

Proyecto Final Maquina dispensadora

MAESTRO:

Abelardo Gómez Andrade

ALUMNO:

Rodríguez Tabares Juan

CODIGO:

215615699



CARRERA:

Ingeniería en Computación

MATERIA:

Teoría de la computación

HORARIO:

Martes y jueves

11:00 – 13:00

SECCION:

D07

Planteamiento del problema

Para esta actividad se decidió usar un autómata finito para solucionar un problema muy conocido y posiblemente todos lo hemos usado alguna vez, estamos hablando de una maquina dispensadora de refrescos la cual aceptará monedas de 5, 10 y 25 pesos, para que el usuario pueda comprar el producto que más le agrade.

Esta máquina expendedora hará sentir al usuario una experiencia diferente ya que se dirigirá a él de una manera amigable y siempre explicándole los posibles productos que puede comprar con el dinero que él mismo ha introducido a la máquina, también ayudara al usuario explicándole el proceso que se lleva acabo y previniendo que gaste de más ya que la máquina expendedora en este caso no puede dar cambio.

Objetivo

La misión principal del proyecto final en la asignatura de Teoría de la computación es poner a prueba los conocimientos de los alumnos acerca de la implementación de los autómatas finitos en objetos o actividades que se encuentren en la vida real.

La implementación de nuestro modelo de máquina dispensadora se hará construyendo un **software** que emule el funcionamiento de una máquina dispensadora ordinaria que ofrezca productos empacados, principalmente dulces. Dicho software será **desarrollado** utilizando el lenguaje de programación C++ y el entorno de desarrollo Code-Blocks, este lenguaje de programación tiene la capacidad de simular autómatas y máquinas de Turing con mucha facilidad, únicamente se necesitan utilizar arreglos (o listas) que ayuden a describir los estados y transiciones del autómata obtenido. Por otro lado, la consola permite al equipo mostrar mensajes al usuario y guiarlo hasta el final del proceso de compra, brindando así una experiencia muy similar a la de una máquina dispensadora real.

El autómata finito se mostrará visualmente mediante un diagrama de transiciones hecho en el programa JFLAP que fue proporcionado por el profesor, este diagrama ayuda a entender de manera más clara cómo funciona el autómata de nuestro proyecto, e inclusive realizar algunas pruebas con posibles cadenas que introduzca el usuario y la respuesta que obtendrá (aceptar o rechazar).

Justificación

El profesor mencionaba que los autómatas pueden emular los funcionamientos de todas aquellas cosas que acepten o rechazan “entradas” y será un autómata finito aquel que evalúe las entradas de un lenguaje regular. En nuestro caso el proyecto que escogimos encaja en ambas descripciones porque rechazara todas las “entradas” (en la realidad serían monedas) hasta llegar a un mínimo que representa la oportunidad de comprar alguno de los productos y posteriormente regresar al estado inicial donde se encuentra sin dinero.

Marco teórico

Programación Orientada a Objetos

La programación Orientada a objetos se define como un paradigma de la programación, una manera de programar específica, donde se organiza el código en unidades denominadas clases de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

Diagrama de transición de estados

Un diagrama de transición de estados muestra el comportamiento dependiente del tiempo de un sistema de información. El cual representa los estados que puede tomar un componente o un sistema y una muestra de los eventos que implican el cambio de un estado a otro.

Autómata Finito

Un Autómata Finito, también llamado Autómata de Estado Finito, es toda Máquina de Estado Finito en la que el conjunto de símbolos de salida es exclusivamente $O = \{0, 1\}$ y donde el estado actual determina cuál de ellos fue el último dato de salida. Aquellos estados para los cuales el último dato de salida fue 1, se denominan estados de aceptación. En todo Autómata Finito, representado como A , debe haber cuando menos un estado de aceptación y por sentido común se recomienda que no todos lo sean.

“Se podría decir que un Autómata Finito es un aceptador o rechazador de cadenas.”

Metodología

Para realizar la resolución del problema propuesto como “Proyecto final” de la materia usamos el paradigma de la programación orientada a objetos para poder separar en clases y en una de crear un autómata de estado finito el cual es abstractamente un diagrama de transición de estados que representara el comportamiento de una maquina dispensadora y para poder crear un autómata finito usaremos un software de programación para su creación usando arreglos multidimensionales que simulan estados de transición dependiendo el digito que se le introduzca de los aceptados por el mismo los cuales son “monedas de 5, 10 y 25 pesos” y rechazando los diferentes a los anteriores mencionados, que dependiendo la cantidad que le introduzcas pasara de estado a estado y al pasar al último estado donde se han introducido un total de 25 pesos es el estado de aceptación y al llegar al mismo pasara a ser otra función el programa el que mostrara un menú de los productos que puedes canjear.

Seudocódigo el cual permite el cambio de estados, comprobación de las entradas y validación del acumulador para poder ofrecer productos todo esto para la compra de productos que hace el usuario:

```
1      funcion cautomata::comprarProducto(cproducto* productos)
2      {
3          entero coin;
4          entero opc;
5          entero acumCoins(0);
6
7          hacer
8          {
9              Si (acumCoins >= 30)
10             {
11                 imprimir <<"|Tienes "<<acumCoins<<" pesos, Debes comprar algo|" <<
12                 Salto de linea;
13                 llamada a funcion menuProductos(&acumCoins, productos);
14                 break;
15             }
16             imprimir <<" |Maquina Dispensadora |"<<Salto de linea;
17             imprimir <<" |*-*-*Bienvenido *-*-*|"<<Salto de linea;
18             imprimir <<"Ingrese monedas [5] [10] [25]: ";
19             guardar >> coin;
20
21             Si (coin!=5 && coin!=10 && coin!=25)
22             {
23                 imprimir <<" |*-*-*No se que moneda es, intenta con modenas de [5],
24                 [10] o [25] *-*-*|" << Salto de linea;
```

```

23         }
24         Si no
25         {
26         hacer
27         {
28         entero pos = 1;
29         desde (int i = 0; i < 5; i++)
30         {
31         Si (coin == matrizAutomata[pos][i])
32         {
33         acumCoins = acumCoins+matrizAutomata[pos][i];
34         pos++;
35         imprimir <<"*-----*"<<Salto de linea;
36         imprimir <<"|*-*Tienes: " << acumCoins << " pesos*-
37         **|"<<Salto de linea;
38         imprimir <<"*-----*"<<Salto de linea;
39         }
40     }
41     Si (acumCoins == 10 o acumCoins == 15 o acumCoins == 25) 42
42     {
43     llamada a funcion mostrarOfertas(&acumCoins, productos);
44     imprimir << "|Comprar o Depositar?   |" << endl; 45
45     imprimir << "|*-*NOTA*-*"           |" << endl;
46     imprimir << "|*-*No doy cambio*-*|" << endl;
47     imprimir << "|(1) Comprar           |" << endl;
48     imprimir << "|(0) seguir depositando |" << endl;
49     guardar >> opc;
50     Si (opc == 1)
51     {
52     menuProductos(&acumCoins, productos);
53     regresar;
54     }
55     }
56     }
57     mientras (1);
58     }
59     }
60     mientras (1);
61

```

f/g	q0	q1	q2	q3	q4	q5	q6
q0	X	5	10	X	X	25	X
q1	X	X	5	10	X	X	25
q2	X	X	X	5	10	X	25
q3	X	X	X	X	5	10	25
q4	X	X	X	X	X	5	10,25
q5	X	X	X	X	X	X	5,10,25
q6	X	X	X	X	X	X	5,10,25

Ilustración 1 Tabla de transición del proyecto,

Pruebas y resultados

```
*-----*
|Maquina Dispensadora |
|*-*-*Bienvenido *-*-*|
*-----*
Ingrese monedas [5] [10] [25]: 5
```

Ilustración 2 Bienvenida de la máquina dispensadora.

```
*-----*
|Maquina Dispensadora |
|*-*-*Bienvenido *-*-*|
*-----*
Ingrese monedas [5] [10] [25]: 10_
```

Ilustración 3 Validamos otra moneda con la denominación de 10 pesos.

```
*-----*
|-*-*Tienes: 15 pesos*-*-*|
*-----*
Presione una tecla para continuar . . .
```

Ilustración 4 Conteo de la cantidad de monedas acumuladas que llevamos.

```
*-----*
|*-*-Tienes: 15 pesos, puedes comprar:*-*-|
*-----*
```

Ilustración 5 En ciertas cantidades la máquina de ofrecerá los productos si así lo deseas.

Menú

```
|Producto: Gansito
|Codigo: 11
|Costo: 12
|Peso: 200g
|
|-----|
|Producto: Negrito
|Codigo: 22
|Costo: 10
|Peso: 250g
|
|-----|
|Producto: Chips moradas
|Codigo: 33
|Costo: 14
|Peso: 300g
|
|-----|
|Producto: Sabritas
|Codigo: 44
|Costo: 15
|Peso: 250g
|
|-----|
```

```
|Producto: Agua en polvo
|Codigo: 66
|Costo: 9
|Peso: 600g
|
|-----|
|Producto: Barritas
|Codigo: 77
|Costo: 10
|Peso: 400g
|
|-----|
|Producto: Mazapan
|Codigo: 88
|Costo: 5
|Peso: 50g
|
|-----|
|Producto: Cacahuates
|Codigo: 99
|Costo: 11
|Peso: 600g
|
|-----|
```

```

*-----*
|Comprar o Depositar?|
|*-*-*NOTA*-*-*|
|*-*-*No doy cambio*-*-*|
*-----*

*-----*
|(1) Comprar|
|(0) seguir depositando|
*-----*

```

Ilustración 6 Después de mostrar el menú la maquina preguntara si deseas seguir depositando o si ya necesitas comprar algo.

```

*-----*
|Tienes 30 pesos, Debes comprar algo|
*-----*
Presione una tecla para continuar . . .

```

Ilustración 7 Cuando se llega a una cantidad de 30 pesos o más se deben de comprar los productos sin poder acumular más dinero.

```

*-----*
|*-*-*Ingrese el codigo*-*-*|
*-----*
44_

```

Ilustración 8 Aquí se ingresa el código del producto que deseas comprar.

```

*-----*
|Disfruta |
*-----*

_____|
|Producto: Sabritas
|Codigo: 44
|Costo: 15
|Peso: 250g
|
_____|
Presione una tecla para continuar . . . _

```

Ilustración 9 Producto final entregado.


```

*-----*
|Maquina Dispensadora|
|*-*-*Bienvenido *-*-*|
*-----*
Ingrese monedas [5] [10] [25]: 25_

```

Ilustración 10 Otra moneda de diferente denominación que fue introducida pero sigue siendo valida.

```

*-----*
|-*-*Tienes: 25 pesos*-*-*|
*-----*
Presione una tecla para continuar . . .

```

Ilustración 11 Moneda introducida y aceptada.

```

*-----*
|Maquina Dispensadora|
|*-*-*Bienvenido *-*-*|
*-----*
Ingrese monedas [5] [10] [25]: 2
*-----*
|-*-*No se que moneda es, intenta con modenas de [5], [10] o [25] *-*-*|
*-----*
Presione una tecla para continuar . . .

```

Ilustración 12 En caso de una moneda diferente se le hará saber al usuario que no puede reconocer ese tipo de monedas.

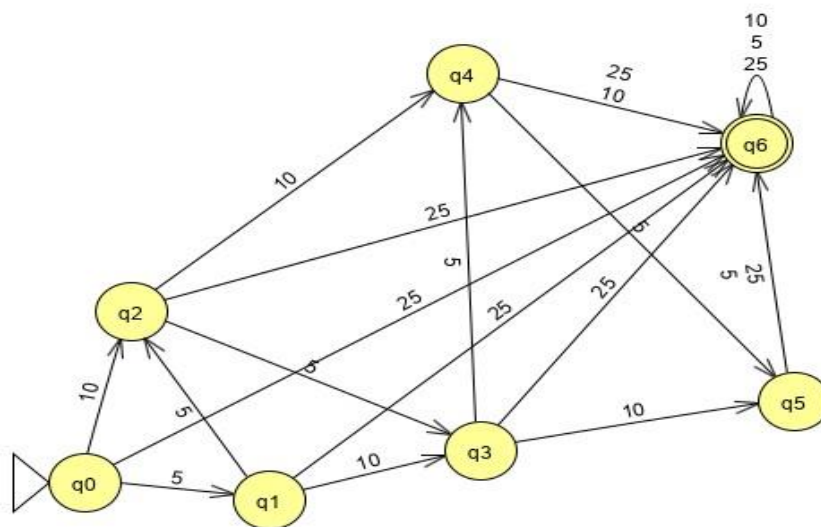


Ilustración 13 Diseño de transición del proyecto

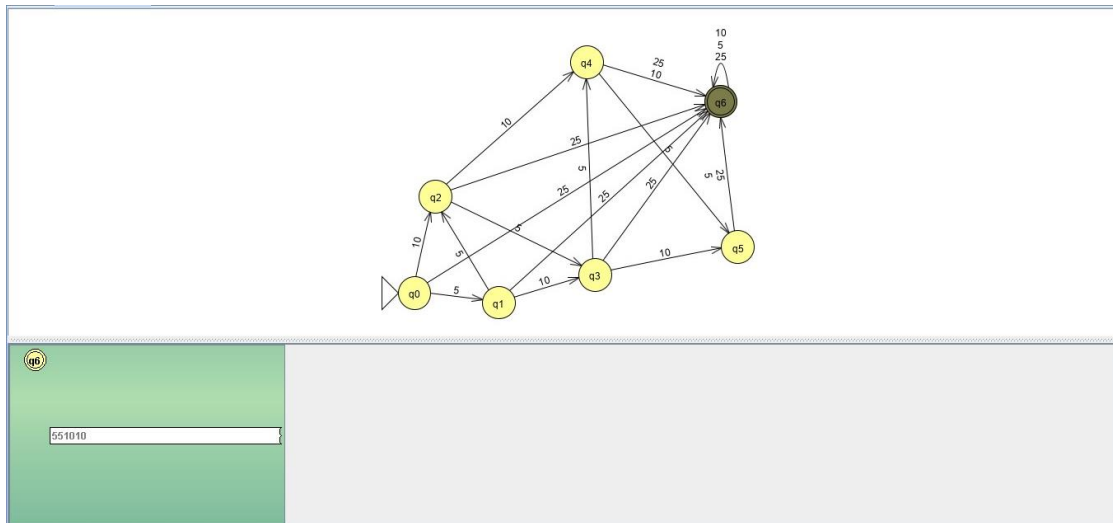


Ilustración 14 Cadena aceptada "5 5 10 10"

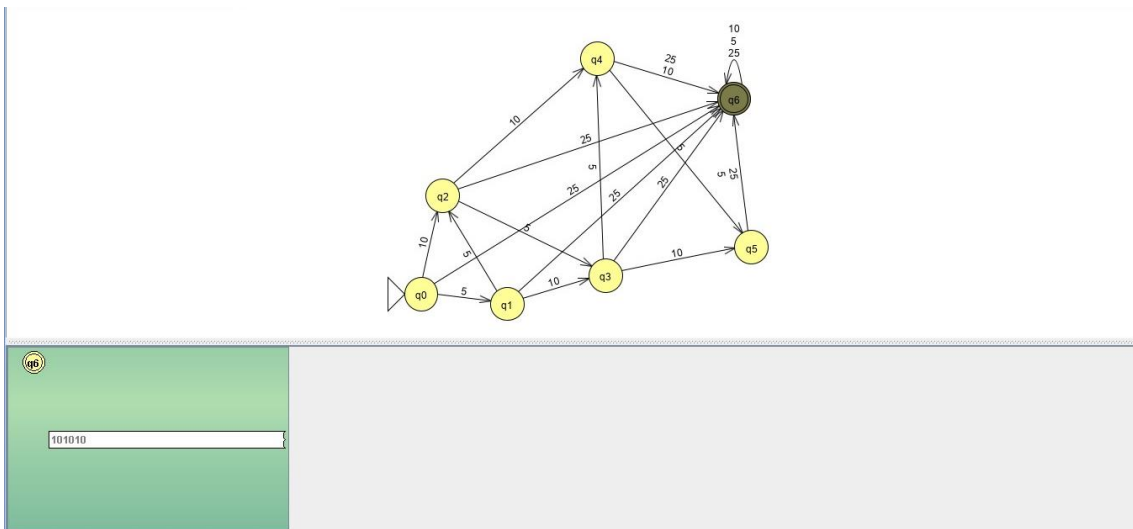


Ilustración 15 Cadena aceptada "10 10 10"

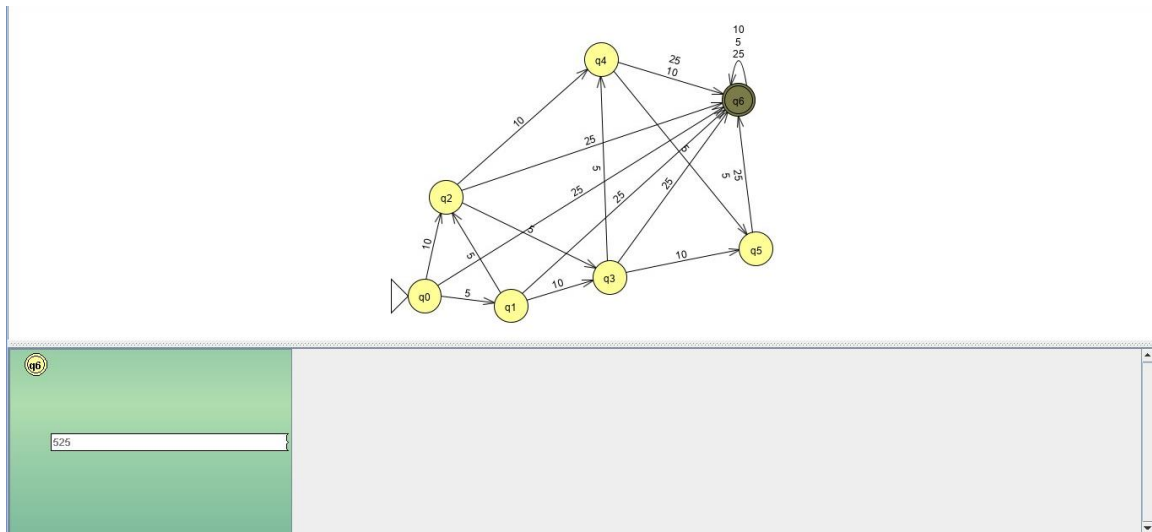


Ilustración 16 Cadena aceptada "5 25"

Conclusiones

Después de realizare todo el trabajo y finalizarlo de una manera eficiente, mis compañeros y yo nunca nos pudimos haber imaginado que algo así como lo que vimos en el transcurso de las clases se pudiera realizar de una forma tan compleja en la vida cotidiana dándonos resultados tan asombrosos como el funcionamiento de máquinas expendedoras, cajeros automáticos y entre otra infinidad de cosas en las que nos sirven los autómatas y máquinas de estado finito, sin duda alguna este trabajo ha sido uno de los que más nos ha ayudado a entender e implementar nuestra enseñanza en la cosas bastante comunes en la vida diaria logrando un resultado exitosos en la ejecución de este pequeño proyecto.

Posiblemente este proyecto se pueda mejorar de una manera más eficiente e implementando nuevas tecnologías en un futuro, pero mientras que descubrimos como hacerlo podemos concluir que las expectativas de este proyecto fueron totalmente cubiertas.

Referencias bibliográficas

Gomez, A. (2014). Introducción a la teoría de autómatas y lenguajes formales. Tlaquepaque Jalisco: TRAUCO.

Hopcroft, J., Motowani R., Ullman J. (2008). la teoría de autómatas, lenguajes y computación. Madrid: PEARSON EDUCACION S.A.