



Instituto Politécnico Nacional
Escuela Superior de Cómputo

Teoría Computacional

Practica 5:Automata de Pila

Alumna: Ramírez Galindo Karina

Profesor: Rosas Trigueros Jorge Luis

Fecha de realización: 28-10-2019

Fecha de entrega: 15-11-2019



Marco Teórico

AUTOMATA DE PILA

El autómata a pila es un autómata finito no determinista con transiciones- ϵ y una capacidad adicional: una pila en la que se puede almacenar una cadena de “símbolos de pila”. La presencia de una pila significa que, a diferencia del autómata finito, el autómata a pila puede “recordar” una cantidad infinita de información. [1]

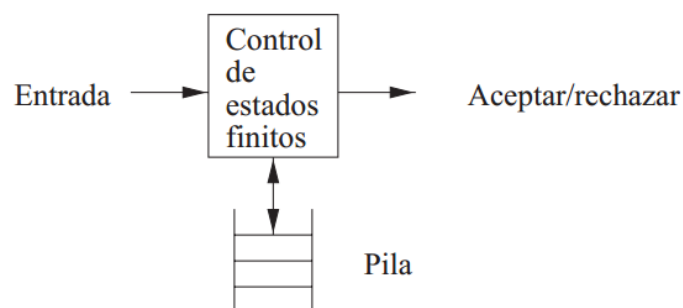


Imagen 1

Autómata de Pila

El autómata a pila puede observar el símbolo colocado en la parte superior de la pila y llevar a cabo su transición basándose en el estado actual, el símbolo de entrada y el símbolo que hay en la parte superior de la pila. Alternativamente, puede hacer una transición “espontánea”, utilizando ϵ como entrada en lugar de un símbolo de entrada. En una transición, el autómata a pila:

1. Consume de la entrada el símbolo que usa en la transición. Si como entrada se utiliza ϵ , entonces no se consume ningún símbolo de entrada.
2. Pasa a un nuevo estado, que puede o no ser el mismo que el estado anterior.
3. Reemplaza el símbolo de la parte superior de la pila por cualquier cadena. La cadena puede ser ϵ , lo que corresponde a una extracción de la pila. Podría ser el mismo símbolo que estaba anteriormente en la cima de la pila; es decir, no se realiza ningún cambio en la pila. También podría reemplazar el símbolo de la cima de la pila por otro símbolo, lo que cambiaría la cima de la pila pero no añade ni extrae ningún símbolo. Por último, el símbolo de la cima de la pila podría ser reemplazado por dos o más símbolos, lo que (posiblemente) tendría el efecto de cambiar el símbolo de la cima de la pila, añadiendo después uno o más nuevos símbolos a la pila. [1] [2]

🚦 AUTOMATA DE PILA NO DETERMINISTA (ADPND)

Un autómata de pila no determinista P es una colección de siete elementos [3] [1]

$$P=(Q,\Sigma,\Gamma,\Delta,s,F,z)$$

Donde:

- Q es un conjunto finito de **estados**, denotados por $q_0, q_1, q_2, \dots, q_0, q_1, q_2, \dots$
- Σ es el alfabeto de entrada
- Γ es un alfabeto llamado “alfabeto de pila”
- Δ es una regla de transición
- $s \in Q$ es el estado inicial
- $z \in \Gamma$ es el símbolo inicial de la pila
- $F \subseteq Q$ es el conjunto de estados finales

Δ se define por medio de ternas de la forma (q, σ, γ)

Donde:

- q es un estado de Q
- σ es un símbolo de $\Sigma \cup \{\epsilon\}$ y $\gamma \in \Gamma$

El resultado es una colección de pares (p, w) donde $p \in Q$ es el estado siguiente y $w \in \Gamma^*$ es la cadena que se introducirá en la pila en lugar del símbolo γ que estaba antes allí.

La terna (q, w, u) , donde q es el estado actual, w es la cadena de entrada restante y u el contenido de la pila (con el símbolo de la cima en el extremo de la izquierda), se llama descripción instantánea del autómata.

$$(q_1, aw, bx) \vdash (q_2, w, yx)$$

Representa el resultado de realizar una transición en la que $(q_2, y) \in (q_1, a, b)$

➤ **Ejemplo:**

Diseñar un ADPND para el siguiente lenguaje

$$\{a^{n+2}b^n\}$$

$$Q=\{q_0, q_1, q_2, q_3\}$$

$$Z=A$$

$$\Sigma=\{a, b\}$$

$$F=\{q_3\}$$

$$\Gamma=\{A, B\}$$

$$S=q_0$$

Posibles cadenas:

$$W = \{aaab, aaaabb, aaaaabbb, \dots\}$$

Tabla de transiciones:

Δ	(a,Z)	(a,A)	(b,A)	(ϵ ,A)	(ϵ ,Z)
q0	$\{(q_0, AZ)\}$	$\{(q_0, AA)\}$	$\{(q_1, \epsilon)\}$		
q1			$\{(q_1, \epsilon)\}$	$\{(q_2, \epsilon)\}$	
q2				$\{(q_2, \epsilon)\}$	$\{(q_2, Z)\}$

Representación de transiciones:

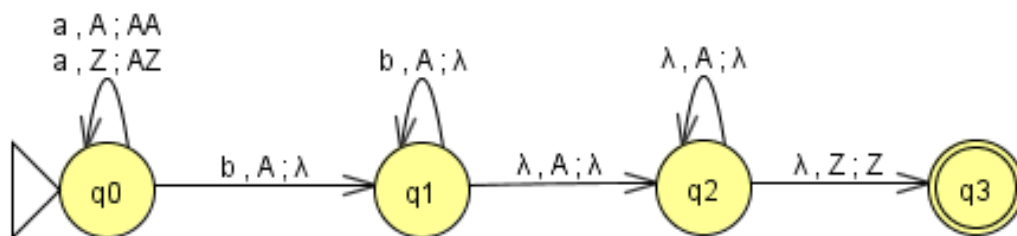


Imagen 2

Autómata de pila para $\{a^{n+2}b^n\}$

Probando la cadena “aaaabb”

$(q_0, aaaabb, z) \vdash (q_0, aaabb, AZ)$

$\vdash (q_0, aabb, AAZ)$

$\vdash (q_0, abb, AAAZ)$

$\vdash (q_0, bb, AAAAZ)$

$\vdash (q_1, b, AAAZ)$

$\vdash (q_2, \varepsilon, AAZ)$

$\vdash (q_2, \varepsilon, AZ)$

$\vdash (q_3, \varepsilon, Z)$

Material y equipo

🚦 Sistema Operativo Ubuntu

🚦 Python 3

Desarrollo de la práctica

Objetivo: hacer un programa para procesar autómatas de pila (ADPND).

Ejercicios:

1) Hacer un programa en python que reconozca el siguiente autómata de pila

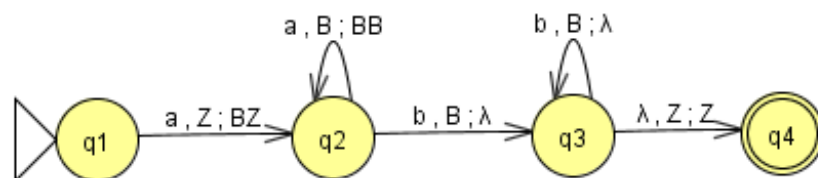


Imagen 3

ADPND del ejercicio 1

$Q = \{q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b\}$

$\Gamma = \{A, B, Z\}$

$Z = Z$

$F = \{q_4\}$

$S = q_0$

Tabla de transiciones:

Δ	(a,Z)	(a,B)	(b,B)	(ϵ ,Z)
q1	{{(q2,BZ)}}			
q2		{{(q2,BB)}}	{{(q3, ϵ)}}	
q3			{{(q3, ϵ)}}	{{(q4,Z)}}

Programa en python:

```

1 Sigma=['a','b']
2 #Q=['q0','q1']
3 F=['q4']
4 s='q1'
5 z='Z'
6
7 delta={ ('q1','a','Z') : ('q2','BZ'),
8         ('q2','a','B') : ('q2','BB'),
9         ('q2','b','B') : ('q3',''),
10        ('q3','b','B') : ('q3',''),
11        ('q3','','Z') : ('q4','Z')
12        }
13
14 pila=[z]
15
16 def transicion(estado,sigma,gamma):
17     global delta,pila
18
19     if (estado,sigma,gamma) not in delta.keys():
20         print("Error")
21         exit(0)
22     t=delta[(estado,sigma,gamma)]
23     print( (estado,sigma,gamma), "->", t)
24     for g in t[1][::-1]:
25         pila.append(g)
26     print("pila:",pila[::-1])
27     return t[0]
28
29 def procesa_cadena(w):
30     global s,pila
31     estado=s
32     pila=[z]
33     for sigma in w:
34         # print("1",pila)
35         gamma=pila.pop()
36         # print("2",pila)
37         estado= transicion(estado,sigma,gamma)
38         gamma=pila.pop()
39         if (estado,'',gamma) in delta.keys():
40             t=delta[(estado,'',gamma)]
41             print( (estado,'',gamma), "->", t)
42             estado=t[0]
43
44         for g in t[1][::-1]:
45             pila.append(g)
46         print("pila",pila[::-1])
47
48     if estado in F:
49         print("\t",w,"PERTENECE AL LENGUAJE")
50     else:
51         print("\t",w,"NO PERTENECE AL LENGUAJE")
52
53 cadenas=['a','ab','aaabbb','aabb','ba']
54
55 for w in cadenas:
56     procesa_cadena(w)

```

Imagen 4

programa en pyhton del ejercicio 1

```

22     t=delta[(estado,sigma,gamma)]
23     print( (estado,sigma,gamma), "->", t)
24     for g in t[1][::-1]:
25         pila.append(g)
26     print("pila:",pila[::-1])
27     return t[0]
28
29 def procesa_cadena(w):
30     global s,pila
31     estado=s
32     pila=[z]
33     for sigma in w:
34         # print("1",pila)
35         gamma=pila.pop()
36         # print("2",pila)
37         estado= transicion(estado,sigma,gamma)
38         gamma=pila.pop()
39         if (estado,'',gamma) in delta.keys():
40             t=delta[(estado,'',gamma)]
41             print( (estado,'',gamma), "->", t)
42             estado=t[0]
43
44         for g in t[1][::-1]:
45             pila.append(g)
46         print("pila",pila[::-1])
47
48     if estado in F:
49         print("\t",w,"PERTENECE AL LENGUAJE")
50     else:
51         print("\t",w,"NO PERTENECE AL LENGUAJE")
52
53 cadenas=['a','ab','aaabbb','aabb','ba']
54
55 for w in cadenas:
56     procesa_cadena(w)

```

Imagen 5

programa en python del ejercicio 1

Posibles cadenas:

$W = \{ab, aabb, aaabbb, \dots\}$

Dichas cadenas se muestran en la imagen 5 incluyendo algunas que no son aceptadas para comprobar que el programa funcione correctamente.

```
ubuntu@ubuntu: ~/Documentos/practica5PUSHDOWNAUTOMATA$ python3 pila_aut.py
('q1', 'a', 'Z') -> ('q2', 'BZ')
pila: ['B', 'Z']
a NO PERTENECE AL LENGUAJE
('q1', 'a', 'Z') -> ('q2', 'BZ')
pila: ['B', 'Z']
('q2', 'b', 'B') -> ('q3', '')
pila: ['Z']
('q3', '', 'Z') -> ('q4', 'Z')
pila ['Z']
ab PERTENECE AL LENGUAJE
('q1', 'a', 'Z') -> ('q2', 'BZ')
pila: ['B', 'Z']
('q2', 'a', 'B') -> ('q2', 'BB')
pila: ['B', 'B', 'Z']
('q2', 'a', 'B') -> ('q2', 'BB')
pila: ['B', 'B', 'B', 'Z']
('q2', 'b', 'B') -> ('q3', '')
pila: ['B', 'B', 'Z']
('q3', 'b', 'B') -> ('q3', '')
pila: ['B', 'Z']
('q3', 'b', 'B') -> ('q3', '')
pila: ['Z']
('q3', '', 'Z') -> ('q4', 'Z')
pila ['Z']
aaabbb PERTENECE AL LENGUAJE
('q1', 'a', 'Z') -> ('q2', 'BZ')
pila: ['B', 'Z']
('q2', 'a', 'B') -> ('q2', 'BB')
pila: ['B', 'B', 'Z']
('q2', 'b', 'B') -> ('q3', '')
pila: ['B', 'Z']
('q3', 'b', 'B') -> ('q3', '')
pila: ['Z']
('q3', '', 'Z') -> ('q4', 'Z')
pila ['Z']
aabb PERTENECE AL LENGUAJE
Error
ubuntu@ubuntu: ~/Documentos/practica5PUSHDOWNAUTOMATA$
```

Imagen 6

Compilación del autómata del ejercicio 1

Como se observa en la imagen 6, la cadena “a” no pertenece al lenguaje, mientras que las cadenas “ab”, “aaabbb” son aceptadas, y cuando el programa lee la cadena “b” nos marca un error debido a que dicho autómata solo acepta el mismo número de a’s y de b’s siempre y cuando las cadenas empiecen con la letra “a”.

Conclusiones

Con esta práctica aprendí el funcionamiento de un ADPND que, como se mencionó anteriormente es una autómata finito no determinista con transiciones $-\epsilon$ cuyo funcionamiento depende del estado de una pila asociada al autómata, además el ADPND a diferencia de autómatas vistos en prácticas pasadas, acepta lenguajes no regulares.

Referencias

- [1] J. E. Hopcroft, R. Motwani y J. D. Ullman, Introducción a la teoría de autómatas, lenguajes y computación, Madrid: PEARSON EDUCACIÓN S.A, 2007.
- [2] D. Kelley, Teoría de autómatas y lenguajes formales, Madrid: PEARSON EDUCACIÓN, S. A, 1995.
- [3] D. G. P. Luis Migel Pardo Vasallo, «Teoria de Automatas Finitos,» [En línea]. Available: https://ocw.unican.es/pluginfile.php/1516/course/section/1946/2-1_Introduccion.pdf.