Turing Machine Documentation

Ryan Peruski, Maria Hernandez

November 17, 2023

1 Roles

- Ryan Peruski: Created initial stack design and initial templates for the documentation
- Maria Hernandez

2 Definition

This Turing machine uses the power of the stack to do a simple check for C-code. It checks for correct parenthesis, square bracket, and curly brace placement. It will accept any code that has correct placement of these characters, and it will reject any code that does not.

3 States, Transitions, Image

The Turing machine operates by moving between states and performing transitions on the tape. The states and transitions are labelled as follows:

- *Red* are all the reject states (q_{reject})
- *Green* are all the accept states (q_{accept})
- *Blue* is the Queue's add States
- *Yellow* is the Queue's remove States (1st part: *deletion*)
- *Orange* is the Queue's remove States (2nd part: *movement*)
- *Magenta* is the Stack with the add states
- *Cyan* is the Stack with the remove states
- *Black* are the initial states to set up the #
- *Beige* transition states to go to either stack or queue / validity of the string

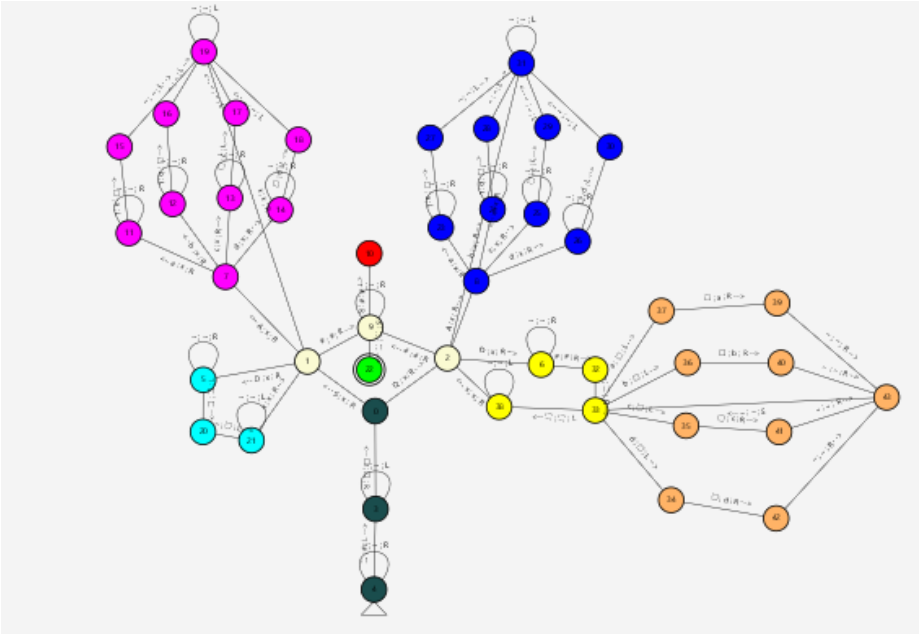

Figure 1: Turing Machine.

The transitions can be summarized as follows:

- xxxxxx
- xxxxxx

Finally, the Turing Machine is shown in Figure 1.

4 Instructions

This Turing Machine can input any C-code (or code in general that uses parenthesis) and it will place left parenthesis on the stack and pop right parenthesis. If it does not see a parenthesis of the right type or none at all, it will reject. If the code ends with anything on the stack, it will reject. If the code is empty, it will accept.

The only thing that the machine will reject on is a #, as this is a marker between the code and the stack.

Note that machine CANNOT handle whitespace ANYWHERE Also, when using a backslash within quotes, only the quotes, newlines, and null characters are accepted. Of course, we could add more support for backslash characters, but we didn't want to make the machine too needlessly complicated.

5 Examples

Here are some examples of input and output for the Turing machine:

- `Input: v[i]=a[i][j];, Output:  xxxxxxxxxxxxx#bqa`
- `Input: for(;;)break;, Output:  xxxxxxxxxxxxx#bqa`
- `Input: {x=7}, Output: xxxxx#bqa`
- `Input: for(i=0;i<v.size();i++)v[i]=2;, Output: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx#bqa`
- `Input: v[i]=a[i, Output:  xxxxxxxx#Rqr`
- `Input: for(;;, Output:  xxxxxx#Rqr`
- `Input: printf("HelloWorld!\n");, Output:  xxxxxxxxxxxxxxxxxxxxxx#bqa`
- `Input: printf("Hereisa{(");, Output:  xxxxxx#qa`
- `Input: """, Output: xxx#qr`
- `Input: 'a''\n', Output: xxxxxxx#bqa`
- `Input: /*WeSupportComments!*/, Output:  xxxxxxxxxxxxxxxxxxxx#bqa`
- `Input: /*WeSupportComments!{{{{*/, Output:  xxxxxxxxxxxxxxxxxxxxxxx#bqa`

6 Conclusion

The Turing machine is a powerful tool for performing computations on input tapes. It has applications in computer science, mathematics, and other fields.