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.

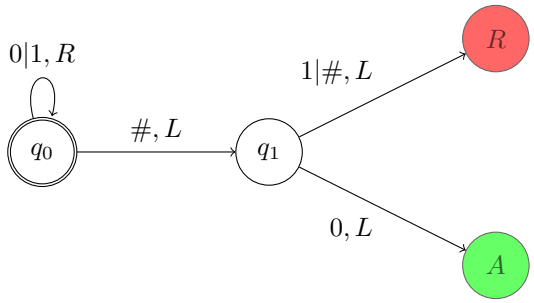
- `isDiv2`:

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

- `isDiv2Rec`:

```
// checks whether a binary number is divisible by 2 recursively
alphabet = {0, 1}
module isDiv2Rec {
    if 0, 1 {
        move right
        goto isDiv2
    } if blank {
        move left
        if 0 {
            accept
        } if 1, blank {
            reject
        }
    }
}
```

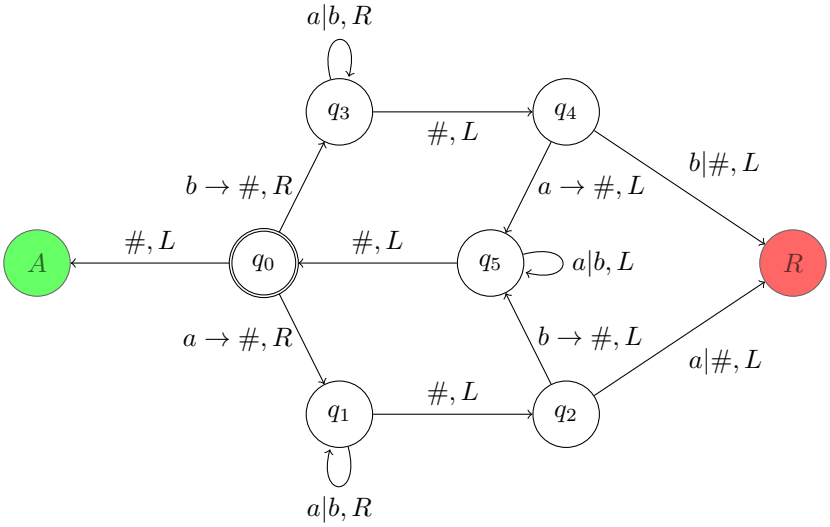
Both `isDiv2` and `isDiv2Rec` correspond to the following Turing Machine (TM):



- palindrome:

```
// checks whether the given string is a palindrome, i.e. the string equals its reverse
alphabet = {a, b}
module palindrome {
    if blank {
        accept
    }
    // starts with a => check ends with a
    if a {
        changeto blank
        move right
        // move to the end
        while a, b {
            move right
        } if blank {
            move left
            // cannot end with a b
            if b {
                reject
            } if a, blank {
                changeto blank
                move left
                goto restart
            }
        }
    }
    // starts with b => check ends with b
    if b {
        changeto blank
        move right
        // move to the end
        while a, b {
            move right
        } if blank {
            move left
            // cannot end with an a
            if a {
                reject
            } if b, blank {
                changeto blank
                move left
                goto restart
            }
        }
    }
}
// go to the start and restart
module restart {
    while a, b {
        move left
    } if blank {
        move right
        goto palindrome
    }
}
```

The program `palindrome` 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 mystery {
    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. $2 = 10$

      Solution:

      ii. $1 = 1$

      Solution:

      iii. $4 = 100$

      Solution:

      iv. $5 = 101$

      Solution:

      v. $6 = 110$

      Solution:

   (b) Describe the values this program accepts.

   Solution:

2. Consider the following TML program:

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

(a) Does the program accept the values:

i. *ab*

Solution:

ii. *aab*

Solution:

iii. *abb*

Solution:

iv. *abba*

Solution:

v. *abab*

Solution:

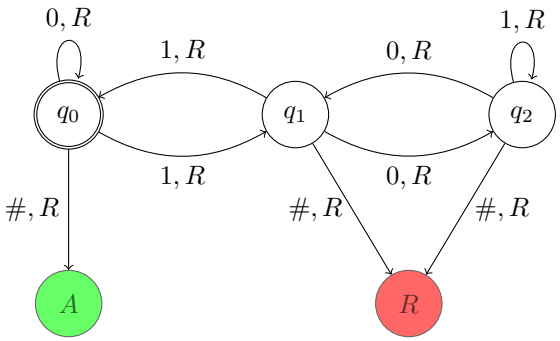
(b) Describe the values this program accepts.

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.

(a) Does the TM accept the values:

i. $2 = 10$

Solution:

ii. $1 = 1$

Solution:

iii. $6 = 100$

Solution:

iv. $5 = 101$

Solution:

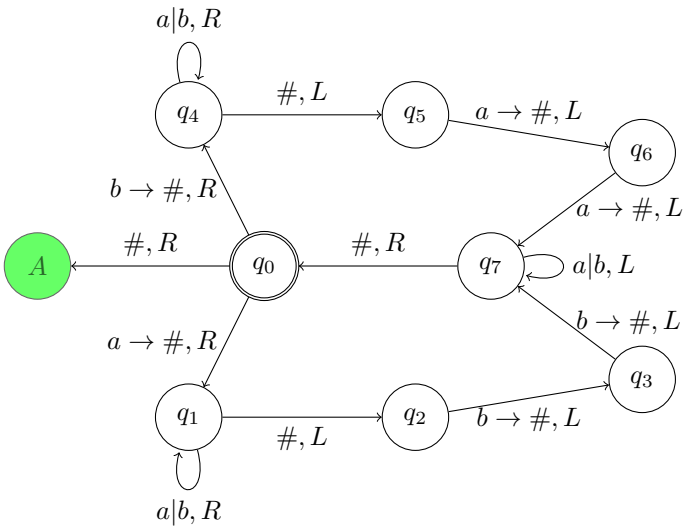
v. $8 = 110$

Solution:

(b) Describe the values this program accepts.

Solution:

2. Consider the following TM FSM:



NOTE: The missing transitions go to the reject state, i.e. q_2, q_3 to $a|\#$ and q_5, q_6 to $b|\#$ are rejected. You are given a basic representation of this FSM as code in Teams.

(a) Does this TM accept the values:

i. *ab*

Solution:

ii. *abb*

Solution:

iii. *aabb*

Solution:

iv. *bbaaaa*

Solution:

v. *abba*

Solution:

vi. *abab*

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:

3. check all a's come before the b's

Solution:

4. strings of the form $a^n b^n$ [HINT: this is a special version of palindrome.]

Solution:

5. HARD: check there are same number of a's and b's

Solution: