3 Build an interpreter

The following grammar in the BackusNaur form(BNF) describes a Turing Machine and its input tape. It is stored in a txt file that you can download from bCourses. The Input $< TM >$ is composed by a $< machine >$ and an $< input\_list >$. The $< machine >$ may expand into a

$< initial\ state >$, a $< final\_state\_list >$ and a $< transition\_list >$. The $< transition\_list >$ may contain a list of $< transition > s$. The $< transition >$ consists of a current state, an input, an output, a movement and a next state. The $< input\_list >$ contains all of inputs.

```
<TM>:=’{‘ <machine> ’},{’ <input_list> ’}’
<machine> := ‘{’<initial_state> ‘},{’<final_state_list> ‘},{’<transition_list> ‘}’
<initial_state>:= ‘q0’
<final_state_list> := <final_state> ’,’ <final_state_list> | <final_state>
<final_state> := identifier
<transition_list>:= <transition> ‘,’ <transition_list> |  <transition>
<transition> := ‘{’ <state> ’,’ <input> ‘,’ <output> ‘,’ <move> ‘,’ <state> ’}’
<state> := identifier
<output> := identifier
<move> := ‘R’|’L’
```

Write a program in your computer to read the input($< TM >$) as above and come up with your output.

Here are examples for you to test your code. Example 1 is a unary-add-one TM with a input tape bb111bb. It is also stored in the txt file that you can download from bCourses.

```
{{q0},{qf},{{q0,b,b,R,q0},{q0,1,1,R,q1},{q1,1,1,R,q1},{q1,b,1,R,qf}}},{b,b,1,1,1,b,b}
```

Your program should read the txt file, execute the TM and output the following string:

```
{b,b,1,1,1,1,b},{qf}
```

Here is example 2 for you to test your code. It is a unary-minus-one TM with a input tape bb111bb.

```
{{q0},{qf},{{q0,b,b,R,q0},{q0,1,1,R,q1},{q1,1,1,R,q1},{q1,b,b,L,q2},{q2,1,b,R,qf}}},
{b,b,1,1,1,b,b}
```

Your program should read the txt file, execute the TM and output the following string:

```
{b,b,1,1,b,b,b},{qf}
```

Here is example 3 for you to test your code. You take a binary number bb111bb as input and add 1.

```
{{q0},{qf},{{q0,b,b,R,q0},{q0,1,1,R,q1},{q0,0,0,R,q1},{q1,1,1,R,q1},{q1,0,0,R,q1},
{q1,b,b,L,q2},{q2,1,0,L,q2},{q2,0,1,R,qf},{q2,b,1,R,qf}}},{b,b,1,1,1,b,b}
```

Your program should read the txt file, execute the TM and output the following string:

```
{b,1,0,0,0,b,b},{qf}
```

The turing machine accepts the number which only contains 1, and any number contains 0 will be considered unacceptable. If the number only contains 1, the final state will be qa, otherwise it will be qu.

```
{{q0},{qa,qu},{{q0,b,b,R,q0},{q0,1,1,R,q1},{q0,0,0,R,qu},{q1,1,1,R,q1},{q1,b,b,L,qa},
{q1,0,0,R,qu}}},{b,b,1,1,1,b,b}
```

Your program should read the txt file, execute the TM and output the following string:

```
{b,b,1,1,1,b,b},{qa}
```