



Escuela de Computación
Ingeniería en Computación
IC1400 - Fundamentos de Organización de
Computadoras

Máquinas de Turing

Máquinas abstractas de Turing

Sebastián Fernández Valerio
sebasr.ferval@estudiantec.cr
2021144538

Juleisy Porras Díaz
j.porrasd03@estudiantec.cr
2021123254

Tamara Rodríguez Luna
nicolerodriguezluna@estudiantec.cr
2021077818

San José, Costa Rica
Febrero 2021

Índice general

1. Introducción	2
2. Ejercicios	3
2.1. Ejercicio 1	3
2.2. Ejercicio 2	4
2.3. Ejercicio 3	8
2.4. Ejercicio 4	13
2.5. Ejercicio 5	19
2.6. Ejercicio 6	21
3. Conclusiones	23

Capítulo 1

Introducción

En el presente informe se plantean soluciones a los ejercicios propuestos acerca de las máquinas de turing. Estas son maquinas abstractas que funcionan mediante una cinta donde se escriben y transcriben diferentes datos. Estos datos son administrados por medio de transiciones entre estados, cada uno con su propia función.

Las presentes soluciones abarcan desde soluciones a operaciones numéricas como sumas, hasta verificaciones entre palabras y caracteres, cada uno desde un enfoque distinto. Todo esto con el propósito de reforzar los conocimientos previamente adquiridos en clase.

Capítulo 2

Ejercicios

2.1. Ejercicio 1

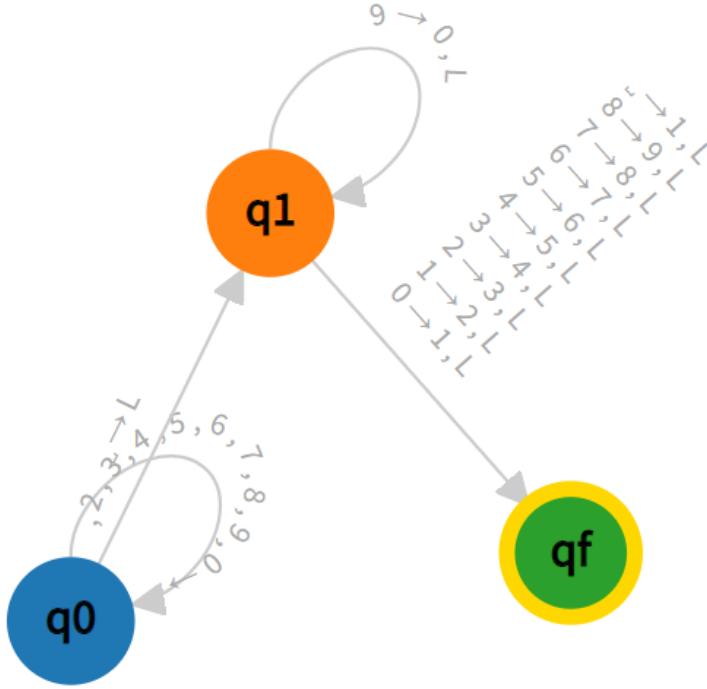
Es una máquina abstracta de Turing que recibe un número entero positivo de cualquier tamaño y le suma uno. Al finalizar su ejecución la cinta contiene únicamente el resultado de la operación, es decir, el nuevo número.

<https://gist.github.com/Juless03/fa7ba666d8326c0589cc6ae34b4b5445>

```
input: '19'
blank: ' '
start state: q0
table:
q0:
[1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,0]: R
' ': {L: q1}

q1:
' ': {write: 1,L: qf}
0 : {write: 1, L: qf}
1 : {write: 2, L: qf}
2 : {write: 3, L: qf}
3 : {write: 4, L: qf}
4 : {write: 5, L: qf}
5 : {write: 6, L: qf}
6 : {write: 7, L: qf}
7 : {write: 8, L: qf}
8 : {write: 9, L: qf}
9 : {write: 0, L: q1}

qf:
```



2.2. Ejercicio 2

Es una máquina abstracta de Turing que lee una palabra de la cinta y la copia al lado separado de un espacio en blanco. La palabra debe estar conformada únicamente por letras minúsculas, no debe llevar tildes ni ñ.

La cinta al finalizar debe contener la palabra de entrada, un espacio, y la palabra de salida.

<https://gist.github.com/Juless03/849a660853d204e2835b6fc2bdc6469e>

input: 'saldo'

blank: ' '

start state: Q0

table:

Q0:

[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z]: R
' ': {write: 0, L: Q1}

Q1:

[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z]: L
' ': {R: Checkletter}

Checkletter:

```

a: {write: A, R: CopyA}
b: {write: B, R: CopyB}
c: {write: C, R: CopyC}
d: {write: D, R: CopyD}
e: {write: E, R: CopyE}
f: {write: F, R: CopyF}
g: {write: G, R: CopyG}
h: {write: H, R: CopyH}
i: {write: I, R: CopyI}
j: {write: J, R: CopyJ}
k: {write: K, R: CopyK}
l: {write: L, R: CopyL}
m: {write: M, R: CopyM}
n: {write: N, R: CopyN}
o: {write: O, R: CopyO}
p: {write: P, R: CopyP}
q: {write: Q, R: CopyQ}
r: {write: R, R: CopyR}
s: {write: S, R: CopyS}
t: {write: T, R: CopyT}
u: {write: U, R: CopyU}
v: {write: V, R: CopyV}
w: {write: W, R: CopyW}
x: {write: X, R: CopyX}
y: {write: Y, R: CopyY}
z: {write: Z, R: CopyZ}
0: {L: GoBack2}

```

CopyA:

```
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,0]: R
' ': {write: a, R: GoBack}
```

CopyB:

```
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,0]: R
' ': {write: b, R: GoBack}
```

CopyC:

```
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,0]: R
' ': {write: c, R: GoBack}
```

CopyD:

```
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,0]: R
' ': {write: d, R: GoBack}
```

CopyE:

```
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,0]: R
' ': {write: e, R: GoBack}
```

CopyF:

```
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,0]: R
' ': {write: f, R: GoBack}
```

CopyG:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: g, R: GoBack}

CopyH:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: h, R: GoBack}

CopyI:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: i, R: GoBack}

CopyJ:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: j, R: GoBack}

CopyK:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: k, R: GoBack}

CopyL:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: l, R: GoBack}

CopyM:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: m, R: GoBack}

CopyN:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: n, R: GoBack}

CopyO:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: o, R: GoBack}

CopyP:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: p, R: GoBack}

CopyQ:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: q, R: GoBack}

CopyR:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: r, R: GoBack}

CopyS:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: s, R: GoBack}

CopyT:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: t, R: GoBack}

CopyU:
 [a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
 ' ': {write: u, R: GoBack}

CopyV:

```

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
' ': {write: v, R: GoBack}
CopyW:
[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
' ': {write: w, R: GoBack}
CopyX:
[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
' ': {write: x, R: GoBack}
CopyY:
[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
' ': {write: y, R: GoBack}
CopyZ:
[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: R
' ': {write: z, R: GoBack}

GoBack:
[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z ,0]: {L: GoBack}
' ': L
[A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z]: {R: Checkletter}

GoBack2:
A: {write: a, L: GoBack2}
B: {write: b, L: GoBack2}
C: {write: c, L: GoBack2}
D: {write: d, L: GoBack2}
E: {write: e, L: GoBack2}
F: {write: f, L: GoBack2}
G: {write: g, L: GoBack2}
H: {write: h, L: GoBack2}
I: {write: i, L: GoBack2}
J: {write: j, L: GoBack2}
K: {write: k, L: GoBack2}
L: {write: l, L: GoBack2}
M: {write: m, L: GoBack2}
N: {write: n, L: GoBack2}
O: {write: o, L: GoBack2}
P: {write: p, L: GoBack2}
Q: {write: q, L: GoBack2}
R: {write: r, L: GoBack2}
S: {write: s, L: GoBack2}
T: {write: t, L: GoBack2}
U: {write: u, L: GoBack2}
V: {write: v, L: GoBack2}
W: {write: w, L: GoBack2}
X: {write: x, L: GoBack2}
Y: {write: y, L: GoBack2}

```

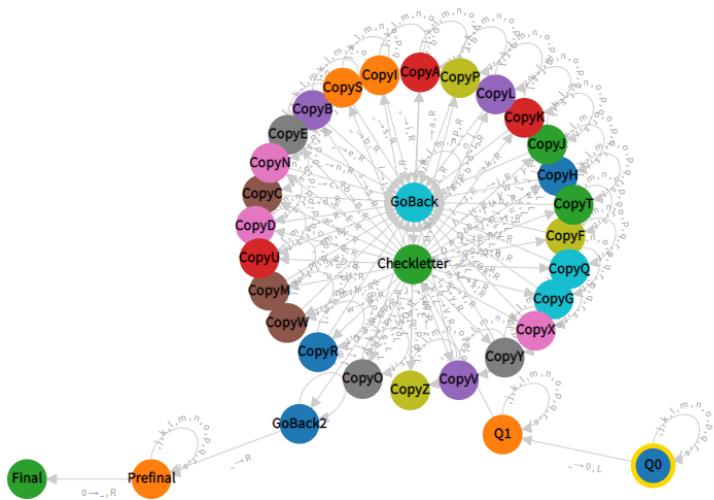
```

Z: { write: z , L: GoBack2}
' ': {R: Prefinal}

Prefinal:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
0: {write: ' ', R: Final}

```

Final :



2.3. Ejercicio 3

Es una máquina de Turing que recibe dos números enteros positivos. La máquina debe sumar ambos números. El primer número puede ser de cualquier tamaño. El segundo número debe ser de únicamente un dígito de cero a nueve.

Al finalizar la cinta debe contener un sólo número con el resultado de la suma.
<https://gist.github.com/Juless03/74a3be4d07314427e8ff014643b036d1>

```

input: '99999 1'
blank: ' '
start state: q0
table:
q0:
[1,2,3,4,5,6,7,8,9,0]: R
' ': {R: q1}
Back0:
' ': {L: add0}
Back1:

```

```

' ': {L: add1}
Back2:
' ': {L: add2}
Back3:
' ': {L: add3}
Back4:
' ': {L: add4}
Back5:
' ': {L: add5}
Back6:
' ': {L: add6}
Back7:
' ': {L: add7}
Back8:
' ': {L: add8}
Back9:
' ': {L: add9}

q1:
0: {write: ' ', L: Back0}
1: {write: ' ', L: Back1}
2: {write: ' ', L: Back2}
3: {write: ' ', L: Back3}
4: {write: ' ', L: Back4}
5: {write: ' ', L: Back5}
6: {write: ' ', L: Back6}
7: {write: ' ', L: Back7}
8: {write: ' ', L: Back8}
9: {write: ' ', L: Back9}

add0:
0: {write: 0, R: q0}
1: {write: 1, R: q0}
2: {write: 2, R: q0}
3: {write: 3, R: q0}
4: {write: 4, R: q0}
5: {write: 5, R: q0}
6: {write: 6, R: q0}
7: {write: 7, R: q0}
8: {write: 8, R: q0}
9: {write: 9, R: q0}

add1:
0: {write: 1, R: q0}
1: {write: 2, R: q0}
2: {write: 3, R: q0}

```

```

3: {write: 4, R: q0}
4: {write: 5, R: q0}
5: {write: 6, R: q0}
6: {write: 7, R: q0}
7: {write: 8, R: q0}
8: {write: 9, R: q0}
9: {write: 0, L: decenas}

add2:
0: {write: 2, R: q0}
1: {write: 3, R: q0}
2: {write: 4, R: q0}
3: {write: 5, R: q0}
4: {write: 6, R: q0}
5: {write: 7, R: q0}
6: {write: 8, R: q0}
7: {write: 9, R: q0}
8: {write: 0, L: decenas}
9: {write: 1, L: decenas}

add3:
0: {write: 3, R: q0}
1: {write: 4, R: q0}
2: {write: 5, R: q0}
3: {write: 6, R: q0}
4: {write: 7, R: q0}
5: {write: 8, R: q0}
6: {write: 9, R: q0}
7: {write: 0, L: decenas}
8: {write: 1, L: decenas}
9: {write: 2, L: decenas}

add4:
0: {write: 4, R: q0}
1: {write: 5, R: q0}
2: {write: 6, R: q0}
3: {write: 7, R: q0}
4: {write: 8, R: q0}
5: {write: 9, R: q0}
6: {write: 0, L: decenas}
7: {write: 1, L: decenas}
8: {write: 2, L: decenas}
9: {write: 3, L: decenas}

add5:
0: {write: 5, R: q0}

```

```
1: {write: 6, R: q0}
2: {write: 7, R: q0}
3: {write: 8, R: q0}
4: {write: 9, R: q0}
5: {write: 0, L: decenas}
6: {write: 1, L: decenas}
7: {write: 2, L: decenas}
8: {write: 3, L: decenas}
9: {write: 4, L: decenas}

add6:
0: {write: 6, R: q0}
1: {write: 7, R: q0}
2: {write: 8, R: q0}
3: {write: 9, R: q0}
4: {write: 0, L: decenas}
5: {write: 1, L: decenas}
6: {write: 2, L: decenas}
7: {write: 3, L: decenas}
8: {write: 4, L: decenas}
9: {write: 5, L: decenas}

add7:
0: {write: 7, R: q0}
1: {write: 8, R: q0}
2: {write: 9, R: q0}
3: {write: 0, L: decenas}
4: {write: 1, L: decenas}
5: {write: 2, L: decenas}
6: {write: 3, L: decenas}
7: {write: 4, L: decenas}
8: {write: 5, L: decenas}
9: {write: 6, L: decenas}

add8:
0: {write: 8, R: q0}
1: {write: 9, R: q0}
2: {write: 0, L: decenas}
3: {write: 1, L: decenas}
4: {write: 2, L: decenas}
5: {write: 3, L: decenas}
6: {write: 4, L: decenas}
7: {write: 5, L: decenas}
8: {write: 6, L: decenas}
9: {write: 7, L: decenas}
```

```

add9:
0: {write: 9, R: q0}
1: {write: 0, L: decenas}
2: {write: 1, L: decenas}
3: {write: 2, L: decenas}
4: {write: 3, L: decenas}
5: {write: 4, L: decenas}
6: {write: 5, L: decenas}
7: {write: 6, L: decenas}
8: {write: 7, L: decenas}
9: {write: 8, L: decenas}

```

```

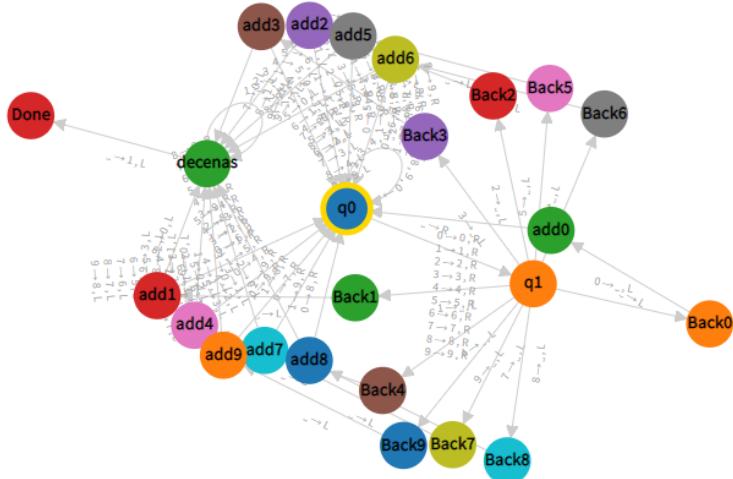
decenas:
' ': {write: 1, L: Done}
1: {write: 2, L}
2: {write: 3, L}
3: {write: 4, L}
4: {write: 5, L}
5: {write: 6, L}
6: {write: 7, L}
7: {write: 8, L}
8: {write: 9, L}
9: {write: 0, L}

```

```

Done:

```



2.4. Ejercicio 4

Es una máquina de Turing que determina si una palabra es un palíndromo. La palabra debe estar escrita en letras minúsculas, sin tildes ni ñ.

La cinta debe terminar con la misma información en que inició, pero la máquina debe terminar en un estado ACEPTADO si la palabra es palíndromo, RECHAZADO si la palabra no es un palíndromo.

<https://gist.github.com/Juless03/779ccba010bd404fa244ce6acad21da1>

```
input: 'abba'
blank: ' '
start state: start

table:
  start:
    a: {R: q1 a}
    b: {R: q2 b}
    c: {R: q3 c}
    d: {R: q4 d}
    e: {R: q5 e}
    f: {R: q6 f}
    g: {R: q7 g}
    h: {R: q8 h}
    i: {R: q9 i}
    j: {R: q10 j}
    k: {R: q11 k}
    l: {R: q12 l}
    m: {R: q13 m}
    n: {R: q14 n}
    o: {R: q15 o}
    p: {R: q16 p}
    q: {R: q17 q}
    r: {R: q18 r}
    s: {R: q19 s}
    t: {R: q20 t}
    u: {R: q21 u}
    v: {R: q22 v}
    w: {R: q23 w}
    x: {R: q24 x}
    y: {R: q25 y}
    z: {R: q26 z}

q1 a:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
': {L: a final}
```

a final:
 a: {R: Aceptado}
 [b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q2 b:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ': {L: b final}

b final:
 b: {R: Aceptado}
 [a,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q3 c:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ': {L: c final}

c final:
 c: {R: Aceptado}
 [a,b,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q4 d:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ': {L: d final}

d final:
 d: {R: Aceptado}
 [a,b,c,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q5 e:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ': {L: e final}

e final:
 e: {R: Aceptado}
 [a,b,c,d,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q6 f:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ': {L: f final}

f final:
 f: {R: Aceptado}
 [a,b,c,d,e,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q7 g:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R

```

' ': {L: g final}

g final:
g: {R: Aceptado}
[a,b,c,d,e,f,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q8 h:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
' ': {L: h final}

h final:
h: {R: Aceptado}
[a,b,c,d,e,f,g,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q9 i:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
' ': {L: i final}

i final:
i: {R: Aceptado}
[a,b,c,d,e,f,g,h,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q10 j:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
' ': {L: j final}

j final:
j: {R: Aceptado}
[a,b,c,d,e,f,g,h,i,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q11 k:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
' ': {L: k final}

k final:
k: {R: Aceptado}
[a,b,c,d,e,f,g,h,i,j,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q12 l:
[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
' ': {L: l final}

l final:
l: {R: Aceptado}
[a,b,c,d,e,f,g,h,i,j,k,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

```

q13 m:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ' ': {L: m final}

m final:
 m: {R: Aceptado}
 [a,b,c,d,e,f,g,h,i,j,k,l,n,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q14 n:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ' ': {L: n final}

n final:
 n: {R: Aceptado}
 [a,b,c,d,e,f,g,h,i,j,k,l,m,o,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q15 o:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ' ': {L: o final}

o final:
 o: {R: Aceptado}
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,p,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q16 p:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ' ': {L: p final}

p final:
 p: {R: Aceptado}
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,q,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q17 q:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ' ': {L: q final}

q final:
 q: {R: Aceptado}
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,r,s,t,u,v,w,x,y,z]: {R: Rechazado}

q18 r:
 [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]: R
 ' ': {L: r final}

r final:
 r: {R: Aceptado}

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,s ,t ,u ,v ,w ,x ,y ,z]: {R: Rechazado}

q19 s:

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z]: R
' ': {L: s final}

s final:

s: {R: Aceptado}

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,t ,u ,v ,w ,x ,y ,z]: {R: Rechazado}

q20 t:

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z]: R
' ': {L: t final}

t final:

t: {R: Aceptado}

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,u ,v ,w ,x ,y ,z]: {R: Rechazado}

q21 u:

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z]: R
' ': {L: u final}

u final:

u: {R: Aceptado}

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,v ,w ,x ,y ,z]: {R: Rechazado}

q22 v:

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z]: R
' ': {L: v final}

v final:

v: {R: Aceptado}

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,w ,x ,y ,z]: {R: Rechazado}

q23 w:

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z]: R
' ': {L: w final}

w final:

w: {R: Aceptado}

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,x ,y ,z]: {R: Rechazado}

q24 x:

[a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,o ,p ,q ,r ,s ,t ,u ,v ,w ,x ,y ,z]: R
' ': {L: x final}

x final:
x: {R: Aceptado}
[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, y, z]: {R: Rechazado}

q25 y:
[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z]: R
': {L: y final}

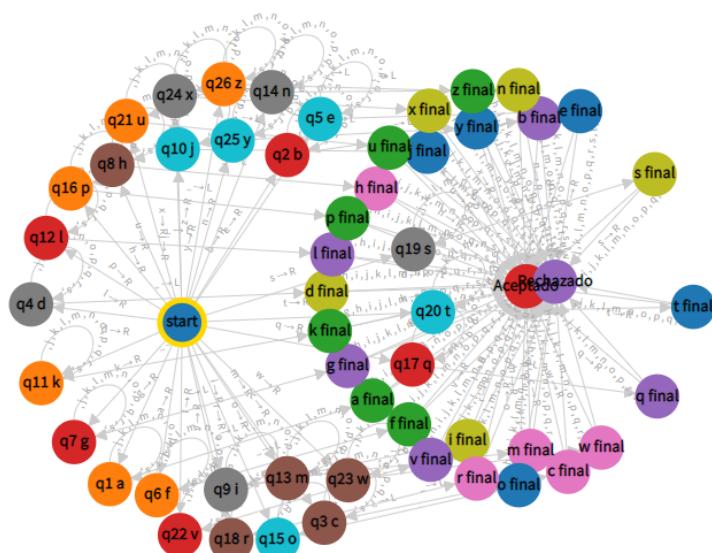
y final:
y: {R: Aceptado}
[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, z]: {R: Rechazado}

q26 z:
[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z]: R
': {L: z final}

z final:
z: {R: Aceptado}
[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y]: {R: Rechazado}

Aceptado:

Rechazado:



2.5. Ejercicio 5

Es una máquina de Turing que recibe dos números enteros positivos y determina si el primer número es mayor, igual, o menor que el segundo número.

La máquina debe terminar en el estado MAYOR si el primer número es mayor, IGUAL si ambos números son iguales, o MENOR si el primer número es menor.

<https://gist.github.com/Juless03/31b94b1e562eda9e5d302547b0061721>

```
input: '6682 125'
blank: '
start state: Change1

table:
GoRight:
[1,2,3,4,5,6,7,8,9,0]: R
': {R: Change2}
[A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z]: {R: Change2}

GoLeft:
': L
[1,2,3,4,5,6,7,8,9,0]: L
[A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z]: {R: Change1}

Change1:
0: {write: A, R: GoRight}
1: {write: B, R: GoRight}
2: {write: C, R: GoRight}
3: {write: D, R: GoRight}
4: {write: E, R: GoRight}
5: {write: F, R: GoRight}
6: {write: G, R: GoRight}
7: {write: H, R: GoRight}
8: {write: I, R: GoRight}
9: {write: J, R: GoRight}

A: L
B: L
C: L
D: L
E: L
F: L
G: L
H: L
I: L
```

J: L
K: L
L: L
M: L
N: L
O: L
P: L
Q: L
R: L
S: L
T: L
U: L
V: L
W: L
X: L
Y: L
Z: L

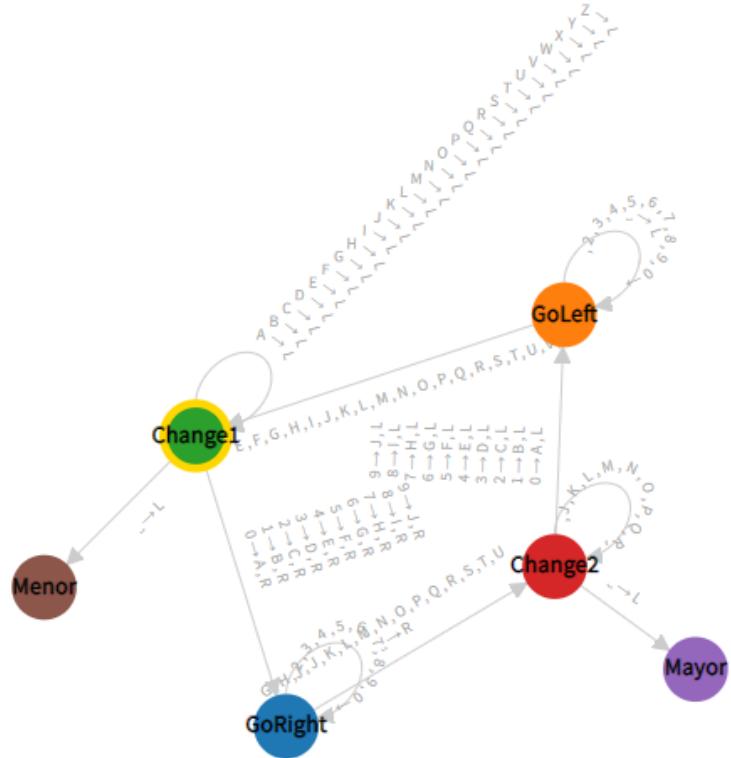
' ': {L: Menor}

Change2:

[A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z]: R
0: {write: A, L: GoLeft}
1: {write: B, L: GoLeft}
2: {write: C, L: GoLeft}
3: {write: D, L: GoLeft}
4: {write: E, L: GoLeft}
5: {write: F, L: GoLeft}
6: {write: G, L: GoLeft}
7: {write: H, L: GoLeft}
8: {write: I, L: GoLeft}
9: {write: J, L: GoLeft}
' ': {L: Mayor}

Mayor:

Menor:



2.6. Ejercicio 6

Es una máquina de Turing que lee una dirección de correo electrónico y verifica si la dirección es válida o no. La dirección debe estar compuesta de letras minúsculas, sin tildes ni ñ, puede incluir puntos, guiones bajos, guiones, debe tener una única @ y debe cumplir la siguiente estructura:

frase@frase.frase

donde una frase es cualquier combinación de símbolos válidos.

La cinta debe terminar con la misma información en que inició, pero la máquina debe terminar en un estado de ACEPTADO o RECHAZADO según corresponda.

<https://gist.github.com/Juless03/7a6cca76640ac221f295dce4d53a498d>

```
input: 'frase@frase.frase'
```

```
blank: ' '
```

```
start state: Inicio
```

```
Synonym:
```

```
table:
```

```
Inicio:
```

$[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z] : R$
 $"., -, -" : R$
 $"@" : \{R: Prueba\}$
 $' ' : \{L: RECHAZADO\}$

Prueba :

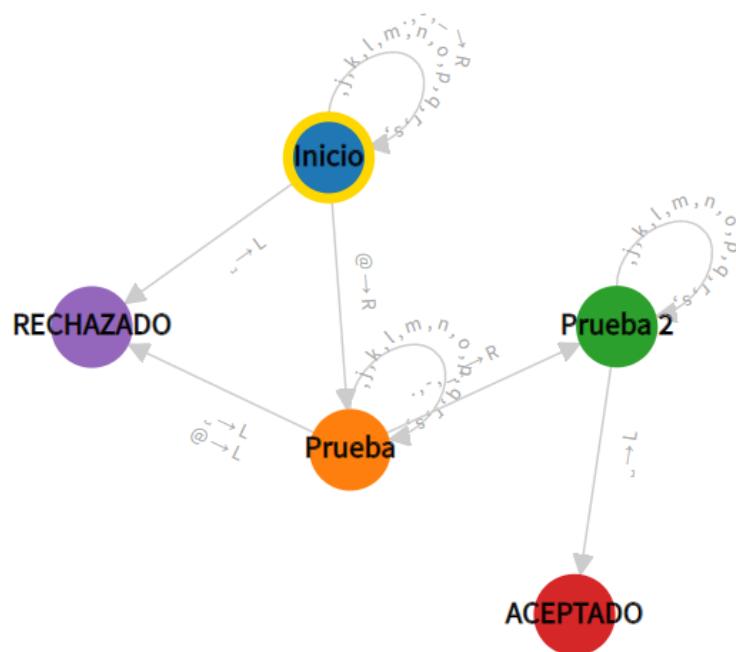
$[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z] : R$
 $"., -, -" : \{R: Prueba 2\}$
 $' ' : \{L: RECHAZADO\}$
 $'@' : \{L: RECHAZADO\}$

Prueba 2:

$[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z] : R$
 $' ' : \{L: ACEPTADO\}$

ACEPTADO:

RECHAZADO:



Capítulo 3

Conclusiones

Se realizaron ejercicios con una máquina abstracta de Turing, la cual es más que solo un dispositivo o instrumento, es toda una estructura que nos permite encontrar soluciones de problemas simples o complicados de distinta naturaleza. En el presente trabajo se buscó analizar cada caso de manera integral, buscar sus soluciones a partir de las condiciones establecidas. En base a esto se logró dar solución a 5 de los 6 ejercicios planteados, e excepción del número 5. Las máquinas tuvieron códigos cortos y largos, donde se puso en acción las soluciones a los problemas respetando las características propias de los mismos y las restricciones anteriormente expuestas como requisito a la hora de llevar a cabo la tarea.