

1 Introduction to Turing Machine Language

In this section, you are given some programs in Turing Machine Language (TML). They will be used to explain the syntax of the programming language and how they can be run on tapes.

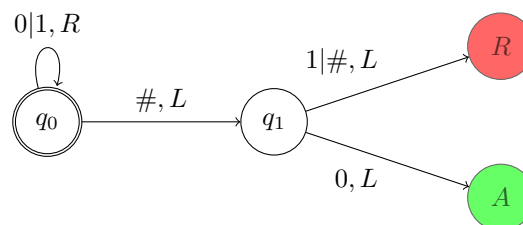
- isDiv2:

```
1 // checks whether a binary number is divisible by 2
2 alphabet = {0, 1}
3 module isDiv2 {
4     // move to the end
5     while 0, 1 {
6         move right
7     } if blank {
8         move left
9         // check last letter is 0
10        if 0 {
11            accept
12        } if 1, blank {
13            reject
14        }
15    }
16 }
```

- isDiv2Rec:

```
1 // checks whether a binary number is divisible by 2 recursively
2 alphabet = {0, 1}
3 module isDiv2Rec {
4     // recursive case: not at the end => move closer to the end
5     if 0, 1 {
6         move right
7         goto isDiv2Rec
8     }
9     // base case: at the end => check final letter 0
10    if blank {
11        move left
12        if 0 {
13            accept
14        } if 1, blank {
15            reject
16        }
17    }
18 }
```

Both isDiv2 and isDiv2Rec correspond to the following Turing Machine (TM):



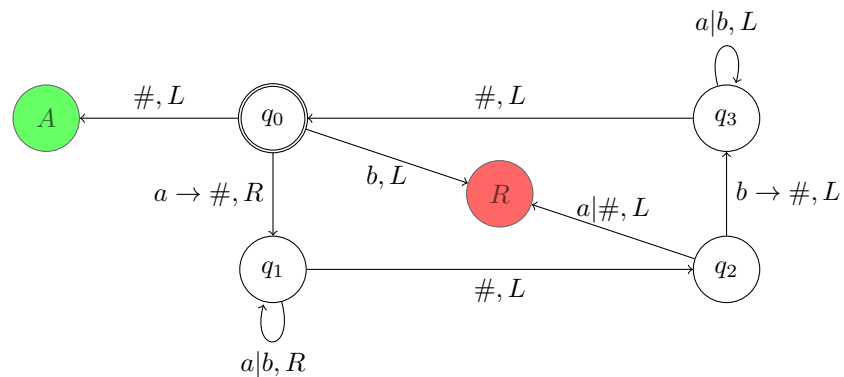
- aNbN:

```

1 // accepts strings of the form anbn for some n ≥ 0
2 // e.g. blank, ab, aabb, aaabbb, etc.
3 alphabet = {a, b}
4 module aNbN {
5     // empty string => base case, accept
6     if blank {
7         accept
8     }
9     // cannot start with b
10    if b {
11        reject
12    }
13    // starts with a => remove the last b
14    if a {
15        changeto blank
16        move right
17        // move to the end
18        while a, b {
19            move right
20        } if blank {
21            move left
22            // the last letter must be a b
23            if a, blank {
24                reject
25            } if b {
26                changeto blank
27                move left
28                // move back and restart
29                while a, b {
30                    move left
31                } if blank {
32                    move right
33                    goto aNbN
34                }
35            }
36        }
37    }
38 }

```

The program aNbN corresponds to the following TM:



2 Identifying TML Programs

In this section, you are presented with TML programs. You will be given some tape values to run the program in and decode what values the program accepts. You are encouraged to use the website to try and solve this.

1. Consider the following TML Program:

```
1  alphabet = {0, 1}
2  module mystery1 {
3      while 0, 1 {
4          move right
5      } if blank {
6          move left
7          if blank, 0 {
8              reject
9          } if 1 {
10             move left
11             if blank, 1 {
12                 reject
13             } if 0 {
14                 accept
15             }
16         }
17     }
18 }
```

- (a) Does the program accept the values:

- i. 10 (NOTE: This is 2 in decimal)

Solution:

- ii. 1

Solution:

- iii. 100 (NOTE: This is 4 in decimal)

Solution:

- iv. 101 (NOTE: This is 5 in decimal)

Solution:

- v. 110 (NOTE: This is 6 in decimal)

Solution:

- (b) Describe the values this program accepts.

Solution:

2. Consider the following TML program:

```
1  alphabet = {a, b}
2  module mystery2 {
3      if blank {
4          accept
5      } if b {
6          reject
7      } if a {
8          changeto blank
9          move right
10         if b, blank {
11             reject
12         } if a {
13             changeto blank
14             move right
15             while a, b {
16                 move right
17             } if blank {
18                 move left
19                 if a, blank {
20                     reject
21                 } if b {
22                     changeto blank
23                     move left
24                     if a, blank {
25                         reject
26                     } if b {
27                         changeto blank
28                         move left
29                         while a, b {
30                             move left
31                         } if blank {
32                             move right
33                             goto mystery2
34                         }
35                     }
36                 }
37             }
38         }
39     }
40 }
```

(a) Does the program accept the values:

i. *ab*

Solution:

ii. *aabb*

Solution:

iii. *abba*

Solution:

iv. *bab*

Solution:

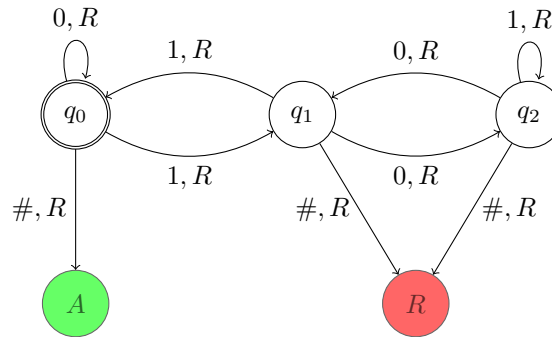
(b) Describe the values this program accepts.

Solution:

3 Identifying TMs

In this section, you are presented with TMs. You will be given some tape values to run the program in and decode what values the program accepts. Since the website can only execute TML programs, you are also given the TML program for the code, but it is not comprehensible like the previous programs; you will likely find it easier to understand the TM than the program (which you should do!).

1. Consider the following TM FSM:



You are given a basic representation of this FSM as code in Teams. The file is called mystery3.

- (a) Does the TM accept the values:

- i. 11 (NOTE: This is 3 in decimal)

Solution:

- ii. 10 (NOTE: This is 2 in decimal)

Solution:

- iii. 1

Solution:

- iv. 110 (NOTE: This is 6 in decimal)

Solution:

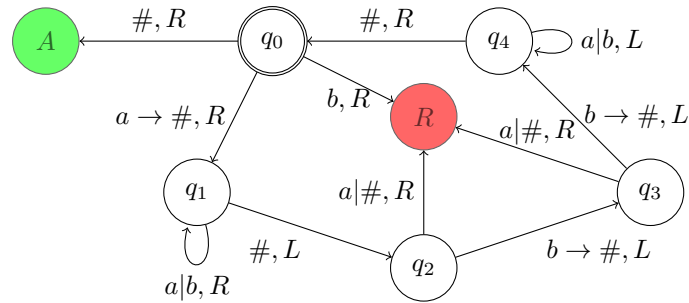
- v. 1001 (NOTE: This is 9 in decimal)

Solution:

- (b) Describe the values this program accepts.

Solution:

2. Consider the following TM FSM:



You are given a basic representation of this FSM as code in Teams. The file is called mystery4.

(a) Does this TM accept the values:

i. *ab*

Solution:

ii. *abb*

Solution:

iii. *aabbbb*

Solution:

iv. *bab*

Solution:

v. *abba*

Solution:

(b) Describe the values this program accepts.

Solution:

4 Writing TML Programs

Following a similar syntax to the code given above, write the following programs. You are free to use the website to check the accuracy of the program while writing the programs.

1. divisibility by 4 in binary iteratively [HINT: Go to the end and check for 2 zeros. Allow 0 as well.]

Solution:

2. divisibility by 4 in binary, recursively.

Solution:

The remaining questions are optional. If you attempt them, you can answer them in the survey.

3. strings of the form $a^n b^m c^{n+m}$
4. strings of the form $a^n b^n c^n$
5. HARD: check there are same number of a 's and b 's

5 Appendix- Links

Below are the links to the website and the survey:

- Website: [Link to Website](#)
- Survey: [Link to Survey](#)