Turing Machine to accept Palindromes

Ankan Bal

September 23, 2020

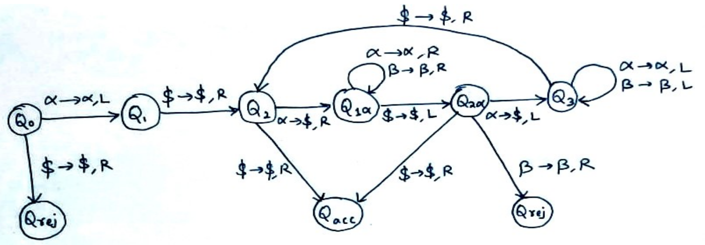
1 Statement

Given a string, we need to construct a Turing Machine that can accept palindrome strings. The strings can be made of English alphabets, digits, space symbol or combinations of them.

2 Idea

The idea behind our Palindrome Checker is that we will convert one character from beginning to blank symbol and same character from end to blank symbol at a time. As such if all characters are converted to blank symbols that means the string is a palindrome. If at any moment we find a character that is not same at the extreme ends of a string that means it is not a palindrome. The Turing machine that can do that is shown with a diagram in the next section.

3 Diagram



Turing Machine $M = (Q, \Sigma, \tau, \delta, Q_0, \$, F)$

$Q = \{Q_0, Q_1, Q_2, Q_3, Q_{acc}, Q_{rej}, Q_{1\alpha}, Q_{2\alpha}\}$

$\Sigma = \{0 \cdots 9, a \cdots z, Z\}$

$\$$ = Blank Symbol

Q_0 = Initial State

$F = \{Q_{acc}\}$

$\tau = \{\$\}$

where

$\alpha \in \Sigma$

$\beta \in \Sigma - \{\alpha\}$

$Z \to$ Space symbol

Here we used the symbol α and β to represent that the two states between Q2 and Q3 can change depending on the character we discover in Q2, hence it is like storing the character and comparing it to the character at the end of the string. Therefore the number of states Q has in total is 80 (2 states Q1α and Q2α for all 37 characters each in the input alphabets and 6 states common to all of them).

The tape alphabet τ consists of only \$ since that is the only symbol we used to replace the input alphabets.

In the expression the Z symbol is represented as a space symbol, but of course in the program while giving a sentence input we are supposed to give space as such and not as Z.

The transition function is explained in the below table.

curr. state	curr. char.	new char.	next state	direction
Q0	$\neq$ \$	same	Q1	L
	\$	same	Qrej	R
Q1	\$	same	Q2	R
Q2	$\neq$ \$	\$	Q1α	R
	\$	same	Qacc	R
Q1α	$\neq$ \$	same	Q1α	R
	\$	same	Q2α	L
Q2α	$\neq$ \$	\$	Q3	L
	\$	same	Qacc	R
	$\neq$ \$ and $\neq \alpha$	same	Qrej	R
Q3	$\neq$ \$	same	Q3	L
	\$	same	Q2	R

4 Explanation

The Turing machine we have created here is based on the following transition rules:-

- **Q2** -> The third state where we read some character α which belongs to the set of input alphabets, converts it to blank symbol and then move the tape head to the right, note that if we find this symbol blank that means all characters are matched and as such it was an even palindrome and we reach accepting state.
- **Q1**α -> The next state is the state after reading some character α which belongs to the input alphabets, we keep going right while keeping the alphabets as it is until we reach the blank symbol, then we move left by one and keep the blank symbol as it as.
- **Q2**α -> The next state is the state we reach after moving left by one, here three things can happen; if we encounter the α character then we change it to blank symbol and move left by one, if we find a blank symbol then it was an odd palindrome which completely matched and we reach

the accepting state, if however we encounter a symbol β which belongs to the input alphabets but not equal to α then it means there is a character mismatch and we reach the rejecting state.

- **Q3** -> This is the state we reach when symbol α is found in the state Q2α, here we keep moving left while keeping the alphabets as it is until we reach the blank symbol, then move right by one and keep the blank symbol as it is, reaching the state Q2.
- **Qacc** -> This is the accepting/final state as to when we finish matching every character for palindrome.
- **Qrej** -> This is the rejecting state as to when there is a character mismatch and hence the string is not palindrome.

The state Q2 has a problem as to if we put an empty string it is still going to accept it as a valid palindrome since we put an accepting state from it when it finds a blank symbol. So in order to solve that we put the first two states which will basically check if the string is empty or not. Rules are as follows:

- **Q0** -> This is where the Turing machine starts where it has two transitions, if it encounters a blank symbol then it will reach reject state since it means there is no string in the first place, else it will move tape head by left once and reach state Q1.
- **Q1** -> This is the state after we find an input alphabet in the tape indicating the presence of a string and move the tape head by right once so as to position it in the beginning of the tape string and thus reach state Q2.

5 Example

5.1 Example 1

String:- malayalam
Blank Symbol:- $

$$alayalam$ m->$,R
$$alayalam$ a->a,R
$$alayalam$ l->l,R
$$alayalam$ a->a,R
$$alayalam$ y->y,R
$$alayalam$ a->a,R
$$alayalam$ l->l,R
$$alayalam$ a->a,R
$$alayalam$ m->m,R
$$alayalam$ $->$,L
$$alayala$$ m->$,L

```
$$alayala$$ a->a,L
$$alayala$$ l->l,L
$$alayala$$ a->a,L
$$alayala$$ y->y,L
$$alayala$$ a->a,L
$$alayala$$ l->l,L
$$alayala$$ a->a,L
$$alayala$$ $->$,R
$$$layala$$ a->$,R
$$$layala$$ l->l,R
$$$layala$$ a->a,R
$$$layala$$ y->y,R
$$$layala$$ a->a,R
$$$layala$$ l->l,R
$$$layala$$ a->a,R
$$$layala$$ $->$,L
$$$layal$$$ a->$,L
$$$layal$$$ l->l,L
$$$layal$$$ a->a,L
$$$layal$$$ y->y,L
$$$layal$$$ a->a,L
$$$layal$$$ l->l,L
$$$layal$$$ $->$,R
$$$$ayal$$$ l->$,R
$$$$ayal$$$ a->a,R
$$$$ayal$$$ y->y,R
$$$$ayal$$$ a->a,R
$$$$ayal$$$ l->l,R
$$$$ayal$$$ $->$,L
$$$$aya$$$$ l->$,L
$$$$aya$$$$ a->a,L
$$$$aya$$$$ y->y,L
$$$$aya$$$$ a->a,L
$$$$aya$$$$ $->$,R
$$$$$ya$$$$ a->$,R
$$$$$ya$$$$ y->y,R
$$$$$ya$$$$ a->a,R
$$$$$ya$$$$ $->$,L
$$$$$y$$$$$ a->$,L
$$$$$y$$$$$ y->y,L
$$$$$y$$$$$ $->$,R
$$$$$$$$$$$ y->$,R
String accepted, all characters converted to blank
```

5.2 Example 2

String:- derived
Blank Symbol:- $

```
$$erived$ d->$,R
$$erived$ e->e,R
$$erived$ r->r,R
$$erived$ i->i,R
$$erived$ v->v,R
$$erived$ e->e,R
$$erived$ d->d,R
$$erived$ $->$,L
$$erive$$ d->$,L
$$erive$$ e->e,L
$$erive$$ v->v,L
$$erive$$ i->i,L
$$erive$$ r->r,L
$$erive$$ e->e,L
$$erive$$ $->$,R
$$$rive$$ e->$,R
$$$rive$$ r->r,R
$$$rive$$ i->i,R
$$$rive$$ v->v,R
$$$rive$$ e->e,R
$$$rive$$ $->$,L
$$$riv$$$ e->$,L
$$$riv$$$ v->v,L
$$$riv$$$ i->i,L
$$$riv$$$ r->r,L
$$$riv$$$ $->$,R
$$$$iv$$$ r->$,R
$$$$iv$$$ i->i,R
$$$$iv$$$ v->v,R
$$$$iv$$$ $->$,L
```

The character v did not match the character r as such the string is not palindrome.