

LENGUAJES FORMALES

PRÁCTICA 2:

ANALIZADOR SINTÁCTICO CON BISON

Juego de Pruebas

Departament d'Enginyeria



Informàtica i
Matemàtiques



Alumnes:	Bové Gómara, Marc López Mellina, Alejandro Mege Barriola, Gwenaëlle
Grado:	3r Ingeniería Informática
Curso:	2017/2018
Profesor:	Riaño Ramos, David

Índice

1. Identificación de autómatas finitos deterministas y no deterministas	2
2. Reconocimiento de alfabetos sin símbolos	2
3. Reconocimiento de AF con ningún o más de un estado inicial	3
4. Reconocimiento de AF sin estados finales	3
5. Reconocimiento del uso de números de estado o símbolos inexistentes en transiciones	3
6. Reconocimiento de símbolos repetidos en alfabetos	4
7. Reconocimiento de AF no deterministas	4
8. Implementación en C	5

1. Identificación de autómatas finitos deterministas y no deterministas:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
AF determinista. Ej: AF con un estado y dos transiciones.	<pre>int transicion (int estado, char símbolo) { int sig; if ((estado==0)&&(símbolo==' a')) sig = 1; if ((estado==0)&&(símbolo =='b')) sig= 0; return (sig); }</pre>	<pre>int transicion (int estado, char símbolo) { int sig; if ((estado==0)&&(símbolo==' a')) sig = 1; if ((estado==0)&&(símbolo =='b')) sig= 0; return (sig); }</pre>	SI
AF no determinista. Ej: AF con dos estados y dos transiciones.	<pre>int * transicion (int estado, char símbolo) { static int sig[num-estados+1], n=0; if ((estado==0)&&(símbolo =='a')) sig[n++] = 0; if ((estado==1)&&(símbolo =='b')) sig[n++] = 0; sig[n]= -1; /* centinella */ return (sig); }</pre>	<pre>int * transicion (int estado, char símbolo) { static int sig[num-estados+1], n=0; if ((estado==0)&&(símbolo =='a')) sig[n++] = 0; if ((estado==1)&&(símbolo =='b')) sig[n++] = 0; sig[n]= -1; /* centinella */ return (sig); }</pre>	SI

2. Reconocimiento de alfabetos sin símbolos:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
Alfabeto con símbolos.	-	-	SI
Alfabeto con un símbolo.	-	-	SI
Alfabeto sin símbolos.	[ERROR]: El alfabeto debe contener uno o más símbolos.	[ERROR]: El alfabeto debe contener uno o más símbolos.	SI

3. Reconocimiento de AF con ningún o más de un estado inicial:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
AF con un estado inicial.	-	-	SI
AF con más de un estado inicial.	[ERROR]: Los Autómatas Finitos solo deben tener un estado final.	[ERROR]: Los Autómatas Finitos solo deben tener un estado final.	SI
AF sin ningún estado inicial.	[ERROR]: Los Autómatas Finitos solo deben tener un estado final.	[ERROR]: Los Autómatas Finitos solo deben tener un estado final.	SI

4. Reconocimiento de AF sin estados finales:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
AF con uno o mas estados finales.	-	-	SI
AF sin estados finales.	[ERROR]: Los Autómatas Finitos deben tener algún estado final	[ERROR]: Los Autómatas Finitos deben tener algún estado final	SI

5. Reconocimiento del uso de números de estado o símbolos inexistentes en transiciones:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
Símbolos correspondientes a estados y a símbolos del alfabeto válidos. Ej. Transiciones { (0 , a ; 1) }	-	-	SI
Dos símbolos correctos y letra de alfabeto incorrecta. Ej. Transiciones { (0 , Z; 1) }	[ERROR] El símbolo Z de la transición (0,Z;1) es desconocido.	[ERROR] El símbolo Z de la transición (0,Z;1) es desconocido.	SI

El símbolo inicial no es válido, el final y el del alfabeto sí. Ej. Transiciones { (99 , a; 1) }	[ERROR] El estado 99 de la transición (99,a;1) es desconocido.	ERROR] El estado 9 de la transición (99,a;1) es desconocido.	SI
El símbolo final no es válido, el inicial y el del alfabeto sí. Ej. Transiciones { (1 , a; 99) }	[ERROR] El estado 99 de la transición (1,a;99) es desconocido.	ERROR] El estado 99 de la transición (1,a;99) es desconocido.	SI
El símbolo final e inicial no son válidos y el del alfabeto sí. Ej. Transiciones { (451 , a; 99), (0 , a ; 1) }	[ERROR] El estado 451 de la transición (451,a;99) es desconocido. [ERROR] El estado 99 de la transición (451,a;99) es desconocido.	[ERROR] El estado 451 de la transición (451,a;99) es desconocido. [ERROR] El estado 99 de la transición (451,a;99) es desconocido.	SI

6. Reconocimiento de símbolos repetidos en alfabetos:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
Todos los símbolos del alfabeto son diferentes. Ej. Alfabeto { a , b }	-	-	SI
Hay símbolos repetidos en el alfabeto. Ej. Alfabeto { a , b , a }	[AVISO]: EL símbolo a ya existe.	[AVISO]: EL símbolo a ya existe.	SI

7. Reconocimiento de AF no deterministas:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
AF determinista.	-	-	SI
AF no determinista.	[AVISO] Se ha detectado que el AF es no determinista.	[AVISO] Se ha detectado que el AF es no determinista.	SI

8. Implementación en C:

Datos de entrada	Datos de salida esperados	Datos de salida reales	Correcto
AF determinista. Ej: AF con un estado y dos transiciones.	Se crea un fichero transicion.h ¹ Se crea un fichero transicion.c ²	Se crea un fichero transicion.h ¹ Se crea un fichero transicion.c ²	SI
AF no determinista. Ej: AF con dos estados y dos transiciones.	Se crea un fichero transicion.h ¹ Se crea un fichero transicion.c ³	Se crea un fichero transicion.h ¹ Se crea un fichero transicion.c ³	SI

1. transicion.h

```
int transiciondet(int estado, char simbolo);

int * transicion(int estado, char simbolo);
```

2. transicion.c (Determinista)

```
#include "transicion.h"

int transicion (int estado, char simbolo)
{
    int sig;
    if ((estado==0)&&(simbolo=='a')) sig = 1;
    if ((estado==0)&&( simbolo =='b')) sig= 0;

    return (sig);
}
```

3. transicion.c (Indeterminista)

```
#include "transicion.h"

int * transicion (int estado, char simbolo)
{
    static int sig[num-estados+1], n=0;
    if ((estado==0)&&( simbolo =='a')) sig[n++] = 0;
    if ((estado==1)&&( simbolo =='b')) sig[n++] = 0;

    sig[n]= -1; /* centinella */
    return (sig);
}
```