Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la computación
CC3102-1 Teoría de la Computación

Tarea n°2

Integrantes: Tomás Cortez

Profesor: Alejandro Hevia
Auxiliares: Ivana Bachmann Espinoza.
Rodrigo Fuentes Z.
Vicente Rojas C.

Fecha de entrega: 6 de diciembre de 2019
Santiago, Chile

1. Problema 1

1.1.

Para este problema se partió pensando en las posibles formas de hacer el string, naturalmente el string vacío y el que esté formado solamente por unos son aceptados y fueron los primeros en hacerse parte de la GLC.

$S \to \varepsilon$

$S \to US$

$U \to 1$

Luego se pensó principalmente en tres maneras de formar strings que incorporasen ceros. Se sabe que teniendo cuatro ceros continuos se rompe la regla, por lo tanto estas formas tienen un cero, dos ceros o tres ceros cotinuos. Se termina en 1 para evitar romper la regla.

Primero se tiene la forma con un cero, presenta el substring 01, concatenar 01 no rompe la regla y podemos decir que los próximos substring que vienen pueden contener cualquiera de las formas que se verán más adelante,ya que la cantidad de ceros y de unos es la misma en el substring, por ejemplo 01000 es válido. También se puede terminar en un 0.

Así se obtiene:

$S \to A$

$A \to 01|01A|0$

Para el segundo caso se tiene la forma con dos ceros, presenta el substring 001, aquí se puede decir que se vuelve al nivel anterior, a menos que se coloque un 1 extra y se vuelve al primer nivel.

$S \to B$

$B \to 001|001A|0011S|00$

Para el tercer caso se tiene la forma con tres ceros, aquí se pueden agregar unos para volver a la forma base o a la anterior, ya que agregar un cero sería romper la regla

$S \to C$

$C \to 0001|000111S|00011B|000$

2. Problema 2

Para crear el AP que reconociera el lenguaje L se siguió la siguiente lógica: Primero se agrega un signo $ a la pila para indicar el comienzo, luego por cada 2 leído en la entrada se agregan dos a consecutivas y por cada 1 se agrega una a. Luego al encontrar el símbolo = por cada 5 leído se desapilan 5 a consecutivas, de no poderse desapilar el autómata rechaza.

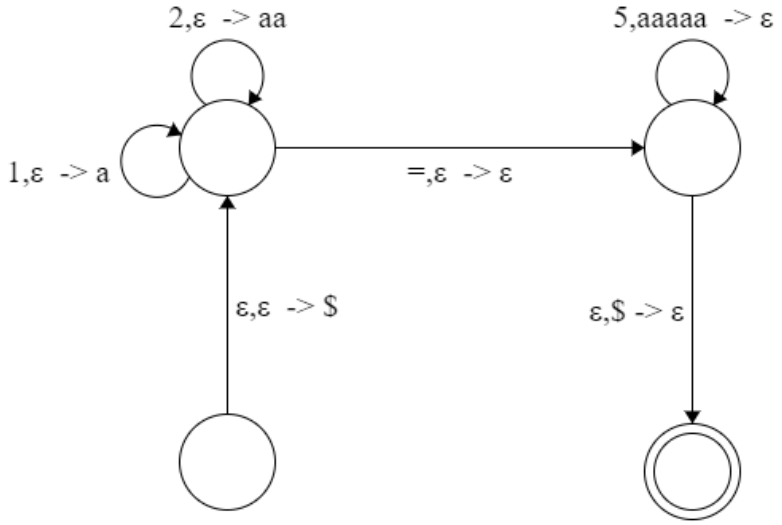

Figura 2.1: Esquema AP

3. Problema 3

3.1.

$0^N1^N0^N1^N \forall n \geq 0$

Se utilizará lema de bombeo para demostrar que el lenguaje no es libre de contexto.

$\forall N \geq 1$ existe $w = 0^N1^N0^N1^N$, $|w| \geq N$

Se puede escribir el string w =xuyvz

tal que $|uyv| \leq N$ (Restricción I)

$|uv| \geq 1$ (Restricción II)

Luego se tienen dos casos:

i) u y v contienen símbolos iguales, para este caso u y v contienen solamente 0 o solamente 1, se puede bombear con p = 0 y se obtendrá un string desbalanceado, ya que se cambiará el número de símbolos solamente en una de las cuatro partes (por que se debe bombear al menos un símbolo por la restricción II) y todas deben mantener la misma cantidad de símbolos.

ii) u y v contienen símbolos distintos, para este caso u y v contienen 0s y 1s o 1s y 0s, se puede bombear nuevamente con p=0 y se obtendrá un string desbalanceado, ya que se cambiará el número de símbolos solamente en dos de las cuatro partes (por que se debe bombear al menos un símbolo por la restricción II) y todas deben mantener la misma cantidad de símbolos.

No se puede cambiar el número de símbolos en más de dos partes, ya que cada una de las partes del string es de tamaño N y por la restricción I no se puede lograr.

Por lo tanto queda demostrado que el lenguaje no cumple el lema de bombeo, por ende no es libre de contexto.

3.2.

$t_1\#t_2\#...\#t_k | k \geq 2$, cada $t_i \in a, b^*$, y $t_i = t_j$, para algún $i \neq j$

Se utilizará lema de bombeo para demostrar que el lenguaje no es libre de contexto.

$\forall N \geq 1$ existe $w = 0^N1^N\#0^N1^N$, $|w| \geq N$

Se puede escribir el string w =xuyvz

Tal que $|uyv| \leq N$ (Restricción I)

$|uv| \geq 1$ (Restricción II)

Luego se tienen dos casos:

i) uyv está de un lado del símbolo #, por lo tanto si bombeamos con p = 0, $t_1 \neq t_2$, ya que se obtiene $0^{N-p}1^{N-p'}\#0^N1^N$ con $p \vee p' \geq 1$ por la restricción II, por tanto la palabra sale del lenguaje.Es análogo para el otro lado del símbolo gato

ii) uyv toma el símbolo # y para este caso también hay dos casos, si u o v contiene #, basta con bombear a 0 y la palabra saldrá del lenguaje, ya que solamente contaría como una palabra. Si no lo contiene bastará con bombear con p = 0, lo que resultará en un string de la forma $0^N1^{N-p}\#0^{N-p'}1^N$ con $p \vee p' \geq 1$ por la restricción II, por tanto sale del lenguaje.

No se pueden bombear todos los ceros y todos los unos a la vez por la restricción I.

Por lo tanto queda demostrado que el lenguaje no cumple el lema de bombeo, por ende no es libre de contexto.

4. Problema 4

Demostrar los Lenguajes libres de contexto son cerrados bajo el operador SUFFIX(A).

Idea: Se tendrá dos copias del autómata que reconoce A, una sin modificaciones y otra copia que todas sus transiciones son leyendo ϵ, pero manteniendo los cambios en la pila, además esta copia tendrá transiciones que leen ϵ sin modificar la pila que lo llevarán al estado correspondiente del primer autómata mencionado.

Sea A un lenguaje LC, éste tiene un AP $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ que lo reconoce. Luego se construye el autómata $M' = (Q', \Sigma, \Gamma, \delta', q_0^*, F)$

Donde:

$Q' = Q \cup Q^*$ donde Q^* son los mismos estados de Q, pero etiquetados de otra forma.$\forall q_i \in Q, \ni q_i^* \in Q^*$

$$\delta(q', w', p') = \begin{cases} \delta(q_i, w, p) & si, q' = q_i \in Q, w' = w, p' = p \\ (q_i, \epsilon) & si, q' = q_i^* \in Q^*, w = \epsilon, p = \epsilon \\ (q_{i+1}^*, r) & q' = q_i^* \in Q^*, w' = \epsilon, p' = p, \delta(q_i, w, p) = (q_{i+1}, r) \end{cases}$$

La relación de transición anterior logra hacer todo lo que se esperaba de la idea.

Cuando se lee una palabra primero se avanza por el 'primer' autómata llenando el stack sin leer nada, luego no determinísticamente el autómata sabe cuando pasar al 'segundo' autómata para aceptar los sufijos de A. Si una palabra no es sufijo de A, llegará el momento en el que están en algún estado lea un símbolo y rechace.

5. Problema 5

Para reconocer si efectivamente si $w \doteq t$ se utilizará un autómata de tres cintas el cual tendrá una cinta por cada símbolo en el diccionario de las palabra, como es un diccionario binario se utilizan dos cintas. Se escribe # en las otras dos cintas, luego se recorre la entrada y por cada símbolo leído se deposita una marca en la cinta correspondiente a éste. Una vez se llega a # se coloca un # en todas las cintas, luego todas las cintas menos la principal se mueven hacia la izquierda hasta encontrar # (el que fue escrito al comienzo) para después avanzar una vez a la derecha , luego la cinta principal lee la segunda palabra y por cada símbolo leído se lee en la cinta correspondiente, si se lee # se rechaza Si no se rechaza se avanza a la derecha |esto se repite hasta que se llega a blanco en la cinta principal, una vez pasa esto se ven las otras cintas y si todas leen # se acepta, ya que indica que la cantidad de símbolo no fue menor ni mayor.

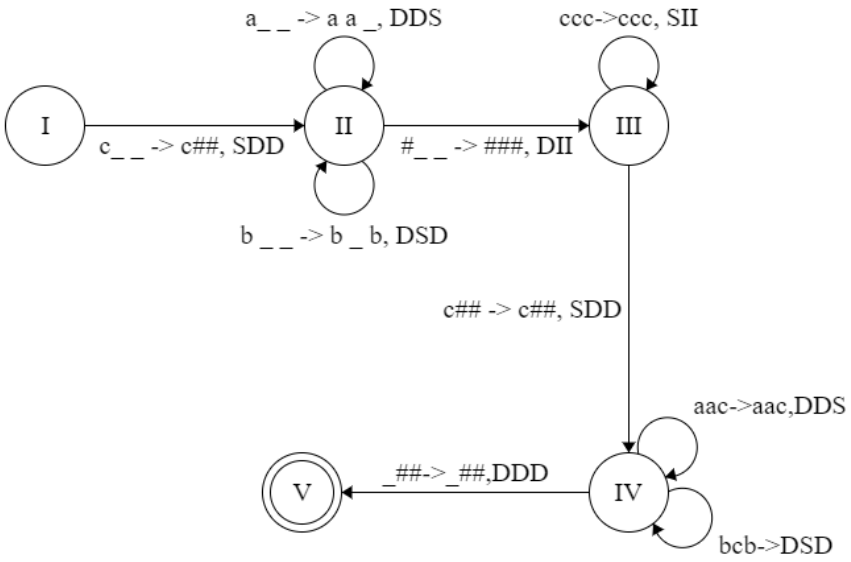

Figura 5.1: Esquema

En el esquema se utiliza la letra a para indicar un elemento del diccionario binario y b para el otro. C se utiliza para cualquiera de los dos.