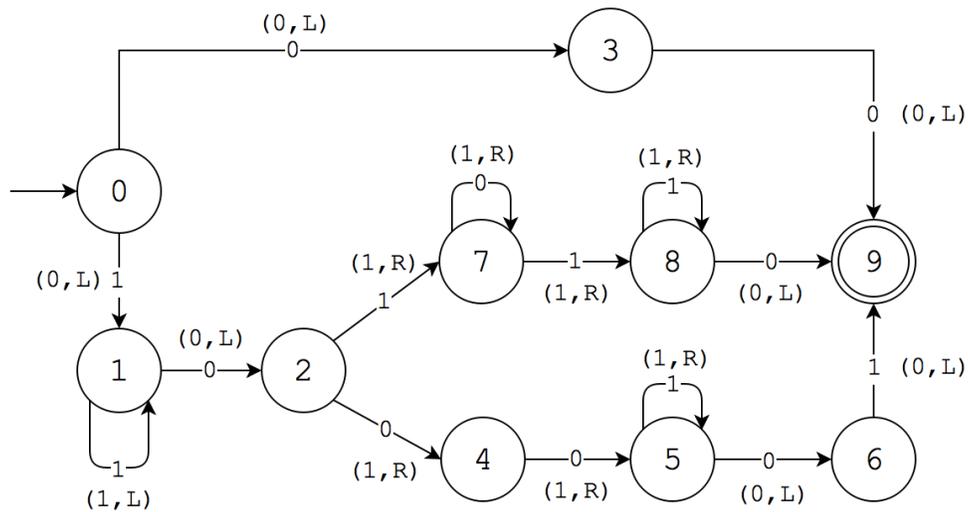


Figure 1: Addition Automaton in different form