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.

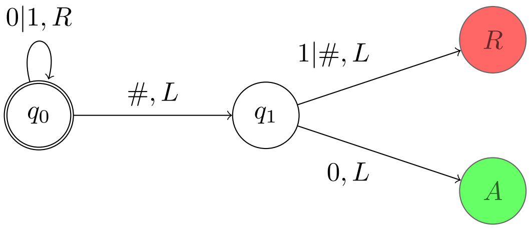
- `isDiv2`:

```
// checks whether a binary number is divisible by 2
alphabet = {0, 1}
module isDiv2 {
    // move to the end
    while 0, 1 {
        move right
    } if blank {
        move left
        // check last letter is 0
        if 0 {
            accept
        } if 1, blank {
            reject
        }
    }
}
```

- `isDiv2Rec`:

```
// checks whether a binary number is divisible by 2 recursively
alphabet = {0, 1}
module isDiv2Rec {
    // recursive case: not at the end => move closer to the end
    if 0, 1 {
        move right
        goto isDiv2Rec
    }
    // base case: at the end => check final letter 0
    if blank {
        move left
        if 0 {
            accept
        } if 1, blank {
            reject
        }
    }
}
```

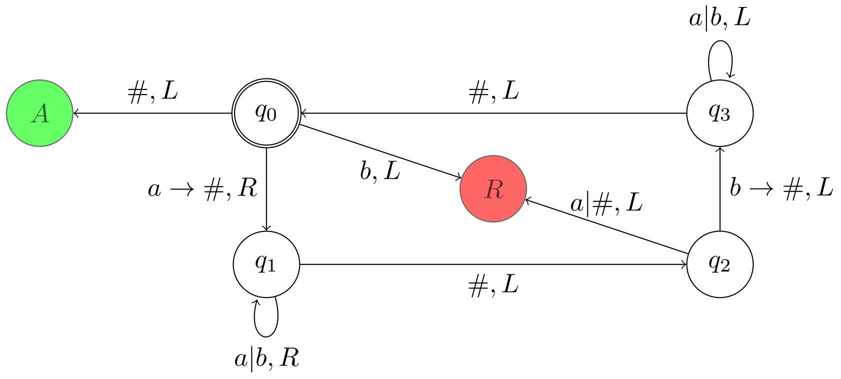
Both `isDiv2` and `isDiv2Rec` correspond to the following Turing Machine (TM):



- `aNbN`:

```
// accepts strings of the form a^nb^n for some n >= 0
// e.g. blank, ab, aabb, aaabbb, etc.
alphabet = {a, b}
module aNbN {
    // empty string => base case, accept
    if blank {
        accept
    }
    // cannot start with b
    if b {
        reject
    }
    // starts with a => remove the last b
    if a {
        changeto blank
        move right
        // move to the end
        while a, b {
            move right
        } if blank {
            move left
            // the last letter must be a b
            if a, blank {
                reject
            } if b {
                changeto blank
                move left
                // move back and restart
                while a, b {
                    move left
                } if blank {
                    move right
                    goto aNbN
                }
            }
        }
    }
}
```

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:

```
alphabet = {0, 1}
module mystery1 {
    while 0, 1 {
        move right
    } if blank {
        move left
        if blank, 0 {
            reject
        } if 1 {
            move left
            if blank, 1 {
                reject
            } if 0 {
                accept
            }
        }
    }
}
```

(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:

```
alphabet = {a, b}
module mystery2 {
    if blank {
        accept
    } if b {
        reject
    } if a {
        changeto blank
        move right
        if b, blank {
            reject
        } if a {
            changeto blank
            move right
            while a, b {
                move right
            } if blank {
                move left
                if a, blank {
                    reject
                } if b {
                    changeto blank
                    move left
                    if a, blank {
                        reject
                    } if b {
                        changeto blank
                        move left
                        while a, b {
                            move left
                        } if blank {
                            move right
                            goto mystery2
                        }
                    }
                }
            }
        }
    }
}
```

(a) Does the program accept the values:

i. *ab*

Solution:

ii. *aabb*

Solution:

iii. *abba*

Solution:

iv. *bab*

Solution:

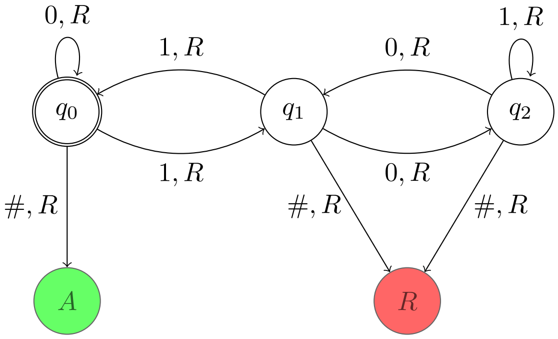
(b) Describe the values this program accepts.

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:

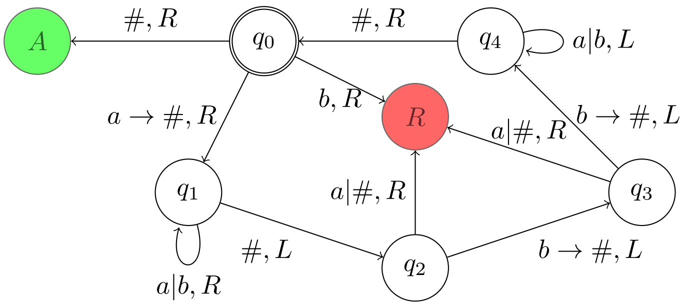
v. 1001 (NOTE: This is 9 in decimal)

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