ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Primer Parcial — **DIS**
Juan Francisco Cardona M — **Febrero 25 de 2010**

Nombre: ____________________
Código: ____________________

(10 %) 1. **Lenguajes y expresiones regulares**
Determine el lenguaje que representa a cada una de las siguientes expresiones regulares:

1. $r_1 = a \mid b$.
2. $r_2 = ab^* \mid bc^*$.

Solución:

1. $L(r_1) = \{a, b\}$
2. $L(r_2) = \{ab^n, bc^n \mid 0 \leq n\}$

(20 %) 2. **Expresiones regulares y lenguajes**
Encuentre una expresion regular que describa a cada uno de los siguientes lenguajes:

1. $L(r_3) = \{a, b, c\}$
2. $L(r_4) = \{\epsilon, a, abb, abbbb, \ldots, ab^{2n}, \ldots\}, n \in \mathbb{N}$

Solución:

1. $r_3 = a \mid b \mid c$
2. $r_4 = (a(bb)^* \mid \epsilon)$

(20 %) 3. **Gramáticas independientes de contexto**
Encuentre una gramática para cada uno de los siguientes lenguajes.

1. $L(G_1) = \{bb, bbbb, bbbbbb, \ldots\} = \{(bb)^{n+1} \mid n \in \mathbb{N}\}$.
2. $L(G_2) = \{\epsilon, ac, abc, abbc, \ldots, ab^n c, \ldots\} = \{ab^n c \mid n \in \mathbb{N}\}$.

Solución:

1. $G_1 = (\{S\}, \{b\}, \{S \to bbS \mid bb\}, S)$
2. $G_2 = (\{S, D\}, \{a, b, c\}, \{S \to \epsilon \mid aDc, D \to b \mid bD\}, S)$

(20 %) 4. **Ambiguedad**

1. Muestre que siguiente gramática es ambigua: $G_3 = (\{S\}, \{a, b\}, \{S \rightarrow a \mid SbS\}, S)$
2. Sí la anterior gramática es ambigua, escriba una nueva (G_3') que elimine la ambiguedad.

Solución:

1. Primera derivacion: $S \Rightarrow SbS \Rightarrow SbSbS \Rightarrow abSbS \Rightarrow ababS \Rightarrow ababa$
   Segunda derivación: $S \Rightarrow SbS \Rightarrow SbSbS \Rightarrow SbSba \Rightarrow Sbaba \Rightarrow ababa$
2. $G_3' = (\{S\}, \{a, b\}, \{S \rightarrow a \mid abS\}, S)$

(30 %) 5. **Transformación de gramáticas**
Considere la siguiente gramática $G_4 = (\{S, A, B, C, D, E, F\}, \{a, b, c\}, P, S)$, donde el conjunto P está definido como:

$$
\begin{array}{rcl}
S & \rightarrow & aABb \mid AE \\
A & \rightarrow & aAbE \mid \epsilon \\
B & \rightarrow & abB \\
C & \rightarrow & ABC \mid c \\
D & \rightarrow & AAB \\
E & \rightarrow & CA \\
F & \rightarrow & FA \mid a
\end{array}
$$

Responda las siguientes preguntas:

1. Identifique los no-terminales indefinidos de la gramática G_4.
2. Identifique los no-terminales no alcanzables de la gramática G_4.
3. Transforme la gramática G_4 a una gramática G_4' sin reglas inutiles.

Solución:

1. Calculamos el conjunto DEF Paso 1: Las siguientes producciones cumplen con la condición $(A \rightarrow u) \in P$ y $u \in \Sigma^*$: $A \rightarrow \epsilon$, $C \rightarrow c$ y $F \rightarrow a$.

$$DEF \coloneqq \{A, C, F\}$$

Paso 2: La siguiente producción cumple con la condición $(B \rightarrow D_1 D_2 \ldots D_n) \in P$ donde cada D_i es un terminal o un no-terminal presente en DEF: $E \rightarrow CA$.

$$DEF \coloneqq DEF \cup \{E\} = \{A, C, F, E\}$$

Paso 3: La siguiente producción cumple con la condición del paso 2: $S \rightarrow AE$.

$$DEF \coloneqq DEF \cup \{S\} = \{A, C, F, E, S\}$$

El conjunto de los no-terminales indefinidos es:

$$UNDEF = V \setminus DEF = \{S, A, B, C, D, E, F\} \setminus \{A, C, F, E, S\} = \{B, D\}$$

2. Se determina el grafo de produce a y encontramos.

$$
\begin{array}{rcl}
S & \rightarrow & A \\
S & \rightarrow & B \\
S & \rightarrow & E \\
E & \rightarrow & C \\
C & \rightarrow & B \\
C & \rightarrow & A \\
F & \rightarrow & A \\
D & \rightarrow & A \\
D & \rightarrow & B
\end{array}
$$

Observamos que los no-terminales $\{D, F\}$ no son alcanzables desde el axioma.

3. De los puntos anteriores obtenemos la siguiente gramática:

$$\begin{array}{lcl} S & \rightarrow & AE \\ A & \rightarrow & aAbE \mid \epsilon \\ C & \rightarrow & c \\ E & \rightarrow & CA \\ F & \rightarrow & FA \mid a \end{array}$$

Pregunta	Puntos	Obtenido
1	10	
2	20	
3	20	
4	20	
5	30	
Total:	100	

Nombre: ______________________________
Código: ______________________________

(35 %) 1. **Construcción de autómatas a partir de una expresión regular**

1. Utilice el método estructural de Thompson para convertir la expresión regular

$$re_1 = (a \mid b)^*(a \mid b)^*$$

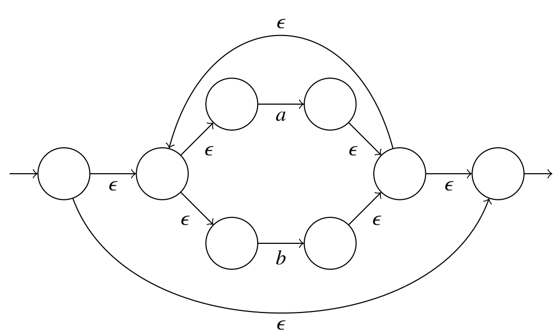
en un AFN (AFN_ϵ_1) con movimientos espontáneos.

2. Elimine en el autómata (AFN_ϵ_1) los movimientos espontáneos y conviértalo en un autómata no determinista sin movimientos espontáneos AFN (AFN_1).
3. Convierta el AFN (AFN_1) en un AFD (AFD_1) utilizando el método de los conjuntos de potencia.

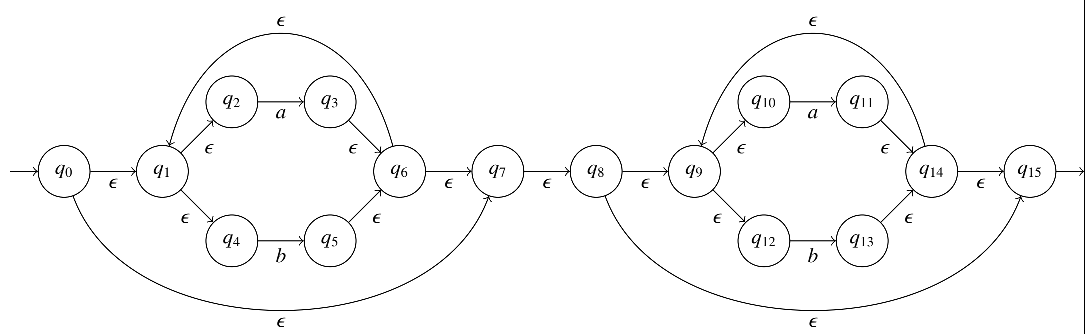
Solución:

1. En primer lugar creamos utilizando el método de Thompson. Observamos primero que la expresión regular se compone de dos sub-expresiones similares, $(a \mid b)^*$. Podemos construir el autómata de una de ellas y luego las unimos.

   Para la sub-expresión $(a \mid b)^*$ obtenemos el siguiente autómata con movimientos expontáneos.



   Ahora unimos las dos sub-expresiones que están concatenadas, las numeramos y nos produce el siguiente autómata:



2. Ahora creamos la gramática independiente de contexto del autómata que obtuvimos en el punto anterior.

$$
\begin{array}{lcl}
q_0 & \rightarrow & q_1 \\
q_0 & \rightarrow & q_7 \\
q_1 & \rightarrow & q_2 \\
q_1 & \rightarrow & q_4 \\
q_2 & \rightarrow & aq_3 \\
q_3 & \rightarrow & q_6 \\
q_4 & \rightarrow & bq_5 \\
q_5 & \rightarrow & q_6 \\
q_6 & \rightarrow & q_1 \\
q_6 & \rightarrow & q_7 \\
q_7 & \rightarrow & q_8
\end{array}
\qquad
\begin{array}{rcl}
q_8 & \rightarrow & q_9 \\
q_8 & \rightarrow & q_{15} \\
q_9 & \rightarrow & q_{12} \\
q_{10} & \rightarrow & aq_{11} \\
q_{11} & \rightarrow & q_{14} \\
q_{12} & \rightarrow & bq_{13} \\
q_{13} & \rightarrow & q_{14} \\
q_{14} & \rightarrow & q_9 \\
q_{14} & \rightarrow & q_{15} \\
q_{15} & \rightarrow & \epsilon
\end{array}
$$

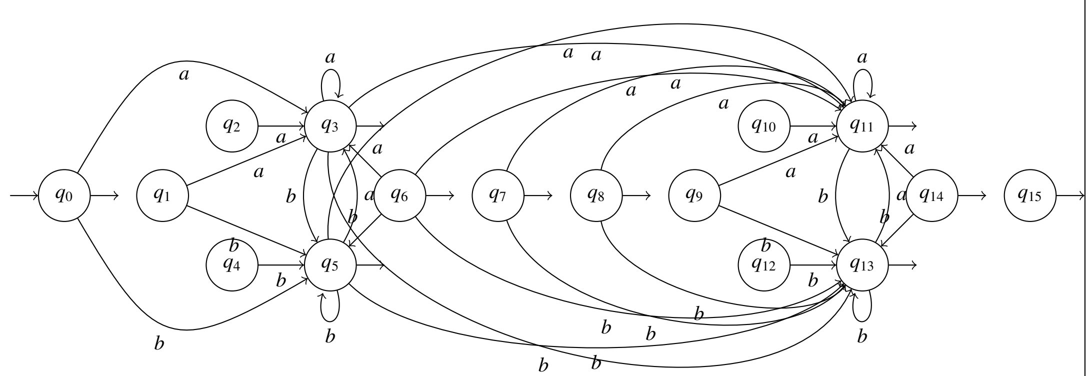
De esta gramática obtenemos el conjunto de copia:

Estado	**Conjunto de copia**
q_0	$q_0, q_1, q_2, q_4, q_7, q_8, q_{15}, q_9, q_{10}, q_{12}$
q_1	q_1, q_2, q_4
q_2	q_2
q_3	$q_3, q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_4	q_4
q_5	$q_5, q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_6	$q_6, q_1, q_2, q_4, q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_7	$q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_8	$q_8, q_9, q_{10}, q_{12}, q_{15}$
q_9	q_9, q_{10}, q_{12}
q_{10}	q_{10}
q_{11}	$q_{11}, q_{14}, q_9, q_{10}, q_{12}, q_{15}$
q_{12}	q_{12}
q_{13}	$q_{13}, q_{14}, q_9, q_{10}, q_{12}, q_{15}$
q_{14}	$q_{14}, q_9, q_{10}, q_{12}, q_{15}$
q_{15}	q_{15}

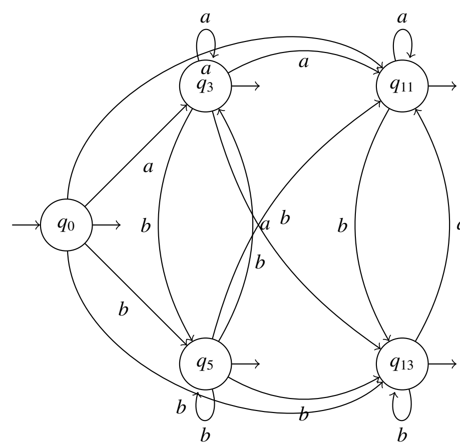
Del conjunto de copia redefinimos la gramática:

$$
\begin{array}{lll}
q_0 \rightarrow aq_3 & q_5 \rightarrow aq_3 & q_9 \rightarrow bq_{13} \\
q_0 \rightarrow bq_5 & q_5 \rightarrow bq_5 & q_{10} \rightarrow aq_{11} \\
q_0 \rightarrow aq_{11} & q_5 \rightarrow aq_{11} & q_{11} \rightarrow aq_{11} \\
q_0 \rightarrow bq_{13} & q_5 \rightarrow bq_{13} & q_{11} \rightarrow bq_{13} \\
q_0 \rightarrow \epsilon & q_5 \rightarrow \epsilon & q_{11} \rightarrow \epsilon \\
q_1 \rightarrow aq_3 & q_6 \rightarrow aq_3 & q_{12} \rightarrow bq_{13} \\
q_1 \rightarrow bq_5 & q_6 \rightarrow bq_5 & q_{13} \rightarrow aq_{11} \\
q_1 \rightarrow \epsilon & q_7 \rightarrow aq_{11} & q_{13} \rightarrow bq_{13} \\
q_2 \rightarrow aq_3 & q_7 \rightarrow bq_{13} & q_{13} \rightarrow \epsilon \\
q_3 \rightarrow aq_3 & q_7 \rightarrow \epsilon & q_{14} \rightarrow aq_{11} \\
q_3 \rightarrow bq_5 & q_8 \rightarrow aq_{11} & q_{14} \rightarrow bq_{13} \\
q_3 \rightarrow aq_{11} & q_8 \rightarrow bq_{13} & q_{14} \rightarrow \epsilon \\
q_3 \rightarrow bq_{13} & q_8 \rightarrow \epsilon & q_{15} \rightarrow \epsilon \\
q_3 \rightarrow \epsilon & q_9 \rightarrow aq_{11} & \\
q_4 \rightarrow bq_5 & & \\
\end{array}
$$

Ahora con la gramática redefinida, construimos el autómata que corresponde a esta gramática:



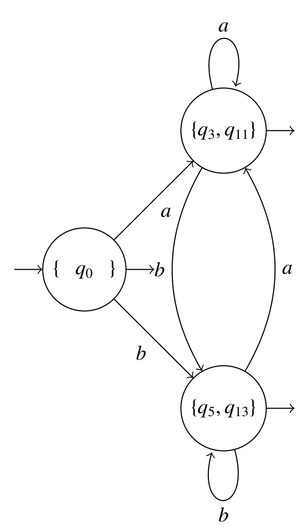
Pero este autómata tiene algunos estados que son no accesibles: q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{14}. Y también tiene un autómata en un estado no postaccesible: q_{15}. Por lo tanto hay que hacer limpieza del autómata, el autómata correspondiente es el siguiente:



3. A partir del anterior autómata no determinista, podemos pasar a generar el autómata determinista. Primero construimos el conjunto potencia de las transiciones:

$$
\begin{array}{rcl}
\{q_0\} & \xrightarrow{a} & \{q_3, q_{11}\} \\
\{q_0\} & \xrightarrow{b} & \{q_5, q_{13}\} \\
\{q_3, q_{11}\} & \xrightarrow{a} & \{q_3, q_{11}\} \\
\{q_3, q_{11}\} & \xrightarrow{b} & \{q_5, q_{13}\} \\
\{q_5, q_{13}\} & \xrightarrow{a} & \{q_3, q_{11}\} \\
\{q_5, q_{13}\} & \xrightarrow{b} & \{q_5, q_{13}\}
\end{array}
$$

Con estas transiciones podemos generar el autómata determinista:



(15 %) 2. **Seguimiento de autómatas no deterministas**
Para el autómata finito no determinista (AFN_2) de la figura 1, indique 3 caminos etiquetados como *aabb*. ¿El AFN_2 acepta la cadena *aabb*?
Nota: En cada caso explicar su respuesta con una derivación.

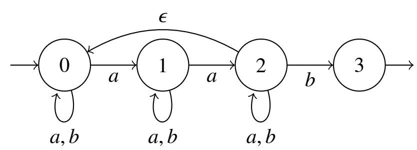


Figura 1: Un reconocedor de estado finito no-determinista (AFN_2)

Solución:

Con el anterior autómata se puede construir los siguientes caminos:

1	$0 \xrightarrow{a} 0 \xrightarrow{a} 0 \xrightarrow{b} 0 \xrightarrow{b} 0$
2	$0 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{b} 1 \xrightarrow{b} 1$
3	$0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{b} 2 \xrightarrow{b} 3$

Del tercer camino, al llegar al final del mismo se encuentra en un estado de aceptación por lo tanto la cadena es aceptada.

(35 %) 3. **Generación de autómatas utilizando el método de *Berry* y *Sethi***
Dada la siguiente expresión regular (re_2), generar un autómata deterministas utilizando el método de *Berry* y *Sethi* (BC).

$$re_2 = (a \mid b)^* b(ac)^+$$

Solución:

En primer lugar renumeramos la expresion re_2 y definimos la nueva:

$$re'_2 = (a_1 \mid b_2)^* b_3(a_4c_5)^+$$

Ahora, calculamos el conjunto de los digrafos de $Dig(re'_2)$:

$$\begin{aligned} Dig(re'_2) &= Dig((a_1 \mid b_2)^* b_3(a_4c_5)^+) \\ &= Dig((a_1 \mid b_2)^*) \cup Dig(b_3(a_4c_5)^+) \cup Fin((a_1 \mid b_2)^*)Ini(b_3(a_4c_5)^+) \end{aligned} \quad (1)$$

Tenemos que (1) se subdivide en tres subexpresiones: (2), (3), (4).

$$Dig((a_1 \mid b_2)^*) \quad (2)$$

$$Dig(b_3(a_4c_5)^+) \tag{3}$$

$$Fin((a_1 \mid b_2)^*)Ini(b_3(a_4c_5)^+) \tag{4}$$

Resolvamos (2):

$$\begin{aligned}
&= Dig((a_1 \mid b_2)^*) \\
&= Dig(a_1 \mid b_2) \cup Fin(a_1 \mid b_2)Ini(a_1 \mid b_2) \\
&= Dig(a_1) \cup Dig(b_2) \cup (Fin(a_1) \cup Fin(b_2))(Ini(a_1) \cup Ini(a_2)) \\
&= \emptyset \cup \emptyset \cup (\{a_1\} \cup \{b_2\})(\{a_1\} \cup \{b_2\}) \\
&= \{a_1, b_2\}\{a_1, b_2\} \\
&= \{a_1a_1, a_1b_2, b_2a_1, b_2b_2\}
\end{aligned}$$

La solución de (2) es (5).

$$\{a_1a_1, a_1b_2, b_2a_1, b_2, b_2\} \tag{5}$$

Resolvamos ahora (3):

$$\begin{aligned}
&= Dig(b_3(a_4c_5)^+) \\
&= Dig(b_3) \cup Dig((a_4c_5)^+) \cup Fin(b_3)Ini(a_4c_5)^+) \\
&= \emptyset \cup (Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5)) \cup \{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4)) \\
&= (Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5)) \cup \{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4))
\end{aligned}$$

Para resolver (3) la dividimos en dos subexpresiones (6) y (7):

$$(Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5)) \tag{6}$$

$$\{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4)) \tag{7}$$

Resolvamos (6):

$$\begin{aligned}
&= Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5) \\
&= (Dig(a_4) \cup Dig(c_5) \cup Fin(a_4)Ini(c_5)) \cup (Fin(c_5) \cup Fin(a_4)Null(c_5))(Ini(a_4) \cup Null(a_4)Ini(c_5)) \\
&= (\emptyset \cup \emptyset \cup \{a_4\}\{c_5\}) \cup (\{c_5\} \cup \{a_4\}\emptyset)(\{a_4\} \cup \emptyset\{c_5\}) \\
&= \{a_4c_5\} \cup (\{c_5\})(\{a_4\}) \\
&= \{a_4c_5\} \cup \{c_5a_4\} \\
&= \{a_4c_5, c_5a_4\}
\end{aligned}$$

La solución de (6) es (8)

$$\{a_4c_5, c_5a_4\} \tag{8}$$

Resolvamos (7):

$$\begin{aligned}
&= \{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4)) \\
&= \{b_3\}(\{a_4\} \cup \emptyset\{a_4\}) \\
&= \{b_3\}(\{a_4\}) \\
&= \{b_3a_4\}
\end{aligned}$$

La solución de (7) es (9):

$$\{b_3a_4\} \tag{9}$$

Para resolver a (3), tenemos las soluciones de (6) es (8) y (7) es (9):

$$
\begin{aligned}
&= \{a_4c_5, c_5a_4\} \cup \{b_3a_4\} \\
&= \{a_4c_5, c_5a_4, b_3a_4\}
\end{aligned}
$$

La solución de (3) es (10):

$$\{a_4c_5, c_5a_4, b_3a_4\} \tag{10}$$

Nos falta solucionar (4):

$$
\begin{aligned}
&= Fin((a_1 \mid b_2)^*)Ini(b_3(a_4c_5)^+) \\
&= Fin(a_1 \mid b_2)(Ini(b_3) \cup Null(b_3)Ini((a_4c_5)^+) \\
&= (Fin(a_1) \cup Fin(b_2))(\{b_3\} \cup \emptyset Ini(a_4c_5)) \\
&= (\{a_1\} \cup \{b_2\})(\{b_3\} \cup \emptyset) \\
&= (\{a_1, b_2\})\{b_3\} \\
&= \{a_1b_3, b_2b_3\}
\end{aligned}
$$

La solución de (4) es (11):

$$\{a_1b_3, b_2b_3\} \tag{11}$$

De (5), (10) y (11) podemos resolver a (1):

$$
\begin{aligned}
&= \{a_1a_1, a_1b_2, b_2a_1, b_2b_2\} \cup \{a_4c_5, c_5a_4, b_3a_4\} \cup \{a_1b_3, b_2b_3\} \\
&= \{a_1a_1, a_1b_2, b_2a_1, b_2b_2, a_1b_3, b_2b_3, a_4c_5, c_5a_4, b_3a_4\}
\end{aligned}
$$

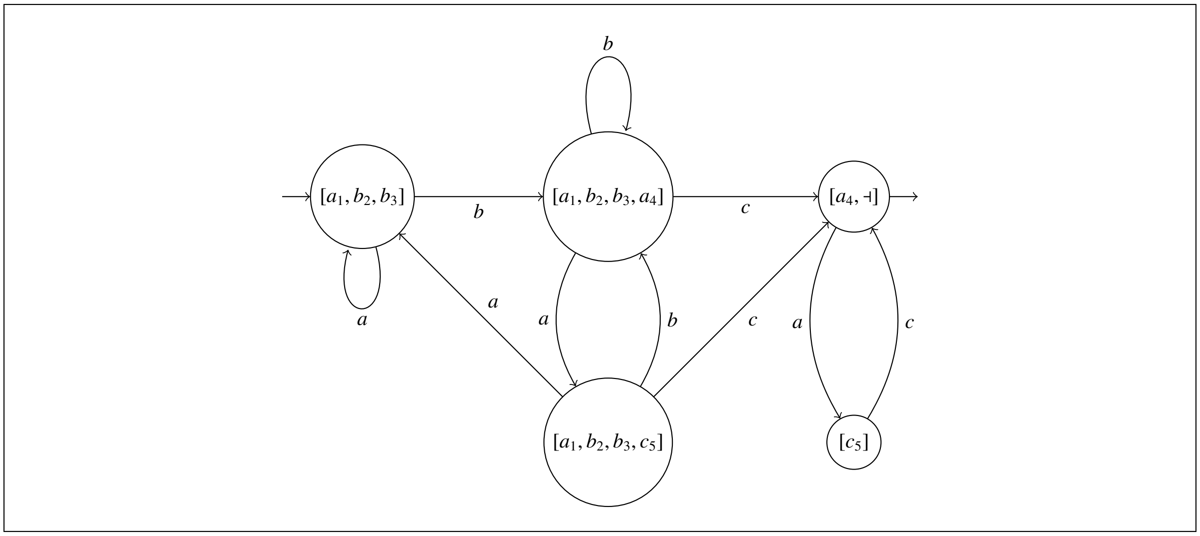
Entonces,

$$Dig(re_2') = \{a_1a_1, a_1b_2, b_2a_1, b_2b_2, a_1b_3, b_2b_3, a_4c_5, c_5a_4, b_3a_4\} \tag{12}$$

De (12) construyamos el conjunto Fol:

$$
\begin{array}{c|l}
 & \text{siguientes} \\
\hline
a_1 & a_1, b_2, b_3 \\
b_2 & a_1, b_2, b_3 \\
b_3 & a_4 \\
a_4 & c_5 \\
c_5 & a_4, \dashv
\end{array}
\tag{13}
$$

El autómata deterministico es mostrado a continuación:


$[a_1, b_2, b_3]$
$[a_1, b_2, b_3, a_4]$
$[a_4, \dashv]$
$[a_1, b_2, b_3, c_5]$
$[c_5]$
a
b
c

(15 %) 4. **Programación**
El siguiente programa en Java es la implementación de una autómata determinista obtenido en el punto 3 utilizando el método *Berry* y *Sethi*. Esta implementación cuenta con varios errores, responda la siguientes preguntas:

1. Identifique los errores que hay en la implementación del autómata.
2. Re-escriba las secciones de código donde los errores ocurren, para que el autómata se ejecute según su especificación obtenida en el punto 3.

```
/**
 * Codigo utilizado como parte del segundo parcial
 */
public class AutomataBerrySethi {

    /*
     * Constantes q0 hasta q4 representa los estados,
     * el estado inicial es q0, el de aceptación es q3,
     * el estado qe, es el estado de error.
     * La variable privada int state contiene el estado actual
     * de la máquina
     */
    private static final int q0 = 0;
    private static final int q1 = 1;
    private static final int q2 = 2;
    private static final int q3 = 3;
    private static final int q4 = 4;
    private static final int qe = 5;

    private int state;

    /**
     * La función de transición
     * @param s codigo de estado (un int)
     * @param c carácter para hacer una transición
     * @return el siguiente estado del código
     */
    static private int[][] delta
       = {{q0,q1},{q2,q1},{q0,q1},{q4,qe},{qe,qe},{qe,qe}};
    /**
     * Re-establece el actual estado al estado de inicio
     */
    public void reset() {
       state = q0;
    }

    /**
     * Realiza una transición en cada carácter en la cadena
     * dada.
     * @param en la cadena a utilizar
     */
    public void process(String in) {

       for (int i = 0; i < in.length(); i++) {
          char c = in.charAt(i);
          try {
             state = delta[state][c];
          }
          catch (ArrayIndexOutOfBoundsException ex) {
             state = qe;
          }
       }
    }

    /**
     * Examina si el DFA aceptó la cadena.
     * @return true si el estado final fue de aceptación
     */
    public boolean accepted() {
       return state == q3;
    }
}
```

Solución:

1. Hay dos conjuntos de errores:

   a) en la definición de la función de transición, si la observamos encontramos que están definidos únicamente por cada estado, dos posibles transiciones,

   ```
   static private int[][] delta
      = {{q0,q1},{q2,q1},{q0,q1},{q4,qe},{qe,qe},{qe,qe}};
   ```

   pero el alfabeto de la gramática consta de tres elementos $\Sigma = \{a, b, c\}$, por lo tanto, nos falta adicionar a cada fila de la tabla un elemento para el tercer carácter.

   b) el segundo error está en la instrucción que selecciona el estado.

   ```
   state = delta[state][c];
   ```

El valor de `c` selecciona con base el carácter y el estado actual `state` el nuevo estado siguiente, el problema, es que el valor de `c` es un carácter y requiere de un número para hacerlo.

2. Soluciones:

   a) El siguiente código corrige el primer error:

   ```
   static private int[][] delta
      = {{q0,q1,qe},{q2,q1,q3},{q0,q1,q3},{q4,qe,qe}
        ,{qe,qe,q3},{qe,qe,qe}};
   ```

   b) El siguiente código corrige el segundo tipo de error:

   ```
   state = delta[state][c-'a'];
   ```

Pregunta	Puntos	Obtenido
1	35	
2	15	
3	35	
4	15	
Total:	100	

Nombre: ______________________________
Código: ______________________________

(35 %) 1. **Construcción de autómatas a partir de una expresión regular**

1. Utilice el método estructural de Thompson para convertir la expresión regular

$$re_1 = (a \mid b)^*(a \mid b)^*$$

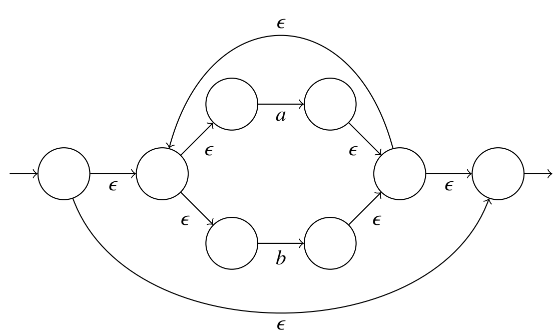
en un AFN (AFN_ϵ_1) con movimientos espontáneos.

2. Elimine en el autómata (AFN_ϵ_1) los movimientos espontáneos y conviértalo en un autómata no determinista sin movimientos espontáneos AFN (AFN_1).
3. Convierta el AFN (AFN_1) en un AFD (AFD_1) utilizando el método de los conjuntos de potencia.

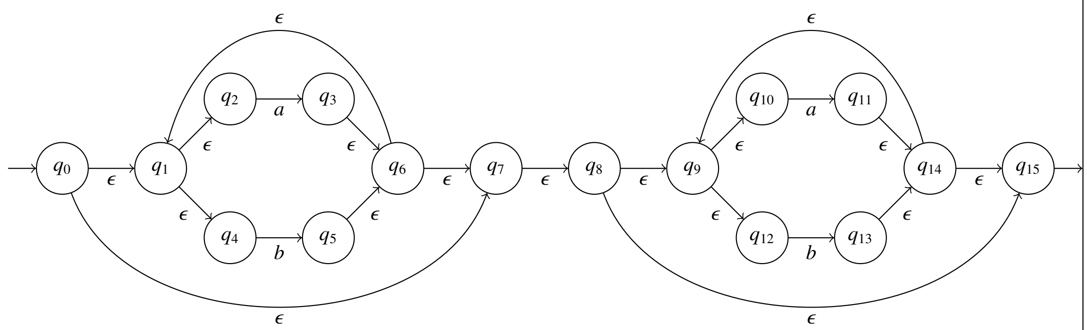
Solución:

1. En primer lugar creamos utilizando el método de Thompson. Observamos primero que la expresión regular se compone de dos sub-expresiones similares, $(a \mid b)^*$. Podemos construir el autómata de una de ellas y luego las unimos.

   Para la sub-expresión $(a \mid b)^*$ obtenemos el siguiente autómata con movimientos expontáneos.



   Ahora unimos las dos sub-expresiones que están concatenadas, las numeramos y nos produce el siguiente autómata:



2. Ahora creamos la gramática independiente de contexto del autómata que obtuvimos en el punto anterior.

$$
\begin{array}{lcl}
q_0 & \rightarrow & q_1 \\
q_0 & \rightarrow & q_7 \\
q_1 & \rightarrow & q_2 \\
q_1 & \rightarrow & q_4 \\
q_2 & \rightarrow & aq_3 \\
q_3 & \rightarrow & q_6 \\
q_4 & \rightarrow & bq_5 \\
q_5 & \rightarrow & q_6 \\
q_6 & \rightarrow & q_1 \\
q_6 & \rightarrow & q_7 \\
q_7 & \rightarrow & q_8
\end{array}
\qquad
\begin{array}{lcl}
q_8 & \rightarrow & q_9 \\
q_8 & \rightarrow & q_{15} \\
q_9 & \rightarrow & q_{12} \\
q_{10} & \rightarrow & aq_{11} \\
q_{11} & \rightarrow & q_{14} \\
q_{12} & \rightarrow & bq_{13} \\
q_{13} & \rightarrow & q_{14} \\
q_{14} & \rightarrow & q_9 \\
q_{14} & \rightarrow & q_{15} \\
q_{15} & \rightarrow & \epsilon
\end{array}
$$

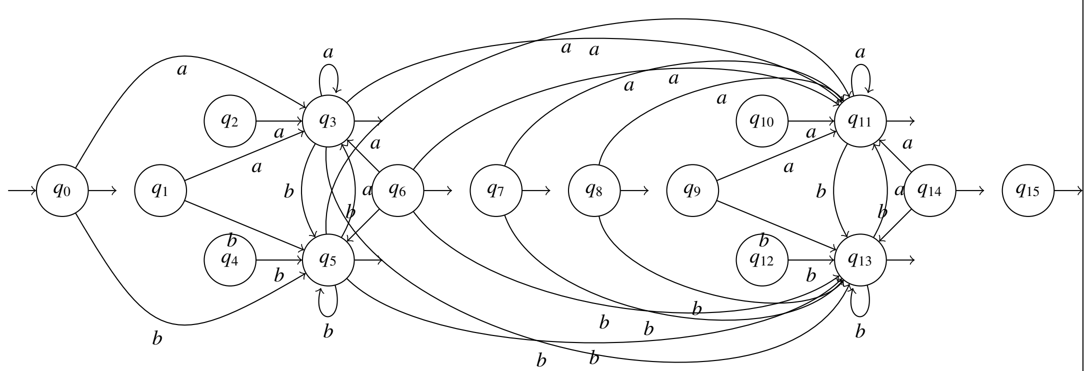
De esta gramática obtenemos el conjunto de copia:

Estado	**Conjunto de copia**
q_0	$q_0, q_1, q_2, q_4, q_7, q_8, q_{15}, q_9, q_{10}, q_{12}$
q_1	q_1, q_2, q_4
q_2	q_2
q_3	$q_3, q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_4	q_4
q_5	$q_5, q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_6	$q_6, q_1, q_2, q_4, q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_7	$q_7, q_8, q_9, q_{10}, q_{12}, q_{15}$
q_8	$q_8, q_9, q_{10}, q_{12}, q_{15}$
q_9	q_9, q_{10}, q_{12}
q_{10}	q_{10}
q_{11}	$q_{11}, q_{14}, q_9, q_{10}, q_{12}, q_{15}$
q_{12}	q_{12}
q_{13}	$q_{13}, q_{14}, q_9, q_{10}, q_{12}, q_{15}$
q_{14}	$q_{14}, q_9, q_{10}, q_{12}, q_{15}$
q_{15}	q_{15}

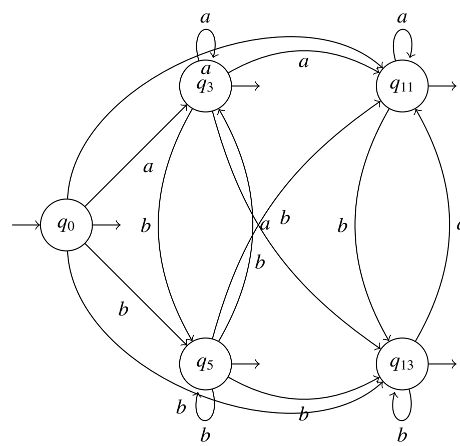
Del conjunto de copia redefinimos la gramática:

$$\begin{aligned}
q_0 &\rightarrow aq_3 \\
q_0 &\rightarrow bq_5 \\
q_0 &\rightarrow aq_{11} \\
q_0 &\rightarrow bq_{13} \\
q_0 &\rightarrow \epsilon \\
q_1 &\rightarrow aq_3 \\
q_1 &\rightarrow bq_5 \\
q_1 &\rightarrow \epsilon \\
q_2 &\rightarrow aq_3 \\
q_3 &\rightarrow aq_3 \\
q_3 &\rightarrow bq_5 \\
q_3 &\rightarrow aq_{11} \\
q_3 &\rightarrow bq_{13} \\
q_3 &\rightarrow \epsilon \\
q_4 &\rightarrow bq_5
\end{aligned}
\qquad
\begin{aligned}
q_5 &\rightarrow aq_3 \\
q_5 &\rightarrow bq_5 \\
q_5 &\rightarrow aq_{11} \\
q_5 &\rightarrow bq_{13} \\
q_5 &\rightarrow \epsilon \\
q_6 &\rightarrow aq_3 \\
q_6 &\rightarrow bq_5 \\
q_7 &\rightarrow aq_{11} \\
q_7 &\rightarrow bq_{13} \\
q_7 &\rightarrow \epsilon \\
q_8 &\rightarrow aq_{11} \\
q_8 &\rightarrow bq_{13} \\
q_8 &\rightarrow \epsilon \\
q_9 &\rightarrow aq_{11}
\end{aligned}
\qquad
\begin{aligned}
q_9 &\rightarrow bq_{13} \\
q_{10} &\rightarrow aq_{11} \\
q_{11} &\rightarrow aq_{11} \\
q_{11} &\rightarrow bq_{13} \\
q_{11} &\rightarrow \epsilon \\
q_{12} &\rightarrow bq_{13} \\
q_{13} &\rightarrow aq_{11} \\
q_{13} &\rightarrow bq_{13} \\
q_{13} &\rightarrow \epsilon \\
q_{14} &\rightarrow aq_{11} \\
q_{14} &\rightarrow bq_{13} \\
q_{14} &\rightarrow \epsilon \\
q_{15} &\rightarrow \epsilon
\end{aligned}$$

Ahora con la gramática redefinida, construimos el autómata que corresponde a esta gramática:



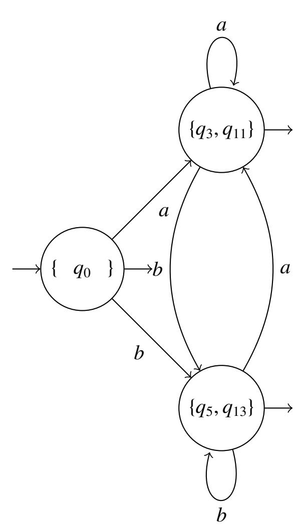
Pero este autómata tiene algunos estados que son no accesibles: q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{14}. Y también tiene un autómata en un estado no postaccesible: q_{15}. Por lo tanto hay que hacer limpieza del autómata, el autómata correspondiente es el siguiente:



3. A partir del anterior autómata no determinista, podemos pasar a generar el autómata determinista. Primero construimos el conjunto potencia de las transiciones:

$$
\begin{aligned}
\{q_0\} &\xrightarrow{a} \{q_3, q_{11}\} \\
\{q_0\} &\xrightarrow{b} \{q_5, q_{13}\} \\
\{q_3, q_{11}\} &\xrightarrow{a} \{q_3, q_{11}\} \\
\{q_3, q_{11}\} &\xrightarrow{b} \{q_5, q_{13}\} \\
\{q_5, q_{13}\} &\xrightarrow{a} \{q_3, q_{11}\} \\
\{q_5, q_{13}\} &\xrightarrow{b} \{q_5, q_{13}\}
\end{aligned}
$$

Con estas transiciones podemos generar el autómata determinista:



(15 %) 2. **Seguimiento de autómatas no deterministas**
Para el autómata finito no determinista (AFN_2) de la figura 1, indique 3 caminos etiquetados como *aabb*. ¿El AFN_2 acepta la cadena *aabb*?
Nota: En cada caso explicar su respuesta con una derivación.

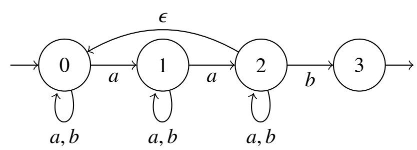


Figura 1: Un reconocedor de estado finito no-determinista (AFN_2)

Solución:

Con el anterior autómata se puede construir los siguientes caminos:

1	$0 \xrightarrow{a} 0 \xrightarrow{a} 0 \xrightarrow{b} 0 \xrightarrow{b} 0$
2	$0 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{b} 1 \xrightarrow{b} 1$
3	$0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{b} 2 \xrightarrow{b} 3$

Del tercer camino, al llegar al final del mismo se encuentra en un estado de aceptación por lo tanto la cadena es aceptada.

(35 %) 3. **Generación de autómatas utilizando el método de *Berry* y *Sethi***
Dada la siguiente expresión regular (re_2), generar un autómata deterministas utilizando el método de *Berry* y *Sethi* (BC).

$$re_2 = (a \mid b)^* b(ac)^+$$

Solución:

En primer lugar renumeramos la expresion re_2 y definimos la nueva:

$$re'_2 = (a_1 \mid b_2)^* b_3(a_4c_5)^+$$

Ahora, calculamos el conjunto de los digrafos de $Dig(re'_2)$:

$$\begin{aligned} Dig(re'_2) &= Dig((a_1 \mid b_2)^* b_3(a_4c_5)^+) \\ &= Dig((a_1 \mid b_2)^*) \cup Dig(b_3(a_4c_5)^+) \cup Fin((a_1 \mid b_2)^*) Ini(b_3(a_4c_5)^+) \end{aligned} \tag{1}$$

Tenemos que (1) se subdivide en tres subexpresiones: (2), (3), (4).

$$Dig((a_1 \mid b_2)^*) \tag{2}$$

$$Dig(b_3(a_4c_5)^+) \tag{3}$$

$$Fin((a_1 \mid b_2)^*)Ini(b_3(a_4c_5)^+) \tag{4}$$

Resolvamos (2):

$$\begin{aligned}
&= Dig((a_1 \mid b_2)^*) \\
&= Dig(a_1 \mid b_2) \cup Fin(a_1 \mid b_2)Ini(a_1 \mid b_2) \\
&= Dig(a_1) \cup Dig(b_2) \cup (Fin(a_1) \cup Fin(b_2))(Ini(a_1) \cup Ini(a_2)) \\
&= \emptyset \cup \emptyset \cup (\{a_1\} \cup \{b_2\})(\{a_1\} \cup \{b_2\}) \\
&= \{a_1, b_2\}\{a_1, b_2\} \\
&= \{a_1a_1, a_1b_2, b_2a_1, b_2b_2\}
\end{aligned}$$

La solución de (2) es (5).

$$\{a_1a_1, a_1b_2, b_2a_1, b_2, b_2\} \tag{5}$$

Resolvamos ahora (3):

$$\begin{aligned}
&= Dig(b_3(a_4c_5)^+) \\
&= Dig(b_3) \cup Dig((a_4c_5)^+) \cup Fin(b_3)Ini(a_4c_5)^+) \\
&= \emptyset \cup (Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5)) \cup \{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4)) \\
&= (Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5)) \cup \{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4))
\end{aligned}$$

Para resolver (3) la dividimos en dos subexpresiones (6) y (7):

$$(Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5)) \tag{6}$$

$$\{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4)) \tag{7}$$

Resolvamos (6):

$$\begin{aligned}
&= Dig(a_4c_5) \cup Fin(a_4c_5)Ini(a_4c_5) \\
&= (Dig(a_4) \cup Dig(c_5) \cup Fin(a_4)Ini(c_5)) \cup (Fin(c_5) \cup Fin(a_4)Null(c_5))(Ini(a_4) \cup Null(a_4)Ini(c_5)) \\
&= (\emptyset \cup \emptyset \cup \{a_4\}\{c_5\}) \cup (\{c_5\} \cup \{a_4\}\emptyset)(\{a_4\} \cup \emptyset\{c_5\}) \\
&= \{a_4c_5\} \cup (\{c_5\})(\{a_4\}) \\
&= \{a_4c_5\} \cup \{c_5a_4\} \\
&= \{a_4c_5, c_5a_4\}
\end{aligned}$$

La solución de (6) es (8)

$$\{a_4c_5, c_5a_4\} \tag{8}$$

Resolvamos (7):

$$\begin{aligned}
&= \{b_3\}(Ini(a_4) \cup Null(a_4)Ini(a_4)) \\
&= \{b_3\}(\{a_4\} \cup \emptyset\{a_4\}) \\
&= \{b_3\}(\{a_4\}) \\
&= \{b_3a_4\}
\end{aligned}$$

La solución de (7) es (9):

$$\{b_3a_4\} \tag{9}$$

Para resolver a (3), tenemos las soluciones de (6) es (8) y (7) es (9):

$$\begin{aligned} &= \{a_4c_5, c_5a_4\} \cup \{b_3a_4\} \\ &= \{a_4c_5, c_5a_4, b_3a_4\} \end{aligned}$$

La solución de (3) es (10):

$$\{a_4c_5, c_5a_4, b_3a_4\} \tag{10}$$

Nos falta solucionar (4):

$$\begin{aligned} &= Fin((a_1 \mid b_2)^*)Ini(b_3(a_4c_5)^+) \\ &= Fin(a_1 \mid b_2)(Ini(b_3) \cup Null(b_3)Ini((a_4c_5)^+)) \\ &= (Fin(a_1) \cup Fin(b_2))(\{b_3\} \cup \emptyset Ini(a_4c_5)) \\ &= (\{a_1\} \cup \{b_2\})(\{b_3\} \cup \emptyset) \\ &= (\{a_1, b_2\})\{b_3\} \\ &= \{a_1b_3, b_2b_3\} \end{aligned}$$

La solución de (4) es (11):

$$\{a_1b_3, b_2b_3\} \tag{11}$$

De (5), (10) y (11) podemos resolver a (1):

$$\begin{aligned} &= \{a_1a_1, a_1b_2, b_2a_1, b_2b_2\} \cup \{a_4c_5, c_5a_4, b_3a_4\} \cup \{a_1b_3, b_2b_3\} \\ &= \{a_1a_1, a_1b_2, b_2a_1, b_2b_2, a_1b_3, b_2b_3, a_4c_5, c_5a_4, b_3a_4\} \end{aligned}$$

Entonces,

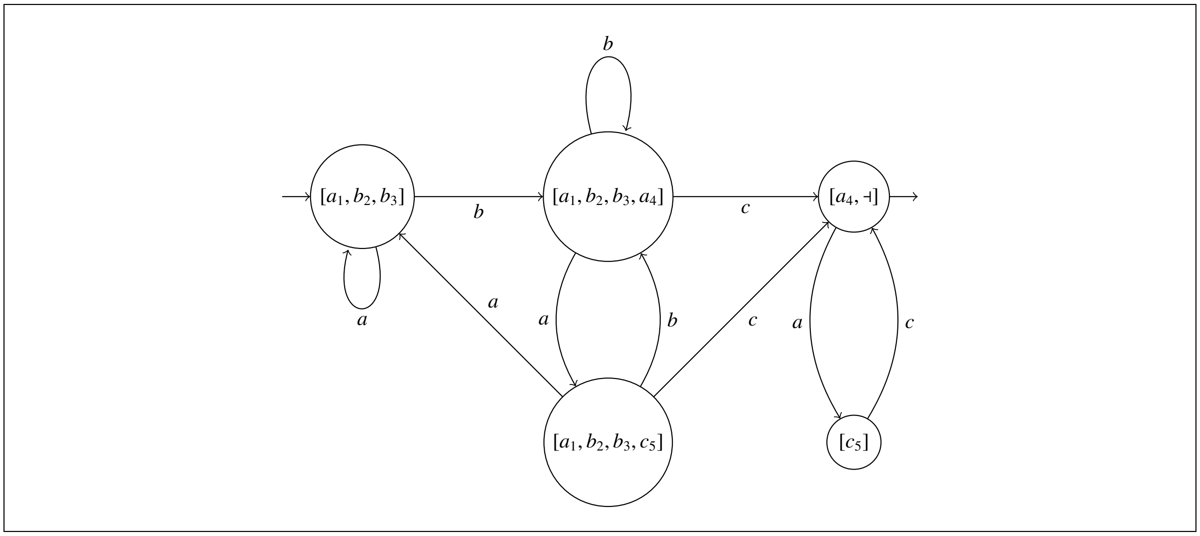
$$Dig(re'_2) = \{a_1a_1, a_1b_2, b_2a_1, b_2b_2, a_1b_3, b_2b_3, a_4c_5, c_5a_4, b_3a_4\} \tag{12}$$

De (12) construyamos el conjunto Fol:

	siguientes
a_1	a_1, b_2, b_3
b_2	a_1, b_2, b_3
b_3	a_4
a_4	c_5
c_5	$a_4, \dashv$

(13)

El autómata deterministico es mostrado a continuación:


b
[a1, b2, b3]
b
[a1, b2, b3, a4]
c
[a4, ⊣]
a
a
a
b
c
a
c
[a1, b2, b3, c5]
[c5]

(15 %) 4. **Programación**
El siguiente programa en Java es la implementación de una autómata determinista obtenido en el punto 3 utilizando el método *Berry* y *Sethi*. Esta implementación cuenta con varios errores, responda la siguientes preguntas:

1. Identifique los errores que hay en la implementación del autómata.
2. Re-escriba las secciones de código donde los errores ocurren, para que el autómata se ejecute según su especificación obtenida en el punto 3.

```
/**
 * Codigo utilizado como parte del segundo parcial
 */
public class AutomataBerrySethi {

    /*
     * Constantes q0 hasta q4 representa los estados,
     * el estado inicial es q0, el de aceptación es q3,
     * el estado qe, es el estado de error.
     * La variable privada int state contiene el estado actual
     * de la máquina
     */
    private static final int q0 = 0;
    private static final int q1 = 1;
    private static final int q2 = 2;
    private static final int q3 = 3;
    private static final int q4 = 4;
    private static final int qe = 5;

    private int state;

    /**
     * La función de transición
     * @param s codigo de estado (un int)
     * @param c carácter para hacer una transición
     * @return el siguiente estado del código
     */
    static private int[][] delta
       = {{q0,q1},{q2,q1},{q0,q1},{q4,qe},{qe,qe},{qe,qe}};
    /**
     * Re-establece el actual estado al estado de inicio
     */
    public void reset() {
       state = q0;
    }

    /**
     * Realiza una transición en cada carácter en la cadena
     * dada.
     * @param en la cadena a utilizar
     */
    public void process(String in) {

       for (int i = 0; i < in.length(); i++) {
          char c = in.charAt(i);
          try {
             state = delta[state][c];
          }
          catch (ArrayIndexOutOfBoundsException ex) {
             state = qe;
          }
       }
    }

    /**
     * Examina si el DFA aceptó la cadena.
     * @return true si el estado final fue de aceptación
     */
    public boolean accepted() {
       return state == q3;
    }
}
```

Solución:

1. Hay dos conjuntos de errores:

   a) en la definición de la función de transición, si la observamos encontramos que están definidos únicamente por cada estado, dos posibles transiciones,

   ```
   static private int[][] delta
      = {{q0,q1},{q2,q1},{q0,q1},{q4,qe},{qe,qe},{qe,qe}};
   ```

   pero el alfabeto de la gramática consta de tres elementos $\Sigma = \{a, b, c\}$, por lo tanto, nos falta adicionar a cada fila de la tabla un elemento para el tercer carácter.

   b) el segundo error está en la instrucción que selecciona el estado.

   ```
   state = delta[state][c];
   ```

El valor de `c` selecciona con base el carácter y el estado actual `state` el nuevo estado siguiente, el problema, es que el valor de `c` es un carácter y requiere de un número para hacerlo.

2. Soluciones:

   a) El siguiente código corrige el primer error:

   ```
   static private int[][] delta
      = {{q0,q1,qe},{q2,q1,q3},{q0,q1,q3},{q4,qe,qe}
        ,{qe,qe,q3},{qe,qe,qe}};
   ```

   b) El siguiente código corrige el segundo tipo de error:

   ```
   state = delta[state][c-'a'];
   ```

Pregunta	Puntos	Obtenido
1	35	
2	15	
3	35	
4	15	
Total:	100	

ST0270 Lenguajes formales y compiladores | **Universidad EAFIT**
Supletorio segundo parcial | **DIS**
Juan Francisco Cardona M | **Abril 26 de 2010**

Nombre: ____

Código: ____

(40 %) 1. **Construcción de autómatas a partir de una expresión regular**

1. Utilice el método estructural de Thompson para convertir la expresión regular

$$re_1 = (ab \mid cd)^*$$

en un AFN (AFN_ϵ_1) con movimientos espontáneos.

2. Elimine en el autómata (AFN_ϵ_1) los movimientos espontáneos y conviértalo en un autómata no determinista sin movimientos espontáneos AFN (AFN_1).

3. Convierta el AFN (AFN_1) en un AFD (AFD_1) utilizando el método de los conjuntos de potencia.

(40 %) 2. **Generación de autómatas utilizando el método de *Berry* y *Sethi***
Dada la siguiente expresión regular (re_2), generar un autómata deterministas utilizando el método de *Berry* y *Sethi* (BC).

$$re_2 = (ab \mid cd)^*$$

(20 %) 3. **Transformación de un autómata en una expresión regular** Tome el resultado del autómata determinista del primer punto o segundo punto y conviertala en una expresión regular.

Pregunta	Puntos	Obtenido
1	40	
2	40	
3	20	
Total:	100	

ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Tercer Parcial — **DIS**
Juan Francisco Cardona M — **Mayo 20 de 2010**

Nombre: ______________________________
Código: ______________________________

(35 %) 1. **Condicción** $LL(1)$
Probar que la siguiente gramática G_1 cumple con la condición $LL(1)$.

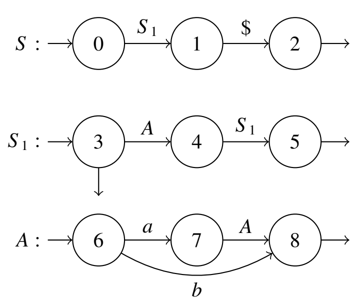
$$G_1 = (\{S, S_1, A\}, \{\$, a, b\}, R, S)$$

donde R define las producciones de la gramática G_1:

$$\begin{array}{rcl} S & \rightarrow & S_1\$ \\ S_1 & \rightarrow & AS_1 \mid \epsilon \\ A & \rightarrow & aA \mid b \end{array}$$

Solución:

En primer lugar construimos la red de máquinas:



Calculamos los conjuntos guías de las bifurcaciones,

$$\begin{aligned} &Gui(3 \xrightarrow{A} 4) = Ini(L(6)L(4)) = \{a, b\} \\ &Gui(3 \rightarrow) = Fol(S_1) = \{\$\} \\ &Gui(6 \xrightarrow{a} 7) = \{a\} \\ &Gui(6 \xrightarrow{b} 8) = \{b\} \end{aligned}$$

No hay elementos comunes entre las bifurcaciones que tiene un mismo origen. Por lo tanto la gramática es $LL(1)$.

(35 %) 2. **Transformación de gramáticas**
La gramática G_2 no cumple con la condición $LL(1)$.

$$G_2 = (\{S\}, \{a, b\}, \{S \rightarrow aaSb \mid ab \mid bb\}, S)$$

Transforme la gramática G_2 utilizando los mecanismos de transformación vistos en clase en una gramática G_2' que cumpla con la condición $LL(1)$. Demuestre que la gramática G_2' cumple con la condición $LL(1)$.

Solución:

Factorizamos la gramática G_2 :

$$S \rightarrow a(aSb \mid b) \mid bb$$

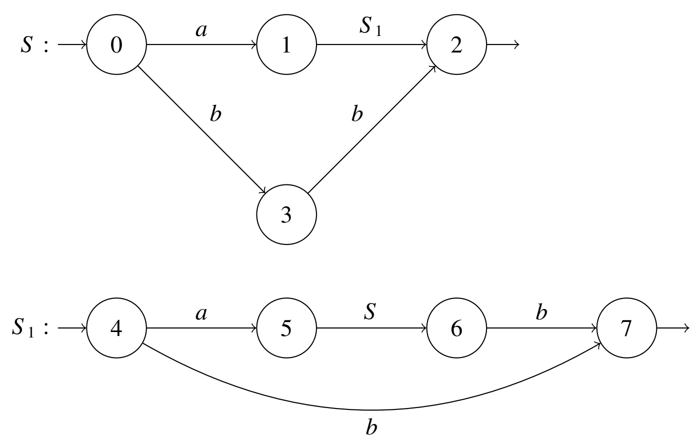
Encontramos que a es común a dos de las tres alternativas originales, la parte no común de las dos alternativas factorizadas creamos un nuevo no-terminal que la contendrá S_1 y nos produce las siguientes reglas:

$$S \rightarrow aS_1 \mid bb$$
$$S_1 \rightarrow aSb \mid b$$

De estas reglas no es posible factorizar más, por lo tanto definimos la nueva gramática G_2':

$$G_2' = (\{S, S_1\}, \{a, b\}, \{S \rightarrow aS_1 \mid bb, S_1 \rightarrow aSb \mid b\}, S)$$

Ahora probemos que G_2' cumple con la condición $LL(1)$. En primer lugar, creamos la red de máquinas generada por esta gramática:



Hay dos bifurcaciones importantes, calculamos la guía:

$$Gui(0 \xrightarrow{a} 1) = \{a\}$$
$$Gui(0 \xrightarrow{b} 3) = \{b\}$$
$$Gui(4 \xrightarrow{a} 5) = \{a\}$$
$$Gui(4 \xrightarrow{b} 7) = \{b\}$$

Como se puede observar las guías con nodos de origen común, son separadas, es decir, no tiene elementos comunes. Por lo tanto, la gramática G_2' cumple con la condición $LL(1)$.

(30 %) 3. **Reconocimiento de cadenas** Utilizando la gramática G_1 definida en 1 y el reconocedor $LL(1)$, probar que las siguientes cadenas son válidas, es decir, cuál es reconocida por el reconocedor $LL(1)$ y cuál no.

1. $bb\$$
2. $ba\$$

Para cada verificación llenar un tabla con los siguientes campos:

Pila	**cadena**	**Comentarios**

Solución:

En primer lugar debemos calcular todos los conjuntos guías de la red de automátas que se encuentran en la red del punto 1:

$$
\begin{aligned}
Gui(0 \xrightarrow{S_1} 1) &= Ini(L(3)L(1)) = \{a, b, \$\} \\
Gui(1 \xrightarrow{\$} 2) &= \{\$\} \\
Gui(2 \rightarrow) &= Fol(S) = \{\dashv\} \\
Gui(3 \xrightarrow{A} 4) &= Ini(L(3)L(4)) = \{a, b\} \\
Gui(3 \rightarrow) &= Fol(S_1) = \{\$\} \\
Gui(4 \xrightarrow{S_1} 5) &= Ini(L(3)L(5)) \cup Fol(S_1) = \{a, b, \$\} \\
Gui(5 \rightarrow) &= Fol(S_1) = \{\$\} \\
Gui(6 \xrightarrow{a} 7) &= \{a\} \\
Gui(6 \xrightarrow{b} 8) &= \{b\} \\
Gui(7 \xrightarrow{A} 8) &= Ini(L(6)L(8)) = \{a, b\} \\
Gui(8 \rightarrow) &= Fol(A) = \{a, b, \$\}
\end{aligned}
$$

1. Para la cadena *bb*$:

Pila	**cadena**	**Comentario**
0	$bb\$ \dashv$	Movimiento de invocación
03	$bb\$ \dashv$	Movimiento de invocación
036	$bb\$ \dashv$	Movimiento de examén
038	$b\$ \dashv$	Movimiento de retorno
04	$b\$ \dashv$	Movimiento de invocación
043	$b\$ \dashv$	Movimiento de invocación
0436	$b\$ \dashv$	Movimiento de examén
0438	$\$ \dashv$	Movimiento de retorno
044	$\$ \dashv$	Movimiento de invocación
0443	$\$ \dashv$	Movimiento de retorno
045	$\$ \dashv$	Movimiento de retorno
05	$\$ \dashv$	Movimiento de retorno
1	$\$ \dashv$	Movimiento de examén
2	$\dashv$	Aceptación de la cadena

Por lo tanto la cadena *bb*$ es aceptada por el reconocedor, esta cadena es válida en G_1.

2. Para la cadena *ba*$:

Pila	**cadena**	**Comentario**
0	$ba\$ \dashv$	Movimiento de invocación
03	$ba\$ \dashv$	Movimiento de invocación
036	$ba\$ \dashv$	Movimiento de examén
038	$a\$ \dashv$	Movimiento de retorno
04	$a\$ \dashv$	Movimiento de invocación
043	$a\$ \dashv$	Movimiento de invocación
0436	$a\$ \dashv$	Movimiento de examén
0437	$\$ \dashv$	falló, no hay más movimientos

Por lo tanto la cadena *ba*$ no es una cadena válida de G_1.

Pregunta	Puntos	Obtenido
1	35	
2	35	
3	30	
Total:	100	

ST0270 Lenguajes formales y compiladores | **Universidad EAFIT**
Tercer Parcial (Supletorio) | **DIS**
Juan Francisco Cardona M | **Mayo 28 de 2010**

Nombre: ____________________
Código: ____________________

(35 %) 1. **Condicción** $LL(1)$
Probar que la siguiente gramática G_1 cumple con la condición $LL(1)$.

$$G_1 = (\{S, A\}, \{a, b\}, R, S)$$

donde R define las producciones de la gramática G_1:

$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow aA \mid bA \mid \epsilon \end{aligned}$$

(35 %) 2. **Transformación de gramáticas**
La gramática G_2 no cumple con la condición $LL(1)$.

$$G_2 = (\{S\}, \{a, b\}, R, S)$$

donde R es:

$$\begin{aligned} S &\rightarrow SA \mid \epsilon \\ A &\rightarrow Aa \mid b \end{aligned}$$

Transforme la gramática G_2 utilizando los mecanismos de transformación vistos en clase en una gramática G_2' que cumpla con la condición $LL(1)$. Demuestre que la gramática G_2' cumple con la condición $LL(1)$.

(30 %) 3. **Reconocimiento de cadenas** Utilizando la gramática G_1 definida en 1 y el reconocedor $LL(1)$, probar que las siguientes cadenas son válidas, es decir, cuál es reconocida por el reconocedor $LL(1)$ y cuál no.

1. $ab\$$
2. $ba\$$

Para cada verificación llenar un tabla con los siguientes campos:

Pila	**cadena**	**Comentarios**

Pregunta	Puntos	Obtenido
1	35	
2	35	
3	30	
Total:	100	

ST0270 Lenguajes formales y compiladores | **Universidad EAFIT**
Primer Parcial | **DIS**
Juan Francisco Cardona M | **Agosto 19 de 2010**

Nombre: ______________________________

Código: ______________________________

(15 %) 1. **Lenguajes y expresiones regulares**
Determine el lenguaje que representa cada una de las siguientes expresiones regulares:

1. $r_1 = (z \mid y)^* x$.
2. $r_2 = xx^* \mid yy^*$.

> **Solución:**
>
> 1. $L(r_1) = \{ux \ \mid \ u \in \{z, y\}^*\}$
> 2. $L(r_2) = \{x^n \mid n \geq 1\} \cup \{y^m \mid m \geq 1\}$

(20 %) 2. **Lenguajes de las gramáticas independientes de contexto**
Encuentre el lenguaje para cada una de las gramáticas independiente de contexto:

1. $G_1 = (\{S\}, \{x, y, z\}, \{S \to xSz, S \to y\}, S)$.
2. $G_2 = (\{S\}, \{x, y\}, \{S \to xSy, S \to y\}, S)$.

> **Solución:**
>
> 1. $L(G_1) = \{x^n y z^n \ \mid \ n \geq 0\}$
> 2. $L(G_2) = \{x^n y^{n+1} \ \mid n \geq 0\}$

(35 %) 3. **Transformación de gramáticas - recursividad por la izquierda**
La siguiente gramática independiente de contexto tiene problemas de recursividad inmediata e indirecta, transforme la gramática en una gramática sin recursividad inmediata e indirecta por la izquierda.

$$G_3 = (\{A_1, A_2\}, \{a, b, c, d, e\}, \{A_1 \to A_1 a \ \mid \ A_2 b \ \mid \ c, A_2 \to A_1 d \mid e\}, A_1)$$

Solución:

Primera iteración. No se aplica la primera iteración, pero se debe aplicar una transformación de recursividad inmediata. Reordenamos las producciones de A_1

$$\begin{array}{lll} A_1 & \rightarrow & A_1 a \\ A_1 & \rightarrow & A_2 b \mid c \end{array}$$

Aplicamos la transformación directa:

$$\begin{array}{lll} A_1 & \rightarrow & A_2 b A_1' \mid cA_1' \mid A_2 b \mid c \\ A_1' & \rightarrow & aA_1' \mid a \end{array}$$

Pasamos a la siguiente iteración y encontramos que se cumple $A_i \rightarrow A_j \alpha$ con $i > j$:

$$A_2 \rightarrow A_1 d$$

Por lo tanto expandimos y encontramos lo siguiente:

$$A_2 \rightarrow A_2 b A_1' d \mid cA_1' d \mid A_2 bd \mid cd$$

Encontramos recursividad inmediata y aplicamos de nuevo su correspodiente transformación. Ordenamos:

$$\begin{array}{lll} A_2 & \rightarrow & A_2 b A_1' d \mid A_2 bd \\ A_2 & \rightarrow & cA_1' d \mid cd \mid e \end{array}$$

Aplicamos la transformación directa y obtenemos:

$$\begin{array}{lll} A_2 & \rightarrow & cA_1' dA_2' \mid cdA_2' \mid eA_2' \mid cA_1' d \mid cd \mid e \\ A_2' & \rightarrow & bA_1' dA_2' \mid bdA_2' \mid bA_1' d \mid bd \end{array}$$

(10 %) 4. **Análisis**

En el algoritmo de limpieza de la gramática, este tiene dos fases, en la primera parte se calcula el conjunto DEF, el conjunto de no-terminal bien definidos.
El conjunto DEF es inicializado con los no-terminales que cumplen con una condición. Se ha cambiado dicha inicialización por la siguiente:

$$DEF \ := \ \{A \mid (A \rightarrow u) \in P, \text{ con } u \in (\Sigma \cup V)^*\}$$

El resto del algoritmo permanece igual. Responder:
Preguntas

1. ¿Qué efecto tiene este cambio en el cálculo final del conjunto DEF?
2. ¿Cuáles son los terminales que pertenecen al conjunto $V \setminus DEF$?

Solución:

1. El conjunto DEF es inicializado con todos los no-terminales que se encuentran al lado derecho de la producción, incluyendo aquellos que generan secuencia de terminales y también los que no generan directamente. Esto difiere de la inicialización que solamente toma los que genera directamente terminales.
   Al calcular nuevamente el conjunto DEF este no cambia, puesto que todos cumple la condición. Así, que al final van a quedar tanto los que derivan a una secuencia de terminales como los que no.

2. $$\begin{aligned} DEF &:= \{A \mid (A \to u) \in P\} \\ V \setminus DEF &:= \{B \mid \forall A.(A \to u) \in P \ \wedge B \in V \ \wedge \ A \neq B\} \end{aligned}$$

Pregunta	Puntos	Obtenido
1	15	
2	20	
3	35	
4	10	
Total:	80	

Nombre: ____________________
Código: ____________________

(40 %) 1. **Construcción de autómatas a partir de una expresión regular**

a. Utilice el método estructural de Thompson para convertir la expresión regular

$$re_1 = (a \mid (a \mid b))$$

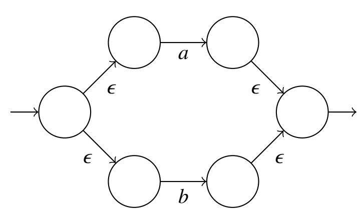
en un AFN (AFN_ϵ_1) con movimientos espontáneos.

b. Elimine en el autómata (AFN_ϵ_1) los movimientos espontáneos y conviértalo en un autómata no determinista sin movimientos espontáneos AFN (AFN_1).

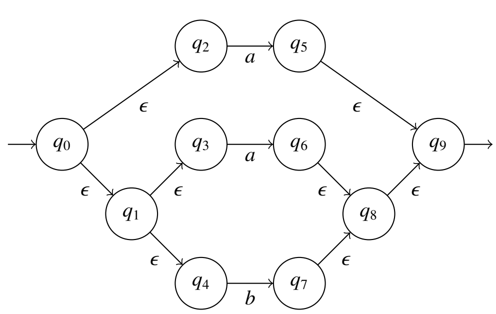
c. Convierta el AFN (AFN_1) en un AFD (AFD_1) utilizando el método de los conjuntos de potencia.

Solución:

a. En primer lugar utilizando el método de Thompson definimos algunos sub-automátas. Observamos primero que la expresión regular se compone de dos sub-expresiones similares, $(a \mid b)$. Podemos construir el autómata de una de ellas y luego las unimos. Para la sub-expresión $(a \mid b)$ obtenemos el siguiente autómata con movimientos expontáneos.



Unimos la sub-expresión y su automáta; con la sub-expresión a y obtenemos el automáta no deterministico con movimientos expontáneos.



b. Del anterior automáta definimos su gramática lineal por la derecha.

$$
\begin{array}{lll}
q_0 \rightarrow q_1 & q_2 \rightarrow aq_5 & q_6 \rightarrow q_8 \\
q_0 \rightarrow q_2 & q_3 \rightarrow aq_6 & q_7 \rightarrow q_8 \\
q_1 \rightarrow q_3 & q_4 \rightarrow bq_7 & q_8 \rightarrow q_9 \\
q_1 \rightarrow q_4 & q_5 \rightarrow q_9 & q_9 \rightarrow \epsilon
\end{array}
$$

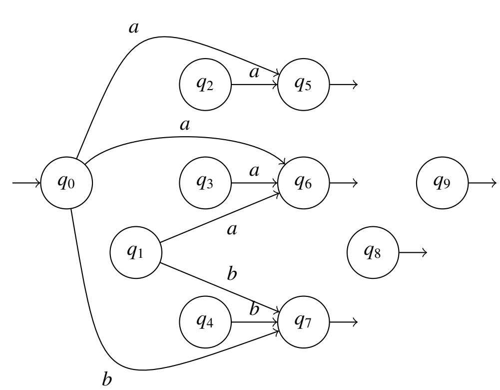
La gramática anterior tiene producciones de copia, obtenemos su correspondiente conjunto de copia:

Estado	**Conjunto de copia**
q_0	q_0, q_1, q_2
q_1	q_1, q_3, q_4
q_2	q_2
q_3	q_3
q_4	q_4
q_5	q_5, q_9
q_6	q_6, q_8, q_9
q_7	q_7, q_8, q_9
q_8	q_8, q_9
q_9	q_9

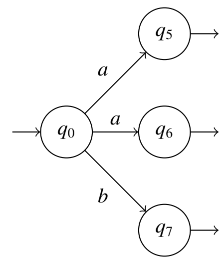
Con el conjunto de copia redefinimos la gramática anterior:

$$
\begin{array}{lll}
q_0 \rightarrow aq_5 & q_1 \rightarrow bq_7 & q_5 \rightarrow \epsilon \\
q_0 \rightarrow aq_6 & q_2 \rightarrow aq_5 & q_6 \rightarrow \epsilon \\
q_0 \rightarrow bq_7 & q_3 \rightarrow aq_6 & q_7 \rightarrow \epsilon \\
q_1 \rightarrow aq_6 & q_4 \rightarrow bq_7 & q_8 \rightarrow \epsilon \\
 & & q_9 \rightarrow \epsilon
\end{array}
$$

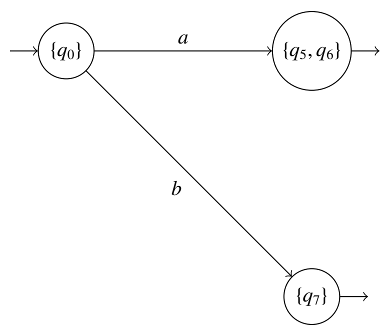
Lo que nos produce el siguiente automáta:



Eliminando los estados inalcanzables e inaccesibles se obtiene el siguiente automáta:



c. El anterior automáta es no determinista, se aplica el conjunto de potencias para eliminar el no determinismo. Y se obtiene el siguiente automáta:



(30 %) 2. **Generación de autómatas utilizando el método de *Berry* y *Sethi***
Dada la siguiente expresión regular

$$re_2 = (ab \mid ba)^*$$

y utilizando el método de *Berry* y *Sethi* (BC), calcular:

a. los conjuntos: $Ini(re_2)$, $Fin(re_2)$, $Null(re_2)$, $Dig(re_2)$, $Follow(re_2)$.
b. el automáta determinista correspondiente a la expresión regular.

Solución:

a. En primer lugr se numera la expresion re_2:

$$re_2' = (a_1 b_2 \mid b_3 a_4)^*$$

El conjunto $Null((a_1 b_2 \mid b_3 a_4)^*) = \epsilon$

El conjunto $Ini((a_1b_2 \mid b_3a_4)^*)$:

$$\begin{aligned}
Ini((a_1b_2 \mid b_3a_4)^*) &= Ini(a_1b_2 \mid b_3a_4)\\
&= Init(a_1b_2) \cup Init(b_3a_4)\\
&= Init(a_1) \cup Null(a_1)Init(b_2) \cup Ini(b_3) \cup Null(b_3)Ini(a_4)\\
&= \{a_1\} \cup \emptyset Init(b_2) \cup \{b_3\} \cup \emptyset Ini(a_4)\\
&= \{a_1\} \cup \{b_3\}\\
&= \{a_1, b_3\}
\end{aligned}$$

El conjunto $Fin((a_1b_2 \mid b_3a_4)^*)$:

$$\begin{aligned}
Fin((a_1b_2 \mid b_3a_4)^*) &= Fin(a_1b_2 \mid b_3a_4)\\
&= Fin(a_1b_2) \cup Fin(b_3a_4)\\
&= Fin(b_2) \cup Null(b_2)Fin(a_1) \cup Fin(a_4) \cup Null(a_4)Fin(b_3)\\
&= \{b_2\} \cup \emptyset Fin(a_1) \cup \{a_4\} \cup \emptyset Fin(b_3)\\
&= \{b_2\} \cup \{a_4\}\\
&= \{b_2, a_4\}
\end{aligned}$$

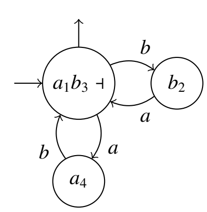
El conjunto $Dig((a_1b_2 \mid b_3a_4)^*)$

$$\begin{aligned}
Dig((a_1b_2 \mid b_3a_4)^*) &= Dig(a_1b_2 \mid b_3a_4) \cup Fin(a_1b_2 \mid b_3a_4)Ini(a_1b_2 \mid b_3a_4)\\
&= Dig(a_1b_2) \cup Dig(b_3a_4) \cup \{b_2, a_4\}\{a_1, b_3\}\\
&= Dig(a_1) \cup Dig(b_2) \cup Fin(a_1)Ini(b_2) \cup Dig(b_3) \cup Dig(a_4)\\
&\quad \cup \{b_2, a_4\}\{a_1, b_3\}\\
&= \emptyset \cup \emptyset \cup \{a_1\}\{b_2\} \cup \emptyset \cup \emptyset \cup \{b_2a_1, b_2b_3, a_4a_1, a_4b_3\}\\
&= \{a_1b_2, b_2a_1, b_2b_3, a_4a_1, a_4b_3\}
\end{aligned}$$

Calculamos el conjunto $Ini(re'_2 \dashv) = \{a_1b_3 \dashv\}$. Luego calculamos el conjunto de $Follow(c_i)$ para cada carácter c_i:

carácter	*Follow*
a_1	b_2
b_2	$a_1b_3 \dashv$
b_3	a_4
a_4	$a_1b_3 \dashv$

b. El correspondiente automáta es:



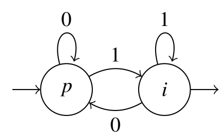
(30 %) 3. **Generación de automátas y su expresión correspondiente**

a. Construir un automáta que reconozca de una cadena binaria ($\Sigma = \{0, 1\}$) todas aquellas cadenas con valores impares.

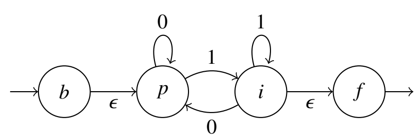
b. Una vez construido el automáta obtener su expresión regular por medio del método BMC.

Solución:

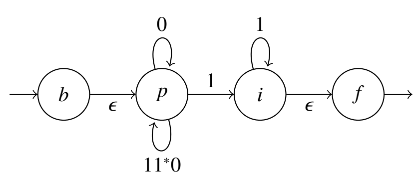
a. Definimos un automáta que detecten los valores impares:



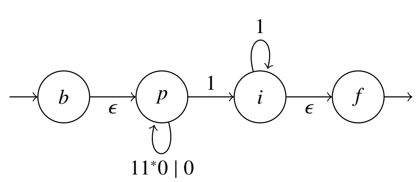
b. Utilizando el método BMC calculamos la expresión regular. Primero añadimos nuevos estados de inicio y de fin:



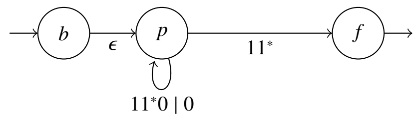
Calculamos una primera trayectoria entre $p \to i \to p$, observe que hay otra trayectoria entre $p \to i \to f$.



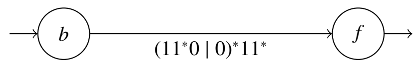
Renombramos los arcos que entran y salen del mismo p:



Tomamos ahora la trayectoria entre $p \to i \to f$:



Finalmente analizamos las trayectorias entre $b \to p \to f$:



La expresión regular para nuestro automáta es: $(11^*0 \mid 0)^*11^*$.

Pregunta	Puntos	Valor obtenido
1	40	
2	30	
3	30	
Total:	100	

ST0270 Lenguajes formales y compiladores | **Universidad EAFIT**
Segundo parcial | **DIS**
Juan Francisco Cardona M | **Septiembre 23 de 2010**

Nombre: ______________________________
Código: ______________________________

(40 %) 1. **Construcción de autómatas a partir de una expresión regular**

a. Utilice el método estructural de Thompson para convertir la expresión regular

$$re_1 = (a \mid (a \mid b))$$

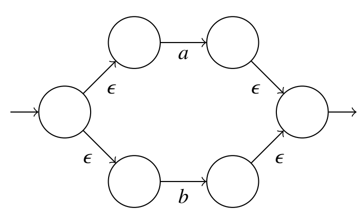
en un AFN (AFN_ϵ_1) con movimientos espontáneos.

b. Elimine en el autómata (AFN_ϵ_1) los movimientos espontáneos y conviértalo en un autómata no determinista sin movimientos espontáneos AFN (AFN_1).

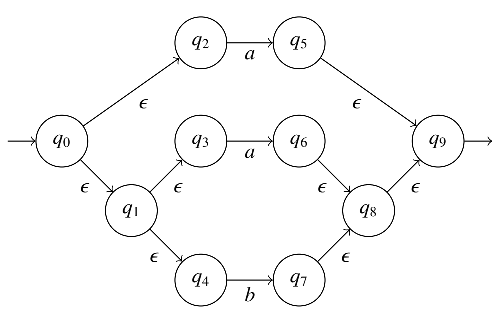
c. Convierta el AFN (AFN_1) en un AFD (AFD_1) utilizando el método de los conjuntos de potencia.

Solución:

a. En primer lugar utilizando el método de Thompson definimos algunos sub-automátas. Observamos primero que la expresión regular se compone de dos sub-expresiones similares, $(a \mid b)$. Podemos construir el autómata de una de ellas y luego las unimos. Para la sub-expresión $(a \mid b)$ obtenemos el siguiente autómata con movimientos expontáneos.



Unimos la sub-expresión y su automáta; con la sub-expresión a y obtenemos el automáta no deterministico con movimientos expontáneos.



b. Del anterior automáta definimos su gramática lineal por la derecha.

$$\begin{array}{lll} q_0 \rightarrow q_1 & q_2 \rightarrow aq_5 & q_6 \rightarrow q_8 \\ q_0 \rightarrow q_2 & q_3 \rightarrow aq_6 & q_7 \rightarrow q_8 \\ q_1 \rightarrow q_3 & q_4 \rightarrow bq_7 & q_8 \rightarrow q_9 \\ q_1 \rightarrow q_4 & q_5 \rightarrow q_9 & q_9 \rightarrow \epsilon \end{array}$$

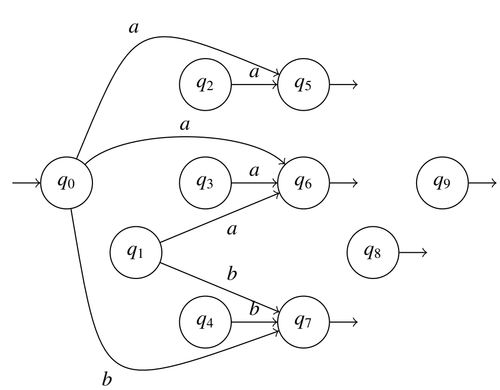
La gramática anterior tiene producciones de copia, obtenemos su correspondiente conjunto de copia:

Estado	**Conjunto de copia**
q_0	q_0, q_1, q_2
q_1	q_1, q_3, q_4
q_2	q_2
q_3	q_3
q_4	q_4
q_5	q_5, q_9
q_6	q_6, q_8, q_9
q_7	q_7, q_8, q_9
q_8	q_8, q_9
q_9	q_9

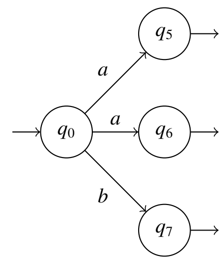
Con el conjunto de copia redefinimos la gramática anterior:

$$\begin{array}{lll} q_0 \rightarrow aq_5 & q_1 \rightarrow bq_7 & q_5 \rightarrow \epsilon \\ q_0 \rightarrow aq_6 & q_2 \rightarrow aq_5 & q_6 \rightarrow \epsilon \\ q_0 \rightarrow bq_7 & q_3 \rightarrow aq_6 & q_7 \rightarrow \epsilon \\ q_1 \rightarrow aq_6 & q_4 \rightarrow bq_7 & q_8 \rightarrow \epsilon \\ & & q_9 \rightarrow \epsilon \end{array}$$

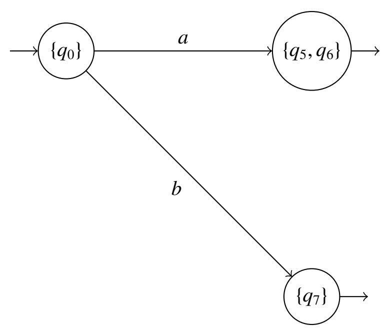
Lo que nos produce el siguiente automáta:



Eliminando los estados inalcanzables e inaccesibles se obtiene el siguiente automáta:



c. El anterior automáta es no determinista, se aplica el conjunto de potencias para eliminar el no determinismo. Y se obtiene el siguiente automáta:



(30 %) 2. **Generación de autómatas utilizando el método de *Berry* y *Sethi***
Dada la siguiente expresión regular

$$re_2 = (ab \mid ba)^*$$

y utilizando el método de *Berry* y *Sethi* (BC), calcular:

a. los conjuntos: $Ini(re_2)$, $Fin(re_2)$, $Null(re_2)$, $Dig(re_2)$, $Follow(re_2)$.
b. el automáta determinista correspondiente a la expresión regular.

Solución:

a. En primer lugr se numera la expresion re_2:

$$re_2' = (a_1b_2 \mid b_3a_4)^*$$

El conjunto $Null((a_1b_2 \mid b_3a_4)^*) = \epsilon$

El conjunto $Ini((a_1b_2 \mid b_3a_4)^*)$:

$$\begin{aligned}
Ini((a_1b_2 \mid b_3a_4)^*) &= Ini(a_1b_2 \mid b_3a_4)\\
&= Init(a_1b_2) \cup Init(b_3a_4)\\
&= Init(a_1) \cup Null(a_1)Init(b_2) \cup Ini(b_3) \cup Null(b_3)Ini(a_4)\\
&= \{a_1\} \cup \emptyset Init(b_2) \cup \{b_3\} \cup \emptyset Ini(a_4)\\
&= \{a_1\} \cup \{b_3\}\\
&= \{a_1, b_3\}
\end{aligned}$$

El conjunto $Fin((a_1b_2 \mid b_3a_4)^*)$:

$$\begin{aligned}
Fin((a_1b_2 \mid b_3a_4)^*) &= Fin(a_1b_2 \mid b_3a_4)\\
&= Fin(a_1b_2) \cup Fin(b_3a_4)\\
&= Fin(b_2) \cup Null(b_2)Fin(a_1) \cup Fin(a_4) \cup Null(a_4)Fin(b_3)\\
&= \{b_2\} \cup \emptyset Fin(a_1) \cup \{a_4\} \cup \emptyset Fin(b_3)\\
&= \{b_2\} \cup \{a_4\}\\
&= \{b_2, a_4\}
\end{aligned}$$

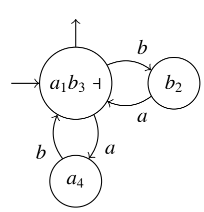
El conjunto $Dig((a_1b_2 \mid b_3a_4)^*)$

$$\begin{aligned}
Dig((a_1b_2 \mid b_3a_4)^*) &= Dig(a_1b_2 \mid b_3a_4) \cup Fin(a_1b_2 \mid b_3a_4)Ini(a_1b_2 \mid b_3a_4)\\
&= Dig(a_1b_2) \cup Dig(b_3a_4) \cup \{b_2, a_4\}\{a_1, b_3\}\\
&= Dig(a_1) \cup Dig(b_2) \cup Fin(a_1)Ini(b_2) \cup Dig(b_3) \cup Dig(a_4)\\
&\quad \cup \{b_2, a_4\}\{a_1, b_3\}\\
&= \emptyset \cup \emptyset \cup \{a_1\}\{b_2\} \cup \emptyset \cup \emptyset \cup \{b_2a_1, b_2b_3, a_4a_1, a_4b_3\}\\
&= \{a_1b_2, b_2a_1, b_2b_3, a_4a_1, a_4b_3\}
\end{aligned}$$

Calculamos el conjunto $Ini(re'_2 \dashv) = \{a_1b_3 \dashv\}$. Luego calculamos el conjunto de $Follow(c_i)$ para cada carácter c_i:

carácter	*Follow*
a_1	b_2
b_2	$a_1b_3 \dashv$
b_3	a_4
a_4	$a_1b_3 \dashv$

b. El correspondiente automáta es:



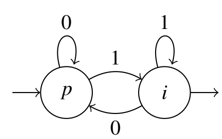
(30 %) 3. **Generación de automátas y su expresión correspondiente**

a. Construir un automáta que reconozca de una cadena binaria ($\Sigma = \{0, 1\}$) todas aquellas cadenas con valores impares.

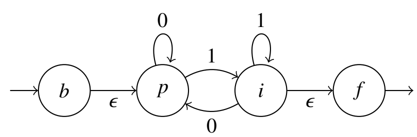
b. Una vez construido el automáta obtener su expresión regular por medio del método BMC.

Solución:

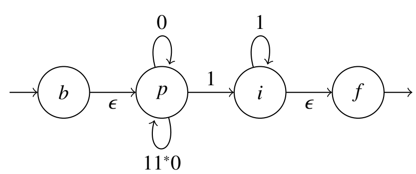
a. Definimos un automáta que detecten los valores impares:



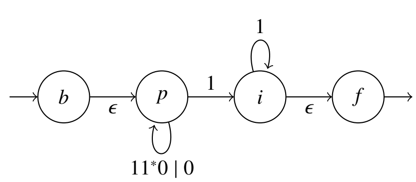
b. Utilizando el método BMC calculamos la expresión regular. Primero añadimos nuevos estados de inicio y de fin:



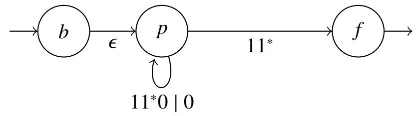
Calculamos una primera trayectoria entre $p \to i \to p$, observe que hay otra trayectoria entre $p \to i \to f$.



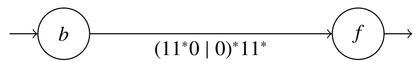
Renombramos los arcos que entran y salen del mismo p:



Tomamos ahora la trayectoria entre $p \to i \to f$:



Finalmente analizamos las trayectorias entre $b \rightarrow p \rightarrow f$:



La expresión regular para nuestro automáta es: $(11^*0 \mid 0)^*11^*$.

Pregunta	Puntos	Valor obtenido
1	40	
2	30	
3	30	
Total:	100	

Nombre: ______________________________

Código: ______________________________

(60 %) 1. **Condicción** $LL(1)$

La siguiente gramática G_1 no cumple con la condición $LL(1)$.

$$G_1 = (\{S\}, \{a, b\}, P, S)$$

Donde P son las producciones de la gramática G_1:

$$\begin{array}{rcl} S & \rightarrow & S\ S\ + \\ & | & S\ S\ * \\ & | & a \end{array}$$

Transforme la gramática G_1 en la gramática G_1' que cumpla con la condición $LL(1)$.

Solución:

Es obvio que la gramática no es el $LL(1)$ debido a la recursividad por la izquierda. Pero en vez de comenzar con eliminar la recursivad realizamos la factorización de la gramática original:

$$\begin{array}{rcl} S & \rightarrow & S\ S\ + \\ & | & S\ S\ * \\ & | & a \end{array}$$

Encontramos cuáles son los factores:

$$\begin{array}{rcl} S & \rightarrow & S\ S\ (+ \mid *) \\ & | & a \end{array}$$

Factorizamos:

$$\begin{array}{rcl} S & \rightarrow & S\ S\ S' \\ & | & a \\ S' & \rightarrow & + \\ S' & \rightarrow & * \end{array}$$

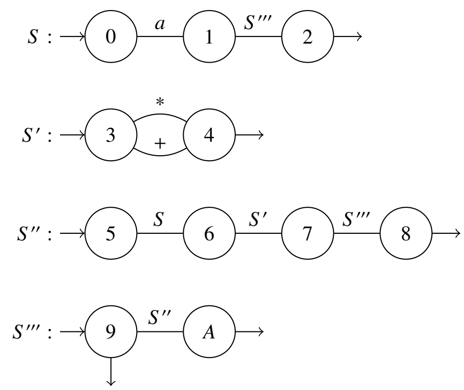
Ahora eliminamos la recursividad por la izquierda:

$$\begin{array}{rcl} S & \rightarrow & a \mid a\ S'' \\ S' & \rightarrow & + \mid * \\ S'' & \rightarrow & S\ S' \mid S\ S'\ S'' \end{array}$$

Pero esta gramática no es $LL(1)$. Debemos factorizar de nuevo y obtenemos:

$$\begin{array}{rcl} S & \rightarrow & a\ S''' \\ S' & \rightarrow & + \mid * \\ S'' & \rightarrow & S\ S'\ S''' \\ S''' & \rightarrow & \epsilon \mid S'' \end{array}$$

Ahora verificamos que esta gramática es $LL(1)$.



Calculamos los conjuntos guías de las bifurcaciones,

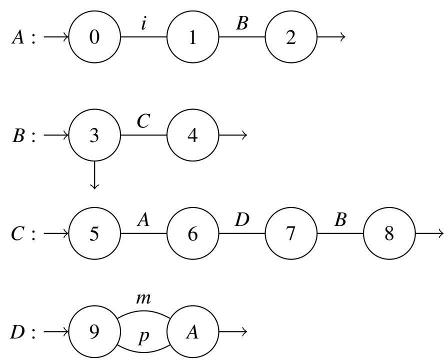
$$
\begin{aligned}
&Gui(3 \xrightarrow{+} 4) = \{+\}\\
&Gui(3 \xrightarrow{*} 4) = \{*\}\\
&Gui(9 \rightarrow) = Fol(S''') = \{+, -, \dashv\}\\
&Gui(9 \xrightarrow{S''} A) = Ini(L(5)L(A)) = \{a\}
\end{aligned}
$$

No hay elementos comunes entre las bifurcaciones que tiene un mismo origen. Por lo tanto la gramática es $LL(1)$.

(40 %) 2. **Reconocimiento de cadenas** La gramática G_2:

$$G_2 = (\{A, B, C, D\}, \{i, m, p\}, \{A \rightarrow iB, B \rightarrow \epsilon \mid C, C \rightarrow ADB, D \rightarrow m \mid p\}, A)$$

Es $LL(1)$, como se puede inferir de su red de máquinas y sus conjuntos guías:



Conjunto guía:

$$\begin{aligned}
Gui(0 \xrightarrow{i} 1) &= Ini(L(0)L(1)) = \{i\} \\
Gui(1 \xrightarrow{B} 2) &= Ini(L(3)L(2)) \cup Fol(A) = \{i, m, p, \dashv\} \\
Gui(2 \rightarrow) &= Fol(A) = \{m, p, \dashv\} \\
Gui(3 \rightarrow) &= Fol(B) = \{m, p, \dashv\} \\
Gui(3 \xrightarrow{C} 4) &= Ini(L(5)L(4)) = \{i\} \\
Gui(4 \rightarrow) &= Fol(B) = \{i, m, p, \dashv\} \\
Gui(5 \xrightarrow{A} 6) &= Ini(L(0)L(6)) = \{i\} \\
Gui(6 \xrightarrow{D} 7) &= Ini(L(9)L(6)) = \{m, p\} \\
Gui(7 \xrightarrow{B} 8) &= Ini(L(3)L(8)) \cup Fol(C) = \{i, m, p, \dashv\} \\
Gui(8 \rightarrow) &= Fol(C) = \{m, p, \dashv\} \\
Gui(9 \xrightarrow{m} A) &= \{p\} \\
Gui(9 \xrightarrow{p} A) &= \{m\} \\
Gui(A \rightarrow) &= Fol(D) = \{i, m, p, \dashv\}
\end{aligned}$$

Reconocer las siguientes cadenas:

a. $i\ i\ p\ i\ m \dashv$

b. $i\ p\ i\ \dashv$

Para cada verificación llenar un tabla con los siguientes campos:

Pila	**cadena**	**Comentarios**

Solución:

En primer lugar debemos calcular todos los conjuntos guías de la red de automátas que se encuentran en la red del punto 1:

a. Para la cadena $i\ i\ p\ i\ m\ \dashv$:

Pila	cadena	Comentario
0	$iipim \dashv$	Movimiento de examén
1	$ipim \dashv$	Movimiento de invocación
13	$ipim \dashv$	Movimiento de invocación
135	$ipim \dashv$	Movimiento de invocación
1350	$pim \dashv$	Movimiento de examén
1351	$pim \dashv$	Movimiento de invocación
13513	$pim \dashv$	Movimiento de acceptación
1352	$pim \dashv$	Movimiento de acceptación
136	$pim \dashv$	Movimiento de invocación
1369	$pim \dashv$	Movimiento de examén
136A	$im \dashv$	Movimiento de acceptación
137	$im \dashv$	Movimiento de invocación
1373	$im \dashv$	Movimiento de invocación
13735	$im \dashv$	Movimiento de invocación
137350	$im \dashv$	Movimiento de examén
137351	$m \dashv$	Movimiento de invocación
1373513	$m \dashv$	Movimiento de acceptación
137352	$m \dashv$	Movimiento de acceptación
13736	$m \dashv$	Movimiento de invocación
137369	$m \dashv$	Movimiento de examén
13736A	$\dashv$	Movimiento de acceptación
13737	$\dashv$	Movimiento de invocación
137373	$\dashv$	Movimiento de acceptación
13738	$\dashv$	Movimiento de acceptación
1374	$\dashv$	Movimiento de acceptación
138	$\dashv$	Movimiento de acceptación
14	$\dashv$	Movimiento de acceptación
2	$\dashv$	Movimiento de acceptación
	$\dashv$	Para el autómata

Por lo tanto la cadena $i\ i\ p\ i\ m\ \dashv$ es aceptada por el reconocedor, esta cadena es válida en G_2.

b. Para la cadena $i\ p\ i\ \dashv$:

Pila	**cadena**	**Comentario**
0	$ipi\dashv$	Movimiento de examén
1	$pi\dashv$	Movimiento de invocación
13	$pi\dashv$	Movimiento de acceptación
2	$pi\dashv$	Movimiento de acceptación
	$pi\dashv$	Falló, la pila no está vacía

Por lo tanto la cadena $i\ p\ i\ \dashv$ no es una cadena válida de G_2.

Pregunta	Puntos	Obtenido
1	60	
2	40	
Total:	100	

ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Supletorio segundo parcial — **DIS**
Juan Francisco Cardona M — **Octubre 6 de 2010**

Nombre: ______________________________
Código: ______________________________

(40 %) 1. **Construcción de autómatas a partir de una expresión regular**

a. Utilice el método estructural de Thompson para convertir la expresión regular

$$re_1 = (a(a \mid b))$$

en un AFN (AFN_ϵ_1) con movimientos espontáneos.

b. Elimine en el autómata (AFN_ϵ_1) los movimientos espontáneos y conviértalo en un autómata no determinista sin movimientos espontáneos AFN (AFN_1).

c. Convierta el AFN (AFN_1) en un AFD (AFD_1) utilizando el método de los conjuntos de potencia.

(30 %) 2. **Generación de autómatas utilizando el método de *Berry* y *Sethi***
Dada la siguiente expresión regular

$$re_2 = (ab \mid ca \mid cb)^*$$

y utilizando el método de *Berry* y *Sethi* (BC), calcular:

a. los conjuntos: $Ini(re'_2)$, $Fin(re'_2)$, $Null(re'_2)$, $Dig(re'_2)$, $Follow(re'_2)$.

b. el automáta determinista correspondiente a la expresión regular.

(30 %) 3. **Generación de automátas y su expresión correspondiente**

a. Para el alfabeto $\Sigma = \{a, b, c\}$, construir un automáta que cualquier termine con cualquier cadena de dos cs.

b. Una vez construido el automáta obtener su expresión regular por medio del método BMC.

Pregunta	Puntos	Valor obtenido
1	40	
2	30	
3	30	
Total:	100	

Nombre: ______________________________
Código: ______________________________

(20 %) 1. **Lenguajes y expresiones regulares**
Determine el lenguaje que representa cada una de las siguientes expresiones regulares, tenga en cuenta que el alfabeto Σ es $\{a, b, c\}$.

1. $r_1 = a(a^* \mid b^*)c^*$.
2. $r_2 = ((aa)^* \mid (bbb)^*)c^*$.

Solución:

1. $L(r_1) = \{a^m c^n \mid m \geq 1, n \geq 0\} \cup \{ab^m c^n \mid m \geq 0, n \geq 0\}$
2. $L(r_2) = \{a^{2*n} c^m \mid n \geq 0, m \geq 0\} \cup \{b^{3*n} c^m \mid m \geq 0, n \geq 0\}$

(20 %) 2. **Expresiones regulares y lenguajes**
Encontrar las expresiones regulares r_3 y r_4, que describan respectivamente a cada uno de los lenguajes L_3 y L_4.
L_3 es el conjunto de cadenas del alfabeto $\{a, b, c\}$ que contengan por lo menos una a y por lo menos una b y que además la primera a está antes (precede) a la primera b

$$L_4 = \{\epsilon, c, aa, aab, aabb, aabbb, \ldots, aab^n, \cdots \mid n \in N\}$$

Solución:

1. $r_3 = c^* a(a \mid c)^* b(a \mid b \mid c)^*$
2. $r_4 = aab^* \mid c \mid \epsilon$

(20 %) 3. **Lenguajes de las gramáticas independientes de contexto**
Encontrar los lenguajes $L(G_1)$ y $L(G_2)$ para cada una de las gramáticas independientes de contexto G_1 y G_2 respectivamente:

$$\begin{aligned} G_1 &= (\{S\}, \{a, b\}, \{S \rightarrow aaS, S \rightarrow a, S \rightarrow b\}, S) \\ G_2 &= (\{S, A, B, C\}, \{a, b, c\}, \{S \rightarrow ABC, A \rightarrow a \mid aB \mid bC, B \rightarrow b \mid cC, C \rightarrow c\}, S) \end{aligned}$$

Solución:

$$\begin{aligned} L(G_1) &= \{a^{2n+1} \mid n \geq 0\} \cup \{a^{2n}b \mid n \geq 0\} \\ L(G_2) &= \{abc, accc, abbc, abccc, accbc, accccc, bcbc, bcccc\} \end{aligned}$$

(20 %) 4. **Lenguajes y gramáticas independientes de contexto**
Definir la gramática independiente de contexto G_3 que genera el siguiente lenguaje:

$$L(G_3) = \{a^n b^m c^m b^n \mid n \geq 0,\ m \geq 1\}$$

Solución:

$$G_3 = (\{S, D\}, \{a, b, c\}, \{S \rightarrow aSb \mid D, D \rightarrow bDc \mid bc\}, S)$$

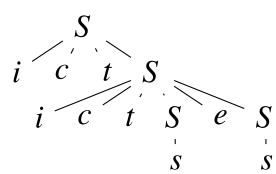
(20 %) 5. **Ambiguedad de gramáticas independientes de contexto**
Se dice que una gramática es ambigua cuando permite más de un árbol de análisis sistático para una sola cadena. Demuestre que la gramática G_4 que se presenta a continuación es ambigua, mostrando que la cadena *ictictses* tiene derivaciones que producen distintos árboles sintácticos.

$$G_4 = (\{S\}, \{i, c, t, e, s\}, \{S \rightarrow ictS \mid ictSeS \mid s\}, S)$$

Solución:

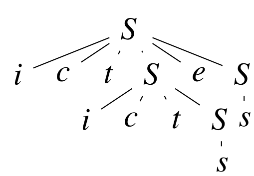
Primera derivación:

$$S \Rightarrow ictS \Rightarrow icticts SeS \Rightarrow ictictseS \Rightarrow ictictses$$



Segunda derivación:

$$S \Rightarrow ictSeS \Rightarrow ictictSeS \Rightarrow ictictseS \Rightarrow ictictses$$



Pregunta	Puntos	Obtenido
1	20	
2	20	
3	20	
4	20	
5	20	
Total:	100	

ST0270 Lenguajes formales y compiladores | **Universidad EAFIT**
Primer Parcial | **DIS**
Cardona, Guarín y Trefftz | **Marzo 31 de 2011**

Nombre: ______________________________

Código: ______________________________

(20 %) 1. **Conversión de recursividades izquierdas en derechas**

Dada la gramática independiente del contexto

$$G = (\{A_1, A_2\}, \{a, b, c\}, P, A)$$

donde el conjunto de reglas P está dado por:

$$\begin{array}{lcl} A_1 & \rightarrow & A_1 a \mid A_1 b \mid cA_2 \\ A_2 & \rightarrow & A_2 a \mid a \mid \epsilon \end{array}$$

Construir una gramática equivalente eliminando todas las recursividades por la izquierda.

Solución:

Obsérvese que hay reglas recursivas por la izquierda con A_1, A_2 en su lado izquierdo. Aplicando la conversión al no terminal A_1 e introduciendo un nuevo no-terminal A_1' se tiene la siguiente gramática transformada

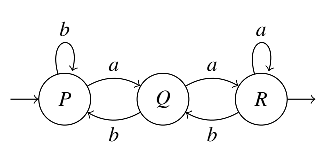
$$\begin{array}{lcl} A_1 & \rightarrow & cA_2 \mid cA_2A_1' \\ A_1' & \rightarrow & a \mid b \mid aA_1' \mid bA_1' \\ A_2 & \rightarrow & A_2 a \mid a \mid \epsilon \end{array}$$

Entonces aplicando la conversión al no terminal A_2 e introduciendo un nuevo no terminal A_2' se obtiene la gramática independiente de contexto, con reglas

$$\begin{array}{lcl} A_1 & \rightarrow & cA_2 \mid cA_2A_1' \\ A_1' & \rightarrow & a \mid b \mid aA_1' \mid bA_1' \\ A_2 & \rightarrow & aA_2' \mid A_2' \mid a \mid \epsilon \\ A_2' & \rightarrow & a \mid aA_2' \end{array}$$

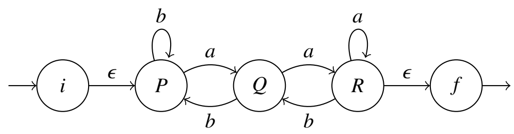
(20 %) 2. **Expresiones regulares y autómatas**

Obtener la expresión regular asociada al lenguaje aceptado por el autómata finito determinista definido por el siguiente diagrama de transición (utilizar el método BMC).

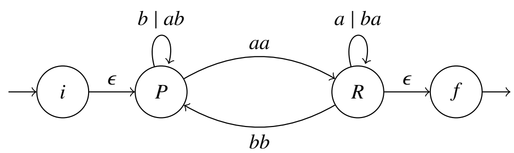


Solución:

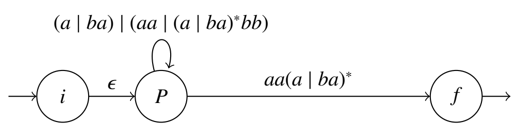
Añadimos dos estados nuevos para reemplazar los estados iniciales y finales:



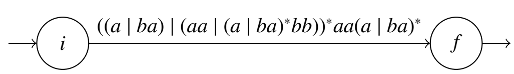
Eliminamos el nodo más interior: Q, teniendo en cuenta todos los posibles caminos.



Ahora eliminamos el nodo R.



Finalmente eliminando P tenemos:

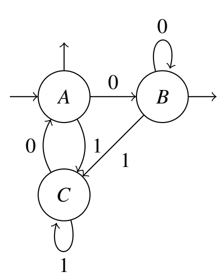


El resultado obtenido es:

$$((a \mid ba) \mid (aa \mid (a \mid ba)^{*}bb))^{*}aa(a \mid ba)^{*}$$

(20 %) 3. **Minimización de autómatas**

Obtener el autómata mínimo del autómata definido por el siguiente diagrama.



Solución:

Verificamos que los estados que estados sean indistinguibles, creando un tabla para analizar la relación. Observamos que por definición cada estado es indistinguible de si mismo:

A	X		
B		X	
C			X
	A	B	C

Comparamos ahora B y A encontramos que ambos estados son indistinguibles:

$$\begin{aligned} \delta(A,0) &= B \\ \delta(B,0) &= B \\ \delta(A,1) &= C \\ \delta(B,1) &= C \end{aligned}$$

Por lo tanto

A	X		
B	$[A,B]$	X	
C			X
	A	B	C

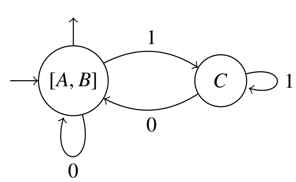
Ahora comparamos C con A y encontramos:

$$\begin{aligned} \delta(A,0) &= B \\ \delta(C,0) &= A \\ \delta(A,1) &= C \\ \delta(C,1) &= C \end{aligned}$$

Ambos estados son distinguibles. Ahora miramos C y B y encontramos:

$$\begin{aligned} \delta(B,0) &= B \\ \delta(C,0) &= A \\ \delta(B,1) &= C \\ \delta(C,1) &= C \end{aligned}$$

Ambos estados son distinguibles. Lo siguiente es reconstruir el autómata teniendo en cuenta, que los estados $[A,B]$ son indistinguibles:

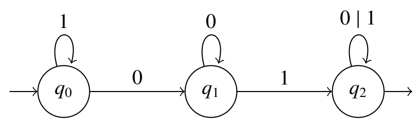


(20 %) 4. **Definición de autómatas deterministas**

Defina una autómata determinista que acepta todas las cadenas que contiene a la subcadena 01, teniendo en cuenta que su alfabeto Σ es {0, 1}.

Solución:

El siguiente es el autómata determinista:



(20 %) 5. **Expresiones regulares para compiladores**

Como parte de un proyecto para la NASA, usted ha sido encargado de crear un lenguaje sencillo que permite hacer expresiones con números enteros y variables.

No es necesario definir las variables antes de usarlas.

La gramática del lenguaje, en ANTLR, es la siguiente:

```
grammar Lenguajito;

prog       :  instr_list  DOT
           ;

instr_list :  instr*
           ;

instr      :  asignacion SEMI
           |  impresion SEMI
           ;

asignacion :  VARIABLE EQ expr
           ;

expr       :  term (PLUS  term | MINUS  term)*
           ;

term       :  factor ( TIMES factor)*
           ;

factor     :  NUMBER
           |  VARIABLE
           ;

impresion  :  PRINTCMD VARIABLE
           ;
```

El siguiente es un ejemplo de un programa en dicho lenguaje:

```
ab11 <- 50;
b55 <- 6;
c <- ab + b55 * 2 - 5;
imprimir c;
.
```

Usted ha sido encargado(a) de crear la especificación, en ANTLR, de los tokens que conforman el lenguaje.

Notas:

- Su especificación debe permitir ignorar espacios en blanco, tabuladores y lenguajes de salto de línea tanto para UNIX (\n) como para DOS (\r\n).
- Los nombres de variables comienzan con letras (mayúsculas o minúsculas) y luego puede venir una combinación de letras o números
- Los números son enteros y pueden ser de varios dígitos.

Solución:

Una posible respuesta sería:

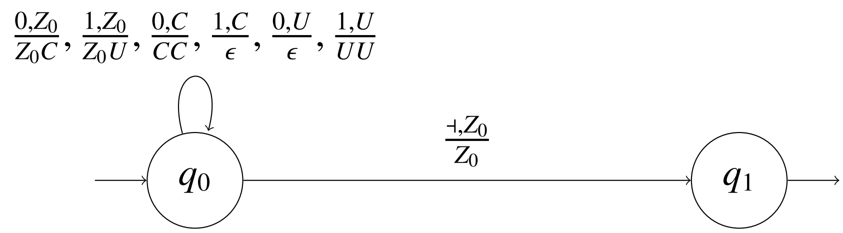
```
NEWLINE   :  '\r'? '\n' {skip();};
WS        :  (' '|'\t')+ {skip();} ;

SEMI      :  ';';
PLUS      :  '+';
MINUS     :  '-';
TIMES     :  '*';
EQ        :  '<-';

PRINTCMD  :  'imprimir';

DOT       :  '.';

NUMBER    :  '0'..'9'+;
VARIABLE  :  ('a'..'z'|'A'..'Z')(('a'..'z'|'A'..'Z')|'0'..'9')*;
```

ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Tercer Parcial — **DIS**
Cardona, Guarín y Trefftz — **Mayo 19 de 2011**

Nombre: ______________________________
Código: ______________________________

(30 %) 1. **Autómata de pilas**

Dado el autómata de pila $P = (Q, \Sigma, \Gamma, S, q_0, Z_0, F)$ definido por el siguiente diagrama de transición



y las cadenas:

1. $x = 01001 \in \Sigma^*$
2. $y = 10101 \in \Sigma^*$

realizar sus computaciones (a partir de la configuración inicial) para determinar su aceptación o no por el autómata.

Solución:

1.
$$\begin{array}{rcl}
(q_0, x, Z_0) & = & (q_0, 01001, Z_0) \\
& \vdash & (q_0, 1001 \dashv, Z_0C) \\
& \vdash & (q_0, 001 \dashv, Z_0) \\
& \vdash & (q_0, 01 \dashv, Z_0C) \\
& \vdash & (q_0, 1 \dashv, Z_0CC) \\
& \vdash & (q_0, 1 \dashv, Z_0C)
\end{array}$$

No hay más movimientos posibles.

2.
$$\begin{array}{rcl}
(q_0, y, Z_0) & = & (q_0, 10101, Z_0) \\
& \vdash & (q_0, 0101 \dashv, Z_0U) \\
& \vdash & (q_0, 101 \dashv, Z_0) \\
& \vdash & (q_0, 01 \dashv, Z_0U) \\
& \vdash & (q_0, 1 \dashv, Z_0) \\
& \vdash & (q_0, \dashv, Z_0U)
\end{array}$$

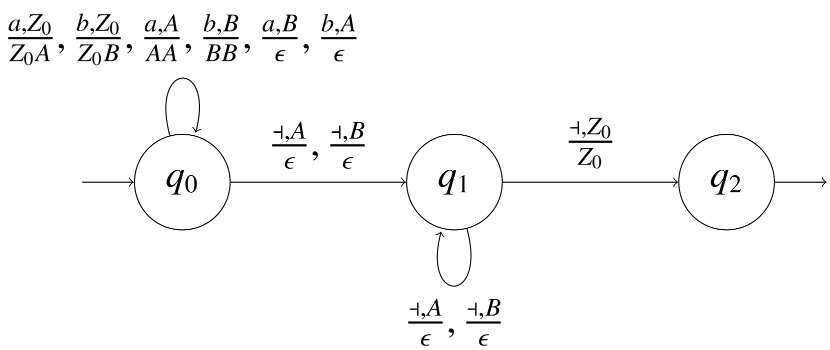
No hay más movimientos posibles.

(30 %) 2. **Autómatas de pila**

Defina un autómata de pila determinista que deja la pila vacía antes de aceptar y que reconozca únicamente el siguiente lenguaje:

$$L_1 = \{x \in \{a, b\}^* \mid n_a(x) \neq n_b(x)\}$$

Solución:



(40 %) 3. **Condición** $LL(1)$

Transformar la siguiente gramática G_1 y probar que la nueva gramática G'_1 cumple con la condición $LL(1)$.

$$G_1 = (\{S, A\}, \{a, b\}, R, S)$$

donde R define las producciones de la gramática G_1:

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & Aa \mid Ab \mid \epsilon \end{array}$$

Solución:

En primer lugar eliminamos la recursividad por la izquierda de la gramática.

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & \epsilon \mid B \\ B & \rightarrow & a \mid aB \mid b \mid bB \end{array}$$

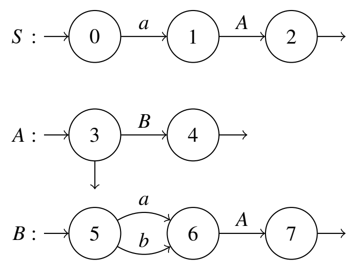
Luego eliminamos realizamos la factorización de la gramática.

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & \epsilon \mid B \\ B & \rightarrow & aC \mid aC \\ C & \rightarrow & \epsilon \mid B \end{array}$$

Se observa que hay dos producciones similares $A \to \epsilon \mid B$ y $C \to \epsilon \mid D$, sustituimos C por A:

$$\begin{array}{rcl} S & \to & aA \\ A & \to & \epsilon \mid B \\ B & \to & aA \mid aA \end{array}$$

Construimos luego la red de autómatas:



Calculamos los conjuntos guías de las bifurcaciones,

$$\begin{array}{l} Gui(3 \xrightarrow{B} 4) = Ini(L(5)L(4)) = \{a, b\} \\ Gui(3 \to) = Fol(A) = Fol(S) = \{\dashv\} \\ Gui(5 \xrightarrow{a} 6) = \{a\} \\ Gui(5 \xrightarrow{b} 6) = \{b\} \end{array}$$

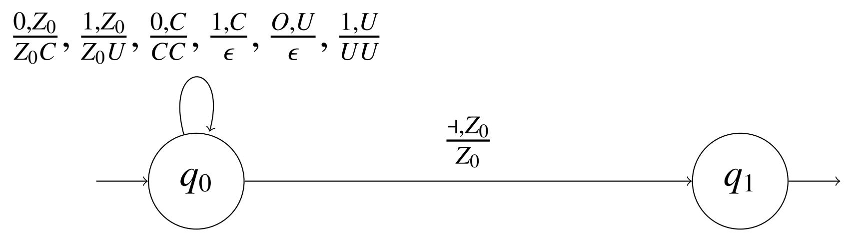
No hay elementos comunes entre las bifurcaciones que tiene un mismo origen. Por lo tanto la gramática es $LL(1)$.

ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Tercer Parcial — **DIS**
Cardona, Guarín y Trefftz — **Mayo 19 de 2011**

Nombre: ______________________________

Código: ______________________________

(30 %) 1. **Autómata de pilas**
Dado el autómata de pila $P = (Q, \Sigma, \Gamma, S, q_0, Z_0, F)$ definido por el siguiente diagrama de transición



y las cadenas:

1. $x = 01001 \in \Sigma^*$
2. $y = 10101 \in \Sigma^*$

realizar sus computaciones (a partir de la configuración inicial) para determinar su aceptación o no por el autómata.

(35 %) 2. **Condición** $LL(1)$
Comprobar que la gramática

$$G_1 = (\{S, A\}, \{a, b\}, R, S)$$

cumple con la condición $LL(1)$. Donde R define las producciones de la gramática G_1:

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & Aa \mid Ab \mid \epsilon \end{array}$$

Generar la tabla del parser descendente. Hagan el seguimiento a la cadena $x = aab$.

(35 %) 3. **Condición** $LR(1)$
Comprobar que la gramática

$$G_2 = (\{S, A\}, \{a, b\}, R, S)$$

cumple con la condición $LR(1)$. Donde R define las producciones de la gramática G_2:

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & Aa \mid Ab \mid \epsilon \end{array}$$

Generar la tabla del parser ascendente. Haga seguimiento a la cadena $y = aab$.

Nombre: ______________________________
Código: ______________________________

(25 %) 1. **Expresiones regulares a autómatas deterministas**
Dada la siguiente expresión regular:

$$r_1 = (ab \mid ac)^*$$

encontrar el autómata determinista que reconozca cadenas x que cumplen con la condición $x \in L(r_1)$.

Nota: Implementar el autómata ***determinista*** utilizando alguno de los métodos: el método estructural Thompson, el algoritmo de Glushkov, McNaughton y Yamada (GMY) y algoritmo de Berry y Sethi. Recuerden que dos de los anteriores métodos puede generar un autómata no-determinista, en cuyo caso deben seguir aplicando otros métodos para convertirlo en determinista.

Solución:

Utilizando el algoritmo de Berry y Sethi

1. Numerar la expresión regular r_1

$$r'_1 = (a_1b_2 \mid a_3c_4)^*$$

2. Computar el conjunto $Null(r'_1)$:

$$\begin{aligned} Null(r'_1) &= Null((a_1b_2 \mid a_3c_4)^*) \\ &= \epsilon \end{aligned}$$

Computar el conjunto $Ini(r'_1)$:

$$\begin{aligned} Ini(r'_1) &= Ini((a_1b_2 \mid a_3c_4)^*) \\ &= Ini((a_1b_2 \mid a_3c_4)) \\ &= Ini(a_1b_2) \cup Ini(a_3c_4) \\ &= Ini(a_1) \cup Null(a_1)Ini(b_2) \cup Ini(a_3) \cup Null(a_3)Ini(c_4) \\ &= \{a_1\} \cup \emptyset \cup \{a_3\} \cup \emptyset \\ &= \{a_1, a_3\} \end{aligned}$$

El conjunto $Fin(r'_1)$:

$$\begin{aligned} Fin(r'_1) &= Fin((a_1b_2 \mid a_3c_4)^*) \\ &= Fin((a_1b_2 \mid a_3c_4)) \\ &= Fin(a_1b_2) \cup Fin(a_3c_4) \\ &= Fin(b_2) \cup Fin(a_1)Null(b_2) \cup Fin(c_4) \cup Fin(a_3)Null(c_4) \\ &= \{b_2\} \cup \emptyset \cup \{c_4\} \cup \emptyset \\ &= \{b_2, c_4\} \end{aligned}$$

El conjunto $Dig(r_1')$:

$$\begin{aligned}
Dig(r_1') &= Dig((a_1b_2 \mid a_3c_4)^*) \\
&= Dig(a_1b_2 \mid a_3c_4) \cup Fin(a_1b_2 \mid a_3c_4)Ini(a_1b_2 \mid a_3c_4) \\
&= Dig(a_1b_2) \cup Dig(a_3c_4) \cup \{b_2, c_4\}\{a_1, a_3\} \\
&= Dig(a_1) \cup Dig(b_2) \cup Fin(a_1)Ini(b_2) \cup Dig(a_3) \cup Dig(c_4) \\
&\cup\ Fin(a_3)Ini(c_4) \cup \{b_2a_1, b_2a_3, c_4a_1, c_4a_3\} \\
&= \emptyset \cup \emptyset \cup \{a_1\}\{b_2\} \cup \emptyset \cup \emptyset \cup \{a_3\}\{c_4\} \cup \{b_2a_1, b_2a_3, c_4a_1, c_4a_3\} \\
&= \{a_1b_2, a_3c_4, b_2a_1, b_2a_3, c_4a_1, c_4a_3\}
\end{aligned}$$

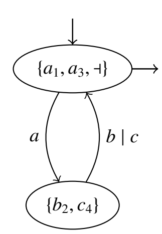
Computar el conjunto de $Fol(a_i) = \{b_j \mid a_ib_j \in Dig(r_1')\}$

$$\begin{aligned}
Fol(a_1) &= \{b_2\} \\
Fol(b_2) &= \{a_1, a_3\} \cup \{\dashv\} \\
Fol(a_3) &= \{c_4\} \\
Fol(c_4) &= \{a_1, a_3\} \cup \{\dashv\}
\end{aligned}$$

Computar el conjunto de $Ini(r_1' \dashv)$:

$$\begin{aligned}
Ini(r_1' \dashv) &= Ini((a_1b_2 \mid a_3c_4)^* \dashv) \\
&= Ini((a_1b_2 \mid a_3c_4)) \cup Null((a_1b_2 \mid a_3c_4))Ini(\dashv) \\
&= Ini(a_1b_2) \cup Ini(a_3c_4) \cup \epsilon\{\dashv\} \\
&= Ini(a_1) \cup Null(a_1)Ini(b_2) \cup Ini(a_3) \cup Null(a_3)Ini(c_4) \cup \{dashv\} \\
&= \{a_1\} \cup \emptyset \cup \{a_3\} \cup \emptyset \cup \{dashv\} \\
&= \{a_1, a_3, \dashv\}
\end{aligned}$$

Este es el estado inicial de allí el siguiente autómata:



(25 %) 2. **Condición** $LL(1)$

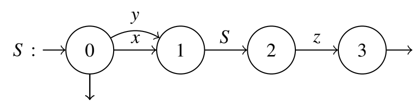
Demostrar que la siguiente gramática G_1

$$\begin{aligned}
S &\rightarrow xSz \\
S &\rightarrow ySz \\
S &\rightarrow \epsilon
\end{aligned}$$

cumple con la condición $LL(1)$.

Solución:

Definimos la red de autómatas:

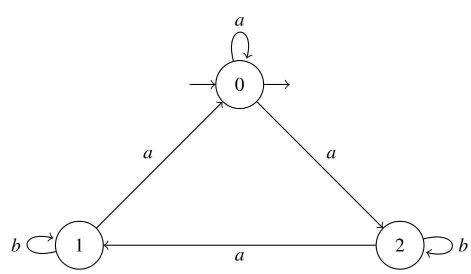


Observamos que existen varias bifucaciones en el estado 0. Calculamos los conjuntos guías de cada bifucarción: $Gui(0 \xrightarrow{y} 1)$, $Gui(0 \xrightarrow{x} 1)$ y $Gui(0 \rightarrow)$.

$$\begin{aligned} Gui(0 \xrightarrow{y} 1) &= \{y\} \\ Gui(0 \xrightarrow{x} 1) &= \{x\} \\ Gui(0 \rightarrow) &= Fol(S) \\ &= \{z, \dashv\} \end{aligned}$$

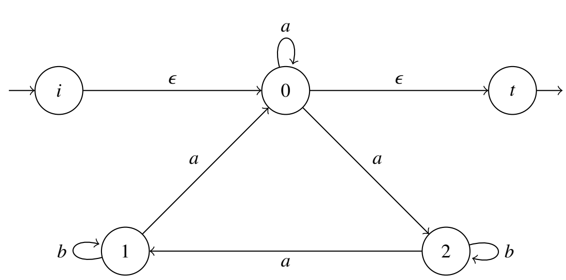
Como se puede observar ninguna de las bifurcaciones se translapa, por lo tanto la gramática es $LL(1)$.

(25 %) 3. **De autómata a expresión regular**
Encontrar la expresión regular que representa el siguiente autómata:

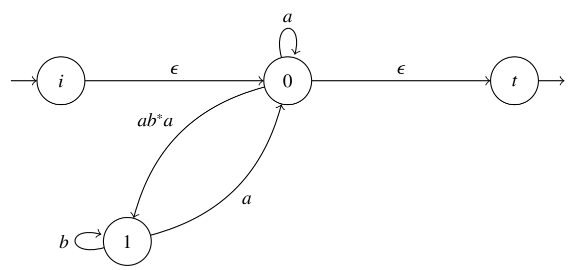


Solución:

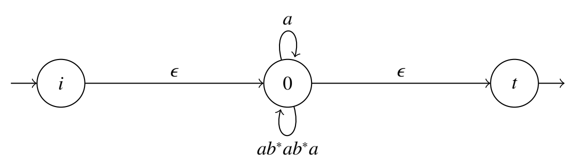
En primer lugar, añadimos los estados auxiliares de inicio i y terminación t:



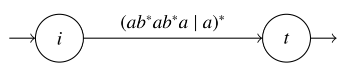
Eliminamos el primer estado intermedio 2:



Luego, eliminamos el estado intermedio 1:



Finalmente, eliminamos el estado intermedio 0:



La expresión regular que representa al autómata A_3 es:

$$(ab^*ab^*a \mid a)^*$$

(25 %) 4. **Transformación de gramáticas y condición** $LL(1)$

La siguiente gramática independiente de contexto $G_4 = (\Sigma = \{\text{id}, \text{'},\text{'}, \text{'int'}, \text{'float'}, \}, V = \{Decl, Tipo, Varlist\}, P = \{$

$$\begin{array}{rcl} Decl & \rightarrow & Tipo\ VarList \\ Tipo & \rightarrow & \text{'int'} \mid \text{'float'} \\ VarList & \rightarrow & \text{id}\ \text{','}\ VarList \mid \text{id} \end{array}$$

$\}, Decl)$ no cumple con la condición $LL(1)$, realize la(s) transformación(es) pertinente(s) para transformarla en un gramática G'_4 que cumple con la condición $LL(1)$. ***Probar*** que la nueva gramática G'_4 cumple con la condición $LL(1)$.

Solución:

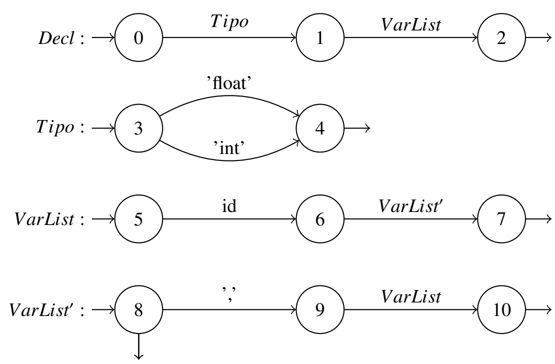
Realizamos primero una factorización por la izquierda, en la producción $VarList$:

$$VarList \rightarrow \text{id } (\epsilon \mid \text{','}\ VarList)$$

Creamos un nuevo no-terminal $VarList' \rightarrow \epsilon \mid \text{','}\ VarList$ y redefinimos la gramática nueva G_2'

$$\begin{aligned}
\Sigma &= \{\text{id}, \text{','}, \text{'int'}, \text{'float'}\} \\
V &= \{Decl, Tipo, Varlist, VarList'\} \\
P &= \{Decl \rightarrow Tipo\ VarList \\
&\quad , Tipo \rightarrow \text{'int'} \mid \text{'float'} \\
&\quad , VarList \rightarrow \text{id}\ VarList' \\
&\quad , VarList' \rightarrow \epsilon \mid \text{','}\ VarList\} \\
Decl &
\end{aligned}$$

Definimos la red de autómatas para la gramática G_4'



Observamos que existen varias bifucaciones. Calculamos los conjuntos guías de cada bifucarción: .

$$\begin{aligned}
Gui(3 \overset{\text{'int'}}{\rightarrow} 4) &= \{\text{'int'}\} \\
Gui(3 \overset{\text{'float'}}{\rightarrow} 4) &= \{\text{'float'}\} \\
Gui(8 \overset{\text{','}}{\rightarrow} 9) &= \{\text{','}\} \\
Gui(8 \rightarrow) &= Fol(VarList') \\
&= Fol(VarList) \\
&= \{\dashv\}
\end{aligned}$$

Como se puede observar ninguna de las bifurcaciones se translapa, por lo tanto la gramática es $LL(1)$.

ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Primer Parcial — **DIS**
Juan Francisco Cardona McCormick — **14 de Agosto de 2012**

Nombre: ______________________________
Código: ______________________________

(20 %) 1. **Expresiones regulares**

Escriba una expresión regular r_1 que defina el lenguaje de los números binarios que tienen *al menos* tres unos seguidos.

> **Solución:**
>
> $$r_1 = (0 \mid 1)^* 111 (0 \mid 1)^*$$

(30 %) 2. **Ambigüedad y transformación de gramáticas**

a) (25 %) Mostrar que la siguiente gramática G_1 es ambigua, dando una prueba de dicha propiedad.

$$\begin{array}{lcl} S & \rightarrow & AB \\ A & \rightarrow & aA \mid bA \mid \epsilon \\ B & \rightarrow & bB \mid cB \mid \epsilon \end{array}$$

> **Solución:**
>
> Observe la siguientes derivaciones:
>
> $$\begin{array}{lcl} S & \Rightarrow & AB \\ & \Rightarrow & aAB \\ & \Rightarrow & abAB \\ & \Rightarrow & ab\epsilon B \\ & \Rightarrow & abbB \\ & \Rightarrow & abb\epsilon \equiv abb \end{array}$$
>
> $$\begin{array}{lcl} S & \Rightarrow & AB \\ & \Rightarrow & aAB \\ & \Rightarrow & a\epsilon B \\ & \Rightarrow & abB \\ & \Rightarrow & abbB \\ & \Rightarrow & abb\epsilon \equiv abb \end{array}$$
>
> Producen los árboles de derivación: Por lo tanto para una misma cadena tenemos dos derivaciones distintas, por lo tanto la gramática es ambigua

b) (5 %) Sin cambiar el lenguaje de la anterior gramática G_1 generar una nueva gramática G_1' no ambigua y que cumpla con la condición $L(G_1) \equiv L(G_1')$.

Solución:

$$\begin{array}{rcl} S & \rightarrow & AB \\ A & \rightarrow & \epsilon \\ & | & A' \\ A' & \rightarrow & a \\ & | & bA' \mid aA' \\ B & \rightarrow & \epsilon \\ & | & bB \mid cB \end{array}$$

(20 %) 3. **Expresiones regulares ambiguas**
Mostrar que la siguiente expresión regular r_2 es ambigua, dando una prueba de dicha propiedad.

$$(a \mid b)^*(b \mid c)^*$$

Solución:

Observamos una posible cadena generada por la expresión regular.
En este caso bb.
Tenemos una nueva versión de la anterior expresion regular renumerando los terminales:

$$r_2 = (a_1 \mid b_2)^*(b_3 \mid c_4)^*$$

Aplicamos la derivación aplicando opciones y obtenemos:

$$\begin{array}{rcl} (a_1 \mid b_2)^*(b_3 \mid c_4)^* & \Rightarrow & \epsilon(b_3 \mid c_4)^* \\ & \Rightarrow & (b_3 \mid c_4)^2 \\ & \Rightarrow & (b_3 \mid c_4)(b_3 \mid c_4) \\ & \Rightarrow & b_3b_3 \\ & \Rightarrow & bb \end{array}$$

Con elección obtemos la misma cadena.

$$\begin{array}{rcl} (a_1 \mid b_2)^*(b_3 \mid c_4)^* & \Rightarrow & (a_1 \mid b_2)^2(b_3 \mid c_4)^* \\ & \Rightarrow & (a_1 \mid b_2)(a_1 \mid b_2)(b_3 \mid c_4)^* \\ & \Rightarrow & (b_2)(b_2)(b_3 \mid c_4)^* \\ & \Rightarrow & b_2b_2\epsilon \\ & \Rightarrow & bb \end{array}$$

Nuestra expresión regular es ambigua.

(30 %) 4. **Recursividad por la izquierda**
La siguiente gramática G_2 es recursiva por la izquierda. Transforme G_2 en un gramática, utilizando mecanismos formales, en una gramática G_2' no recursiva por la izquierda.

$$G_2 = (V = \{A, B\}, \Sigma = \{a, b, c\}, P = \{A \to Bc \mid Ad, B \to Aa \mid a \mid b \mid Bc\}, S = A)$$

Solución:

Tenemos una recursividad por la izquierda inmediata y no inmediata, por lo tanto aplicamos el algoritmo para la recursividad no-inmediata.
En primer lugar establecemos un orden entre los no-terminales, el orden alfabético.
Aplicando el algoritmo en la primera iteración para el no-terminal A, obtenemos la siguiente modificación de la gramática.

$$\begin{array}{rcl} A & \to & Bc \mid BcA' \\ A' & \to & d \mid dA' \\ B & \to & Aa \mid a \mid b \mid Bc \end{array}$$

Comenzamos la segunda iteración expandiendo los lados derechos del no-terminal B que comienze con A.

$$B \to Bca \mid BcA'a \mid a \mid b \mid Bc$$

Aplicamos el algoritmo inmediato al no terminal B y obtenemos:

$$\begin{array}{rcl} B & \to & a \mid aB' \mid b \mid bB' \\ B' & \to & ca \mid caB' \mid cA'a \mid cA'aB' \mid c \mid cB' \end{array}$$

Las producciones de la nueva gramática son:

$$\begin{array}{rcl} A & \to & Bc \mid BcA' \\ A' & \to & d \mid dA' \\ B & \to & a \mid aB' \mid b \mid bB' \\ B' & \to & ca \mid caB' \mid cA'a \mid cA'aB' \mid c \mid cB' \end{array}$$

ST0270 Lenguajes formales y compiladores — **Universidad EAFIT**
Segundo parcial — **DIS**
Juan Francisco Cardona McCormick — **11 de septiembre de 2012**

Nombre: ____
Código: ____

Nota: Toda solución debe ser formalmente escrita para ser aceptada como correcta.

(15 %) 1. **Expresión regular para un AFD**
Obtener una expresión regular para el autómata finito determinista M_1 de la figura 1

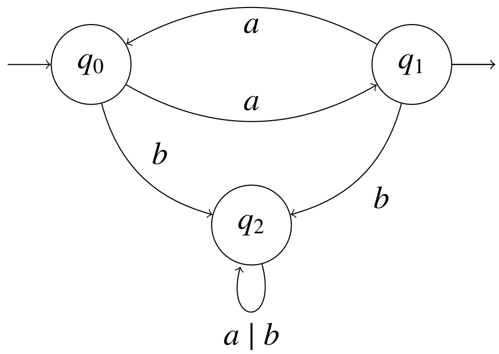
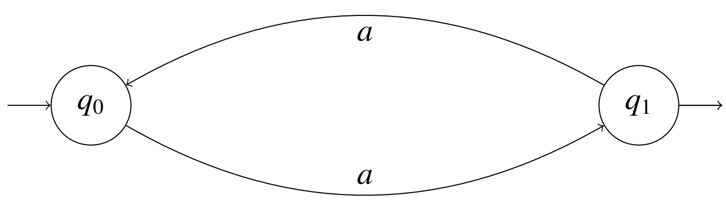
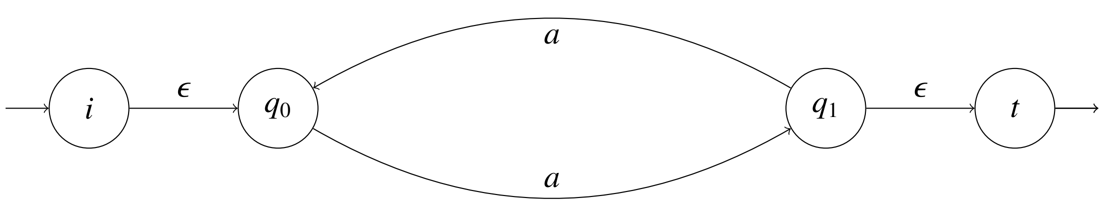
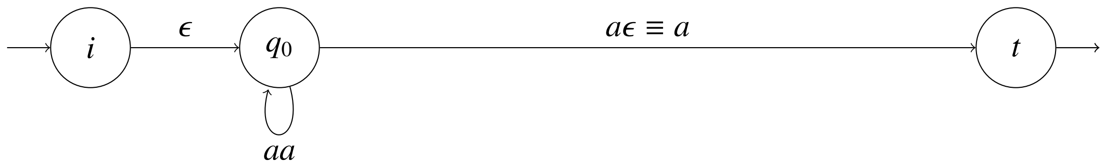


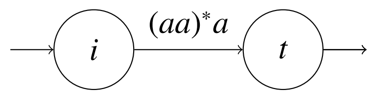
Figura 1: Autómata finito determinista

> **Solución:**
>
> En primer lugar observamos el autómata tiene un estado sumidero q_2, lo eliminamos y obtenemos:
>
> 
>
> Observamos que los estados inicial tiene arcos que entran a él y el final tiene arcos que salen de él. Por lo tanto añadimos dos estados nuevos i y t. Esto nos produce el siguiente autómata:
>
> 

Vamos a eliminar el estado interno q_1, hay dos caminos uno que va directo al estado t y otro que retorna a q_0, por lo tanto transformamos esos dos caminos.



Ahora eliminamos el estado interno q_0 y se obtiene:



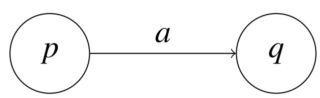
Por lo tanto la expresión regular de autómata es $(aa)^*a$.

(25 %) 2. **Gramáticas uni-lineal derechas a autómata no-determinista**
Dada la siguiente gramática uni-linea G_2 por la derecha, genere el correspondiente autómata no-determinista N_2 que reconoce el mismo lenguaje: $L(G_2) \equiv L(N_2)$

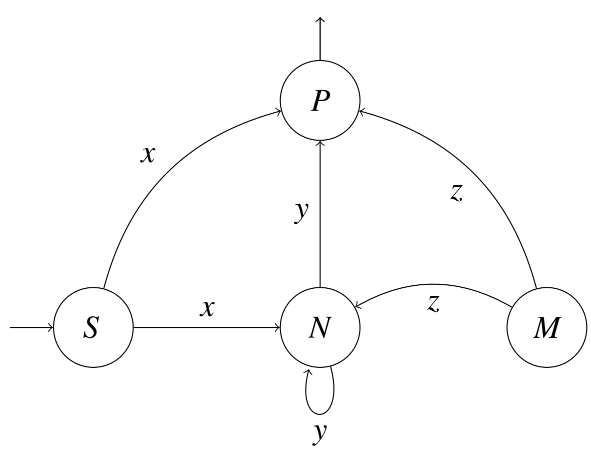
$$\begin{array}{rcl} S & \rightarrow & xN \\ S & \rightarrow & xP \\ N & \rightarrow & yN \\ N & \rightarrow & yP \\ M & \rightarrow & zN \\ M & \rightarrow & zP \\ P & \rightarrow & \epsilon \end{array}$$

Solución:

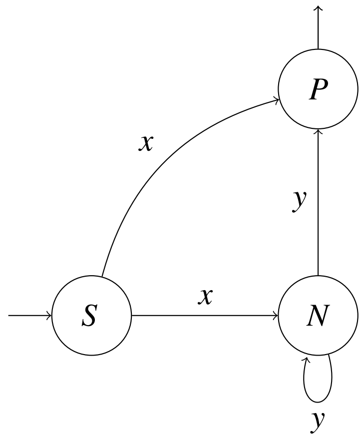
Los estados del autómata son los terminales de la gramática, $V = \{S, N, M, P\}$. El estado inicial q_0 es el axioma de la gramática S. El alfabeto Σ es el conjunto de terminales $\{x, y, z\}$. Por cada producción $p \rightarrow aq$, donde $a \in \Sigma$ y $p, q \in V$:



Esto nos produce el siguiente autómata:



Este es un automáta que no está limpio, el estado M es un estado inalcanzable, se elimina:


P
x
y
S
x
N
y

(25 %) 3. **De un autómata no-determinista a un autómata determinista**
El autómatas obtenido en el punto 2 es un autómata no-determinista N_2, transforme el correspondiente autómata en uno determinista M_3, donde $L(N_2) \equiv L(M_3)$.

Solución:

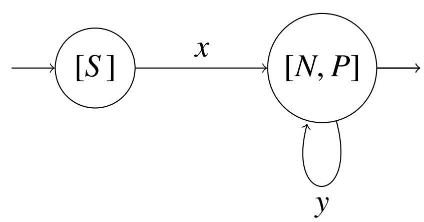
Construimos el conjunto de los estados accesibles, como el axioma del anterior autómata es S, partimos del nuevo estado inicial $[S]$. Calculamos el nuevo δ':

$$\begin{aligned}
\delta'([S], x) &= \delta(S, x) = \{N\} \cup \{P\} = [N, P] \\
\delta'([S], y) &= \delta(S, y) = \emptyset
\end{aligned}$$

Tenemos un nuevo súper-conjunto $[N, P]$, calculamos su nuevo δ':

$$\begin{aligned}
\delta'([N, P], x) &= \delta(N, x) \cup \delta(P, x) = \emptyset \cup \emptyset = \emptyset \\
\delta'([N, P], y) &= \delta(N, y) \cup \delta(P, y) = \{N, P\} \cup \emptyset = [N, P]
\end{aligned}$$

Ya hemos calculado todos los súper conjuntos: Ahora calculamos el autómata a partir del nuevo δ':



(10 %) 4. **Lenguajes locales**
Explique si la siguiente expresión regular r_4 genera o no un lenguaje local:

$$a(ab)^*c$$

Solución:

Calculamos los conjuntos inicial (Ini), final (Fin) y digrafo (Dig).

$$\begin{aligned} Ini &= \{a\} \\ Fin &= \{c\} \\ Dig &= \{aa, ab, ba, bc\} \end{aligned}$$

Se observa que con los conjuntos Ini, Fin y Dig se puede construir la siguiente cadena: $aaabc$. Pero si se observa esa cadena no pertenece al lenguaje generado por la expresión regular r_4. Por lo tanto la expresión regular r_4 no genera un lenguaje local.

(25 %) 5. **Eliminación de movimientos espontáneos**
El siguiente autómata A_5 (figura 2) tiene movimientos espontáneos, genere un autómata no determinista N_5 sin movimientos espontáneos, tal que $L(A_5) \equiv L(N_5)$.

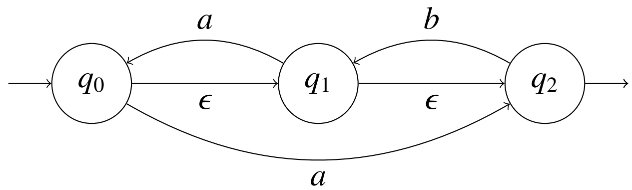


Figura 2: Autómata finito determinista

Solución:

Calculamos la gramática equivalente del anterior autómata:

$$\begin{array}{lcl} q_0 & \rightarrow & q_1 \\ q_0 & \rightarrow & aq_2 \\ q_1 & \rightarrow & q_2 \\ q_1 & \rightarrow & aq_0 \\ q_2 & \rightarrow & bq_1 \\ q_2 & \rightarrow & \epsilon \end{array}$$

Calculamos el conjunto de copia:

	Copy
q_0	q_0, q_1, q_2
q_1	q_1, q_2
q_2	q_2

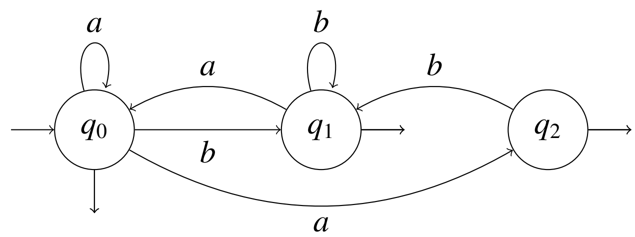
Eliminamos la reglas de copia del gramática:

$$\begin{array}{lcl} q_0 & \rightarrow & aq_2 \\ q_1 & \rightarrow & aq_0 \\ q_2 & \rightarrow & bq_1 \\ q_2 & \rightarrow & \epsilon \end{array}$$

Expandimos con las producciones de los otros no terminales del conjunto de copia:

$$\begin{array}{lcl} q_0 & \rightarrow & aq_2 \\ q_0 & \rightarrow & aq_0 \\ q_0 & \rightarrow & bq_1 \\ q_0 & \rightarrow & \epsilon \\ q_1 & \rightarrow & aq_0 \\ q_1 & \rightarrow & bq_1 \\ q_1 & \rightarrow & \epsilon \\ q_2 & \rightarrow & bq_1 \\ q_2 & \rightarrow & \epsilon \end{array}$$

Esto nos produce el siguiente autómata:



Nombre: ______________________________
Código: ______________________________

Nota: Toda solución debe ser formalmente escrita para ser aceptada como correcta.

(20 %) 1. **Algoritmo Berry y Sethi**
Utilizando la expresión r_1 del definir un autómata estado de finito determinista, *utilizando el algoritmo de Berry y Sethi*.

$$r_1 = a(a \mid b)^* b$$

Solución:

Renumeramos la expresión regular original:

$$r'_1 = a_1(a_2 \mid b_3)^* b_4$$

Calculamos el conjunto $Ini(r'_1)$:

$$\begin{aligned} Ini(r'_1) &= Ini(a_1) \cup Null(a_1) Ini((a_2 \mid b_3)^* b_4) \\ &= Ini(a_1) \cup \emptyset Ini((a_2 \mid b_3)^* b_4) \\ &= \{a_1\} \end{aligned}$$

Calculamos el conjunto $Fin(r'_1)$:

$$\begin{aligned} Fin(r'_1) &= Fin((a_2 \mid b_3)^* b_4) \cup Fin(a_1) Null((a_2 \mid b_3)^* b_4) \\ &= Fin(b_4) \cup Fin((a_2 \mid b_3)^*) Null(b_4) \cup Fin(a_1) Null((a_2 \mid b_3)^* b_4) \\ &= \{b_4\} \cup Fin((a_2 \mid b_3)^*)\emptyset \cup \{a_1\}(Null((a_2 \mid b_3)^*) \cap Null(b_4)) \\ &= \{b_4\} \cup \emptyset \cup \{a_1\}(\epsilon \cap \emptyset) \\ &= \{b_4\} \cup \emptyset \cup \emptyset \\ &= \{b_4\} \end{aligned}$$

Conjunto $Null(r'_1)$:

$$\begin{aligned} Null(r'_1) &= Null(a_1) \cap Null((a_2 \mid b_3)^* b_4) \\ Null(r'_1) &= \emptyset \cap Null((a_2 \mid b_3)^* b_4) \\ Null(r'_1) &= \emptyset \end{aligned}$$

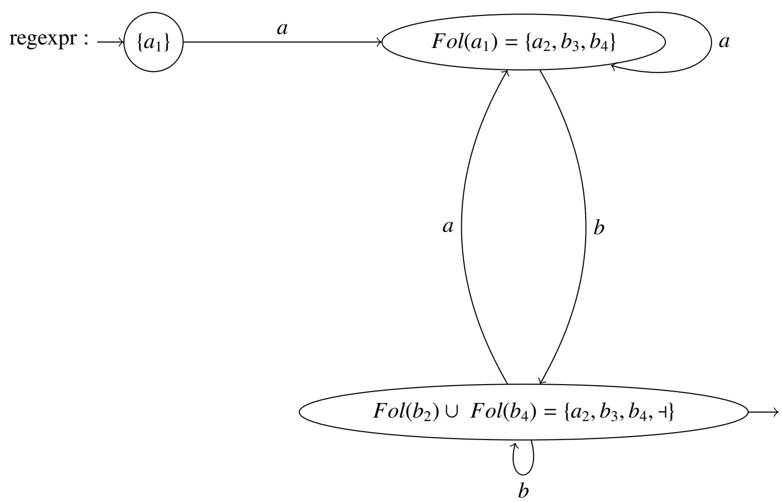
Conjunto $Dig(r_1')$:

$$
\begin{aligned}
Dig(r_1') &= Dig(a_1) \cup Dig((a_2 \mid b_3)^* b_4) \cup Fin(a_1)Ini((a_2 \mid b_3)^* b_4) \\
&= \emptyset \cup Dig((a_2 \mid b_3)^*) \cup Dig(b_4) \cup Fin((a_2 \mid b_3)^*)Ini(b_4) \cup \{a_1\}(Ini((a_2 \mid b_3)^*) \cup Null((a_2 \mid b_3)^*)Ini(b_4)) \\
&= Dig(a_2 \mid b_3) \cup Fin(a_2 \mid b_3)Ini(a_2 \mid b_3) \cup \emptyset \cup Fin(a_2 \mid b_3)\{b_4\} \cup \{a_1\}(Ini(a_2 \mid b_3)) \cup \{b_4\}) \\
&= Dig(a_2) \cup Dig(a_3) \cup (Fin(a_2) \cup Fin(b_3))(Ini(a_2) \cup Fin(b_3)) \cup (Fin(a_2) \cup Fin(b_3))\{b_4\} \cup \{a_1\}(Ini(a_2) \cup Ini(b_3) \cup \{b_4\}) \\
&= \emptyset \cup \emptyset \cup (\{a_2\} \cup \{b_3\})(\{a_2\} \cup \{b_3\}) \cup (\{a_2\} \cup \{b_3\})\{b_4\} \cup \{a_1\}(\{a_2\} \cup \{b_3\} \cup \{b_4\}) \\
&= \{a_2, b_3\}\{a_2, b_3\} \cup \{a_2, b_3\}\{b_4\} \cup \{a_1\}\{a_2, b_3, b_4\} \\
&= \{a_2a_2, a_2b_3, b_3a_2, b_3, b_3\} \cup \{a_2b_4, b_3b_4\} \cup \{a_1a_2, a_1b_3, a_1b_4\} \\
&= \{a_2a_2, a_2b_3, b_3a_2, b_3, b_3, a_2b_4, b_3b_4, a_1a_2, a_1b_3, a_1b_4\}
\end{aligned}
$$

Conjunto Fol:

	Followers
a_1	a_1, b_3, b_4
a_2	a_2, b_3, b_4
b_3	a_2, b_3, b_4
b_4	$\dashv$

Se calcula ahora el autómata y se obtiene:



(20 %) 2. **Autómata de pila no determinista**

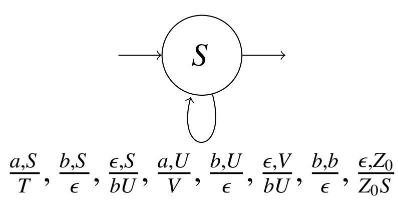
Construir un autómata de pila de la siguiente gramática independiente de contexto G_2, utilizando el algoritmo de transformación de gramáticas independientes de contexto en autómatas de pila:

$$
\begin{aligned}
S &\rightarrow aT \mid b \\
T &\rightarrow Ub \\
U &\rightarrow aV \mid b \\
V &\rightarrow Ub
\end{aligned}
$$

Solución:

	Regla	**Movimiento**
1	$S \rightarrow aT$	Si $cc = a \wedge top = S$ entonces $pop; push(T); shift$
2	$S \rightarrow b$	Si $cc = b \wedge top = S$ entonces $pop; shift$
3	$S \rightarrow Ub$	Si $top = S$ entonces $pop; push(bU);$
4	$U \rightarrow aV$	Si $cc = a \wedge top = U$ entonces $pop; push(V); shift$
5	$U \rightarrow b$	Si $cc = b \wedge top = U$ entonces $pop; shift$
6	$V \rightarrow Ub$	Si $top = V$ entonces $pop; push(bU);$
7		Si $cc = b \wedge top = b$ entonces $pop; shift$
8		Si $cc = \dashv \wedge empty(stack) = true$ entonces $accept; halt$

El siguiente es el automáta de pila obtenido:



(40 %) 3. **Obtención de una gramática** $LL(1)$

La siguiente gramática G_3 no cumple con la condición $LL(1)$:

$$
\begin{array}{rcl}
\text{regexpr} & \rightarrow & \text{regexpr}\ '+'\ \text{rterm} \mid \text{rterm} \\
\text{rterm} & \rightarrow & \text{rterm}\ \text{rfactor} \mid \text{rfactor} \\
\text{rfactor} & \rightarrow & \text{rfactor}\ '*' \mid \text{rprimario} \\
\text{rprimario} & \rightarrow & 'a' \mid' b'
\end{array}
$$

a) Transforme la gramática G_3 en una gramática G_3' que cumpla con la condicción $LL(1)$.

b) Demuestre que la nueva gramática G_3' cumple con la condición $LL(1)$.

Solución:

a) Esta gramática tiene recursividad por la izquierda. Observemos las siguientes roducciónes

$$\text{rfactor} \rightarrow \text{rfactor}\ '*' \mid \text{rprimario}$$

Al eliminar la recursividad por la izquierda, obtenemos:

$$
\begin{array}{rcl}
\text{rfactor} & \rightarrow & \text{rprimario} \mid \text{rprimario rfactor}' \\
\text{rfactor}' & \rightarrow & '*' \mid' *'\ \text{rfactor}'
\end{array}
$$

Estas producciones las factorizamos y obtenemos:

$$
\begin{array}{rcl}
\text{rfactor} & \rightarrow & \text{rprimario rfactor}'' \\
\text{rfactor}' & \rightarrow & '*'\ \text{rfactor}'' \\
\text{rfactor}'' & \rightarrow & \epsilon \mid \text{rfactor}'
\end{array}
$$

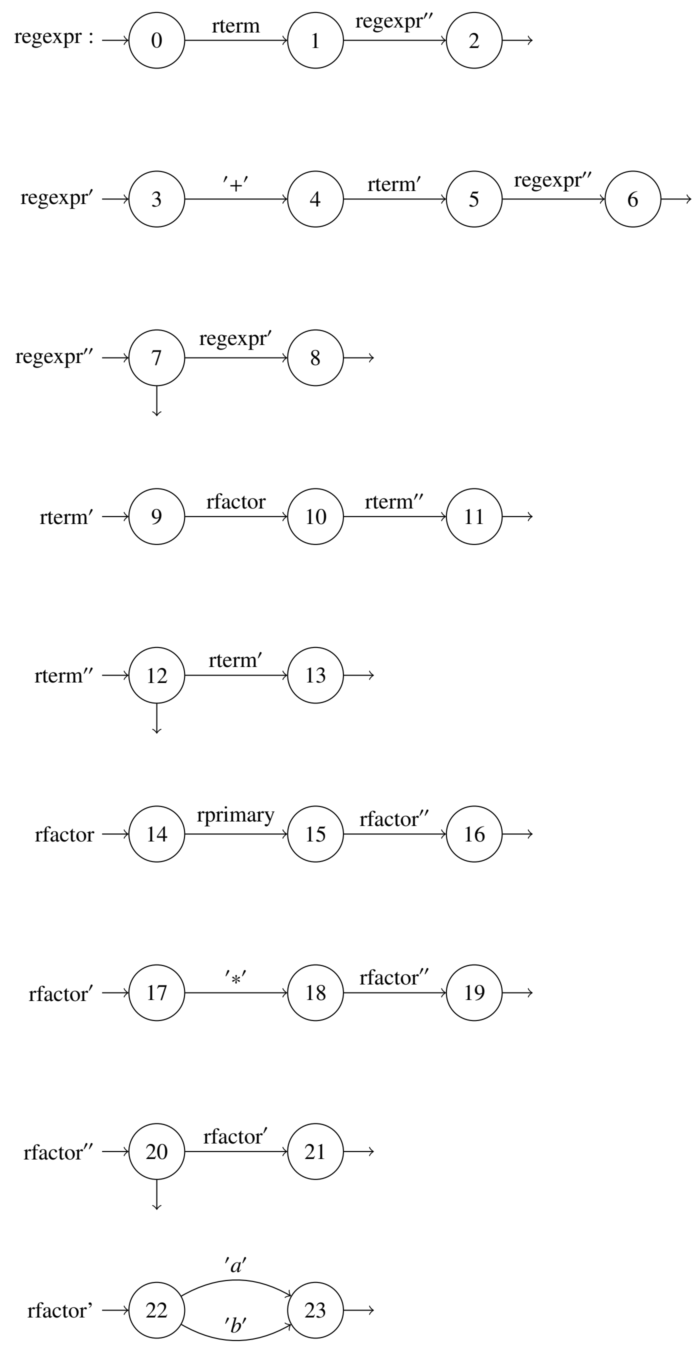
Observe que el mismo procedimiento se debe hacer para las otra dos producciones regexpr y rterm.

$$
\begin{array}{rcl}
\text{regexpr} & \rightarrow & \text{rterm regexpr}'' \\
\text{regexpr}' & \rightarrow & '+'\ \text{rterm regexpr}'' \\
\text{regexpr}'' & \rightarrow & \epsilon \mid \text{regexpr}' \\
\text{rterm} & \rightarrow & \text{rfactor rterm}'' \\
\text{rterm}' & \rightarrow & \text{rfactor rterm}'' \\
\text{rterm}'' & \rightarrow & \epsilon \mid \text{rterm}'
\end{array}
$$

Si se oberva bien las partes derechas de los no-terminales rterm y rterm′ son iguales, por lo tanto debemos sustituir una de las dos para simplificar más la gramática. Se elige sustituir a rterm por rterm′ y se obtiene finalmente la siguiente gramática:

$$
\begin{array}{rcl}
\text{regexpr} & \rightarrow & \text{rterm regexpr}'' \\
\text{regexpr}' & \rightarrow & '+'\ \text{rterm' regexpr}'' \\
\text{regexpr}'' & \rightarrow & \epsilon \mid \text{regexpr}' \\
\text{rterm}' & \rightarrow & \text{rfactor rterm}'' \\
\text{rterm}'' & \rightarrow & \epsilon \mid \text{rterm}' \\
\text{rfactor} & \rightarrow & \text{rprimario rfactor}'' \\
\text{rfactor}' & \rightarrow & '*'\ \text{rfactor}'' \\
\text{rfactor}'' & \rightarrow & \epsilon \mid \text{rfactor}' \\
\text{rprimary} & \rightarrow & 'a' \mid 'b'
\end{array}
$$

b) Construimos la red de máquinas:



Observamos donde están las bifurcaciones para comprobar si los conjuntos *Gui* son disjuntos. Primer caso en el estado 22:

$$
\begin{aligned}
Gui(22 \xrightarrow{'a'} 23) &= \{'a'\} \\
Gui(22 \xrightarrow{'b'} 23) &= \{'b'\}
\end{aligned}
$$

Ambos conjuntos son dijuntos. En el estado 20:

$$\begin{aligned}
Gui(20 \xrightarrow{\text{rfactor}'} 21) &= Ini(L(17)L(21)) \\
&= Ini(L(\text{rfactor}')) \\
&= '*' \\
Gui(20 \rightarrow) &= Fol(\text{rfactor}'') \\
&= Fol(\text{rfactor}') \cup Fol(\text{rfactor}) \\
&= \emptyset \cup Ini(10) \\
&= Ini(12) \\
&= Ini(9) \\
&= \{'a',' b'\}
\end{aligned}$$

Se observan que ambos conjuntos son disjuntos. En el estado 12:

$$\begin{aligned}
Gui(12 \xrightarrow{\text{rterm}'} 13) &= Ini(L(12)L(13)) \\
&= Ini(9) \\
&= \{'a',' b'\} \\
Gui(12 \rightarrow) &= Fol(\text{rterm}'') \\
&= Fol(\text{rterm}') \\
&= Ini(5) \\
&= \{'+'\}
\end{aligned}$$

Ambos conjuntos son disjuntos. En el estado 7:

$$\begin{aligned}
Gui(7 \xrightarrow{\text{regexpr}'} 8) &= \{'+'\} \\
Gui(7 \rightarrow) &= Fol(\text{regexpr}'') \\
&= Fol(\text{regexpr}) \\
&= \{'\dashv'\}
\end{aligned}$$

Ambos conjuntos son disjuntos. Por lo tanto afirmamos que la nueva gramática cumple con la condición $LL(1)$.

(20 %) 4. **Implementación de procedimientos recursivos**
La siguiente gramática G_4 cumple con la condición $LL(1)$

$$\begin{aligned} A &\rightarrow baA' \mid aA' \\ A' &\rightarrow A \mid \epsilon \end{aligned}$$

Construir analizador recursivo descendente basado en procedimientos para ésta gramática.

Solución:

```
proc A () {
   if (cc == b) {
      cc = getNext();
      if (cc == a) {
         cc = getNext();
         call A'()
      }
   }
   else if (cc == a)  {
      cc = getNext();
      call A'();
   }
   else {
      error();
   }
}

proc A'() {
   if (cc == a || cc == b) {
      cc = getNext();
      call A();
   }
   else if (cc == ⊣) {
      return;
   }
   else {
      error();
   }
}
```

Donde `error` es un procedimiento que termina la ejecución del analizador informando de un error.

ST0270 Lenguajes formales y compiladores | **Universidad EAFIT**
Cuarto parcial | **DIS**
Juan Francisco Cardona McCormick | **1 de noviembre de 2012**

Nombre: ______________________________

Código: ______________________________

Importante: Toda respuesta debe estar sustentada por un procedimiento formal. No se aceptan respuestas sin su correspondiente procedimiento. El parcial debe ser el resultado de un trabajo individual, no grupal.

(40 %) 1. **Construcción de pilot** $LR(0)$

Dada la siguiente gramática G_1:

$$\begin{array}{rcl} S_0 & \rightarrow & SheepNoise \dashv \\ SheepNoise & \rightarrow & baa\ \ SheepNoise \\ & | & baa \end{array}$$

a) Construir un piloto para $LR(0)$.

b) A partir del pilot demostrar si la gramática es $LR(0)$. Si la gramática G_1 no cumple con la condición $LR(0)$ mostrar cuales son los estados que no cumplen con dicha condició y por qué.

c) Si la gramática es $LR(0)$ probar que la siguiente cadena es una cadena válida: $x = baabaabaa \dashv$

(60 %) 2. **Construcción de pilotos** $LR(1)$.

Dada la siguiente gramática G_2:

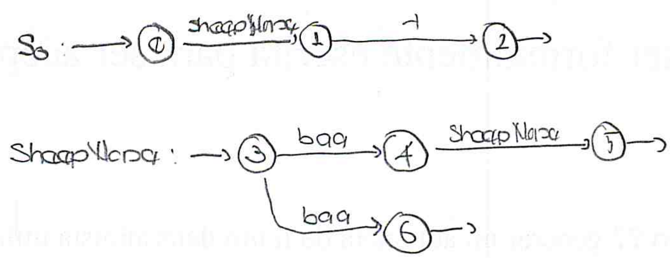
$$\begin{array}{rcl} S_0 & \rightarrow & Stmt\ \dashv \\ Stmt & \rightarrow & 'if'\ expr\ 'then'\ Stmt \\ & | & 'if'\ expr\ 'then'\ Stmt\ 'else'\ Stmt \\ & | & assign \end{array}$$

a) Construir un piloto $LR(1)$.

b) Determinar si la gramática anterior es $LR(1)$ o no. Si no lo es mostrar cuáles estados y por que no cumple con la condición $LR(1)$.

c) Si la gramática es $LR(1)$ probar que la siguiente cadena es una cadena válida: $x = if\ expr\ then\ if\ expr\ then\ assign\ else\ assign \dashv$

1



I_0 = closure(0) = $\cancel{\text{closure}(\{0\})}$ = {0,3}.

$\vartheta(I_0, \text{SheepNoise}) = \text{closure}(\delta(0,\text{SheepNoise}) \cup \delta(3,\text{SheepNoise}))$
$= \text{closure}(\{1\} \cup \{\})$
$= \text{closure}(\{1\}) = \{1\} = I_1$

$\vartheta(I_0, \text{baa}) = \text{closure}(\delta(0,\text{baa}) \cup \delta(3,\text{baa}))$
$= \text{closure}(\{\} \cup \{4,6\})$
$= \text{closure}(\{4,6\}) = \{4,6,3\}. = I_2$

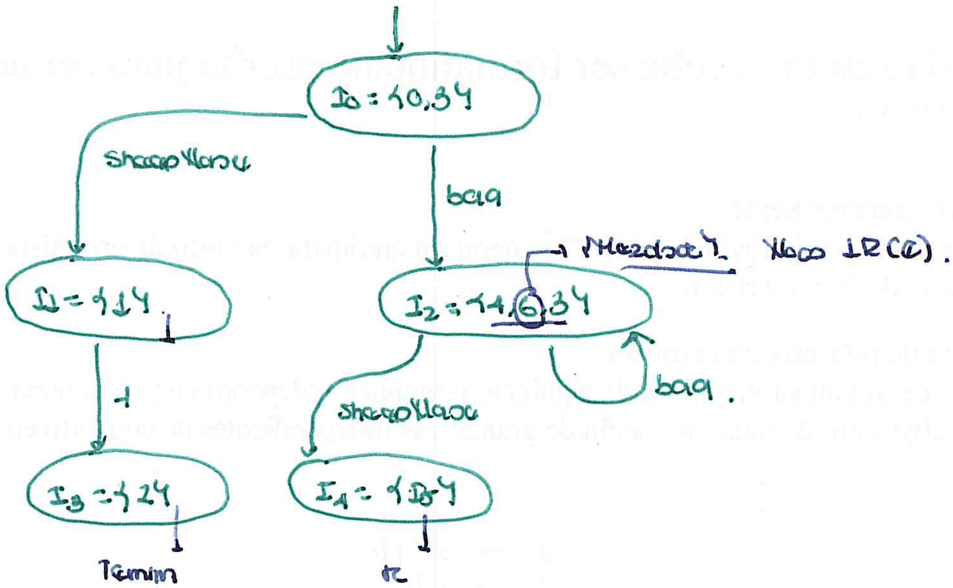
$\vartheta(I_1, \dashv) = \text{closure}(\delta(1,\dashv)) = \text{closure}(\{2\}) = \{2\} = I_3$

$\vartheta(I_2, \text{SheepNoise}) = \text{closure}(\delta(3,\text{SheepNoise}) \cup \delta(4,\text{SheepNoise}) \cup \delta(6,\text{SheepNoise}))$
$= \text{closure}(\{\} \cup \{5\} \cup \{\})$
$= \text{closure}(\{5\}) = \{5\} = I_4$

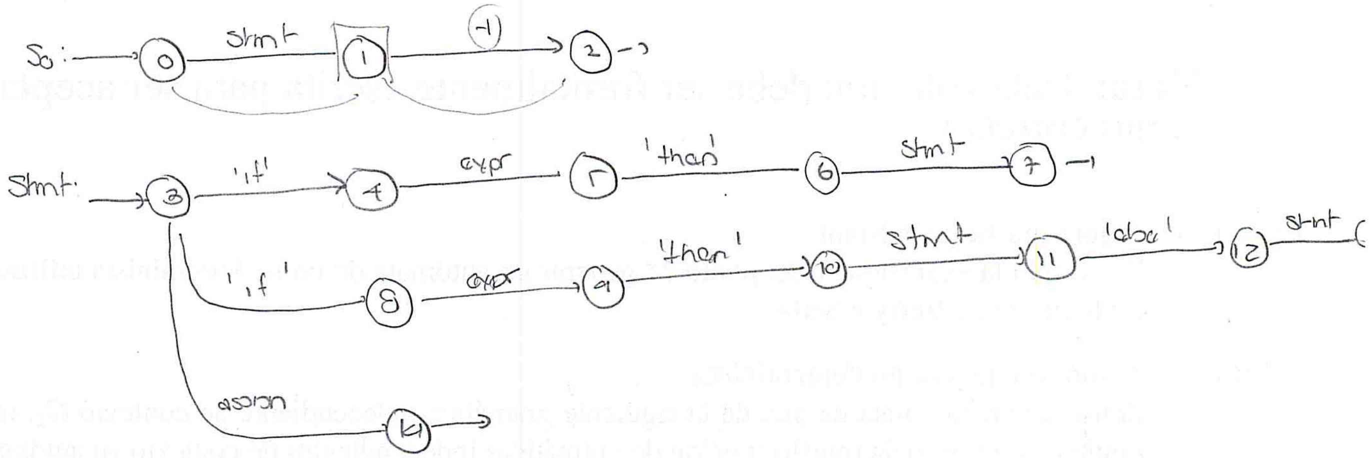
$\vartheta(I_2, \text{baa}) = \text{closure}(\delta(3,\text{baa}) \cup \delta(4,\text{baa}) \cup \delta(6,\text{baa}))$
$= \text{closure}(\{4,6\}) = I_2$

$\vartheta(I_3, ..) =$

$\vartheta(I_4 ...) = ..$


baq
baq
baq
Tamim

2



$$\text{closure}_1(\{(0,\dashv)\}) = \{(0,\dashv),(3,\dashv)\} = I_0$$

$$\vartheta(I_0, \text{Stmt}) = \text{closure}_1((\delta(0,\text{Stmt}),\dashv) \cup (\delta(3,\text{Stmt}),\dashv))$$
$$= \text{closure}_1(\{(1,\dashv)\} \cup \{\})$$
$$= \text{closure}_1(\{(1,\dashv)\}) = \{1,\dashv\} = I_1$$

$$\vartheta(I_0,\text{'if'}) = \text{closure}_1((\delta(0,\text{'if'}),\dashv) \cup (\delta(3,\text{'if'}),\dashv))$$
$$= \text{closure}_1(\{\} \cup (\{4,8\},\dashv))$$
$$= \text{closure}_1(\{(4,\dashv),(8,\dashv)\}) = \boxed{\{(4,\dashv),(8,\dashv)\}} = I_2$$

$$\vartheta(I_0,\text{'assign'}) = \text{closure}_1((\delta(3,\text{'assign'}),\dashv)) =$$
$$= \text{closure}_1(\{(14,\dashv)\}) = \{(14,\dashv)\} = I_3$$

$$\boxed{\vartheta(I_1,\dashv) = \text{closure}_1((\delta(1,\dashv),\dashv)) = \text{closure}_1((2,\dashv)) = \{(2,\dashv)\} = I_4}$$

$$\vartheta(I_2, \text{expr}) = \text{closure}_1((\delta(4,\text{expr}),\dashv) \cup (\delta(8,\text{expr}),\dashv))$$
$$= \text{closure}_1(\{(5,\dashv)\} \cup \{(9,\dashv)\})$$
$$= \text{closure}_1(\{(5,\dashv),(9,\dashv)\}) = \{(5,\dashv),(9,\dashv)\} = I_5$$

$\cancel{\vartheta(I_3}$

$\underline{\underline{\vartheta(I_5, \text{'then'})}} = \cancel{\text{closure}_1\{\ldots\text{'then'}\}}$

$= \text{closure}_1((\delta(5, \text{'then'}), +) \cup (\delta(9, \text{'then'}), +))$

$= \text{closure}_1(\{(6,+)\} \cup \{(10,+)\})$

$= \{(6,+), (3,+), (10,+), \boxed{(3, \text{'else'})}\} = I_6$

$\vartheta(I_6, \text{stmt}) = \text{closure}_1((\delta(6,\text{stmt}),+) \cup (\delta(3,\text{stmt}), \overset{\{else,+\}}{+}) \cup (\delta(10,\text{stmt}, +))$

$= \text{closure}(\{(7,+)\} \cup \{\} \cup \{(11,+)\})$

$= \{(7,+),(11,+)\} = I_7$

$\vartheta(I_6, \text{'if'}) = \text{closure}_1((\delta(3,\text{'if'}),\{else,+\}) \cup (\delta(6,\text{'if'}),+) \cup (\delta(10,\text{'if'}),+)\}.$

$= \text{closure}_1(\{(4,\{else,+\}), (8,\{else,+\})\} \cup \{\} \cup \{\}\}$

$= \cancel{\text{closure}_1}(\{(4,else,+\}, (8,\{else,+\})\}. = I_8$

$\vartheta(I_6, \text{'assign'}) = \text{closure}_1((\delta(6,\text{assign}),+) \cup (\delta(3,\text{assign}),\{+,\text{'else'}\})$
$\cup (\delta(10,\text{assign}),\{+,\text{'else'}\}))$

$= \text{closure}_1(\{\} \cup \{(14,\{+,\text{'else'}\})\})$

$= \cancel{\text{closure}_1}\{(14,\{+,\text{'else'}\})\} \leftarrow = I_9$

$\vartheta(I_7, \text{'else'}) = \text{closure}_1((\delta(7,\text{'else'}),+) \cup (\delta(11,\text{'else'}),+))$

$= \text{closure}_1(\{\} \cup \boxed{\{(12,+)\}.})$

$= \text{closure}(\{(12,+)\}) = \{(12,+),(3,+)\}. = I_{10}$

$\vartheta(I_8, \text{expr}) = \text{closure}_1((\delta(4,\text{expr}),\{else,+\}) \cup (\delta(8,\text{expr}),\{else,+\}))$

$= \text{closure}_1((5,\{else,+\}) \cup \{(9,\{else,+\})\})$

$= \text{closure}_1(\{(5,\{else,+\}),(9,\{else,+\})\})$

$= \{(5,\{else,+\}),(9,\{else,+\})\}. = I_{11}$

$$\vartheta(I_{10}, \mathrm{Stmt}) = \mathrm{closure}_1((\delta(12, \mathrm{stmt}), \dashv) \cup (\delta(3, \mathrm{Stmt}), \dashv))$$
$$= \mathrm{closure}_1(\{(13, \dashv)\} \cup \{\})$$
$$= \mathrm{closure}_1(\{(13, \dashv)\}) = \{(13, \dashv)\} = I_{12}.$$

$$\vartheta(I_{10}, \text{'if'}) = \mathrm{closure}_1((\delta(12, \text{'if'}), \dashv) \cup (\delta(3, \text{'if'}), \dashv))$$
$$= \mathrm{closure}_1(\{\} \cup \{(4, \dashv), (8, \dashv)\})$$
$$= \mathrm{closure}_1(\{(4, \dashv), (8, \dashv)\})$$
$$= \{(4, \dashv), (8, \dashv)\} = I_2$$

$$\vartheta(I_0, \text{'assign'}) = \mathrm{closure}_1((\delta(12, \text{'assign'}), \dashv) \cup (\delta(3, \mathrm{assign}), \dashv))$$
$$= \mathrm{closure}_1(\{\} \cup \{(14, \dashv)\}) = \{(14, \dashv)\} = I_3$$

$$\vartheta(I_{11}, \text{'then'}) = \mathrm{closure}_1((\delta(5, \text{'then'}), \{\mathrm{else}, \dashv\}) \cup (\delta(9, \text{'then'}), \{\mathrm{else}, \dashv\}))$$
$$= \mathrm{closure}_1(\{(6, \{\mathrm{else}, \dashv\})\} \cup \{(10, \{\mathrm{else}, \dashv\})\})$$
$$= \mathrm{closure}_1(\{(6, \{\mathrm{else}, \dashv\}), (10, \{\mathrm{else}, \dashv\})\})$$
$$= \{(6, \{\mathrm{else}, \dashv\}), (10, \{\mathrm{else}, \dashv\}), (3, \{\mathrm{else}, \dashv\})\} = I_{13}$$

$$\vartheta(I_{13}, \mathrm{Stmt}) = \mathrm{closure}_1((\delta(6, \mathrm{stmt}), \{\mathrm{else}, \dashv\}) \cup (\delta(10, \mathrm{Stmt}), \{\mathrm{else}, \dashv\}) \cup (\delta(3, \mathrm{Stmt}), \{\mathrm{else}, \dashv\}))$$
$$= \mathrm{closure}_1(\{(7, \{\mathrm{else}, \dashv\})\} \cup \{(11, \{\mathrm{else}, \dashv\})\})$$
$$= \{(7, \{\mathrm{else}, \dashv\}), (11, \{\mathrm{else}, \dashv\})\} = I_{14}$$

$$\vartheta(I_{13}, \text{'if'}) = \mathrm{closure}_1((\delta(6, \text{'if'}), \{\mathrm{else}, \dashv\}) \cup (\delta(10, \text{'if'}), \{\mathrm{else}, \dashv\}) \cup (\delta(3, \text{'if'}), \{\mathrm{else}, \dashv\}))$$
$$= \mathrm{closure}_1(\{\} \cup \{\} \cup \{(4, \{\mathrm{else}, \dashv\}), (8, \{\mathrm{else}, \dashv\})\}) = I_8$$

$$\vartheta(I_{13}, \mathrm{assign}) = \mathrm{closure}_1((\delta(6, \mathrm{assign}), \{\mathrm{else}, \dashv\}) \cup (\delta(10, \mathrm{assign}), \{\mathrm{else}, \dashv\}) \cup (\delta(3, \mathrm{assign}), \{\mathrm{else}, \dashv\}))$$
$$= \mathrm{closure}_1(\{\} \cup \{\} \cup \{(14, \{\mathrm{else}, \dashv\})\}) = I_9$$

$$\vartheta(I_{14}, \text{'else'}) = \mathrm{closure}_1((\delta(7, \text{'else'}), \{\mathrm{else}, \dashv\}) \cup (\delta(11, \text{'else'}), \{\mathrm{else}, \dashv\}))$$
$$= \mathrm{closure}_1(\{\} \cup \{(12, \{\mathrm{else}, \dashv\})\})$$
$$= \{(12, \{\mathrm{else}, \dashv\}), (3, \{\mathrm{else}, \dashv\})\} = I_{15}$$

$$\theta(I_{17}, Stmt) = closure((\delta(12, Stmt), \{else, \dashv\}) \cup (\delta(3, Stmt), \{else, \dashv\}))$$
$$= closure(\{(13, \{else, \dashv\})\} \cup \{\})$$
$$= \{(13, \{else, \dashv\})\} = I_{16}$$

$$\theta(I_{17}, 'if') = closure((\delta(12, 'if'), \{else, \dashv\}) \cup (\delta(3, 'if'), \{else, \dashv\}))$$
$$= closure(\{\} \cup \{(4, \{else, \dashv\}), (8, \{else, \dashv\})\}) = I_{8}$$

$$\theta(I_{17}, assign) = closure((\delta(12, assign), \{else, \dashv\}) \cup (\delta(3, assign), \{else, \dashv\}))$$
$$= closure(\{\} \cup \{(14, \{else, \dashv\})\}) =$$
$$= \{(14, \{else, \dashv\})\} = I_{9}$$

ST0270 Lenguajes formales y compiladores - 031 | **Universidad EAFIT**
Primer Parcial | **DIS**
Juan Francisco Cardona McCormick | **21 de febrero de 2013**

Nombre: ______________________________

Código: ______________________________

(20 %) 1. **Expresiones regulares**
Dado el alfabeto $\Sigma = \{\boxdot, \odot\}$. Escriba una expresión regular r_1 que defina el siguiente lenguaje: Todas las cadenas comienzan con $\boxdot$ y terminan siempre con $\odot$ y tiene al menos tres $\boxdot$ no necesariamente seguidas.

> **Solución:**
>
> $$r_1 = \boxdot(\boxdot \mid \odot)^* \boxdot (\boxdot \mid \odot)^* \boxdot (\boxdot \mid \odot)^* \odot$$

(20 %) 2. **Gramáticas independientes de contexto**
Escribir la gramática independiente de contexto G_2 que genere el siguiente lenguaje L_2:

$$L_2 = \{a^n b^{n\times 2} a^{k\times 3} b^k \mid n \geq 1, k \geq 0\}$$

> **Solución:**
> Se observa que la gramática es la concatentación de dos lenguajes: $L_2' L_2''$. Donde L_2' es:
>
> $$L_2' = \{a^n b^{n\times 2} \mid n \geq 1\}$$
>
> y L_2'' es:
>
> $$L_2'' = \{a^{k\times 3} b^k \mid k \geq 0\}$$
>
> Ahora definimos el lenguaje para la primera gramática:
>
> $$G_2' = (V = \{A\}, \Sigma = \{a, b\}, P = \{A \to aAbb \mid abb\}, A)$$
>
> Ahora definamos el lenguaje para la segunda gramática:
>
> $$G_2'' = (V = \{B\}, \Sigma = \{a, b\}, P = \{B \to aaaBb \mid \epsilon\}, B)$$
>
> Ahora contatenmos las dos gramáticas:
>
> $$G_2 = (V = \{A, B\}, \Sigma = \{a, b\}, P = \{S \to AB, A \to aAbb \mid abb, B \to aaaBb \mid \epsilon\}, S)$$

(20 %) 3. **Ambigüedad**

a) (10 %) Mostrar la ambigüedad en la siguiente gramática independiente de contexto G_3:

$$G_3 = (V = \{A, B\}, \Sigma = \{a, b, c, d\}, P = \{A \to ABA \mid a, B \to b \mid c \mid d, A)$$

Solución:
Observamos que la gramática tiene un problema de ambigüedad en la primera producción, dado que esta utilizando el no-terminal como B operador y A no tiene un orden de derivación, se sabe que tiene problemas de ambigüedad. Observemos la cadena *abaca*, genera a través de dos derivaciones distintas dos árboles de derivación diferentes.

Primera derivación:

$$\begin{aligned} A &\Rightarrow ABA \\ &\Rightarrow ABABA \\ &\Rightarrow aBABA \\ &\Rightarrow abABA \\ &\Rightarrow abaBA \\ &\Rightarrow ababA \\ &\Rightarrow ababa \end{aligned}$$

Segunda derivación:

$$\begin{aligned} A &\Rightarrow ABA \\ &\Rightarrow aBA \\ &\Rightarrow abA \\ &\Rightarrow abABA \\ &\Rightarrow abaBA \\ &\Rightarrow ababA \\ &\Rightarrow ababa \end{aligned}$$

Esto genera dos árboles de derivación distintos para la misma cadena por lo tanto la gramática es ambigua.

b) (10 %) Mostrar la ambigüedad en la siguiente expresión regular r_3 con alfabeto $\Sigma = \{a, b\}$:

$$r_3 = a^*(a \mid b)^*$$

Solución:
La cadena *aa* puede ser obtenidas por dos derivaciones distintas. Redefinamos la anterior expresion regular y definamos una nueva numerada:

$$r'_3 = a_1^*(a_2 \mid b_3)^*$$

Derivemos la cadena *aa* a partir de la nueva expresion regular.

Primera derivación:

$$\begin{aligned} a_1^*(a_2 \mid b_3)^* &\Rightarrow (a_1)^2(a_2 \mid b_3)^* \\ &\Rightarrow (a_1)^2\epsilon \equiv a_1a_1 \end{aligned}$$

Segunda derivación:

$$\begin{aligned} a_1^*(a_2 \mid b_3)^* &\Rightarrow (a_1)^1(a_2 \mid b_3)^* \\ &\Rightarrow (a_1)(a_2 \mid b_3)^1 \\ &\Rightarrow (a_1)(a_2) \equiv a_1a_2 \end{aligned}$$

Eliminando los número de la primera y segunda derivación obtenemos: aa. Por lo tanto la expresion regular es ambigua.

(20 %) 4. **Transformar gramáticas**
Transforma la siguiente gramática independiente de contexto G_4 a una nueva gramática G_4' en forma normal Chomsky:

$$G_4 = (V = \{E\}, \Sigma = \{+, \times, i\}, P = \{E \to E + E \mid E \times E \mid i\}, E)$$

Solución:

La gramática no cumple con la forma normal Chomsky por que en primer lugar el axioma está en el lado derecho de la producción. Transformemos la gramática eliminando el axioma del lado derecho de todas la producciones generando una nueva gramática G_4''.

$$G_4'' = (V = \{S, E\}, \Sigma = \{+, \times, i\}, P = \{S \to E, E \to E + E \mid E \times E \mid i\}, S)$$

Todavía la gramática nueva G_4''. Tiene problemas por que no está en forma binaria homogenea. Transformamos la gramática G_4'' en la nueva gramática G_4''' añadiendo un nuevo no-terminal al lado derecho del primera producción.

$$G_4''' = (V = \{S, E, C\}, \Sigma = \{+, \times, i\}, P = \{S \to EC, E \to E + E \mid E \times E \mid i, C \to \epsilon\}, S)$$

Todavía la gramática nueva G_4''' tiene producciones que no están forma binaria homogenea. Transformamos la gramática G_4''' en la nueva gramática G_4'''' convirtiendo las producciones con más de dos elementos a dos.

$$G_4'''' = (V = \{S, E, E', E'', C\}, \Sigma = \{+, \times, i\}, P = \{S \to EC, E \to EE' \mid EE'' \mid i, E' \to +E, E'' \to \times E, C \to \epsilon\}, S)$$

Todavía la gramática nueva G_4'''' no cumple debido a que existen terminales en las reglas binarias homogeneas. Transformemos a la gramática G_4'''' eliminando los terminales de las reglas binarias y produciendo nuevos no terminales y obtenemos la gramática G_4':

$$G_4' = (V = \{S, E, E', E'', C, \langle + \rangle, \langle \times \rangle\}, \Sigma = \{+, \times, i\}, P = \{S \to EC, E \to EE' \mid EE'' \mid i, E' \to \langle + \rangle E, E'' \to \langle \times \rangle E,$$

(20 %) 5. **Gramática limpia**
Transformar la siguiente gramática G_5 en una gramática independiente de contexto limpia G'_5:

$$G_5 = (V = \{A, B, C, D, E\}, \Sigma = \{a, b, c, d\},$$
$$P = \{A \to BC \mid AD, B \to aBdd \mid add, C \to ccCb \mid \epsilon, D \to DA, E \to dE \mid d\}, A)$$

Solución:

En primer lugar verificamos si la gramática está bien definida:

$$def_0 = \{B, C, E\}$$

Este es el conjunto inicial de los no-terminales que producen terminales. Iteramos para obtener el siguiente nivel.

$$def_1 = def_0 \cup \{A\} = \{A, B, C, E\}$$

Iteramos de nuevo y obtenemos:

$$def_2 = def_1 \cup \varnothing = \{A, B, C, E\}$$

Ahora encontramos los no-terminales que no están definidos:

$$nodef = V \setminus def = \{A, B, C, D, E\} \setminus \{A, B, C, E\} = \{D\}$$

Generamos una nueva gramática G''_5 elimando el no-terminal D

$$G''_5 = (V = \{A, B, C, E\}, \Sigma = \{a, b, c, d\}, P = \{A \to BC \mid AD, B \to aBdd \mid add, C \to ccCb \mid \epsilon, E \to dE \mid d\}, A)$$

Ahora verificamos que no-terminales son alcanzables desde el axioma:

$$\{B, C\}$$

Por lo tanto generamos una nueva gramática G'_5 eliminando el no-terminal E:

$$G''_5 = (V = \{A, B, C\}, \Sigma = \{a, b, c, d\}, P = \{A \to BC \mid AD, B \to aBdd \mid add, C \to ccCb \mid \epsilon\}, A)$$

Nombre: ______________________________
Código: ______________________________

(20 %) 1. **Expresiones regulares**
Dado el alfabeto $\Sigma_1 = \{\lambda, \gamma, \phi\}$. Escriba una expresión regular r_1 que defina el siguiente lenguaje: cada carácter λ debe ir acompañanda de al menos dos pares de carácteres $\gamma\phi$ no necesariamente consecutivos.

Solución:

$$r_1 = (\gamma \mid \phi)^*(\lambda(\gamma \mid \phi)^*(\gamma\phi)(\gamma \mid \phi)^*(\gamma\phi))^*(\gamma \mid \phi)^*$$

(20 %) 2. **Gramáticas independientes de contexto**
Escribir la gramática independiente de contexto G_2 que genere el siguiente lenguaje L_2:

$$L_2 = \{a^{k\times 2}b^k a^n b^{n\times 3} \mid k \geq 1, n \geq 0\}$$

Solución:
Se observa que la gramática es la concatentención de dos lenguajes: $L_2'L_2''$. Donde L_2' es:

$$L_2' = \{a^{k\times 2}b^k \mid k \geq 1\}$$

y L_2'' es:

$$L_2'' = \{a^n b^{n\times 3} \mid k \geq 0\}$$

Ahora definimos el lenguaje para la primera gramática:

$$G_2' = (V = \{A\}, \Sigma = \{a, b\}, P = \{A \to aaAb \mid aab\}, A)$$

Ahora definamos el lenguaje para la segunda gramática:

$$G_2'' = (V = \{B\}, \Sigma = \{a, b\}, P = \{B \to aBbbb \mid \epsilon\}, B)$$

Ahora contatenmos las dos gramáticas:

$$G_2 = (V = \{A, B\}, \Sigma = \{a, b\}, P = \{S \to AB, A \to aaAb \mid aab, B \to aBbbb \mid \epsilon\}, S)$$

(20 %) 3. **Ambigüedad**

a) (10 %) Mostrar la ambigüedad en la siguiente gramática independiente de contexto G_3:

$G_3 = (V = \{D, T, K\}, \Sigma = \{k, f, l, s, a\}, P = \{D \to DKD \mid DTD \mid a, K \to k \mid f, T \to l \mid s, D)$

Solución:
Observamos que la gramática tiene un problema de ambigüedad en la primera producción, dado que esta utilizando los no-terminales K y T como operadores y D no tiene un orden de derivación, se sabe que tiene problemas de ambigüedad. Observemos la cadena $akafa$, genera a través de dos derivaciones distintas dos árboles de derivación diferentes.

Primera derivación:

$$\begin{aligned} D &\Rightarrow DKD \\ &\Rightarrow DKDKD \\ &\Rightarrow aKDKD \\ &\Rightarrow akDKD \\ &\Rightarrow akaKD \\ &\Rightarrow akafD \\ &\Rightarrow akafa \end{aligned}$$

Segunda derivación:

$$\begin{aligned} A &\Rightarrow DKD \\ &\Rightarrow aKD \\ &\Rightarrow akD \\ &\Rightarrow akDKD \\ &\Rightarrow akaKD \\ &\Rightarrow akafD \\ &\Rightarrow akafa \end{aligned}$$

Esto genera dos árboles de derivación distintos para la misma cadena por lo tanto la gramática es ambigua.

b) (10 %) Mostrar la ambigüedad en la siguiente expresión regular r_3 con alfabeto $\Sigma = \{a, b\}$:

$$r_3 = (a \mid b)^* a^*$$

Solución:
La cadena aa puede ser obtenidas por dos derivaciones distintas. Redefinamos la anterior expresion regular y definamos una nueva numerada:

$$r_3' = (a_1 \mid b_2)^* a_3^*$$

Derivemos la cadena aa a partir de la nueva expresion regular.

Primera derivación:

$$\begin{aligned} (a_1 \mid b_2)^* a_3^* &\Rightarrow (a_1 \mid b_2)^1 a_3^* \\ &\Rightarrow a_1 1 a_3^* \\ &\Rightarrow a_1 1 a_3^1 \equiv a_1 a_3 \end{aligned}$$

Segunda derivación:

$$\begin{aligned} (a_1 \mid b_2)^* a_3^* &\Rightarrow \epsilon a_3^* \\ &\Rightarrow a_3^2 \equiv a_3 a_3 \end{aligned}$$

Eliminando los número de la primera y segunda derivación obtenemos: *aa*. Por lo tanto la expresion regular es ambigua.

(20 %) 4. **Transformar gramáticas**
Transforma la siguiente gramática independiente de contexto G_4 en una gramática sin reglas de copia G_4'.

$$G_4 = (V = \{S, A, Q, L, I, K\},$$
$$\Sigma = \{\text{var}, =, \text{exp}, \text{while}, ;, \text{cond}, \text{do}, \text{if}, \text{then}, \text{else}, \text{fi}, \text{skip}\}$$
$$P = \{S \to A \mid Q \mid L \mid I \mid K,$$
$$A \to \text{var} = \text{exp}, Q \to S\,; S, L \to \text{while cond do } S, I \to \text{if cond then } S \text{ else } S \text{ fi}, K \to \text{skip}\},$$
$$S)$$

Solución:

En primer lugar calculamos el conjunto de copia, en la primera iteración obtenemos.

no-terminales	conjunto de copia
S	$\{S\}$
A	$\{A\}$
Q	$\{Q\}$
L	$\{L\}$
I	$\{I\}$
K	$\{K\}$

Calculamos todos los no-terminales que cumpla con la condición.

$$C \in Copy(A) \text{ sí } B \in Copy(A) \wedge (B \to C \in P)$$

Obtenemos:

no-terminales	conjunto de copia
S	$\{S, A, Q, L, I, K\}$
A	$\{A\}$
Q	$\{Q\}$
L	$\{L\}$
I	$\{I\}$
K	$\{K\}$

So observa que ya se llegó al punto fijo. Ahora nos queda una nueva gramática G'_4

$$
\begin{array}{lllllll}
G'_4 & = & (& V & = & \{ & S, A, Q, L, I, K \quad \}, \\
 & & & \Sigma & = & \{ & \text{var}, =, \text{exp}, \text{while}, ;, \text{cond}, \text{do}, \text{if}, \text{then}, \text{else}, \text{fi}, \text{skip} \quad \}, \\
 & & & P & = & \{ & S \to \text{var} \;=\; \text{exp}, \\
 & & & & & & S \to S ; S, \\
 & & & & & & S \to \text{while cond do } S, \\
 & & & & & & S \to \text{if cond then } S \text{ else } S \text{ fi}, \\
 & & & & & & S \to \text{skip}\}, \\
 & & & P & = & \{ & A \to \text{var} \;=\; \text{exp}, \\
 & & & & & & Q \to S ; S, \\
 & & & & & & L \to \text{while cond do } S, \\
 & & & & & & I \to \text{if cond then } S \text{ else } S \text{ fi}, \\
 & & & & & & K \to \text{skip}\}, \quad \}, \\
 & & & S & = & S & \\
 & &) & & & &
\end{array}
$$

(20 %) 5. **Recursividad por la izquierda**

Transformar la siguiente gramática G_5 en una gramática que elimine la recursividad por la izquierda G'_5:

$$G_5 = (V = \{A, B, C\}, \Sigma = \{a, b, c\}, P = \{A \to Abc \mid aA \mid BC, B \to c \mid bB, C \to Cc \mid c\}, A)$$

Solución:

Se observan dos producciones con recursividades inmediatas A y C. Identifiquemos las producciones en A que son recursivas de las que no:

$$\begin{array}{lcl} A & \to & Abc \\ A & \to & aA \mid BC \end{array}$$

Creamos una auxiliar A' y transformamos las producciones de A:

$$\begin{array}{lcl} A & \to & aAA' \mid BCA' \mid aA \mid BC \\ A' & \to & bcA' \mid bc \end{array}$$

Con el no-terminal C, identifiquemos las producciones recursivas de las que no lo son:

$$\begin{array}{lcl} C & \to & Cc \\ C & \to & c \end{array}$$

Ahora aplicamos el algoritmo de transformación de gramáticas recursivas por la izquierda:

$$\begin{array}{lcl} C & \to & cC' \mid c \\ C' & \to & cC' \mid c \end{array}$$

De lo anterior obtenemos la nueva gramática G'_5:

$$G_5 = (V = \{A, A', B, C, C'\}, \Sigma = \{a, b, c\}, P = \{, A \to aAA' \mid BCA' \mid aA \mid BC, A' \to bcA' \mid bc, B \to c \mid bB, C \to$$

Nombre: ____________________
Código: ____________________

Nota: La solución de los puntos del parcial debe ser formal.

1. De expresión regular a gramática independiente de contexto .... 20 puntos

Transformar la expresión regular $e_1 = b(a \cup b)^* a$ en una gramática independiente de contexto G_1, tal que $L(e_1) \equiv L(G_1)$.

Solución:

Se identifican las sub-expresiones básicas de la expresión regular e_1

$$\begin{array}{lcl} E_0 & \rightarrow & a \\ E_1 & \rightarrow & b \end{array}$$

Luego pasa al siguiente nivel $(a \cup b)$

$$\begin{array}{lcl} E_2 & \rightarrow & E_0 \\ & | & E_1 \end{array}$$

La siguiente sub-expresión $(a \cup b)^*$:

$$\begin{array}{lcl} E_3 & \rightarrow & \epsilon \\ & | & E_2 E_3 \end{array}$$

La sub-expresión $b(a \cup b)^* a$:

$$E_4 \rightarrow E_1 E_3 E_0$$

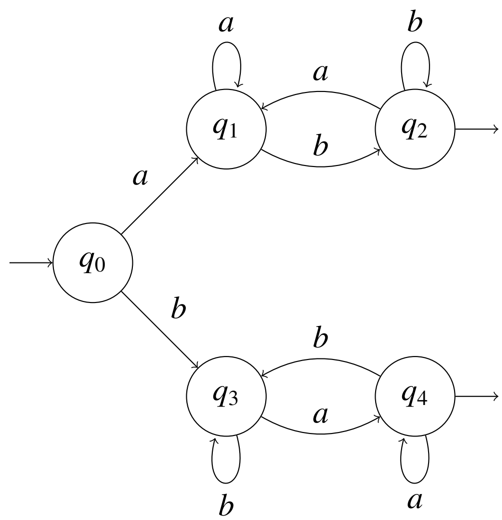
Finalmente la gramática G_1 se define como:

$$\begin{aligned} G_1 = (V = \{E_0, E_1, E_2, E_3, E_4\}, \\ \Sigma = \{a, b\}, \\ P = \{E_4 \rightarrow E_1 E_2 E_0, E_3 \rightarrow \epsilon \mid E_2 E_3, E_2 \rightarrow E_0 \mid E_1, E_1 \rightarrow b, E_2 \rightarrow a\}, \\ S = E_4) \end{aligned}$$

2. Autómata determinista . . . . 20 puntos

Construya una autómata determinista DFA_2 que reconozca el lenguaje L_2 sobre el alfabeto $\Sigma = \{a, b\}$ definido por el siguiente requisito: El primer símbolo por la izquierda es diferente del primero por la derecha. Se debe cumplir la condición $L_2 \equiv L(DFA_2)$.

Solución:



3. De autómata a expresión regular . . . . 20 puntos

En la figura 1 se encuentra un autómata finito determinista DFA_3. Encontrar la expresión regular e_3 que cumpla la siguiente condición $L(DFA_3) \equiv L(e_3)$.

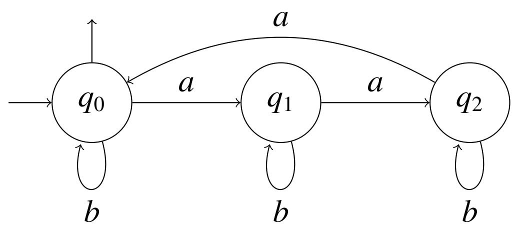
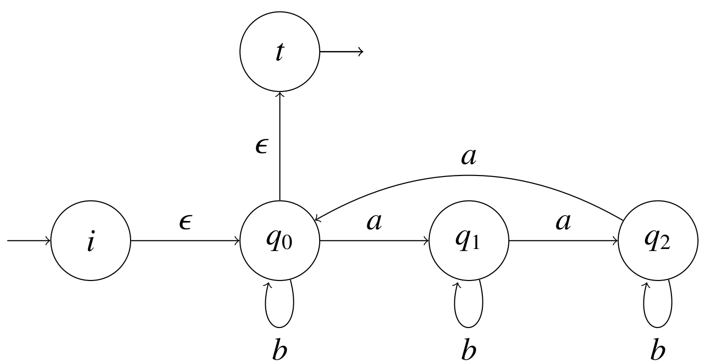


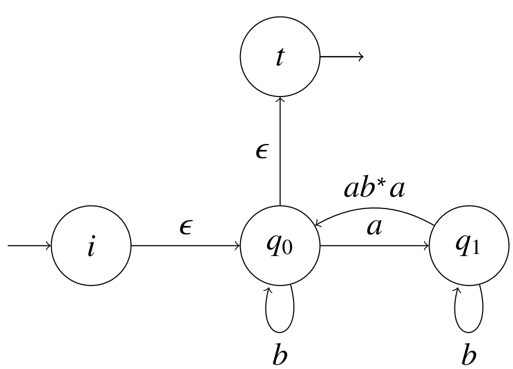
Figura 1: Autómata finito determinista DFA_3

Solución:

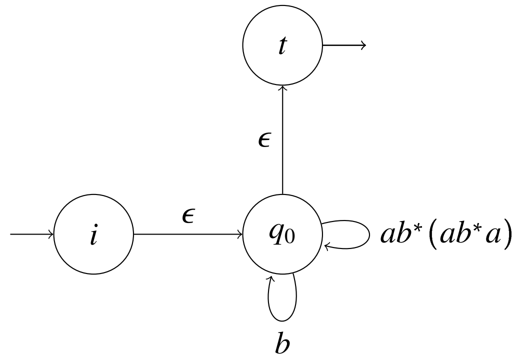
Se observa que el estado inicial no cumple con la condición: es único y no tiene ningún arco llegando a él. Tampoco los estados finales no son únicos y tiene arcos saliendo de ellos, por lo tanto se añade un estado inicial y un final.



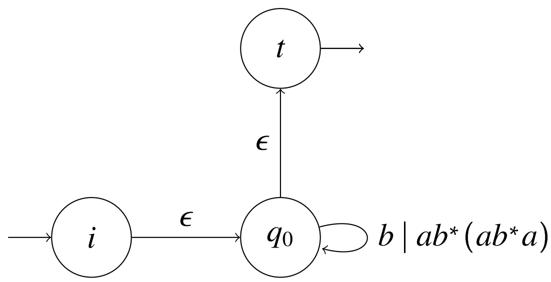
Eliminamos uno de los estados internos q_2. Aplicando BMC, obtenemos:



Todavía quedan estados internos, eliminamos a q_1.

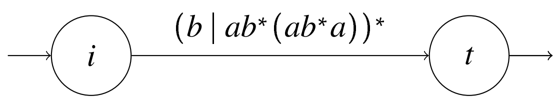


Unimos los dos arcos que salen de q_0 un solo como una alternativa:



Eliminamos el nodo interno q_0 y obtenemos:

Unimos los dos arcos que salen de q_0 un solo como una alternativa:



La expresión regular $e_3 = (b \mid ab^*(ab^*a))^*$.

4. De autómata no determinista a autómata determinista . . . . 20 puntos

En la figura 2 se encuentra definido un autómata finito no determinista NFA_4. Transforme el autómata finito no determinista NFA_4 en un autómata finito determinista DFA_4 tal que: $L(NFA_4) \equiv L(DFA_4)$.

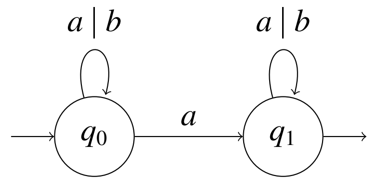


Figura 2: Autómata finito no determinista NFA_4

Solución:

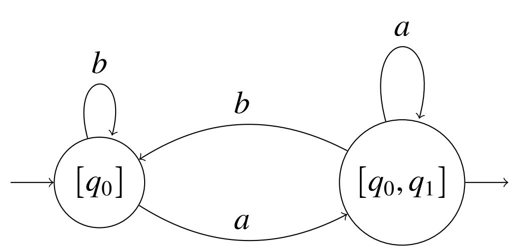
Vamos a calcular el conjunto potencia y comenzamos desde el nuevo estado inicial $[q_0]$. Calculamos el nuevo δ':

$$
\begin{aligned}
\delta'([q_0], a) &= \delta(q_0, a) = \{q_0\} \cup \{q_1\} = \{q_0, q_1\} \\
\delta'([q_0], b) &= \delta(q_0, b) = \{q_0\}
\end{aligned}
$$

Se añade un nuevo macro estado $[q_0, q_1]$.

$$
\begin{aligned}
\delta'([q_0, q_1], a) &= \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \{q_1\} = \{q_0, q_1\} \\
\delta'([q_0, q_1], b) &= \delta(q_0, b) \cup \delta(q_1, b) = \{q_0\} \cup \varnothing = \{q_0\}
\end{aligned}
$$

No se han adicionado más estados obtenemos el nuevo autómata determinista NFA_4:

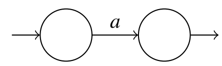


5. Método estructural de Thompson . . . . 20 puntos

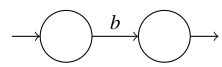
Encuentre el autómata finito no determinista NFA_5 que sea equivalente a la expresión regular e_5, de tal forma que $L(NFA_5) \equiv L(e_5)$. Donde $e_5 = (ab \cup a)^*$.

Solución:

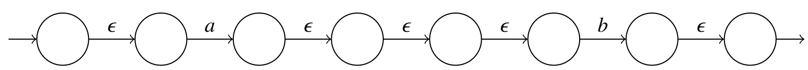
Encontramos las expresiones más simples y las convertimos por el método estructural de Thompson, primero para a:



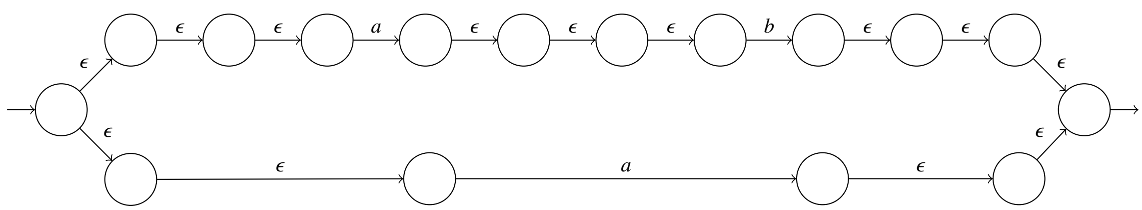
Para la expresion regular b:



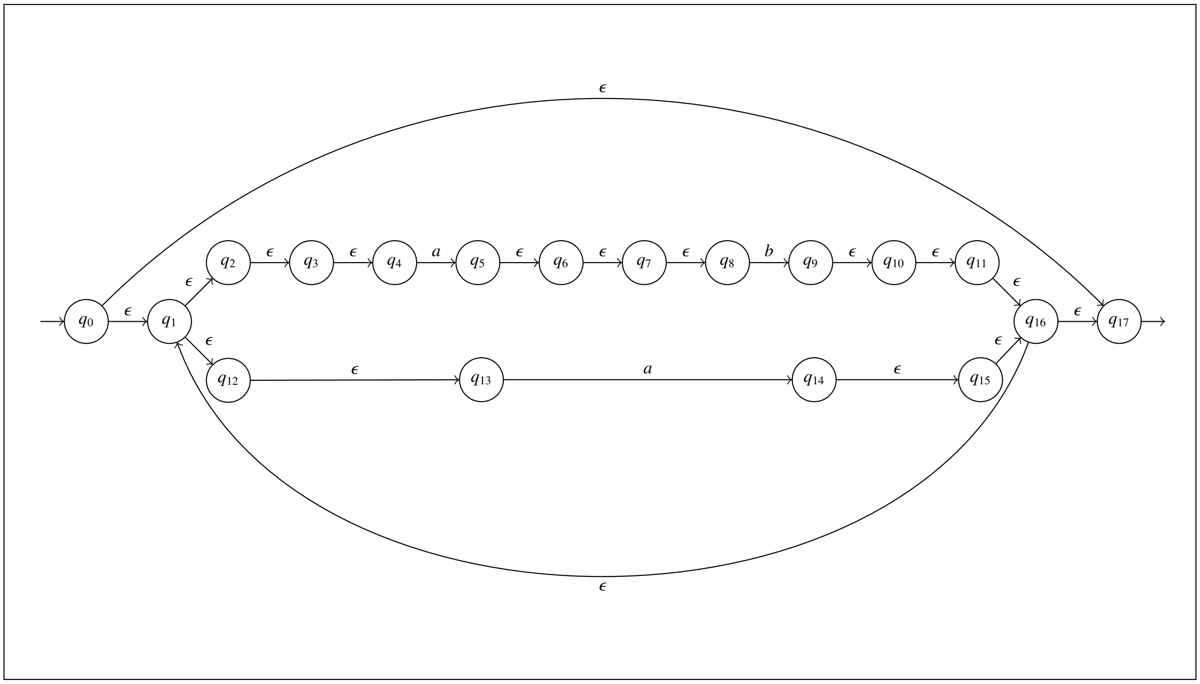
Concatenación ab:



Unión de $(ab \cup a)$



Y por último la expresion $(ab \cup a)^*$


ϵ
q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
q_8
q_9
q_{10}
q_{11}
q_{12}
q_{13}
q_{14}
q_{15}
q_{16}
q_{17}
a
b

Nombre: ____________________
Código: ____________________

1. De automáta a expresiones regular . . . . 20 puntos

Encontrar la expresión regular e_1 equivalente al autómata finito determinista DFA_1 (figura 1); tal que $L(DFA_1) \equiv L(e_1)$.

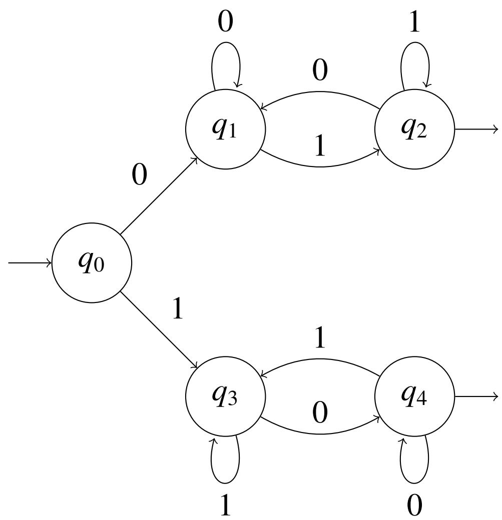
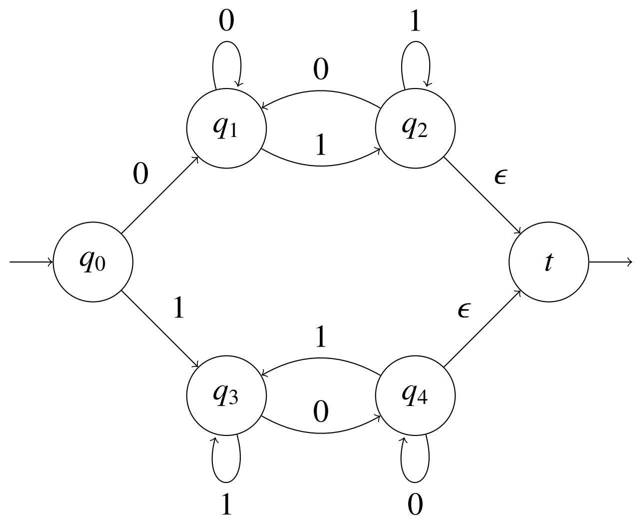


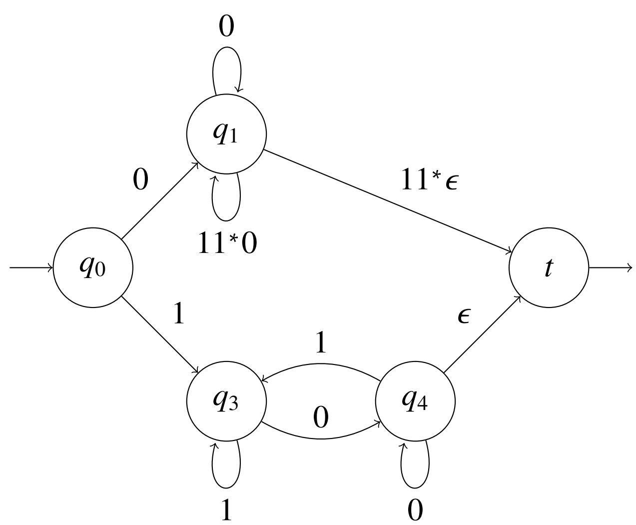
Figura 1: Autómata finito determinista DFA_1

Solución:

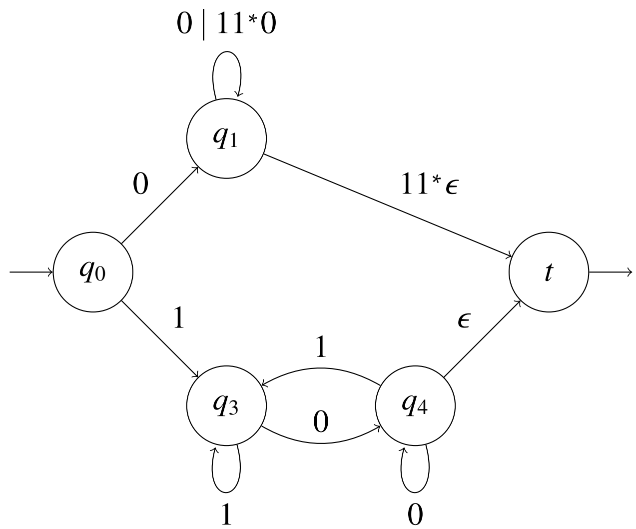
Se observa que el estado inicial q_0 cumple con las condiciones de ser único y ningún arco entra a él; los estados finales q_2 y q_4 no son únicos y tienen arcos saliendo de ellos; por lo tanto se añade un estado final nuevo t y se conectan por medio de movimientos espontáneos.



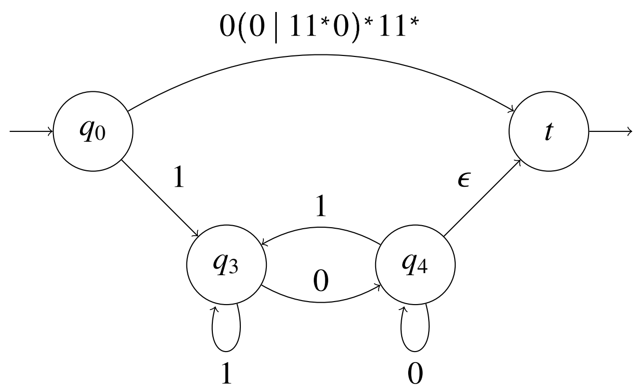
Eliminemos el estado interno q_2, aplicando BMC obtenemos:



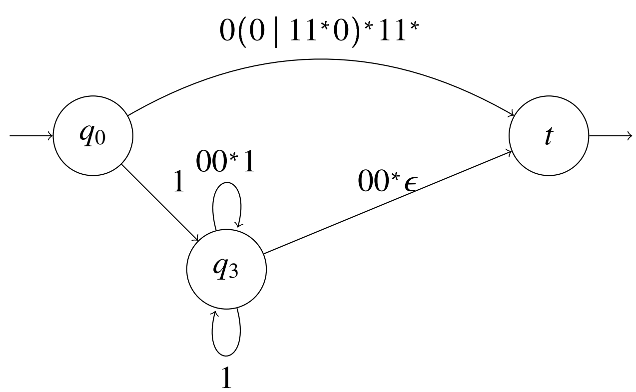
El estado q_1 tiene dos arcos entrando a él los únificamos:



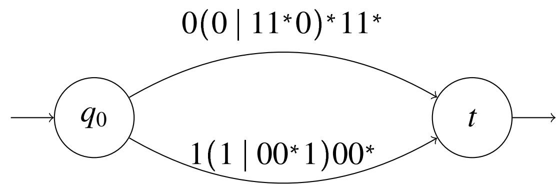
Eliminamos el estado interno q_1:



Eliminamos el estado interno q_4:



En q_3 los arcos que salen y llegan a el se junta y se elimina dicho estado:



Juntando los dos arcos que salen de q_0 y llegan a t obtenemos la expresión regular $e_1 = (0(0 \mid 11^*0)^*11^*) \mid (1(1 \mid 00^*1)00^*)$

2. De expresión regular a gramática independiente de contexto . . . . 20 puntos

Transformar la expresión regular $e_2 = a(a \cup b)^* b$ en una gramática independiente de contexto G_2, tal que $L(e_2) \equiv L(G_2)$.

Solución:

Se identifican las sub-expresiones básicas de la expresión regular e_1

$$\begin{array}{rcl} E_0 & \rightarrow & a \\ E_1 & \rightarrow & b \end{array}$$

Luego pasa al siguiente nivel $(a \cup b)$

$$\begin{array}{rcl} E_2 & \rightarrow & E_0 \\ & | & E_1 \end{array}$$

La siguiente sub-expresión $(a \cup b)^*$:

$$\begin{array}{rcl} E_3 & \rightarrow & \epsilon \\ & | & E_2 E_3 \end{array}$$

La sub-expresión $b(a \cup b)^* a$:

$$E_4 \rightarrow E_0 E_3 E_1$$

Finalmente la gramática G_1 se define como:

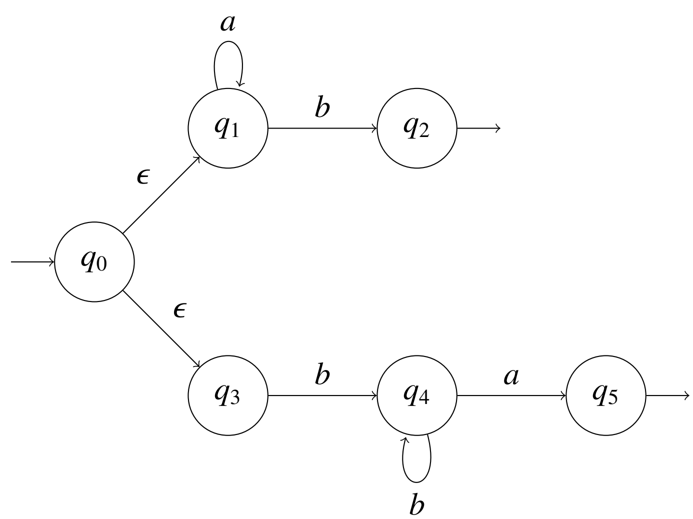
$$\begin{aligned} G_1 = (V &= \{E_0, E_1, E_2, E_3, E_4\}, \\ \Sigma &= \{a, b\}, \\ P &= \{E_4 \rightarrow E_0 E_2 E_1, E_3 \rightarrow \epsilon \mid E_2 E_3, E_2 \rightarrow E_0 \mid E_1, E_1 \rightarrow b, E_2 \rightarrow a\}, \\ S &= E_4) \end{aligned}$$

3. Autómata no determinista . . . . 20 puntos

Encontrar el autómata no determinista NFA_3 que reconozca el lenguaje L_3:

$$L_3 = \{a^n b \mid n \geq 0\} \cup \{b^n a \mid n \geq 1\}$$

Solución:



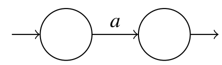
4. Método estructural de Thompson . . . . 20 puntos

Encuentre el autómata finito no determinista NFA_4 que sea equivalente a la expresión regular e_5, de tal forma que $L(NFA_4) \equiv L(e_4)$. Donde $e_4 = (a)^+$. `Nota:` Recuerde que el método estructural de Thompson sólo está definido para los casos básicos: $a \in \Sigma$ y ϵ; y para las operaciones: contatenación, unión y estrella de Kleene, se debe convertir la expresión regular a una expresión que solamente tenga estos operadores.

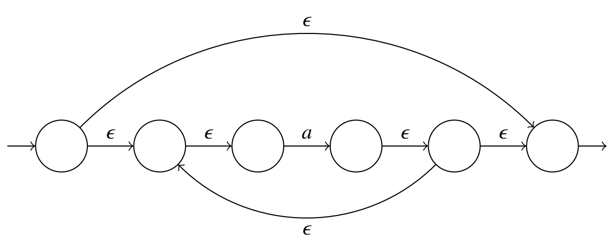
Solución:

En primer lugar expandimos la expresión regular $e_4 \equiv aa^*$.

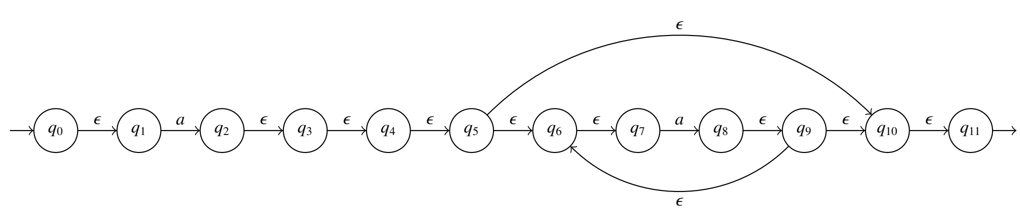
Encontramos las expresiones más simples y las convertimos por el método estructural de Thompson:



El siguiente autómata muestra la sub-expresión regular a^*:



Concatenar aa^*:



5. De autómata no determinista a autómata determinista . . . . 20 puntos

En la figura 2 se encuentra definido un autómata finito no determinista NFA_5. Transforme el autómata finito no determinista NFA_5 en un autómata finito determinista DFA_4 tal que: $L(NFA_5) \equiv L(DFA_5)$.

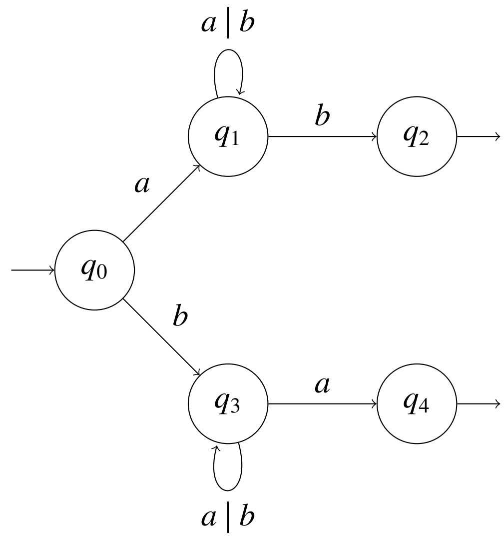


Figura 2: Autómata finito determinista NFA_5

Solución:

Vamos a calcular el conjunto potencia y comenzamos desde el nuevo estado inicial $[q_0]$. Calculamos el nuevo δ':

$$\begin{array}{rcl} \delta'([q_0],a) & = & \delta(q_0,a) = \{q_1\} \\ \delta'([q_0],b) & = & \delta(q_0,b) = \{q_3\} \end{array}$$

Ahora partimos del nuevo estado generado $[q_1]$:

$$\begin{array}{rcl} \delta'([q_1],a) & = & \delta(q_1,a) = \{q_1\} \\ \delta'([q_1],b) & = & \delta(q_1,b) = \{q_1\} \cup \{q_2\} = \{q_1,q_2\} \end{array}$$

Del estado $[q_3]$:

$$\begin{array}{rcl} \delta'([q_3],a) & = & \delta(q_3,a) = \{q_3\} \cup \{q_4\} = \{q_3,q_4\} \\ \delta'([q_3],b) & = & \delta(q_3,b) = \{q_3\} \end{array}$$

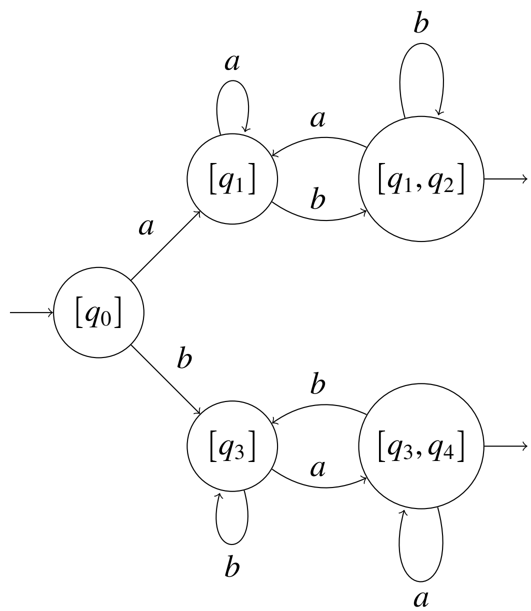
Del estado $[q_1,q_2]$:

$$\begin{array}{rcl}\delta'([q_1,q_2],a) & = & \delta(q_1,a)\cup\delta(q_2,a)=\{q_1\}\cup\varnothing=\{q_1\}\\ \delta'([q_1,q_2],b) & = & \delta(q_1,b)\cup\delta(q_2,b)=\{q_1,q_2\}\cup\varnothing=\{q_1,q_2\}\end{array}$$

Del estado $[q_3,q_4]$:

$$\begin{array}{rcl}\delta'([q_3,q_4],a) & = & \delta(q_3,a)\cup\delta(q_4,a)=\{q_3,q_4\}\cup\varnothing=\{q_3,q_4\}\\ \delta'([q_3,q_4],b) & = & \delta(q_3,b)\cup\delta(q_4,b)=\{q_3\}\cup\varnothing=\{q_3\}\end{array}$$

De lo anterior obtenemos el autómata:



Nombre: ______________________________
Código: ______________________________

Nota: La solución de los puntos del parcial debe ser formal.

1. Algoritmo de Glushkov, McNaughton y Yamada .... 20 puntos
Obtener el autómata de estado finito posiblemente no-determinista de la expresión regular $r_1 = (a \cup b)^+(c \cup b)^*$ aplicando el algoritmo de GMY. Recuerde dejar indicado el calculo de los conjuntos *Null*, *Ini*, *Fin* y *Dig*. Dibuje el autómata obtenido.

2. Algoritmo de Berry y Sethi .... 20 puntos
Utilizando la expresión regular $r_2 = (a \cup b)^*(a \cup c)^+$ definir un autómata de estado finito determinista, *utilizando el algoritmo de Berry y Sethi*. Recuerde dejar indicado el cálculo de los conjuntos *Null*, *Ini*, *Fin*, *Dig* y *Fol*. Dibuje el autómata obtenido.

3. Autómata de pila .... 20 puntos
Definir un autómata de pila posiblemente no determinista que reconozca el siguiente lenguaje:

$$L_3 = \{a^n c^m b^n \mid n > 0, m > 0\}$$

4. Condición $LL(1)$ .... 20 puntos
Determinar si la siguiente siguiente gramática G_4 cumple o no con la condición $LL(1)$. Recuerde calcular las condiciones de $LL(1)$ en las producciones más importantes.

$$G_4 = (\Sigma = \{a, b, c\}, V = \{S, A\}, P = \{S \to aAb \mid \epsilon, A \to bSa \mid c\}, S)$$

5. Reconocedor $LL(1)$ .... 20 puntos
La gramática G_5 cumple con la condición $LL(1)$. Utilizando el algoritmo del Reconocedor determinista $LL(1)$, mostrar si las siguientes cadenas x_i pertenecen o no al lenguaje de la gramática $L(G_5)$.

$$G_5 = (\Sigma = \{a, b\}, V = \{A, B\}, P = \{A \to baB \mid aB, B \to A \mid \epsilon\}, A)$$

a) $x_1 = aba$.
b) $x_2 = abb$.

Recuerde construir la red de autómatas y la secuencia de ejecución del Reconocedor $LL(1)$, con la siguiente tabla:

Pila	**Cadena entrada**	**Acción**

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **26 de abril de 2013**

Nombre: ____________________
Código: ____________________

Nota: La solución de los puntos del parcial debe ser formal.

1. Algoritmo de Glushkov, McNaughton y Yamada . . . . 20 puntos

Obtener el autómada de estado finito posiblemente no-determinista de la expresión regular $r_1 = (a \cup b)^*(a \cup c)^+$ aplicando el algoritmo de GMY. Recuerde dejar indicado el calculo de los conjuntos *Null*, *Ini*, *Fin* y *Dig*, dibuje el autómata obtenido.

2. Algoritmo de Berry y Sethi . . . . 20 puntos

Utilizando la expresión regular $r_2 = (a \cup b)^+(a \cup c)^*$ definir un autómata de estado finito determinista, *utilizando el algorimo de Berry y Sethi*. Recuerde dejar indicado el cálculo de los conjuntos *Null*, *Ini*, *Fin*, *Dig* y *Fol*, dibuje el autómata obtenido.

3. Autómata de pila . . . . 20 puntos

Definir un autómata de pila posiblemente no determinista que reconozca el siguiente lenguaje:

$$L_3 = \{a^{2n}c^m b^n \mid n > 0, m > 0\}$$

4. Condición $LL(1)$ . . . . 20 puntos

Determinar si la siguiente siguiente gramática G_4 cumple o no con la condición $LL(1)$. Recuerde calcular las condiciones de $LL(1)$ en las producciones más importantes.

$$G_4 = (\Sigma = \{d, e\}, V = \{S, L, P\}, \{S \to dPe, P \to dPe \mid L, L \to d \mid dL, S\})\}$$

5. Reconocedor $LL(1)$ . . . . 20 puntos

La gramática G_5 cumple con la condición $LL(1)$. Utilizando el algoritmo del Reconocedor determinista $LL(1)$, mostrar si las siguientes cadenas x_i pertenecen o no al lenguaje de la gramática $L(G_5)$.

$$G_5 = (\Sigma = \{a, b, c\}, V = \{S, A, B, C\}, P = \{S \to aB, A \to bC, B \to Sc \mid C, C \to c \mid Ac\}, S)$$

a) $x_1 = aba$.

b) $x_2 = abcc$.

Recuerde construir la red de autómatas y la secuencia de ejecución del Reconocedor $LL(1)$, con la siguiente tabla:

Pila	Cadena entrada	Acción

Nombre: ______________________________
Código: ______________________________

Importante: Toda respuesta debe estar sustentada por un procedimiento formal. No se aceptan respuestas sin su correspondiente procedimiento. El parcial debe ser el resultado de un trabajo individual, no grupal.

(20 %) 1. **Transformación de gramática** $LL(1)$

La siguiente gramática G_1:

$$\begin{array}{lcl} A & \rightarrow & A\ A \\ & | & (A) \\ & | & () \end{array}$$

no es LL(1). Transforme la gramática G_1 en una nueva gramática G_1' que cumpla con la condición $LL(1)$.

ST0270 Lenguajes formales y compiladores - 031 - 032 **Universidad EAFIT**
Cuarto parcial **DIS**
Juan Francisco Cardona McCormick **16 de mayo de 2013**

Nombre: ____________________
Código: ____________________

Importante: Toda respuesta debe estar sustentada por un procedimiento formal. No se aceptan respuestas sin su correspondiente procedimiento. El parcial debe ser el resultado de un trabajo individual, no grupal.

(35 %) 1. **Construcción del piloto, condición** $LR(0)$ **y análisis sintáctico ascendente** $LR(0)$
Dada la siguiente gramática G_2:

$$\begin{array}{rcl} S_0 & \rightarrow & Stm \dashv \\ Stm & \rightarrow & Stm \; \mathtt{;} \; Asg \\ & | & Asg \\ Asg & \rightarrow & Exp \\ & | & \mathtt{id} :=Exp \\ Exp & \rightarrow & Exp + \mathtt{id} \\ & | & \mathtt{id} \end{array}$$

a) Construir a partir de la gramática G_2 su correspodiente piloto $LR(0)$.

b) Del piloto construido en el punto anterior demostrar si la gramática es o no $LR(0)$. Si la gramática G_2 no cumple con la condición $LR(0)$ mostrar cuáles son los estados que no cumplen con dicha condición y la razón por la cuales dichos estados no cumplen con la condición $LR(0)$.

c) Si la gramática es $LR(0)$, utilizando el correspondiente analizador ascendente generado por el anterior piloto $LR(0)$, si la cadena x:

$$\mathtt{id} := \mathtt{id} + \mathtt{id}$$

pertenece al lenguaje de $L(G_2)$ ($x \in L(G_2)$).

(45 %) 2. **Construcción** $LR(1)$.
Dada la siguiente gramática G_3:

$$\begin{array}{rcl} S_0 & \rightarrow & Stm \dashv \\ Stm & \rightarrow & Stm_E \\ & | & Stm_T \\ Stm_E & \rightarrow & \mathtt{if}\; expr \;\mathtt{then}\; Stm_E \;\mathtt{else}\; Stm_E \\ & | & assign \\ Stm_T & \rightarrow & \mathtt{if}\; expr \;\mathtt{then}\; Stm_E \;\mathtt{else}\; Stm_T \\ & | & Stm \end{array}$$

a) Construir a partir de la gramática G_3 su correspondiente piloto $LR(1)$.

b) Determinar si la gramática G_3 es $LR(1)$ o no. Si no, mostrar cuáles macro-estados y decir por que estos no cumplen con la condición $LR(1)$.

c) Si la gramática es $LR(1)$ probar, utilizando el correspondiente analizador sintáctico ascendente ayudado por su correspodiente piloto, si la cadena:

$$x = \texttt{if}\ expr\ \texttt{then}\ \texttt{if}\ expr\ \texttt{then}\ assign\ \texttt{else}\ assign\ \dashv$$

pertenece al lenguaje de G_3 ($x \in L(G_3)$).

Nombre: ______________________________
Código: ______________________________

(20 %) 1. **Expresiones regulares**
Dado el alfabeto $\Sigma = \{\boxdot, \odot\}$. Escriba una expresión regular r_1 que defina el siguiente lenguaje: Todas las cadenas comienzan con $\odot$ y terminan siempre con $\odot$ y tiene al menos cuatro $\boxdot$ no necesariamente seguidas.

> **Solución:**
>
> $$r_1 = \odot(\boxdot \mid \odot)^* \boxdot (\boxdot \mid \odot)^* \boxdot (\boxdot \mid \odot)^* \boxdot (\boxdot \mid \odot)^* \odot$$

(20 %) 2. **Gramáticas independientes de contexto**
Escribir la gramática independiente de contexto G_2 que genere el siguiente lenguaje L_2:

$$L_2 = \{a^n b^{n\times 3} a^{k\times 2} b^k \mid n \geq 1, k \geq 0\}$$

> **Solución:**
>
> Se observa que la gramática es la concatentación de dos lenguajes: $L_2' L_2''$. Donde L_2' es:
>
> $$L_2' = \{a^n b^{n\times 2} \mid n \geq 1\}$$
>
> y L_2'' es:
>
> $$L_2'' = \{a^{k\times 3} b^k \mid k \geq 0\}$$
>
> Ahora definimos el lenguaje para la primera gramática:
>
> $$G_2' = (V = \{A\}, \Sigma = \{a, b\}, P = \{A \rightarrow aAbbb \mid abbb\}, S)$$
>
> Ahora definamos el lenguaje para la segunda gramática:
>
> $$G_2'' = (V = \{B\}, \Sigma = \{a, b\}, P = \{B \rightarrow aaBb \mid \epsilon\}, B)$$
>
> Ahora contatenmos las dos gramáticas:
>
> $$G_2 = (V = \{S, A, B\}, \Sigma = \{a, b\}, P = \{S \rightarrow AB, A \rightarrow aAbbb \mid abbb, B \rightarrow aaBb \mid \epsilon\}, S)$$

(20 %) 3. **Ambigüedad**

a) (10 %) Mostrar la ambigüedad en la siguiente gramática independiente de contexto G_3:

$$G_3 = (V = \{Z, Y\}, \Sigma = \{z, y, x, w\}, P = \{Z \to ZYZ \mid z, Y \to y \mid x \mid w, Z)$$

Solución:
Observamos que la gramática tiene un problema de ambigüedad en la primera producción, dado que esta utilizando el no-terminal como Y operador y Z no tiene un orden de derivación, se sabe que tiene problemas de ambigüedad. Observemos la cadena *abaca*, genera a través de dos derivaciones distintas dos árboles de derivación diferentes.

Primera derivación:

$$\begin{aligned} Z &\Rightarrow ZYZ \\ &\Rightarrow ZYZYZ \\ &\Rightarrow zYZYZ \\ &\Rightarrow zyZYZ \\ &\Rightarrow zyzYZ \\ &\Rightarrow zyzyZ \\ &\Rightarrow zyzyz \end{aligned}$$

Segunda derivación:

$$\begin{aligned} Z &\Rightarrow ZYZ \\ &\Rightarrow ZYZYZ \\ &\Rightarrow ZYZYz \\ &\Rightarrow ZYZyz \\ &\Rightarrow ZYzyz \\ &\Rightarrow Zyzyz \\ &\Rightarrow zyzyz \end{aligned}$$

Esto genera dos árboles de derivación distintos para la misma cadena por lo tanto la gramática es ambigua.

b) (10 %) Mostrar la ambigüedad en la siguiente expresión regular r_3 con alfabeto $\Sigma = \{h, i\}$:

$$r_3 = i^*(i \mid h)^*$$

Solución:
La cadena *aa* puede ser obtenidas por dos derivaciones distintas. Redefinamos la anterior expresion regular y definamos una nueva numerada:

$$r_3' = i_1^*(i_2 \mid h_3)^*$$

Derivemos la cadena *aa* a partir de la nueva expresion regular.

Primera derivación:

$$\begin{aligned} i_1^*(i_2 \mid h_3)^* &\Rightarrow (i_1)^2(i_2 \mid h_3)^* \\ &\Rightarrow (i_1)^2\epsilon \equiv i_1 i_1 \end{aligned}$$

Segunda derivación:

$$\begin{aligned} i_1^*(i_2 \mid h_3)^* &\Rightarrow (i_1)^1(i_2 \mid h_3)^* \\ &\Rightarrow (i_1)(i_2 \mid h_3)^1 \\ &\Rightarrow (i_1)(u_2) \equiv i_1 i_2 \end{aligned}$$

Eliminando los número de la primera y segunda derivación obtenemos: ii. Por lo tanto la expresion regular es ambigua.

(20 %) 4. **Derivación de gramáticas**
La siguiente gramática G_4:

$$\begin{aligned} V &= \{COLUMN, ROW, ATTACK, SHIP, REACTION, TURN, GAME\}, \\ \Sigma &= \{A, B, C, D, E, F, G, H, I, J, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \\ &\quad battleship, frigate, submarine, destroyer, missed, hit, stunk, lost\_battle\} \\ P &= \{COLUMN \to A \mid B \mid C \mid D \mid E \mid F \mid G \mid H \mid I \mid J \\ &\quad ROW \to 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 10, \\ &\quad ATTACK \to COLUMN\ \ ROW, \\ &\quad SHIP \to battleship \mid frigate \mid submarine \mid destroyer, \\ &\quad REACTION \to missed \mid hit\ \ SHIP \mid sunk\ \ SHIP \mid lost\_battle, \\ &\quad TURN \to ATTACK\ \ REACTION, \\ &\quad GAME \to TURN \mid TURN\ \ GAME\} \\ S &= GAME \end{aligned}$$

Muestra la gramática del popular juego "Batalla naval": Suponga que los jugadores solamente tiene un barco: *destroyer* y este tiene solamente en dos posiciones, suponga que el jugador *Uno* ha puesto su barco en las posiciones $H1$ y $J1$ y que el jugador *Dos* ha puesto su barco en $G7$ y $G8$. Muestre un juego en el cual el jugador *Uno* al hundir el barco de *Dos*, sin que el jugador *Dos* toque siquiera el primer barco. Trate al menos una falla de *Uno*.

Solución:

La siguiente es una posible derivación propuesta:

$$
\begin{aligned}
GAME &\Rightarrow TURN\ GAME \\
&\Rightarrow ATTACK\ REACTION\ GAME \\
&\Rightarrow COLUMN\ ROW\ REACTION\ GAME \\
&\Rightarrow G\ ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,REACTION\ GAME \\
&\Rightarrow G\,6\,missed\ GAME \\
&\Rightarrow G\,6\,missed\ ATTACK\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\ COLUMN\ ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\ TURN\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\ ATTACK\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\ COLUMN\ ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\ ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ SHIP\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ TURN\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ ATTACK\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ COLUMN\ ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\ ROW\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ REACTION\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\ GAME \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\ TURN \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\ ATTACK\ REACTION \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\ COLUMN\ ROW\ REACTION \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\,G\ ROW\ REACTION \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\,G\,8\ REACTION \\
&\Rightarrow G\,6\,missed\,A\,1\,missed\,G\,7\ hit\ destroyer\ A\,2\ missed\,G\,8\ lost\_battle
\end{aligned}
$$

(20 %) 5. **Gramática limpia**
Transformar la siguiente gramática G_5 en una gramática independiente de contexto limpia G'_5:

$$G_5 = (V = \{A, B, C, D, E\}, \Sigma = \{a, b, c, d\},$$
$$P = \{A \to BC \mid AD, B \to aBdd \mid add, C \to ccCb \mid \epsilon, D \to DA, E \to dE \mid d\}, A)$$

Solución:

En primer lugar verificamos si la gramática está bien definida:

$$def_0 = \{B, C, E\}$$

Este es el conjunto inicial de los no-terminales que producen terminales. Iteramos para obtener el siguiente nivel.

$$def_1 = def_0 \cup \{A\} = \{A, B, C, E\}$$

Iteramos de nuevo y obtenemos:

$$def_2 = def_1 \cup \emptyset = \{A, B, C, E\}$$

Ahora encontramos los no-terminales que no están definidos:

$$nodef = V \setminus def = \{A, B, C, D, E\} \setminus \{A, B, C, E\} = \{D\}$$

Generamos una nueva gramática G''_5 elimando el no-terminal D

$G''_5 = (V = \{A, B, C, E\}, \Sigma = \{a, b, c, d\}, P = \{A \to BC \mid AD, B \to aBdd \mid add, C \to ccCb \mid \epsilon, E \to dE \mid d\},$

Ahora verificamos que no-terminales son alcanzables desde el axioma:

$$\{B, C\}$$

Por lo tanto generamos una nueva gramática G'_5 eliminando el no-terminal E:

$$G''_5 = (V = \{A, B, C\}, \Sigma = \{a, b, c, d\}, P = \{A \to BC \mid AD, B \to aBdd \mid add, C \to ccCb \mid \epsilon\}, A)$$

Nombre: ____________________
Código: ____________________

(25 %) 1. **Expresiones regulares**
Dado el alfabeto $\Sigma = \{a, b, c, d\}$. Escriba una expresión regular r_1 que defina el siguiente lenguaje:

$$L(r_1) = \{(ab)^n(cd)^m | n \geq 1,\ m \geq 0\} \cup \{(ab)^n(dc)^m | n \geq 1,\ m \geq 1\}$$

Solución:
Una posible solución es:

$$r_1 = (ab)^+(cd)^* \ \mid \ (ab)^+(dc)^+$$

(25 %) 2. **Gramáticas independientes de contexto**
Escribir la gramática independiente de contexto G_2 que genere el siguiente lenguaje de listas anidadas. Los elementos son de tres tipos: a, b o listas. Las listas pueden estar vacías.
Ejemplos de listas válidas: $[a]$, $[a, b, [b, b, a]]$, $[[[b, a], a], [a, b, b]]$, $[[[]]]$.

Solución:
$G_2 = (V, \Sigma, P, S)$ donde:

$$\begin{array}{rcl}
V & = & \{L, LE, E\} \\
\Sigma & = & \{\ '[',\ ']',\ a,\ b,\ ','\ \} \\
P & = & \{L \rightarrow\ '['\ LE\ ']'\ \mid\ '['\ M\ ']', \\
 & & M \rightarrow\ \epsilon\ \mid\ '['\ M\ ']' \\
 & & LE \rightarrow\ E\ \mid\ E\ ','\ LE, \\
 & & E \rightarrow\ a\ \mid\ b\ \mid\ L\ \} \\
S & = & L
\end{array}$$

(25 %) 3. **Gramática limpia**
Transformar la siguiente gramática G_5 en una gramática independiente de contexto limpia G'_3:

$$G_3 = (V = \{A, B, C, D, E\}, \Sigma = \{a, b, c, d\},$$
$$P = \{A \rightarrow BC \mid AD, B \rightarrow aBdd \mid add, C \rightarrow ccCb \mid \epsilon, D \rightarrow DA, E \rightarrow dE \mid d\}, A)$$

Solución:

En primer lugar verificamos si la gramática está bien definida:

$$def_0 = \{B, C, E\}$$

Este es el conjunto inicial de los no-terminales que producen terminales. Iteramos para obtener el siguiente nivel.

$$def_1 = def_0 \cup \{A\} = \{A, B, C, E\}$$

Iteramos de nuevo y obtenemos:

$$def_2 = def_1 \cup \emptyset = \{A, B, C, E\}$$

Ahora encontramos los no-terminales que no están definidos:

$$nodef = V \setminus def = \{A, B, C, D, E\} \setminus \{A, B, C, E\} = \{D\}$$

Generamos una nueva gramática G''_5 elimando el no-terminal D

$G''_5 = (V = \{A, B, C, E\}, \Sigma = \{a, b, c, d\}, P = \{A \rightarrow BC \mid AD, B \rightarrow aBdd \mid add, C \rightarrow ccCb \mid \epsilon, E \rightarrow dE \mid d\}, A$

Ahora verificamos que no-terminales son alcanzables desde el axioma:

$$\{B, C\}$$

Por lo tanto generamos una nueva gramática G'_5 eliminando el no-terminal E:

$G''_5 = (V = \{A, B, C\}, \Sigma = \{a, b, c, d\}, P = \{A \rightarrow BC \mid AD, B \rightarrow aBdd \mid add, C \rightarrow ccCb \mid \epsilon\}, A)$

(25 %) 4. **Derivación de gramáticas**
La siguiente gramática G_4:

$$
\begin{array}{rcl}
V & = & \{S, A, B, C\}, \\
\Sigma & = & \{\langle, \rangle, a, b, c, \backslash\} \\
P & = & \{\ S \rightarrow \langle s\rangle A\langle\backslash s\rangle, \\
 & & A \rightarrow \langle a\rangle A\langle\backslash a\rangle \mid \langle a\rangle B\langle\backslash a\rangle \\
 & & B \rightarrow \langle b\rangle B\langle\backslash b\rangle \mid \langle b\rangle C\langle\backslash b\rangle \mid \langle b\backslash\rangle \mid \epsilon \\
 & & C \rightarrow \langle c\rangle A\langle\backslash c\rangle \mid \langle c\backslash\rangle\ \} \\
S & = & S
\end{array}
$$

Muestre dos derivaciones (utilizando cualquier orden).

a) Tenga un $\langle c\backslash\rangle$ en el centro.

b) Tenga un $\langle b\backslash\rangle$ en el centro pero circudanda de al menos de un $\langle b\rangle$ y $\langle\backslash b\rangle$.
Por ejemplo: $\ldots\langle b\rangle\ldots\langle b\backslash\rangle\ldots\langle\backslash b\rangle\ldots$.

Solución:

a) Primera derivación:

$$
\begin{array}{rcl}
S & \Rightarrow & \langle s\rangle\ A\ \langle\backslash s\rangle \\
 & \Rightarrow & \langle s\rangle\ \langle a\rangle\ A\ \langle\backslash a\rangle\langle\backslash s\rangle \\
 & \Rightarrow & \langle s\rangle\ \langle a\rangle\ \langle a\rangle B\langle\backslash a\rangle\ \ \langle\backslash a\rangle\langle\backslash s\rangle \\
 & \Rightarrow & \langle s\rangle\ \langle a\rangle\ \langle a\rangle\langle b\rangle\langle c\backslash\rangle\langle\backslash b\rangle\langle\backslash a\rangle\ \ \langle\backslash a\rangle\langle\backslash s\rangle
\end{array}
$$

b) Segunda derivación:

$$
\begin{array}{rcl}
S & \Rightarrow & \langle s\rangle\ A\ \langle\backslash s\rangle \\
 & \Rightarrow & \langle s\rangle\ \langle a\rangle\ A\ \langle\backslash a\rangle\langle\backslash s\rangle \\
 & \Rightarrow & \langle s\rangle\ \langle a\rangle\ \langle a\rangle B\langle\backslash a\rangle\ \ \langle\backslash a\rangle\langle\backslash s\rangle \\
 & \Rightarrow & \langle s\rangle\ \langle a\rangle\ \langle a\rangle\langle b\rangle\langle b\backslash\rangle\langle\backslash b\rangle\ \langle\backslash a\rangle\ \ \langle\backslash a\rangle\langle\backslash s\rangle
\end{array}
$$

Nombre: ______________________________
Código: ______________________________

1. De expresión regular a gramática independiente de contexto . . . . 20 puntos

Transformar la expresión regular $e_1 = b(c \cup d)^* c$ en una gramática independiente de contexto G_1, tal que $L(e_1) \equiv L(G_1)$.

Solución:

Se identifican las sub-expresiones básicas de la expresión regular e_1

$$\begin{array}{rcl} E_0 & \rightarrow & c \\ E_1 & \rightarrow & d \\ E_2 & \rightarrow & b \end{array}$$

Luego pasa al siguiente nivel $(c \cup d)$

$$\begin{array}{rcl} E_3 & \rightarrow & E_0 \\ & | & E_1 \end{array}$$

La siguiente sub-expresión $(c \cup d)^*$:

$$\begin{array}{rcl} E_4 & \rightarrow & \epsilon \\ & | & E_0 E_4 \end{array}$$

La sub-expresión $b(a \cup b)^* a$:

$$E_5 \rightarrow E_3 E_4 E_0$$

Finalmente la gramática G_1 se define como:

$$\begin{array}{c} G_1 = (V = \{E_0, E_1, E_2, E_3, E_4, E_5\}, \\ \Sigma = \{b, c, d\}, \\ P = \{E_5 \rightarrow E_3 E_4 E_0, E_4 \rightarrow \epsilon \mid E_0 E_4, E_3 \rightarrow \epsilon \mid E_0, E_3 \rightarrow E_0 \mid E_1, E_2 \rightarrow b, E_1 \rightarrow d, E_0 \rightarrow c\}, \\ S = E_5) \end{array}$$

2. Transformación de gramáticas . . . . 20 puntos

La siguiente gramática G_2 define el lenguaje formal λ-cálculo.

$$
\begin{aligned}
Expr &\rightarrow x \\
&\mid \lambda x.Expr \\
&\mid Expr\ Expr
\end{aligned}
$$

Donde x pertenece al conjunto de los nombres de variables Var. Se observa que la gramática G_2 es recursiva por la izquierda.

Transformar la gramática G_2 en una nueva gramática G_2' que elimina la recursividad por la izquierda de G_2

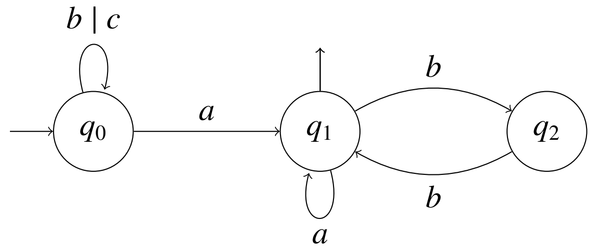
Solución:

La nueva gramática G_2':

$$
\begin{aligned}
Expr &\rightarrow x \\
&\mid \lambda x.Expr \\
&\mid x\ Expr' \\
&\mid \lambda x.Expr\ Expr' \\
Expr' &\rightarrow Expr \\
&\rightarrow Expr\ Expr'
\end{aligned}
$$

3. Autómata total .... 20 puntos

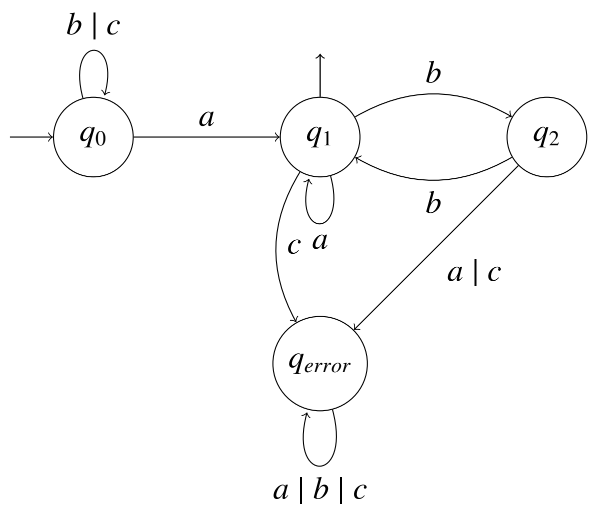
Dado el siguiente autómata A, con $\Sigma = \{a, b, c\}$:



Genere su correspondiente autómata total.

Solución:

Se deben añadir un estado sumidero donde se reciben todas las transiciones que no están definidas en el orignal.



4. Ecuaciones de lenguajes lineales . . . . 20 puntos

Utilizando la identidad de Arden; convierta la siguiente gramática independiente de contexto G_2 en una expresión regular r_2, que cumpla con la condición $L(G_2) \equiv L(r_2)$

$$G_2 = (V = \{S, A\}, \Sigma = \{a, b\}, P = \{S \rightarrow Aa, A \rightarrow aA \mid bA \mid \epsilon\}, S = S)$$

Solución:

En primer lugar observamos generados por las producciones:

$$\begin{array}{rcl} L_S & = & L_A a \\ L_A & = & aL_A \mid bL_B \mid \epsilon \end{array}$$

Factorizamos

$$\begin{array}{rcl} L_S & = & L_A a \\ L_A & = & (a \mid b)L_A \mid \epsilon \end{array}$$

Aplicando la identidad de Arden a L_A obtenemos:

$$\begin{array}{rcl} L_S & = & L_A a \\ L_A & = & (a \mid b)^* \end{array}$$

Aplicando la solución de L_A:

$$\begin{array}{rcl} L_S & = & (a \mid b)^* a \\ L_A & = & (a \mid b)^* \end{array}$$

5. Gramáticas lineales . . . . 20 puntos

Defina la gramática unilineal por la izquierda de la siguiente expresion lineal:

$$(a \mid b)^{+}bb(a \mid b)^{*}a$$

Solución:

La siguiente es la definición de una gramática unlinea para la anterior expresión regular:

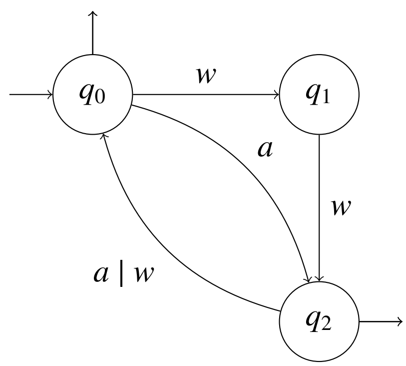
$$\begin{array}{lcl} S & \rightarrow & Ab \\ A & \rightarrow & Aa \mid Ab \mid Bbb \\ B & \rightarrow & Ba \mid Bb \mid a \mid b \end{array}$$

ST0270 Lenguajes formales y compiladores - 032 — **Universidad EAFIT**
Segundo parcial — **DIS**
Juan Francisco Cardona McCormick — **14 de marzo de 2014**

Nombre: ____________________
Código: ____________________

1. Mimización de autómata .... 25 puntos

Encontrar formalmente el autómata mínimo del siguiente autómata finito determinista D_1:



Solución:

Se hace la matriz para evaluar los casos:

q_1		
q_2		
	q_0	q_1

Se compara los estados finales con lo que no los son: q_0 con q_1 y q_2 con q_1. Ambos son distinguibles:

q_1	X	
q_2		X
	q_0	q_1

Nos queda comparar q_2 con q_0. Comparemos con el carácter a:

$$\begin{aligned}\delta(q_0, a) &= q_2\\ \delta(q_2, a) &= q_0\end{aligned}$$

Ambas son finales, con el carácter a son indistiguibles. Comparemos con el carácter w:

$$\begin{aligned}\delta(q_0, w) &= q_1\\ \delta(q_2, w) &= q_0\end{aligned}$$

Con el carácter w, q_0 y q_2 son distinguibles por lo tanto ambas son distiguibles:

q_1	X	
q_2	X	X
	q_0	q_1

Por lo tanto el autómata ya se encuentra minimizado.

2. Transformación de gramáticas..25 puntos

Dada la siguiente gramática G_2:

$$
\begin{aligned}
E &\rightarrow A \rhd E \mid A \\
A &\rightarrow E \lhd A \mid B \\
B &\rightarrow A \ggg C \mid C \\
C &\rightarrow a \mid b
\end{aligned}
$$

Transformar la gramática G_2 en una nueva gramática G_2' que elimina la recursividad por la izquierda de G_2.

Solución:

Se observa que la gramática no es recursiva inmediata, pero si lo es rescursiva no inmediata:

$$
\begin{aligned}
E &\Rightarrow A \rhd E \\
&\Rightarrow E \lhd A \rhd E
\end{aligned}
$$

Se establece un orden entre los no-terminales: $\{E, A, B, C\}$

Analizamos y expandimos si es posible el no-terminal E:

$$
\begin{aligned}
E &\rightarrow A \rhd E \mid A \\
A &\rightarrow E \lhd A \mid B \\
B &\rightarrow A \ggg C \mid C \\
C &\rightarrow a \mid b
\end{aligned}
$$

El no-terminal E no es recursivo inmediato, se continua con el siguiente no-terminal A y se observa que depende directamente de E que está antes, entonces expandimos en A todas aquellas dependencias directas de E y se obtiene la siguiente gramática:

$$
\begin{aligned}
E &\rightarrow A \rhd E \mid A \\
A &\rightarrow A \rhd E \lhd A \mid A \lhd A \mid B \\
B &\rightarrow A \ggg C \mid C \\
C &\rightarrow a \mid b
\end{aligned}
$$

Se observa que A es recursiva directamente. Se elimina la recursividad directa con A y se obtiene la siguiente gramática:

$$
\begin{aligned}
E &\rightarrow A \triangleright E \mid A \\
A &\rightarrow BA' \mid B \\
A' &\rightarrow \triangleright E \triangleleft AA' \mid \triangleright E \triangleleft A \mid \triangleleft AA' \mid \triangleleft A \\
B &\rightarrow A \ggg C \mid C \\
C &\rightarrow a \mid b
\end{aligned}
$$

De la anterior gramática se observa que el siguiente no-terminal B depende de A que está antes en el orden establecido. Entonces expandimos la gramática:

$$
\begin{aligned}
E &\rightarrow A \triangleright E \mid A \\
A &\rightarrow BA' \mid B \\
A' &\rightarrow \triangleright E \triangleleft AA' \mid \triangleright E \triangleleft A \mid \triangleleft AA' \mid \triangleleft A \\
B &\rightarrow BA' \ggg C \mid B \ggg C \mid C \\
C &\rightarrow a \mid b
\end{aligned}
$$

Se observa que B es recursiva inmediata. Se elimina la recursividad directa con B y se obtiene la siguiente gramática:

$$
\begin{aligned}
E &\rightarrow A \triangleright E \mid A \\
A &\rightarrow BA' \mid B \\
A' &\rightarrow \triangleright E \triangleleft AA' \mid \triangleright E \triangleleft A \mid \triangleleft AA' \mid \triangleleft A \\
B &\rightarrow CB' \mid C \\
B' &\rightarrow A' \ggg CB' \mid A' \ggg C \mid \ggg CB' \mid \ggg C \\
C &\rightarrow a \mid b
\end{aligned}
$$

Ahora sigue el último no-terminal C. De la anterior gramática se observa que no depende de los anteriores por lo tanto la gramática anterior permanece sin modificación. Por lo tanto la nueva gramática G_2' es:

$$
\begin{aligned}
E &\rightarrow A \triangleright E \mid A \\
A &\rightarrow BA' \mid B \\
A' &\rightarrow \triangleright E \triangleleft AA' \mid \triangleright E \triangleleft A \mid \triangleleft AA' \mid \triangleleft A \\
B &\rightarrow CB' \mid C \\
B' &\rightarrow A' \ggg CB' \mid A' \ggg C \mid \ggg CB' \mid \ggg C \\
C &\rightarrow a \mid b
\end{aligned}
$$

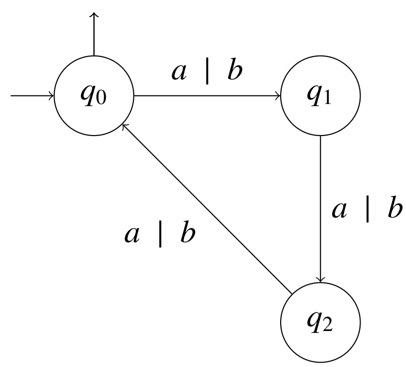
3. Autómata de estado finito . . . . 25 puntos

Encontrar el autómata finito determinista que reconozca el siguiente lenguaje:

$$L_3 = \{x \in \Sigma^* \mid |x| \mod 3 = 0\}$$

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

Solución:



4. Transformación de gramáticas . . . . 25 puntos

Dada la siguiente gramática G_4:

$$
\begin{aligned}
E &\rightarrow A \rhd E \mid A \\
A &\rightarrow E \lhd A \mid B \\
B &\rightarrow a \mid b
\end{aligned}
$$

Transformar la gramática G_4 en una nueva gramática G_4' sin reglas de copia G_4.

Solución:

Encontramos en primer lugar el conjunto de copia:

	Copy
E	E, A, B
A	A, B
B	B

Ahora se reescribe la gramática eliminando las reglas de copia y expandiendo las reglas en cada una de los no-terminales donde pertenece el conjunto de copia:

$$
\begin{aligned}
E &\rightarrow A \rhd E \mid E \lhd A \mid a \mid b \\
A &\rightarrow E \lhd A \mid a \mid b \\
B &\rightarrow a \mid b
\end{aligned}
$$

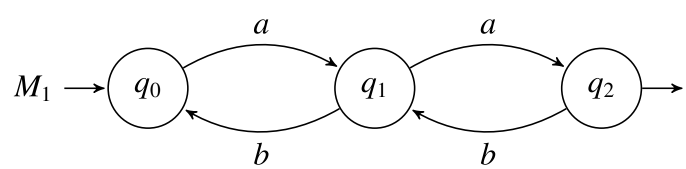
ST0270 Lenguajes formales y compiladores - 031 | **Universidad EAFIT**
Tercer parcial | **DIS**
Juan Francisco Cardona McCormick | **3 de octubre de 2013**

Nombre: ______________________________

Código: ______________________________

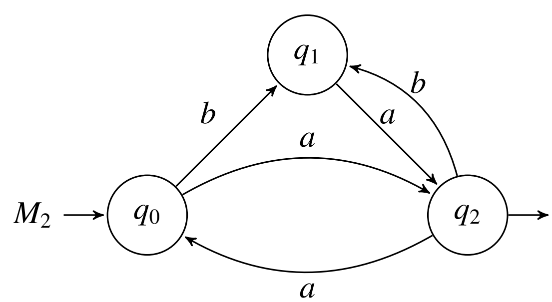
1. De autómata a expresiones regulares . . . . 25 puntos

Obtener la expresión regular definida por el siguiente autómata determinista M_1:



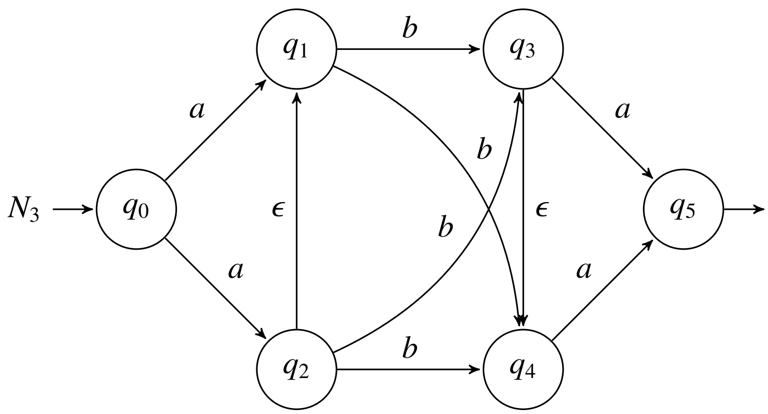
2. Minimizar autómatas . . . . 25 puntos

Minimizar el siguiente autómata M_2:



3. Algoritmo de Berry y Sethi - Aplicado .... 25 puntos

Dado el siguiente no determinista y con movimientos espontáneo N_3, definir un autómata determinista M_3, tal que $L(N_3) \equiv L(M_3)$, utilizando el algoritmo de Berry y Sethi:



4. Autómatas de gramáticas lineales por la izquierda .... 25 puntos

Calcular el autómata de estado finito para una gramática lineal por la izquierda G_4:

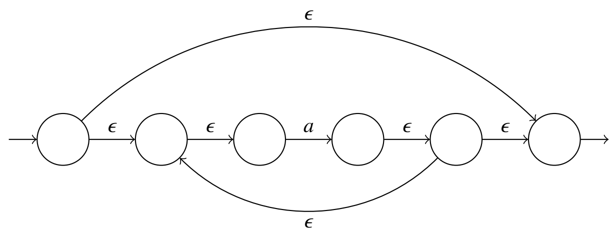
$$
\begin{array}{lcl}
S & \rightarrow & Ab \mid Sb \\
A & \rightarrow & Aa \mid Ab \mid B \\
B & \rightarrow & Bc \mid Bd \mid \epsilon
\end{array}
$$

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **11 de abril de 2013**

Nombre: ______________________________
Código: ______________________________

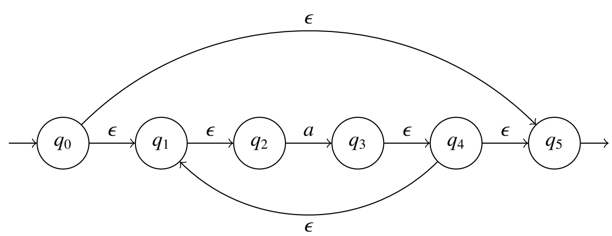
1. Eliminar movimientos espontáneos . . . . 25 puntos

Eliminar los movimientos espontáneos de siguiente autómata no determinista N_1:



Solución:

Enumeramos el automata N_1:



Obtenemos al gramática correspondiente G_1:

$$
\begin{array}{lcl}
q_0 & \rightarrow & q_1 \mid q_5 \\
q_1 & \rightarrow & q_2 \\
q_2 & \rightarrow & a\, q_3 \\
q_3 & \rightarrow & q_4 \\
q_4 & \rightarrow & q_1 \mid q_5 \\
q_5 & \rightarrow & \epsilon
\end{array}
$$

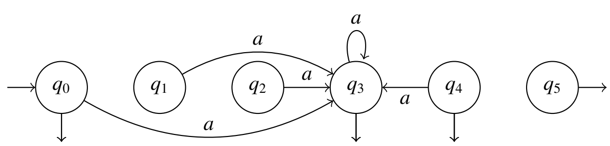
Calculamos el conjunto de copia de G_1:

No terminal	**COPY**
q_0	q_0, q_1, q_2, q_5
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4, q_1, q_2, q_5
q_4	q_4, q_1, q_2, q_5
q_5	q_5

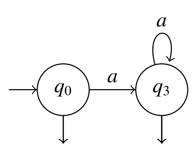
Se reescribe la gramática G_1' eliminando las reglas de copia:

$$
\begin{array}{lcl}
q_0 & \rightarrow & aq_3 \mid \epsilon \\
q_1 & \rightarrow & aq_3 \\
q_2 & \rightarrow & aq_3 \\
q_3 & \rightarrow & aq_3 \mid \epsilon \\
q_4 & \rightarrow & aq_3 \mid \epsilon \\
q_5 & \rightarrow & \epsilon
\end{array}
$$

Esto produce el siguiente autómata:



Limpiando el autómata se obtiene el siguiente nuevo autómata sin movimientos espontáneos:



2. Algoritmo de Berry y Sethi . . . . 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = c(a \mid b)^+b$$

Solución:

El lenguaje no es local por lo tanto se debe renumerar los la expresión regular r_2':

$$r_2' = c_1(a_2 \mid b_3)^+b_4$$

Calculamos el conjunto *Null*:

$$\begin{aligned}
Null(r_2') &= Null(c_1((a_2 \mid b_3)^+b_4)) \\
Null(r_2') &= Null(c_1) \cap Null((a_2 \mid b_3)^+b_4) \\
Null(r_2') &= \emptyset \cap (Null((a_2 \mid b_3))^+) \cap Null(b_4)) \\
Null(r_2') &= \emptyset \cap (Null(a_2 \mid b_3)) \cap \emptyset \\
Null(r_2') &= \emptyset
\end{aligned}$$

Calculamos el conjunto *Ini*:

$$\begin{aligned}
Ini(r_2') &= Ini(c_1((a_2 \mid b_3)^+b_4)) \\
Ini(r_2') &= Ini(c_1) \cup Null(c_1)Ini(((a_2 \mid b_3)^+b_4)) \\
Ini(r_2') &= \{c_1\} \cup \emptyset\ Ini(((a_2 \mid b_3)^+b_4)) \\
Ini(r_2') &= \{c_1\}
\end{aligned}$$

Calculamos el conjunto *Fin*:

$$\begin{aligned}
Fin(r_2') &= Fin(c_1((a_2 \mid b_3)^+b_4)) \\
Fin(r_2') &= Fin(((a_2 \mid b_3)^+b_4)) \cup Fin(c_1)Null(((a_2 \mid b_3)^+b_4)) \\
Fin(r_2') &= (Fin(b_4) \cup Fin((a_2 \mid b_3)^+)Null(b_4)) \cup \{c_1\}(Null(((a_2 \mid b_3)^+) \cap Null(b_4)))) \\
Fin(r_2') &= \{b_4\} \cup \emptyset) \cup \{c_1\}(Null((a_2 \mid b_3)) \cap \emptyset) \\
Fin(r_2') &= \{b_4\} \cup \{c_1\}(\emptyset) \\
Fin(r_2') &= \{b_4\} \cup \emptyset \\
Fin(r_2') &= \{b_4\}
\end{aligned}$$

Calculamos el conjunto *Dig*:

$$
\begin{aligned}
Dig(r_2') &= Dig(c_1((a_2 \mid b_3)^+ b_4)) \\
Dig(r_2') &= Dig(c_1) \cup Dig((a_2 \mid b_3)^+ b_4) \cup Fin(c_1) Ini((a_2 \mid b_3)^+ b_4) \\
Dig(r_2') &= \emptyset \cup (Dig((a_2 \mid b_3)^+) \cup Dig(b_4) \cup Fin((a_2 \mid b_3)^+) Ini(b_4)) \\
&\quad \cup \{c_1\}(Ini((a_2 \mid b_3)^+) \cup Null((a_2 \mid b_3)^+) Ini(b_4)) \\
Dig(r_2') &= ((Dig(a_2 \mid b_3) \cup Fin(a_2 \mid b_3) Ini(a_2 \mid b_3)) \cup \emptyset \\
&\quad \cup Fin(a_2 \mid b_3)\{b_4\}) \cup \{c_1\}(Ini((a_2 \mid b_3)) \cup Null((a_2 \mid b_3)) Ini(b_4)) \\
Dig(r_2') &= ((Dig(a_2) \cup Dig(b_3)) \cup (Fin(a_2) \cup Fin(b_3)))(Ini(a_2) \cup Fin(b_3)))) \\
&\quad \cup (Fin(a_2) \cup Fin(b_3))\{b_4\}) \cup \{c_1\}((Ini(a_2) \cup Ini(b_3))) \cup (Null(a_2) \cup Null(b_3)))\{b_4\}) \\
Dig(r_2') &= ((\emptyset \cup \emptyset) \cup (\{a_2\} \cup \{b_3\})))(\{a_2\} \cup \{b_3\})))) \\
&\quad \cup (\{a_2\} \cup \{b_3\}))\{b_4\}) \cup \{c_1\}((\{a_2\} \cup \{b_3\})) \cup (\emptyset \cup \emptyset))\{b_4\}) \\
Dig(r_2') &= (\emptyset \cup (\{a_2, b_3\}))(\{a_2, b_3\}) \cup (\{a_2, b_3\}\{b_4\}) \cup \{c_1\}((\{a_2, b_3\})) \cup (\emptyset)\{b_4\} \\
Dig(r_2') &= (\{a_2, b_3\}\{a_2, b_3\}) \cup \{a_2 b_4, b_3 b_4\} \cup \{c_1 a_2, c_1 b_3\} \cup \emptyset \\
Dig(r_2') &= \{a_2 a_2, a_2 b_3, b_3 a_2, b_3 b_3\} \cup \{a_2 b_4, b_3 b_4, c_1 a_2, c_1 b_3\} \\
Dig(r_2') &= \{a_2 a_2, a_2 b_3, b_3 a_2, b_3 b_3, a_2 b_4, b_3 b_4, c_1 a_2, c_1 b_3\}
\end{aligned}
$$

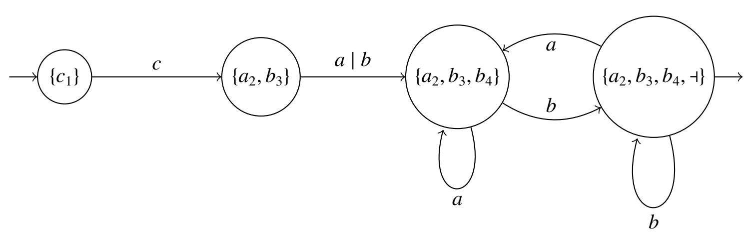
Calculo del conjunto de $Ini(r_2' \dashv)$:

$$
\begin{aligned}
Ini(r_2' \dashv) &= Ini(c_1((a_2 \mid b_3)^+ b_4 \dashv)) \\
Ini(r_2' \dashv) &= Ini(c_1) \cup Null(c_1) Ini(((a_2 \mid b_3)^+ b_4 \dashv)) \\
Ini(r_2' \dashv) &= \{c_1\} \cup \emptyset \; Ini(((a_2 \mid b_3)^+ b_4 \dashv)) \\
Ini(r_2' \dashv) &= \{c_1\}
\end{aligned}
$$

Se calcula el conjunto de los Fol:

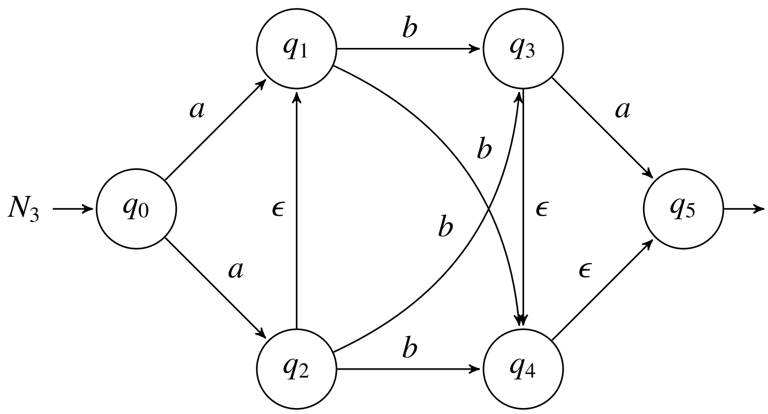
	Following
c_1	a_2, b_3
a_2	a_2, b_3, b_4
b_3	a_2, b_3, b_4
b_4	$\dashv$

Ahora calculamos el autómata determinista M_2:



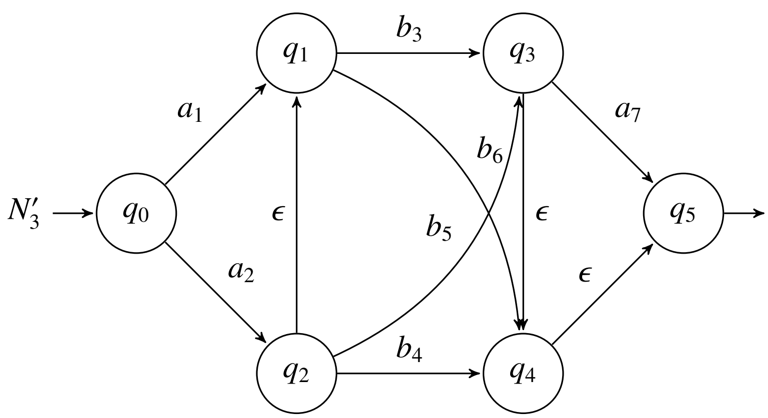
3. Algoritmo de Berry y Sethi - Aplicado . . . . 25 puntos

Dado el siguiente no determinista y con movimientos espontáneo N_3, definir un autómata determinista M_3, tal que $L(N_3) \equiv L(M_3)$, utilizando el algoritmo de Berry y Sethi:



Solución:

Numeramos el autómata N_3 en un nuevo autómata N'_3:



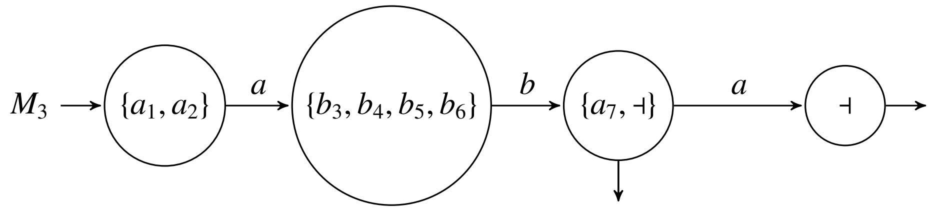
Calculamos el conjunto $Ini(L(N'_3) \dashv)$:

$$Ini(L(N'_3) \dashv) = \{a_1, a_2\}$$

Calculamos el conjunto Fol:

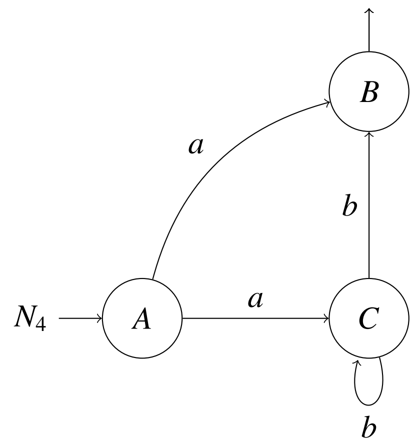
	Following
a_1	b_3, b_6
a_2	b_3, b_4, b_5, b_6
b_3	$a_7, \dashv$
b_4	$\dashv$
b_5	$a_7, \dashv$
b_6	$\dashv$
a_7	$\dashv$

Computamos el autómata M_3:

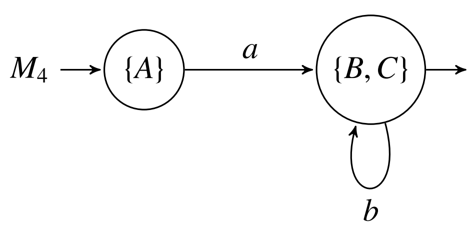


4. De autómata no determinista a autómata determinista . . . . 25 puntos

El siguiente autómata es no-determinista N_4, transforme el correspondiente autómata en uno determinista M_4, donde $L(N_4) \equiv L(M_4)$. Utilizando el conjunto de potencia.



Solución:



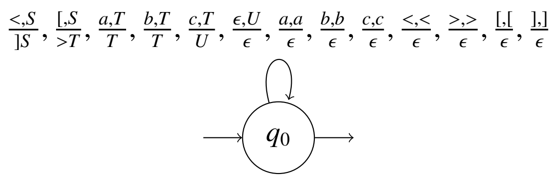
Nombre: ______________________________
Código: ______________________________

1. Gramáticas y Autómatas de pila .... 25 puntos

Dada la siguiente gramática independiente de contexto G_1 construir el correspondiente autómata de pila (posiblemente no determinista) que reconozca cadenas del lenguaje de $L(G_1)$.

$$
\begin{array}{lcl}
S & \rightarrow & <\ T\ ]\ \mid\ [\ T\ > \\
T & \rightarrow & aT\ \mid\ bT\ \mid\ cU \\
U & \rightarrow & \epsilon
\end{array}
$$

Solución:



2. Obtención de una gramática $LL(1)$ .... 50 puntos

La siguiente gramática G_2 no es $LL(1)$. Transformar la gramática G_2 en una gramática G_2' que cumpla con la condición $LL(1)$ y mostrar que la gramática G_2' cumple con la condición $LL(1)$.

$$\begin{array}{lcl} S & \rightarrow & S\,a \mid T \\ T & \rightarrow & aU \mid a \\ U & \rightarrow & b \end{array}$$

Entregar:

a) La tranformación de G_2 a G_2' indicando en cada paso qué tipo de transformación se hizo.

b) La demostración formal que la nueva gramática G_2' cumple con la condicción $LL(1)$.

Solución:

a) La gramática tiene recursividad por la izquierda en el no-terminal S, nos produce la siguiente gramática:

$$\begin{array}{lcl} S & \rightarrow & T \mid TS' \\ S' & \rightarrow & a \mid aS' \\ T & \rightarrow & aU \mid a \\ U & \rightarrow & b \end{array}$$

La gramática se puede factorizar en las no-terminales S S' y T, nos produce la siguiente gramática:

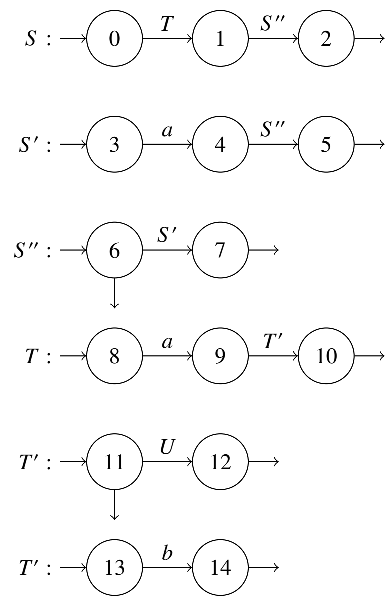
$$\begin{array}{lcl} S & \rightarrow & T\,(\epsilon \mid S') \\ S' & \rightarrow & a\,(\epsilon \mid S') \\ T & \rightarrow & a\,(\epsilon \mid U) \\ U & \rightarrow & b \end{array}$$

Añadimos dos no-terminales auxilares S'' y T' para manejar menor las alternativas:

$$\begin{array}{lcl} S & \rightarrow & T\,S'' \\ S' & \rightarrow & a\,S'' \\ S'' & \rightarrow & \epsilon \mid S' \\ T & \rightarrow & a\,T' \\ T' & \rightarrow & \epsilon \mid U \\ U & \rightarrow & b \end{array}$$

Tenemos ya nuestra gramática G_2'

b) Ahora demostramos que la gramática es G_2' es $LL(1)$. En primer lugar vamos a mostrar la red de autómatas:



Calculamos ahora los conjuntos siguientes de cada uno de los no-terminales:

$$
\begin{aligned}
Fol(S) &= \{\dashv\} \\
Fol(S') &= Fol(S'') = \{\dashv\} \\
Fol(S'') &= Fol(S) \cup Fol(S') = \{\dashv\} \\
Fol(T) &= Ini(S'') \cup Fol(S) = Ini(S') \cup \{\dashv\} = \{a, \dashv\} \\
Fol(T') &= Fol(T) = \{a, \dashv\} \\
Fol(U) &= Fol(T') = \{a, \dashv\}
\end{aligned}
$$

Ahora se calcula los conjuntos guías en la bifurcaciones:

$$
\begin{aligned}
Gui(6 \xrightarrow{S'} 7) &= Ini(L(6)L(7)) = \{a\} \\
Gui(6 \rightarrow) &= Fol(S'') = \{\dashv\} \\
Gui(11 \xrightarrow{U} 12) &= Ini(L(11)L(12)) = \{b\} \\
Gui(11 \rightarrow) &= Fol(T') = \{a, \dashv\}
\end{aligned}
$$

De lo anterior se muestra que la gramática G_2' cumple con la condición $LL(1)$.

3. Implementación de analizadores sintácticos por procedimientos recursivos . . . . . 25 puntos
La siguiente gramática independiente de contexto G_3 cumple con la condición $LL(1)$. Implemente el correspondiente analizador sintáctico utilizando para ello procedimientos recursivos.

$$\begin{array}{lcl} S & \rightarrow & T \mid '(' \ S \ + \ T ')' \\ T & \rightarrow & a \end{array}$$

Solución:

Para el procedimiento del no-terminal S

```
void S() {
   if (cc == '(') {
      cc = getNextToken();
      S();
      if (cc == '+') {
         cc = getNextToken()
         T();
         if (cc == ')') {
            cc = getNextToken()
         }
         else {
            error();
         }
      }
      else {
         error();
      }
   }
   else {
      T();
   }
}
```

Para el procedimiento del no-terminal T:

```
void T() {
   if (cc == 'a') {
      return;
   }
   else {
      error();
   }
}
```

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Aplicación de expresiones regulares**

La *altura de asterisco* de una expresión regular r en Σ, que se denota $sh(r)$, se define como sigue:

$$
\begin{aligned}
&sh(\emptyset) = 0 \\
&sh(\epsilon) = 0 \\
&sh(a) = 0 \; a \in \Sigma \\
&sh((rs) = (sh(r \cup s)) = max(sh(r), sh(s)) \\
&sh((r^*)) = sh(r) + 1.
\end{aligned}
$$

Encuentre la altura del asterisco de las siguientes expresiones regulares:

a) $r_1 = ((a \cup a^*a) \cup a)^*$
b) $r_1' = (aa^*)^*$

Solución:

a) Solución para la expresión $r_1 = ((a \cup a^*a) \cup a)^*$

$$
\begin{aligned}
sh(r_1) &= sh(((a \cup a^*a) \cup a)^*) \\
&= sh(((a \cup a^*a) \cup a)) + 1 \\
&= max(sh((a \cup a^*a)), sh(a)) + 1 \\
&= max(max(sh(a), sh(a^*a)), sh(a)) + 1 \\
&= max(max(sh(a), sh(a^*a)), sh(a)) + 1 \\
&= max(max(sh(a), max(sh(a^*), sh(a))), sh(a)) + 1 \\
&= max(max(sh(a), max(sh(a) + 1, sh(a))), sh(a)) + 1 \\
&= max(max(0, max(0 + 1, 0)), 0) + 1 \\
&= max(max(0, max(1, 0)), 0)) + 1 \\
&= max(max(0, 1), 0) + 1 \\
&= max(1, 0) + 1 \\
&= 1 + 1 \\
&= 2
\end{aligned}
$$

b) Solución para la expresión $r'_1 = (aa^*)^*$

$$\begin{aligned}
r'_1 &= sh((aa^*)^*) \\
&= sh((aa^*)) + 1 \\
&= max(sh(a), sh(a^*)) + 1 \\
&= max(0, sh(a^*)) + 1 \\
&= max(0, sh(a) + 1) + 1 \\
&= max(0, 0 + 1) + 1 \\
&= max(0, 1) + 1 \\
&= 1 + 1 \\
&= 2
\end{aligned}$$

(25 %) 2. **Expresiones regulares**
Encuentre la expresiones regular r_2 que correspondan al lenguaje definidos a continuación en forma recursiva del alfabeto $\Sigma = \{0, 1\}$.

Definición del lenguaje L_2: $\epsilon \in L_2$: si $x \in L_2$, entonces $001x$ y $x11$ son elementos de L; nada es parte de L_2 si no puede obtenerse de esos dos elementos.

Solución:
Una posible solución:

$$r_2 = 001((001)^* \mid (11)*) \mid ((001)^* \mid (11)^*)11 \mid \epsilon$$

(25 %) 3. **Gramáticas independientes de contexto**
Defina la gramática independiente de contexto G_3 para el siguiente lenguaje con alfabeto $\Sigma = \{0, 1\}$.

Definición del lenguaje L_3: $\epsilon \in L_3$: si $x \in L_3$, entonces $001x$ y $x11$ son elementos de L; nada es parte de L_3 si no puede obtenerse de eso dos elementos.

Solución:
Una posible solución sería la siguiente gramática G_3:

$$G_3 = (V = \{S\}, \Sigma = \{0, 1\}, P = \{S \rightarrow \epsilon \mid 001S \mid S11\}, S)$$

(25 %) 4. **Gramática limpia**

Transformar la siguiente gramática G_4 en una gramática independiente de contexto limpia G'_4:

$$G_4 = (V = \{A, B, C, D, S\}, \Sigma = \{a, b, c, d\},$$
$$P = \{S \rightarrow aBb \mid aAb \mid D, A \rightarrow b, B \rightarrow aCBb, C \rightarrow d, D \rightarrow S \mid c\}, S)$$

Solución:

En primer lugar verificamos si la gramática está bien definida:

$$def_0 = \{A, C, D\}$$

Este es el conjunto inicial de los no-terminales que producen terminales. Iteramos para obtener el siguiente nivel.

$$def_1 = def_0 \cup \{S\} = \{A, C, D, S\}$$

Iteramos de nuevo y obtenemos:

$$def_2 = def_1 \cup \emptyset = \{A, C, D, S\}$$

Ahora encontramos los no-terminales que no están definidos:

$$nodef = V \setminus def = \{A, B, C, D, S\} \setminus \{A, C, D, S\} = \{B\}$$

Generamos una nueva gramática G''_5 elimando el no-terminal B

$$G''_4 = (V = \{A, C, D, S\}, \Sigma = \{a, b, c, d\}, P =$$
$$\{S \rightarrow aAb \mid D, A \mid b, C \rightarrow d, D \rightarrow S \mid c\}, S)$$

Ahora verificamos que no-terminales son alcanzables desde el axioma:

$$\{A, D\}$$

Por lo tanto generamos una nueva gramática G'''_4 eliminando el no-terminal C:

$$G'''_4 = (V = \{A, D, S\}, \Sigma = \{a, b, c, d\}, P = \{S \rightarrow aAb \mid D, A \mid b, D \rightarrow S \mid c\}, S)$$

Pero se observa que existe una derivación circular $S \Rightarrow D \Rightarrow S$. Esto implica que se debe eliminar la producción que produce esa circularidad y se obtiene finalmente la gramática G'_4:

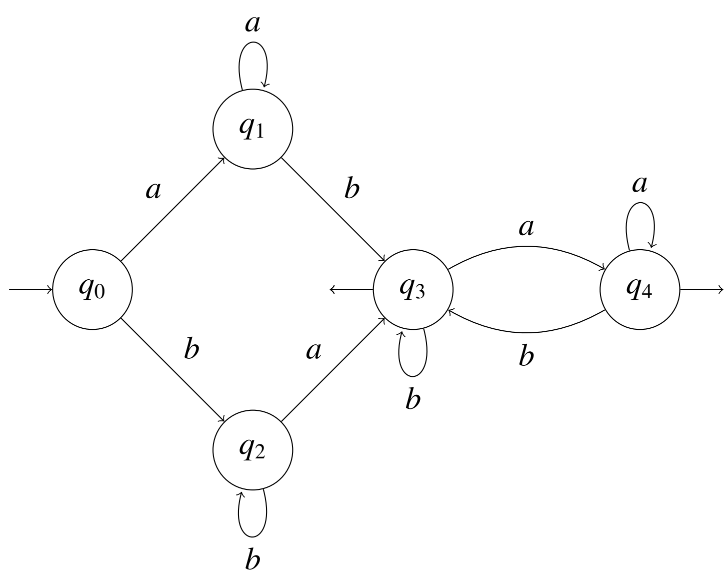
$$G'_4 = (V = \{A, D, S\}, \Sigma = \{a, b, c, d\}, P = \{S \rightarrow aAb \mid D, A \mid b, D \rightarrow c\}, S)$$

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **11 de septiembre de 2014**

Nombre: ______________________________
Código: ______________________________

1. Mimización de autómata . . . . 25 puntos

Encontrar formalmente el autómata mínimo del siguiente autómata finito determinista D_1:



Solución:

Se hace la matriz para evaluar los casos:

q_1				
q_2				
q_3				
q_4				
	q_0	q_1	q_2	q_3

Se compara los estados finales con lo que no los son: q_4 con q_0, q_1, q_2 y q_4 con q_0, q_1, q_2. Todos son distinguibles:

q_1				
q_2				
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3

Ahora comparamos q_0 y q_1. Con el carácter a:

$$\begin{aligned}\delta(q_0, a) &= q_1\\ \delta(q_1, a) &= q_1\end{aligned}$$

Ambos estados son no finales, lucen indistinguibles, pero miramos que pasa con b:

$$\begin{aligned}\delta(q_0, b) &= q_2\\ \delta(q_1, b) &= q_3\end{aligned}$$

Con b ambos estados son distinguibles, por lo tanto los estados q_0 y q_1 son distinguibles.

q_1	X			
q_2				
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3

Ahora con los estados q_0 y q_2, con el carácter a:

$$\begin{aligned}\delta(q_0, a) &= q_1\\ \delta(q_2, a) &= q_3\end{aligned}$$

Los estados ya se comprobaron que son distiguibles por lo tanto q_0 y q_2 son distinguibles:

q_1	X			
q_2	X			
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3

Ahora examinamos q_1 y q_2. en primer lugar miramos con el carácter a y se obtiene:

$$\begin{aligned}\delta(q_1, a) &= q_1\\ \delta(q_2, a) &= q_3\end{aligned}$$

Ambos estados son distiguibles por lo tanto ambos son distinguibles:

q_1	X			
q_2	X	X		
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3

Nos falta examinar q_3 y q_4. En primer lugar miramos con el carácter a, y se obtiene:

$$\begin{aligned} \delta(q_3, a) &= q_4 \\ \delta(q_4, a) &= q_4 \end{aligned}$$

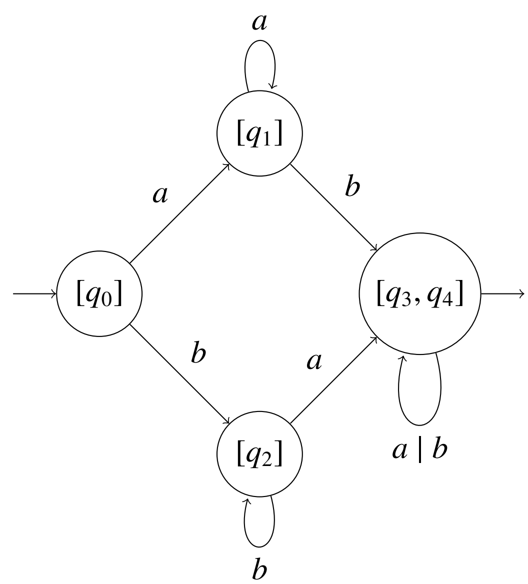
Ambos estados son finales, no se puede distinguir de los dos dos. Ahora con el carácter b:

$$\begin{aligned} \delta(q_3, b) &= q_3 \\ \delta(q_4, b) &= q_3 \end{aligned}$$

Ambas son finales, con el carácter b son indistiguibles. Por lo tanto ambos estados son indistiguibles:

q_1	X			
q_2	X	X		
q_3	X	X	X	
q_4	X	X	X	$[q_3, q_4]$
	q_0	q_1	q_2	q_3

Por lo tanto el autómata no se encuentra minimizado. El nuevo automáta es el siguiente:



2. Transformación de gramáticas....................25 puntos

Dada la siguiente gramática G_2:

$$
\begin{aligned}
S &\rightarrow A \sqcup S \mid A \\
A &\rightarrow S \oplus A \mid B \\
B &\rightarrow A \uplus C \mid C \\
C &\rightarrow [\ S\ ] \mid a \mid b
\end{aligned}
$$

Transformar la gramática G_2 en una nueva gramática G_2' que elimina la recursividad por la izquierda de G_2.

Solución:

En primer lugar organizamos los no-terminales en un orden:

$$S \rightarrow 1,\ A \rightarrow 2,\ B \rightarrow 3,\ C \rightarrow 4$$

Luego comenzamos por el primer no-terminal S y verificamos si tiene recursividad inmediata:

No tiene recursividad inmediata por lo tanto la gramática sigue igual.

$$
\begin{aligned}
S &\rightarrow A \sqcup S \mid A \\
A &\rightarrow S \oplus A \mid B \\
B &\rightarrow A \uplus C \mid C \\
C &\rightarrow [\ S\ ] \mid a \mid b
\end{aligned}
$$

Ahora miramos el segundo no-terminal A y expandimos las reglas que depende de las anteriores:

$$A \rightarrow A \oplus A \mid A \sqcup S \oplus A \mid B$$

Encontramos que hay recursividad por la izquierda; se procede a eliminarla añadiendo un nuevo no-terminal A':

$$\begin{aligned}
S &\rightarrow A \sqcup S \mid A \\
A &\rightarrow B \mid B\,A' \\
A' &\rightarrow \oplus A \mid \oplus A\,A' \mid \sqcup S \oplus A \mid \sqcup S \oplus A\,A' \\
B &\rightarrow A \uplus C \mid C \\
C &\rightarrow [\ S\ ] \mid a \mid b
\end{aligned}$$

Luego miramos el no-terminal B y expandimos la reglas que comiencen con los anteriores no-terminales:

$$B \rightarrow B \uplus C \mid B\,A' \uplus C \mid C$$

Encontramos que hay recursividad por la izquierda; se procede a eliminarla añadiendo un nuevo no-terminal B':

$$\begin{aligned}
S &\rightarrow A \sqcup S \mid A \\
A &\rightarrow B \mid B\,A' \\
A' &\rightarrow \oplus A \mid \oplus A\,A' \mid \sqcup S \oplus A \mid \sqcup S \oplus A\,A' \\
B &\rightarrow C \mid C\,B' \\
B' &\rightarrow \uplus C \mid \uplus C\,B' \mid A' \uplus C \mid A' \uplus C\,B' \\
C &\rightarrow [\ S\ ] \mid a \mid b
\end{aligned}$$

Finalmente, se revisa el último no-terminal C:

$$C \rightarrow [\ S\ ] \mid a \mid b$$

Se observa que no hay dependencia de ningún no-terminal anterior y no hay recursividad inmediata; por lo tanto la nueva gramática queda así:

$$\begin{aligned}
S &\rightarrow A \sqcup S \mid A \\
A &\rightarrow B \mid B\,A' \\
A' &\rightarrow \oplus A \mid \oplus A\,A' \mid \sqcup S \oplus A \mid \sqcup S \oplus A\,A' \\
B &\rightarrow C \mid C\,B' \\
B' &\rightarrow \uplus C \mid \uplus C\,B' \mid A' \uplus C \mid A' \uplus C\,B' \\
C &\rightarrow [\ S\ ] \mid a \mid b
\end{aligned}$$

3. De expresión regular a gramática independiente de contexto . . . . 25 puntos

Transformar la expresión regular $e_3 = a^*(a \cup c)^*(b \cup a)^*$ en una gramática independiente de contexto G_3, tal que $L(e_3) \equiv L(G_3)$.

Solución:

$$
\begin{aligned}
E_0 &\rightarrow\ E_1\ E_2\ E_3 \\
E_1 &\rightarrow\ \epsilon\ \mid\ a\ E_1 \\
E_2 &\rightarrow\ \epsilon\ \mid\ E_3\ E_2 \\
E_3 &\rightarrow\ a\ \mid\ c \\
E_4 &\rightarrow\ \epsilon\ \mid\ E_5\ E_4 \\
E_5 &\rightarrow\ b\ \mid a
\end{aligned}
$$

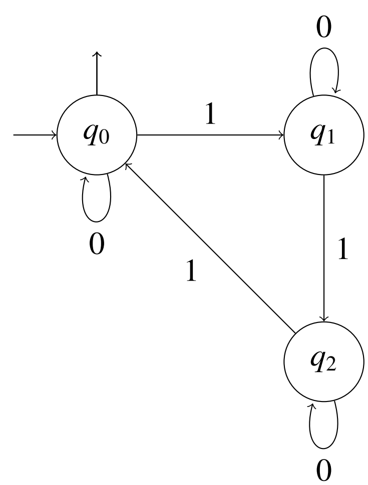
4. Autómata de estado finito . . . . 25 puntos

Encontrar el autómata finito determinista que reconozca el siguiente lenguaje:

$$L_4 = \{x \in \Sigma^* \mid |x|_1 \bmod 3 = 0\}$$

donde $\Sigma = \{0, 1\}$.

Solución:



Nombre: ______________________________
Código: ______________________________

1. Autómatas de pila .... 25 puntos

Implemente una autómata de pila $APND_1$, posiblemente, no determinista cuyo lenguaje $L(APND_1)$ sea igual al siguiente lenguaje L_1:

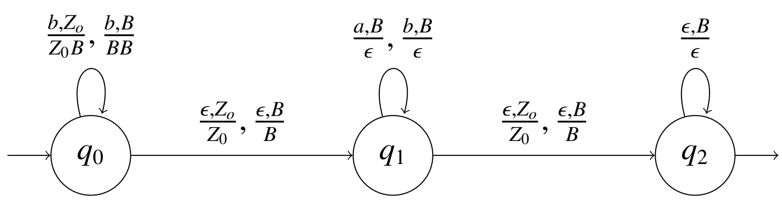
$$L_1 = \{b^n x \mid n \geq 0, x \in \{a, b\}^* \text{ y } |x| \leq n\}$$

Solución:

Son varias la formas de solucionar este punto, vamos a mostrar dos de ellas: directamente y la segunda a través de una gramática.

Directamente

Un estado inicial que recibe b y recuerda cuantas de ellas ha leído. El segundo estado se encarga de eliminar un número menor o igual a los valores b leídos en la primera etapa, la última etapa elimina los valores restantes que están en la pila hasta que la pila se encuentra vacía.



Gramática

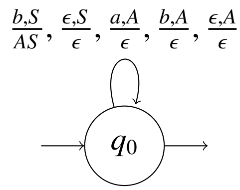
La siguiente es la gramática que crea el lenguaje solicitado:

$$\begin{array}{lcl} S & \rightarrow & bSA \mid \epsilon \\ A & \rightarrow & a \mid b \mid \epsilon \end{array}$$

Luego aplicamos el algoritmo para convertir la gramática en un autómata de pila no determinista:

	Rule	Movimiento
1	$S \rightarrow bSA$	`if` $cc = b \wedge top = S$ `then` $pop; push(AS); shift$
2	$S \rightarrow \epsilon$	`if` $top = S$ `then` pop
3	$A \rightarrow a$	`if` $cc = a \wedge top = A$ `then` $pop; push(\epsilon); shift$
4	$A \rightarrow b$	`if` $cc = b \wedge top = A$ `then` $pop; push(\epsilon); shift$
5	$A \rightarrow \epsilon$	`if` $top = A$ `then` pop

Se obtiene el siguiente autómata:


$\frac{b,S}{AS}, \frac{\epsilon,S}{\epsilon}, \frac{a,A}{\epsilon}, \frac{b,A}{\epsilon}, \frac{\epsilon,A}{\epsilon}$
q_0

2. Berry and Sethi . . . . 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = c(a \mid c \mid \epsilon)^+ b$$

Solución:

En primer lugar numeramos la expresión r_2:

$$r_2' = c_1(a_2 \mid c_3 \mid \epsilon)^+ b_4$$

Calculamos el estado inicial:

$$\begin{aligned}
Ini(r_2' \dashv) &= Ini(c_1((a_2 \mid c_3 \mid \epsilon)^+ b_4 \dashv)) \\
&= Ini(c_1) \cup Null(c_1) Ini((a_2 \mid c_3 \mid \epsilon)^+ b_4 \dashv) \\
&= \{c_1\} \cup \emptyset Ini((a_2 \mid c_3 \mid \epsilon)^+ b_4 \dashv) \\
&= \{c_1\} \cup \emptyset \\
&= \{c_1\}
\end{aligned}$$

Calculamos el conjunto Ini:

$$\begin{aligned}
Ini(r_2') &= Ini(c_1((a_2 \mid c_3 \mid \epsilon)^+ b_4)) \\
&= Ini(c_1) \cup Null(c_1) Ini((a_2 \mid c_3 \mid \epsilon)^+ b_4) \\
&= \{c_1\} \cup \emptyset Ini((a_2 \mid c_3 \mid \epsilon)^+ b_4) \\
&= \{c_1\} \cup \emptyset \\
&= \{c_1\}
\end{aligned}$$

Calculamos el conjunto Fin:

$$\begin{aligned}
Fin(r_2') &= Fin((c_1(a_2 \mid c_3 \mid \epsilon)^+) b_4) \\
&= Fin(b_4) \cup Fin((c_1(a_2 \mid c_3 \mid \epsilon)^+)) Null(b_4) \\
&= \{b_4\} \cup Fin((c_1(a_2 \mid c_3 \mid \epsilon)^+)) \emptyset \\
&= \{b_4\} \cup \emptyset \\
&= \{b_4\}
\end{aligned}$$

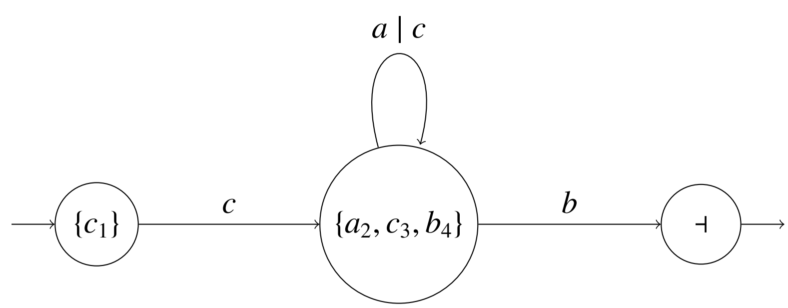
Calculamos el conjunto *Dig*:

$$
\begin{aligned}
Dig(r_2') &= Dig(c_1((a_2 \mid c_3 \mid \epsilon)^{+}b_4)) \\
&= Dig(c_1) \cup Dig((a_2 \mid c_3 \mid \epsilon)^{+}b_4) \cup Fin(c_1)Ini((a_2 \mid c_3 \mid \epsilon)^{+}b_4) \\
&= \emptyset \cup (Dig((a_2 \mid c_3 \mid \epsilon)^{+}) \cup Dig(b_4) \cup Fin((a_2 \mid c_3 \mid \epsilon)^{+})Ini(b_4)) \cup Fin(c_1)Ini((a_2 \mid c_3 \mid \epsilon)^{+}b_4) \\
&= ((Dig(a_2 \mid c_3 \mid \epsilon) \cup Fin(a_2 \mid c_3 \mid \epsilon)Ini(a_2 \mid c_3 \mid \epsilon)) \cup Dig(b_4) \cup Fin((a_2 \mid c_3 \mid \epsilon)^{+})Ini(b_4)) \cup \{c_1\}(Ini \\
&= ((Dig(a_2) \cup Dig(c_3) \cup Dig(\epsilon)) \cup \{a_2, c_3\}\{a_2, c_3\} \cup \emptyset \cup Fin(a_2 \mid c_3 \mid \epsilon)\{b_4\} \cup \{c_1\}(Ini(a_2 \mid c_3 \mid \epsilon) \cup N \\
&= ((\emptyset \cup \emptyset \cup \emptyset) \cup \{a_2a_2, a_2c_3, c_3a_2, c_3c_3\} \cup \{a_2, c_3\}\{b_4\}) \cup \{c_1\}(\{a_2, c_3\} \cup (Null(a_2) \cup Null(c_3) \cup Null(\\
&= (\emptyset \cup \{a_2a_2, a_2c_3, c_3a_2, c_3c_3\} \cup \{a_2b_4, c_3b_4\}) \cup \{c_1\}(\{a_2, c_3\} \cup (\emptyset \cup \emptyset \cup \{\epsilon\})\{b_4\} \\
&= \{a_2a_2, a_2c_3, c_3a_2, c_3c_3, a_2b_4, c_3b_4\} \cup (\{c_1\}(\{a_2, c_3\} \cup \{\epsilon\}\{b_4\}) \\
&= \{a_2a_2, a_2c_3, c_3a_2, c_3c_3, a_2b_4, c_3b_4\} \cup (\{c_1\}(\{a_2, c_3\} \cup \{b_4\}) \\
&= \{a_2a_2, a_2c_3, c_3a_2, c_3c_3, a_2b_4, c_3b_4\} \cup (\{c_1\}(\{a_2, c_3, b_4\}) \\
&= \{a_2a_2, a_2c_3, c_3a_2, c_3c_3, a_2b_4, c_3b_4\} \cup \{c_1a_2, c_1c_3, c_1b_4\} \\
&= \{a_2a_2, a_2c_3, c_3a_2, c_3c_3, a_2b_4, c_3b_4, c_1a_2, c_1c_3, c_1b_4\}
\end{aligned}
$$

Ahora calculamos el conjunto de siguientes:

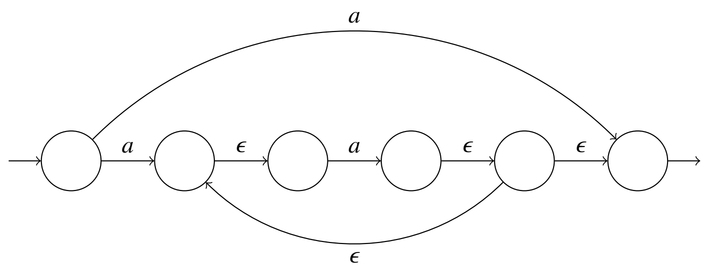
	Following
c_1	a_2, c_3, b_4
a_2	a_2, c_3, b_4
c_3	a_2, c_3, b_4
b_4	$\dashv$

A partir del estado inicial calculamos el autómata:



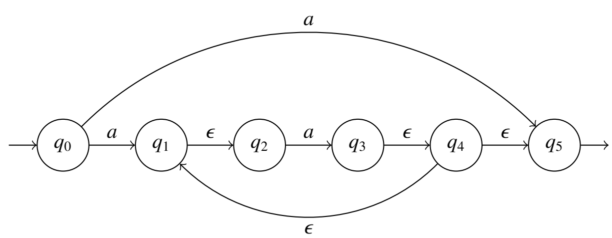
3. Eliminar movimientos espontáneos . . . . 25 puntos

Eliminar los movimientos espontáneos de siguiente autómata no determinista N_3:



Solución:

Enumeramos el automata N_3:



Obtenemos al gramática correspondiente G_3:

$$\begin{array}{lcl} q_0 & \rightarrow & aq_1 \mid aq_5 \\ q_1 & \rightarrow & q_2 \\ q_2 & \rightarrow & a\, q_3 \\ q_3 & \rightarrow & q_4 \\ q_4 & \rightarrow & q_1 \mid q_5 \\ q_5 & \rightarrow & \epsilon \end{array}$$

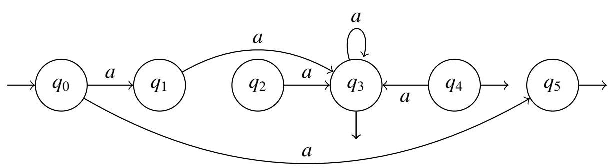
Calculamos el conjunto de copia de G_3:

No terminal	COPY
q_0	q_0
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4, q_1, q_2, q_5
q_4	q_4, q_1, q_2, q_5
q_5	q_5

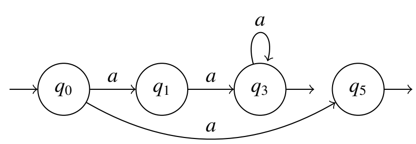
Se reescribe la gramática G'_3 eliminando las reglas de copia:

$$
\begin{array}{lcl}
q_0 & \rightarrow & aq_5 \mid aq_1 \\
q_1 & \rightarrow & aq_1 \mid aq_3 \\
q_2 & \rightarrow & aq_3 \\
q_3 & \rightarrow & aq_3 \mid \epsilon \\
q_4 & \rightarrow & aq_3 \mid \epsilon \\
q_5 & \rightarrow & \epsilon
\end{array}
$$

Esto produce el siguiente autómata:

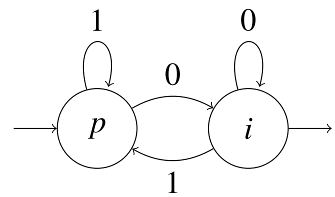


Limpiando el autómata se obtiene el siguiente nuevo autómata sin movimientos espontáneos:



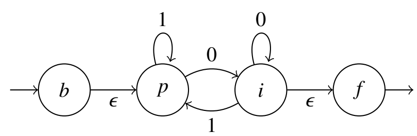
4. Método BMC . . . . 25 puntos

Dado el siguiente autómata determinista N_4 obtener su expresión regular por medio del método BMC:

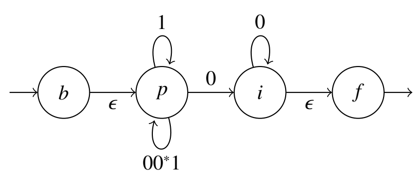


Solución:

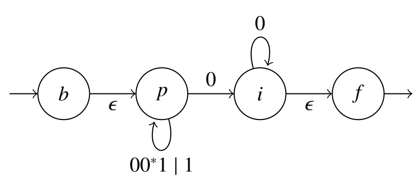
a. Utilizando el método BMC calculamos la expresión regular. Primero añadimos nuevos estados de inicio y de fin:



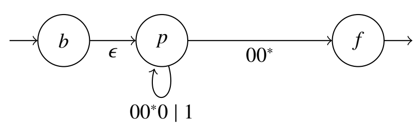
Calculamos una primera trayectoria entre $p \to i \to p$, observe que hay otra trayectoria entre $p \to i \to f$.



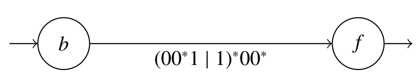
Renombramos los arcos que entran y salen del mismo p:



Tomamos ahora la trayectoria entre $p \to i \to f$:



Finalmente analizamos las trayectorias entre $b \to p \to f$:



Nombre: ______________________________
Código: ______________________________

1. Gramática $LL(1)$ . . . . 20 puntos

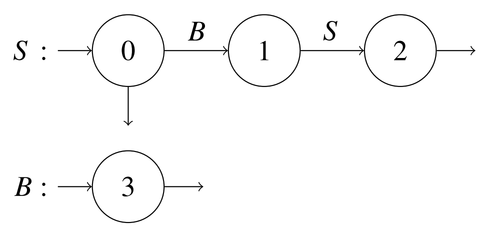
La siguiente gramática G_1: ¿Cumple con la condición $LL(1)$?

$$\begin{array}{rcl} S & \rightarrow & S\ B \mid \epsilon \\ B & \rightarrow & \epsilon \end{array}$$

Muestre la correspondiente evaluación de la condición $LL(1)$.

Solución:

Se obtiene la siguiente red de autómatas:



Se calcula las guía de las bifurcaciones:

$$\begin{array}{rcl} Gui(0 \xrightarrow{B} 1) & = & Ini(L(0)L(2)) \cup Fol(S) \\ & = & Ini(\{\epsilon\}\{\epsilon\}) \cup \{\dashv\} \\ & = & Ini(\{\epsilon\}) \cup \{\dashv\} \\ & = & \emptyset \cup \{\dashv\} \\ Gui(0 \rightarrow) & = & Fol(S) = \{\dashv\} \end{array}$$

De ambas bifurcaciones se encuentra que el resultado es:

$$Gui(0 \xrightarrow{B} 1) \cap Gui(0 \rightarrow) = \{\dashv\}$$

Por lo tanto no es $LL(1)$.

2. $LL(1)$ **y ANTLR3** . . . . 20 puntos

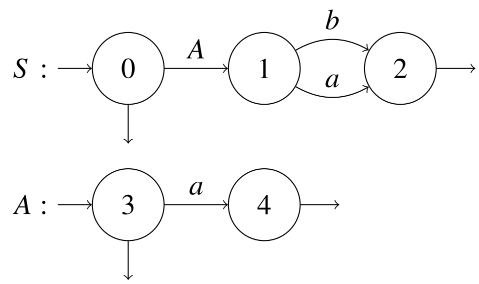
Mostrar si la siguiente gramática G_2 es $LL(1)$ y luego escribir la siguiente gramática utilizando ANTLR3[1], explicar los resultados obtenidos en ambos métodos.

$$\begin{array}{rcl} S & \rightarrow & A\,b \mid A\,c \\ A & \rightarrow & a \mid \epsilon \end{array}$$

Entregar el análisis $LL(1)$ manual y una impresión de la gramática en ANTLR3 y el pantallado de la gramática en ANTLR.

Solución:

Al obtener la red de autómatas:



Calculamos los conjuntos guías:

$$\begin{array}{rcl} Gui(1 \xrightarrow{a} 2) & = & \{a\} \\ Gui(1 \xrightarrow{b} 2) & = & \{b\} \\ Gui(3 \xrightarrow{b} 4) & = & \{a\} \\ Gui(3 \rightarrow) & = & Fol(A) = \{b, c\} \end{array}$$

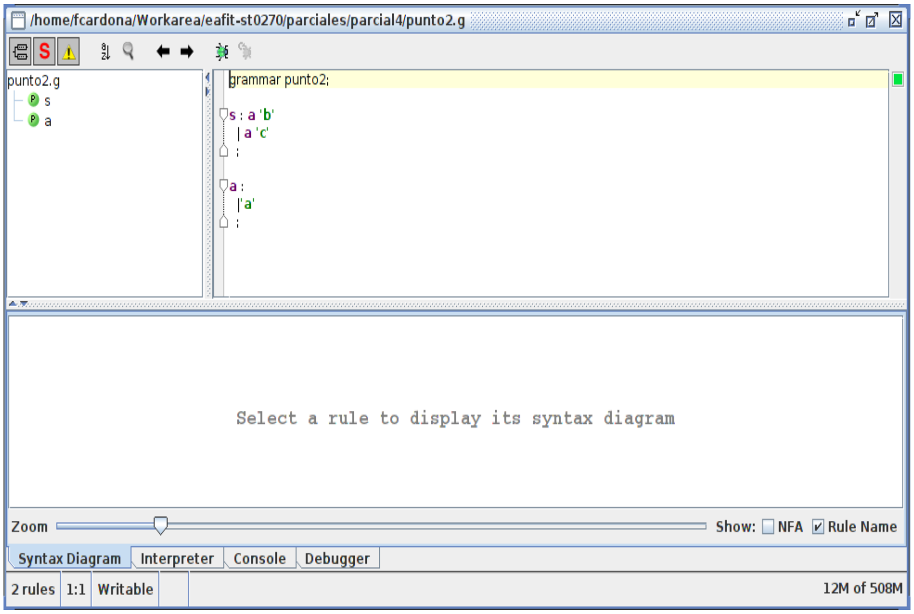
Se obtiene que la gramática es $LL(1)$.

La siguiente es la gramática procesada en ANTLR:

```
grammar punto2;
s : a 'b'
| a 'c'
;

a:
|'a'
;
```

[1] Precisamente ANTLRWorks

La siguiente es la salida de antlrworks que también comprueba que es cumple con las condiciones de una gramática $LL(*)$:



3. Gramática $LL(1)$ . . . . 20 puntos

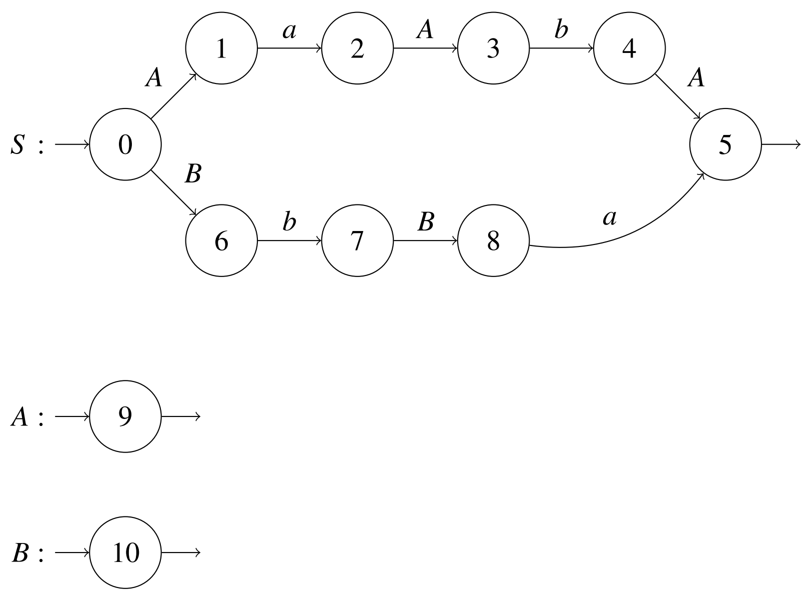
La siguiente gramática G_3: ¿Cumple con la condición $LL(1)$?

$$
\begin{array}{rcl}
S & \rightarrow & A\ a\ A\ bA \ \mid\ B\ b\ B\ a \\
A & \rightarrow & \epsilon \\
B & \rightarrow & \epsilon
\end{array}
$$

Muestre la correspondiente evaluación de la condición $LL(1)$.

Solución:

En primer lugar obtenemos la red de autómatas:



Calculamos los conjuntos guías en las bifurcaciones:

$$
\begin{array}{rcl}
Gui(0 \stackrel{A}{\rightarrow} 1) & = & Init(L(9)L(2)) \\
 & = & \{a\} \\
Gui(0 \stackrel{b}{\rightarrow} 6) & = & Init(L(10)L(6)) \\
 & = & \{b\}
\end{array}
$$

De lo anterior concluimos que la gramática es $LL(1)$.

4. Obtención de una gramática $LL(1)$ . . . . 40 puntos
La siguiente gramática G_4 no es $LL(1)$. Transformar la gramática G_4 en una gramática G'_4 que cumpla con la condición $LL(1)$ y mostrar que la gramática G'_4 cumple con la condición $LL(1)$.

$$\begin{array}{lcl} A & \rightarrow & Ab \mid Bc \mid C \\ B & \rightarrow & bC \mid b \\ C & \rightarrow & c \end{array}$$

Entregar:

a) La tranformación de G_4 a G'_4 indicando en cada paso qué tipo de transformación se hizo.
b) La demostración formal que la nueva gramática G'_4 cumple con la condicción $LL(1)$.

Solución:

a) Expandimos el no-terminal C en los lados derechos de A y B:

$$\begin{array}{lcl} A & \rightarrow & Ab \mid Bc \mid c \\ B & \rightarrow & bc \mid b \\ C & \rightarrow & c \end{array}$$

Limpiamos la gramática obtenida por que el no-terminal C no es alcanzable:

$$\begin{array}{lcl} A & \rightarrow & Ab \mid Bc \mid c \\ B & \rightarrow & bc \mid b \end{array}$$

Expandimos el no-terminal B en A:

$$\begin{array}{lcl} A & \rightarrow & Ab \mid bcc \mid bc \mid c \\ B & \rightarrow & bc \mid b \end{array}$$

Eliminamos el no-terminal B de la gramática pues no es alcanzable:

$$A \rightarrow Ab \mid bcc \mid bc \mid c$$

Factorizamos y creamos un nuevo no-terminal auxiliar D:

$$\begin{array}{lcl} A & \rightarrow & Ab \mid bcD \mid c \\ D & \rightarrow & c \mid \epsilon \end{array}$$

Ahora eliminamos la recursividad por la izquierda:

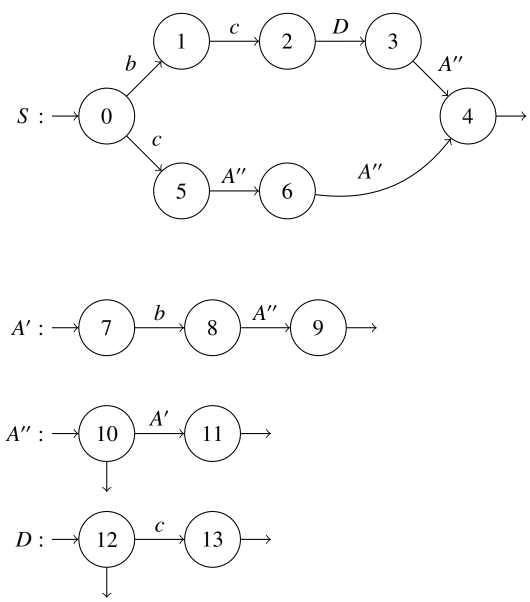
$$\begin{array}{rcl} A & \rightarrow & bcD \mid bcDA' \mid c \mid cA' \\ A' & \rightarrow & b \mid bA' \\ D & \rightarrow & c \mid \epsilon \end{array}$$

Factorizamos y añadimos un auxiliar:

$$\begin{array}{rcl} A & \rightarrow & bcDA'' \mid cA'' \\ A' & \rightarrow & b \mid bA' \\ A'' & \rightarrow & \epsilon \mid A' \\ D & \rightarrow & c \mid \epsilon \end{array}$$

Ya tenemos nuestra gramática G'_4.

b) Ahora demostramos que la gramática es G'_4 es $LL(1)$. En primer lugar vamos a mostrar la red de autómatas:



Ahora se calcula los conjuntos guías en la bifurcaciones:

$$\begin{array}{rcl} Gui(0 \xrightarrow{b} 1) & = & \{b\} \\ Gui(0 \xrightarrow{c} 5) & = & \{c\} \end{array}$$

De lo anterior se demuestra que la bifurcación en 0 es $LL(1)$.

$$\begin{array}{rcl} Gui(10 \xrightarrow{A'} 11) & = & Ini(L(7)L(12)) \\ & = & \{b\} \\ Gui(10 \rightarrow) & = & Fol(A'') \\ & = & Fol(A') \cup Fol(A) \\ & = & Fol(A'') \cup \{\dashv\} \\ & = & \{\dashv\} \end{array}$$

De lo anterior se demuestra que la bifurcación en 10 es $LL(1)$.

$$
\begin{array}{rcl}
Gui(12 \xrightarrow{A'} 13) & = & \{c\} \\
Gui(12 \rightarrow) & = & Fol(D) \\
& = & Init(3) \cup Fol(A) \\
& = & Init(10) \cup \{\dashv\} \\
& = & Init(7) \cup \{\dashv\} \\
& = & \{b\} \cup \{\dashv\} \\
& = & \{b\ ,\dashv\}
\end{array}
$$

De lo anterior se demuestra que la bifurcación en 12 es $LL(1)$.

De lo anterior se muestra que la gramática G_4' cumple con la condición $LL(1)$.

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Primer parcial **DIS**
Juan Francisco Cardona McCormick **14 de agosto de 2014**

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Gramáticas ambiguas**

La gramática independiente de contexto G_1

$$\begin{array}{lcl} \langle\lambda_{term}\rangle & := & x \\ \langle\lambda_{term}\rangle & := & \lambda x.\langle\lambda_{term}\rangle \\ \langle\lambda_{term}\rangle & := & \langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \end{array}$$

a) Demuestre que G_1 es una gramática ambigua.

b) Si G_1 es ambigua genere una gramática G_1', tal que $L(G_1) \equiv L(G_1')$.

Solución:

a) Calculamos dos derivación de la cadena xxx:

$$\begin{array}{lcl} \langle\lambda_{term}\rangle & \Rightarrow & \langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,\langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,x\,\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,x\,x \end{array}$$

$$\begin{array}{lcl} \langle\lambda_{term}\rangle & \Rightarrow & \langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \\ & \Rightarrow & \langle\lambda_{term}\rangle\langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,\langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,x\,\langle\lambda_{term}\rangle \\ & \Rightarrow & x\,x\,x \end{array}$$

Se obtienen dos árboles de derivación:

```
        ⟨λterm⟩                          ⟨λterm⟩
        /     \                          /     \
  ⟨λterm⟩    ⟨λterm⟩               ⟨λterm⟩    ⟨λterm⟩
     |       /     \               /     \        |
     x  ⟨λterm⟩  ⟨λterm⟩       ⟨λterm⟩  ⟨λterm⟩   x
            |        |            |        |
            x        x            x        x
```

b) Eliminar la ambigüedad de la gramática G_1 en la nueva gramática G_1':

$$\begin{array}{lcl} \langle\lambda_{term}\rangle & := & x \\ \langle\lambda_{term}\rangle & := & \lambda x.\langle\lambda_{term}\rangle \\ \langle\lambda_{term}\rangle & := & x\,\langle\lambda_{term}\rangle \\ \langle\lambda_{term}\rangle & := & \lambda x.\,\langle\lambda_{term}\rangle\langle\lambda_{term}\rangle \end{array}$$

(25 %) 2. **Expresiones regulares**
Dado el alfabeto $\Sigma = \{x, y, z\}$

a) Defina una expresión regular $r_{2,1}$ cuyo lenguaje $L(r_{2,1})$ genere cadenas con un número par de y.

b) Defina una expresión regular $r_{2,2}$ cuyo lenguaje $L(r_{2,2})$ genere cadena que no contenga la cadena xyz.

Solución:

1. $((x \mid z)^* y (x \mid z)^* y (x \mid z)^*)^*$
2. $(x \mid y \mid z)^* \setminus (xyz)$

(25 %) 3. **Gramáticas independientes de contexto**
Defina la gramática independiente de contexto G_3 para el siguiente lenguaje con alfabeto $\Sigma = \{a, b\}$.

Definición del lenguaje L_3**:** $\epsilon \in L_3$: si $x \in L_3$, entonces $abxa$ y $axbb$ son elementos de L; nada es parte de L_3 si no puede obtenerse de eso dos elementos.

Solución:
Una posible solución sería la siguiente gramática G_3:

$$G_3 = (V = \{S\}, \Sigma = \{a, b\}, P = \{S \to \epsilon \mid abSa \mid aSbb\}, S)$$

(25 %) 4. **Gramática limpia**

Transformar la siguiente gramática G_4 en una gramática independiente de contexto limpia G'_4:

$$G_4 = (V = \{A, S\}, \Sigma = \{a, b\}, P = \{S \rightarrow aASb, A \rightarrow b\}, S)$$

Solución:

En primer lugar verificamos si la gramática está bien definida:

$$def_0 = \{A\}$$

Este es el conjunto inicial de los no-terminales que producen terminales. Iteramos para obtener el siguiente nivel.

$$def_1 = def_0 \cup \emptyset = \{A\}$$

Ahora encontramos los no-terminales que no están definidos:

$$nodef = V \setminus def = \{A, S\} \setminus \{A\} = \{S\}$$

Generamos una nueva gramática G'_4 eliminando el no-terminal S

$$G'_4 = (V = \{A\}, \Sigma = \{b\}, P = \{A \rightarrow b\}, S)$$

Ahora verificamos que no-terminales son alcanzables desde el axioma:

$$\{A\}$$

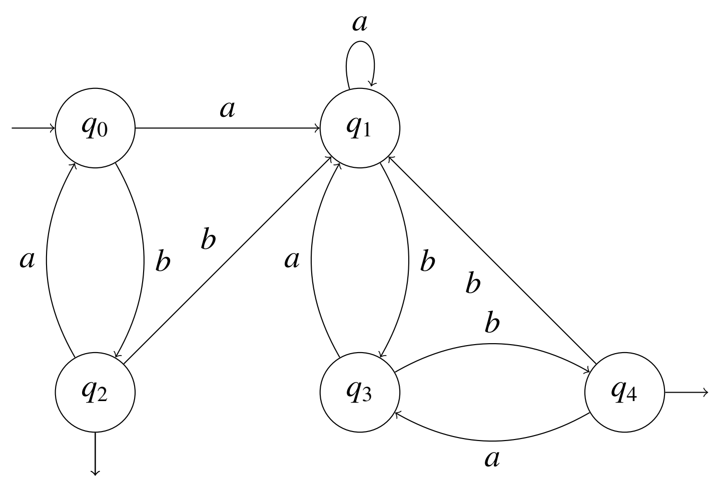
Nuestra gramática final es G'_4.

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **11 de septiembre de 2014**

Nombre: ______________________________
Código: ______________________________

1. Mimización de autómata .... 25 puntos

Encontrar formalmente el autómata mínimo del siguiente autómata finito determinista D_1:



Solución:

Se hace la matriz para evaluar los casos:

q_1				
q_2				
q_3				
q_4				
	q_0	q_1	q_2	q_3

Se compara los estados finales: q_2 y q_4 con cada uno lo que no los son: q_0, q_1 y q_3. Obteniendo que todos ellos son distinguibles.

q_1				
q_2	X	X		
q_3			X	
q_4	X	X		X
	q_0	q_1	q_2	q_3

Ahora comparamos q_1 y q_0. Con el carácter a:

$$\begin{aligned}\delta(q_1,a) &= q_1\\ \delta(q_0,a) &= q_1\end{aligned}$$

Ambos estados son no finales, lucen indistinguibles, pero miramos que pasa con b:

$$\begin{aligned}\delta(q_1,b) &= q_3\\ \delta(q_0,b) &= q_2\end{aligned}$$

Con b ambos estados nuevos son distinguibles, por lo tanto los estados q_0 y q_1 son distinguibles.

q_1	X			
q_2	X	X		
q_3			X	
q_4	X	X		X
	q_0	q_1	q_2	q_3

Ahora con los estados q_3 y q_0, con el carácter a:

$$\begin{aligned}\delta(q_3,a) &= q_1\\ \delta(q_0,a) &= q_1\end{aligned}$$

Ambos estados son indistinguibles. Miremos que pasa con b.

$$\begin{aligned}\delta(q_3,b) &= q_4\\ \delta(q_0,b) &= q_2\end{aligned}$$

Ambos son finales, entonces no podemos distinguir en q_3 y q_0, por lo tanto ambos son indistinguibles.

q_1	X			
q_2	X	X		
q_3	$[q_3,q_0]$		X	
q_4	X	X		X
	q_0	q_1	q_2	q_3

Ahora examinamos q_3 y q_1. en primer lugar miramos con el carácter a y se obtiene:

$$\begin{aligned}\delta(q_3,a) &= q_1\\ \delta(q_1,a) &= q_1\end{aligned}$$

Son indistiguibles. Examinamos ahora con el carácter b y obtenemos:

$$\begin{aligned}\delta(q_3, a) &= q_4 \\ \delta(q_1, a) &= q_3\end{aligned}$$

Ya habíamos encontrado que q_4 y q_3 son distinguibles por lo tanto q_3 y q_1 son distiguibles:

q_1	X			
q_2	X	X		
q_3	$[q_3, q_0]$	X	X	
q_4	X	X		X
	q_0	q_1	q_2	q_3

Nos falta examinar q_2 y q_4. En primer lugar miramos con el carácter a, y se obtiene:

$$\begin{aligned}\delta(q_4, a) &= q_0 \\ \delta(q_2, a) &= q_3\end{aligned}$$

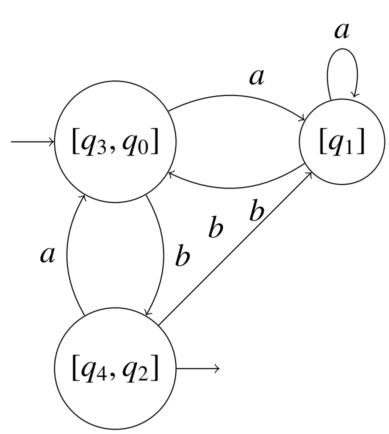
El nuevo estado es indistinguible. Ahora con el carácter b:

$$\begin{aligned}\delta(q_4, b) &= q_1 \\ \delta(q_2, b) &= q_1\end{aligned}$$

q_1 es indistinguible consigo mismo. Por lo tanto q_4 y q_2 son indistiguibles.

q_1	X			
q_2	X	X		
q_3	$[q_3, q_0]$	X	X	
q_4	X	X	$[q_4, q_2]$	X
	q_0	q_1	q_2	q_3

Por lo tanto el autómata se encuentra minimizado. El nuevo automáta es el siguiente:



2. Transformación de gramáticas . . . . 25 puntos a)

Encuentre el autómata D_2 que acepte el lenguaje generado por la gramática regular G_2 (el símbolo inicial es S), tal que $L(D_2) \equiv L(G_2)$:

$$\begin{aligned} S &\rightarrow xN \mid x \\ N &\rightarrow yM \mid y \\ M &\rightarrow zN \mid z \end{aligned}$$

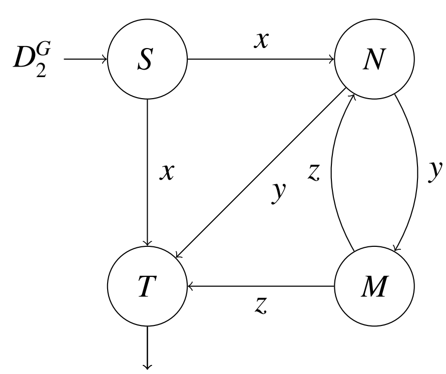
b) Luego, encuentre una expresión regular que represente el mismo lenguaje.

Solución:

a) Aunque la gramática G_2 es una gramática lineal por la derecha tiene un problema con las producciones del siguiente formato: $S \rightarrow x$, $N \rightarrow y$ y $M \rightarrow z$, por que no hay una correspondencia directa entre autómatas y gramáticas lineales por la derecha, por lo tanto es necesario añadir un nuevo no-terminal T y una nueva regla $T \rightarrow \epsilon$.

$$\begin{aligned} S &\rightarrow xN \mid xT \\ N &\rightarrow yM \mid yT \\ M &\rightarrow zN \mid zT \\ T &\rightarrow \epsilon \end{aligned}$$

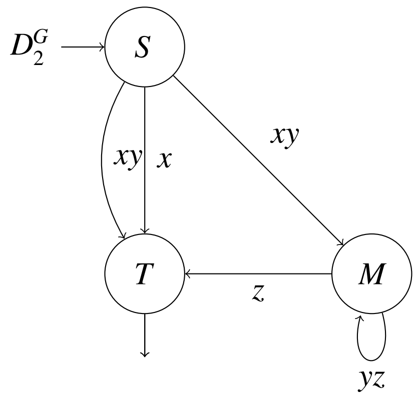
Ahora si se puede transformar directamente la gramática, en el correspondiente autómata.



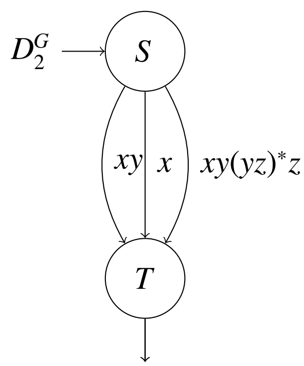
b) Utilizando el anterior autómata D_2 podemos producir una expresión regular a través del método BMC.

El anterior autómata tiene un estado inicial S que es único y no tiene arcos llegando a él y el estado final es T que es único y arcos saliendo de él. Por lo tanto podemos tomar éste autómata y utilizalo con el algoritmo BMC.

Eliminemos en primer lugar el estado N y se obtiene:



Ahora eliminando el nodo M se obtiene:



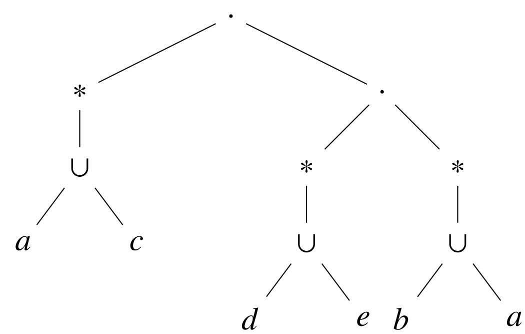
Nos da finalmente la expresión regular: $xy \mid x \mid xy(yz)^*z$

3. De expresión regular a gramática independiente de contexto .... 25 puntos

Transformar la expresión regular $e_3 = (a \cup c)^*(d \cup e)^*(b \cup a)^*$ en una gramática *independiente de contexto líneal por la derecha* G_3, tal que $L(e_3) \equiv L(G_3)$, utilizando únicamente transformación fundamentales.

Solución:

Se obtiene en primer lugar un árbol de sub-expresiones.



$$
\begin{aligned}
E_0 &\rightarrow E_1E_3 \\
E_1 &\rightarrow \epsilon \mid E_2E_1 \\
E_2 &\rightarrow a \mid c \\
E_3 &\rightarrow E_4E_6 \\
E_4 &\rightarrow \epsilon \mid E_5E_4 \\
E_5 &\rightarrow d \mid e \\
E_6 &\rightarrow \epsilon \mid E_7E_6 \\
E_7 &\rightarrow b \mid a
\end{aligned}
$$

La anterior gramática no es lineal, vamos hacer línea a través de transformaciones básicas, expansión de no-terminales y creación de nuevos no terminales agrupando terminales. Son lineales por la derecha: E_7, E_5, E_2. Expandimos E_7 en E_6, E_5 en E_4 y E_2 en E_1 y obtenemos:

$$
\begin{aligned}
E_0 &\rightarrow\ E_1E_3 \\
E_1 &\rightarrow\ \epsilon\ \mid\ aE_1\ \mid\ cE_1 \\
E_2 &\rightarrow\ a\ \mid\ c \\
E_3 &\rightarrow\ E_4E_6 \\
E_4 &\rightarrow\ \epsilon\ \mid\ dE_4\ \mid\ eE_4 \\
E_5 &\rightarrow\ d\ \mid\ e \\
E_6 &\rightarrow\ \epsilon\ \mid\ bE_6\ \mid\ aE_6 \\
E_7 &\rightarrow\ b\ \mid\ a
\end{aligned}
$$

Limpiamos la gramática:

$$
\begin{aligned}
E_0 &\rightarrow\ E_1E_3 \\
E_1 &\rightarrow\ \epsilon\ \mid\ aE_1\ \mid\ cE_1 \\
E_3 &\rightarrow\ E_4E_6 \\
E_4 &\rightarrow\ \epsilon\ \mid\ dE_4\ \mid\ eE_4 \\
E_6 &\rightarrow\ \epsilon\ \mid\ bE_6\ \mid\ aE_6
\end{aligned}
$$

Aun la gramática no es lineal por la derecha. Expandimos E_1 en E_0 y E_4 en E_3:

$$
\begin{aligned}
E_0 &\rightarrow\ E_3 \mid\ aE_1E_3\ \mid\ cE_1E_3 \\
E_1 &\rightarrow\ \epsilon\ \mid\ aE_1\ \mid\ cE_1 \\
E_3 &\rightarrow\ E_6 \mid\ dE_4E_6\ \mid\ eE_4E_6 \\
E_4 &\rightarrow\ \epsilon\ \mid\ dE_4\ \mid\ eE_4 \\
E_6 &\rightarrow\ \epsilon\ \mid\ bE_6\ \mid\ aE_6
\end{aligned}
$$

La gramática luce lineal pero tenemos aun algunas producciones que no son lineales por la derecha: $E_0\ \rightarrow\ aE_1E_3$, $E_o\ \rightarrow\ cE_1E_3$, $E_3\ \rightarrow dE_4E_6$, $E_3\ toeE_4E_6$. Entonces, agrupamos los dos no-terminales: E_1E_3 en $\langle E_1E_3\rangle$, E_4E_6 en $\langle E_4E_6\rangle$ y se obtiene:

$$
\begin{aligned}
E_0 &\rightarrow\ E_3 \mid\ a\langle E_1E_3\rangle\ \mid\ c\langle E_1E_3\rangle \\
\langle E_1E_3\rangle &\rightarrow E_1E_3 \\
E_1 &\rightarrow\ \epsilon\ \mid\ aE_1\ \mid\ cE_1 \\
E_3 &\rightarrow\ E_6 \mid\ d\langle E_4E_6\rangle\ \mid\ e\langle E_4E_6\rangle \\
\langle E_4E_6\rangle &\rightarrow E_4E_6 \\
E_4 &\rightarrow\ \epsilon\ \mid\ dE_4\ \mid\ eE_4 \\
E_6 &\rightarrow\ \epsilon\ \mid\ bE_6\ \mid\ aE_6
\end{aligned}
$$

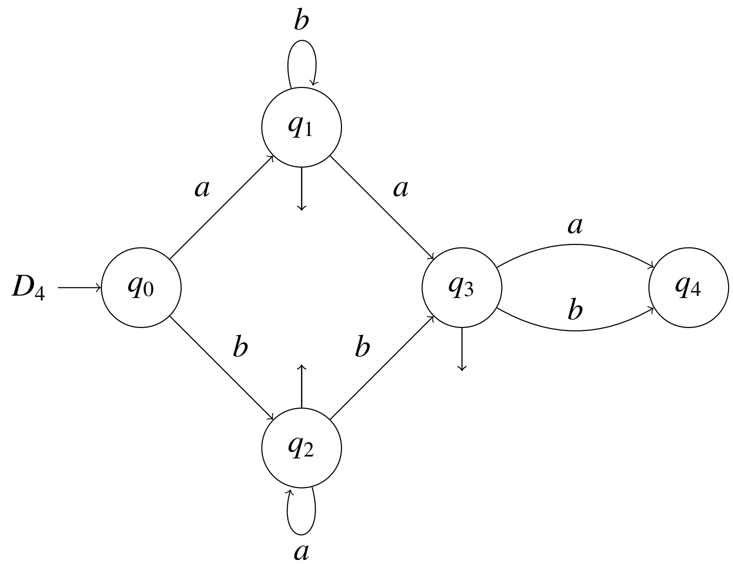
Todavía no es, expandimos E_1 en $\langle E_1E_3\rangle$ y E_4 en $\langle E_4E_6\rangle$ y agrupamos y se obtiene:

$$
\begin{aligned}
E_0 &\rightarrow\ E_3 \mid\ a\langle E_1E_3\rangle\ \mid\ c\langle E_1E_3\rangle \\
\langle E_1E_3\rangle &\rightarrow E_3\ \mid\ a\langle E_1E_3\rangle\ \mid\ c\langle E_1E_3\rangle \\
E_1 &\rightarrow\ \epsilon\ \mid\ aE_1\ \mid\ cE_1 \\
E_3 &\rightarrow\ E_6 \mid\ d\langle E_4E_6\rangle\ \mid\ e\langle E_4E_6\rangle \\
\langle E_4E_6\rangle &\rightarrow\ E_6 \mid\ d\langle E_4E_6\rangle\ \mid\ e\langle E_4E_6\rangle \\
E_4 &\rightarrow\ \epsilon\ \mid\ dE_4\ \mid\ eE_4 \\
E_6 &\rightarrow\ \epsilon\ \mid\ bE_6\ \mid\ aE_6
\end{aligned}
$$

Ahora ya tenemos una gramática lineal por la derecha.

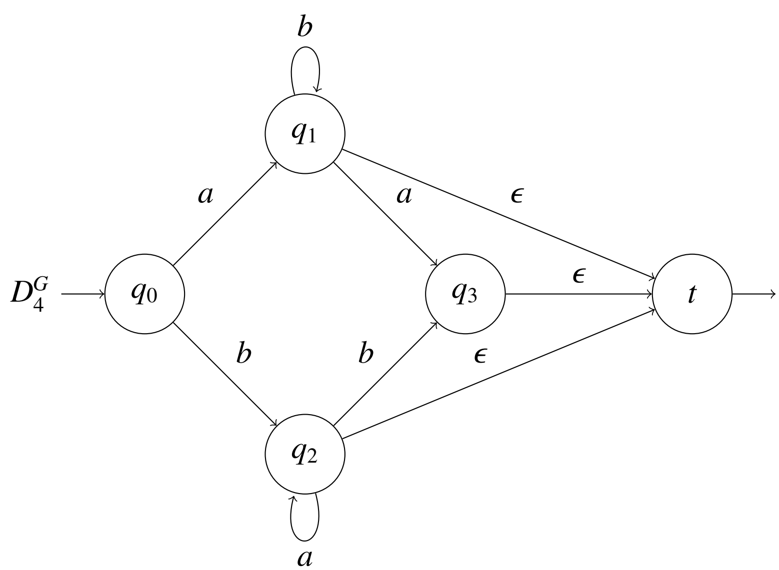
4. De autómata a expresión regular . . . . 25 puntos

Construya la expresión regular r_4 que tenga el mismo lenguaje aceptado por el siguiente autómata determinista D_4, tal que $L(r_4) \equiv L(D_4)$.

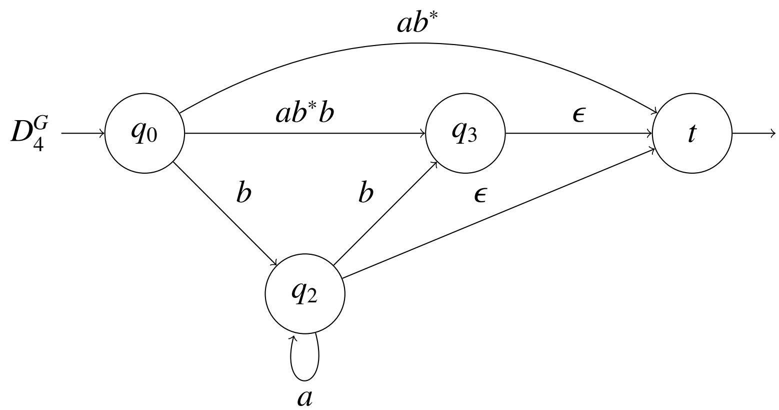


Solución:

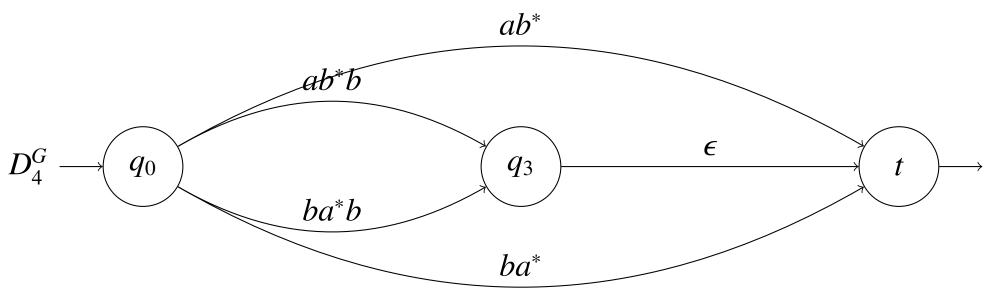
Encontramos que el autómata a excepción del estado q_4 el autómata es completo, y el estado q_4 es un estado sumidero, eliminamos el estado q_4 y añadimos el estado t que es el nuevo estado final t haciendo este el único estado final y construimos un nuevo autómata D_4^G.



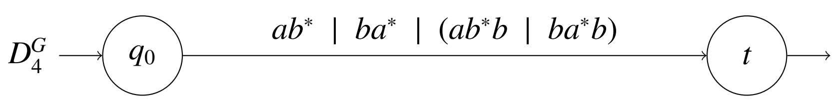
Ahora eliminamos el estado q_1 en D_4^G y obtenemos:



Ahora eliminamos q_2 de D_4^G:



Ahora eliminamos a q_3 en D_4^G y se obtiene:



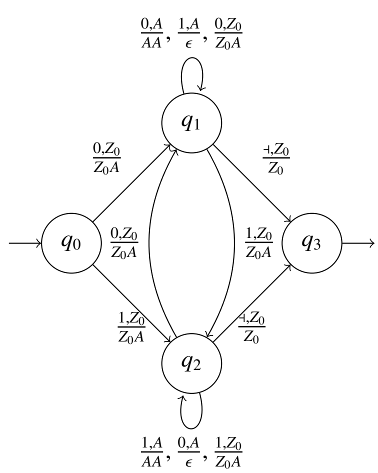
Nombre: ______________________________
Código: ______________________________

1. Autómatas de pila .... 25 puntos

Un *autómata contador* es un autómata de pila que tiene sólo dos símbolos en la pila, A, y Z_0 en el que la cadena en la pila es siempre de la forma $A^n Z_0$ para alguna $n \geq 0$. (En otras palabras, el único cambio posible en el contenido de la pila es le del número de A's de la pila). Y en relación con algunos lenguajes de gramáticas independientes de contexto, como $\{0^i 1^i \mid i \geq 0\}$, el autómata de pila evidente que acepta al lenguaje es, de hecho, un autómata de pila contador. Construya un autómata contador que acepte al lenguaje L_1:

$$L_1 = \{x \in \{0,1\}^* \mid |x|_0 = |x|_1\}$$

Solución:



2. Berry and Sethi .... 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = (ab \mid b)^* ba$$

Solución:

Numeramos r_2:

$$r_2' = (a_0 b_1 \mid b_2)^* b_3 a_4$$

Calculamos $Fin(r_2')$:

$$\begin{aligned}
Fin(r_2') &= Fin((a_0 b_1 \mid b_2)^* b_3 a_4) \\
&= Fin(a_4) \cup Fin((a_0 b_1 \mid b_2)^* b_3) Null(a_4) \\
&= \{a_4\} \cup Fin((a_0 b_1 \mid b_2)^* b_3)\emptyset \\
&= \{a_4\} \cup \emptyset \\
&= \{a_4\}
\end{aligned}$$

Calculamos $Dig(r_2')$:

$$\begin{aligned}
Dig(r_2') &= Dig((a_0 b_1 \mid b_2)^* b_3 a_4) \\
&= Dig((a_0 b_1 \mid b_2)^* b_3) \cup Dig(a_4) \cup Fin((a_0 b_1 \mid b_2)^* b_3) Ini(a_4) \\
&= Dig((a_0 b_1 \mid b_2)^*) \cup Dig(b_3) \cup Fin((a_0 b_1 \mid b_2)^*) Ini(b_3) \cup \emptyset \cup (Fin(b_3) \cup Fin((a_0 b_1 \mid b_2)^*) Null(b_3))\{a_4\} \\
&= Dig(a_0 b_1 \mid b_2) \cup Fin(a_0 b_1 \mid b_2) Ini(a_0 b_1 \mid b_2) \cup \emptyset \cup Fin(a_0 b_1 \mid b_2)\{b_3\} \cup (\{b_3\} \cup Fin(a_0 b_1 \mid b_2)\emptyset)\{a_4\} \\
&= Dig(a_0 b_1) \cup Dig(b_2) \cup (Fin(a_0 b_1) \cup Fin(b_2))(Ini(a_0 b_1) \cup Fin(b_2)) \cup (Fin(a_0 b_1) \cup Fin(b_2))\{b_3\} \cup (\{b_3\} \cup \emptyset)\{a_4\} \\
&= Dig(a_0) \cup Dig(b_1) \cup Fin(a_0) Ini(b_1) \cup \emptyset \cup (Fin(b_1) \cup Fin(a_0) Null(b_1) \cup \{b_2\})(Ini(a_0) \cup Null(a_0) Fin(b_1) \cup \{b_2\}) \cup (Fi \\
&= \emptyset \cup \emptyset \cup \{a_0\}\{b_1\} \cup (\{b_1\} \cup Fin(a_0)\emptyset \cup \{b_2\})(\{a_0\} \cup \emptyset Fin(b_1) \cup \{b_2\}) \cup (\{b_1\} \cup Fin(a_0)\emptyset \cup \{b_2\})\{b_3\} \cup \{b_3 a_4\} \\
&= \{a_0 b_1\} \cup (\{b_1\} \cup \emptyset \cup \{b_2\})(\{a_0\} \cup \emptyset \cup \{b_2\}) \cup (\{b_1\} \cup \emptyset \cup \{b_2\})\{b_3\} \cup \{b_3 a_4\} \\
&= \{a_0 b_1\} \cup (\{b_1\} \cup \{b_2\})(\{a_0\} \cup \{b_2\}) \cup (\{b_1\} \cup \{b_2\})\{b_3\} \cup \{b_3 a_4\} \\
&= \{a_0 b_1\} \cup (\{b_1, b_2\})(\{a_0, b_2\}) \cup (\{b_1, b_2\})\{b_3\} \cup \{b_3 a_4\} \\
&= \{a_0 b_1\} \cup \{b_1 a_0, b_1 b_2, b_2 a_0, b_2 b_2\} \cup \{b_1 b_3, b_2 b_3\} \cup \{b_3 a_4\} \\
&= \{a_0 b_1, b_1 a_0, b_1 b_2, b_2 a_0, b_2 b_2, b_1 b_3, b_2 b_3, b_3 a_4\}
\end{aligned}$$

A partir de $Dig(r_2')$ calculamos Fol (Conjunto de siguientes):

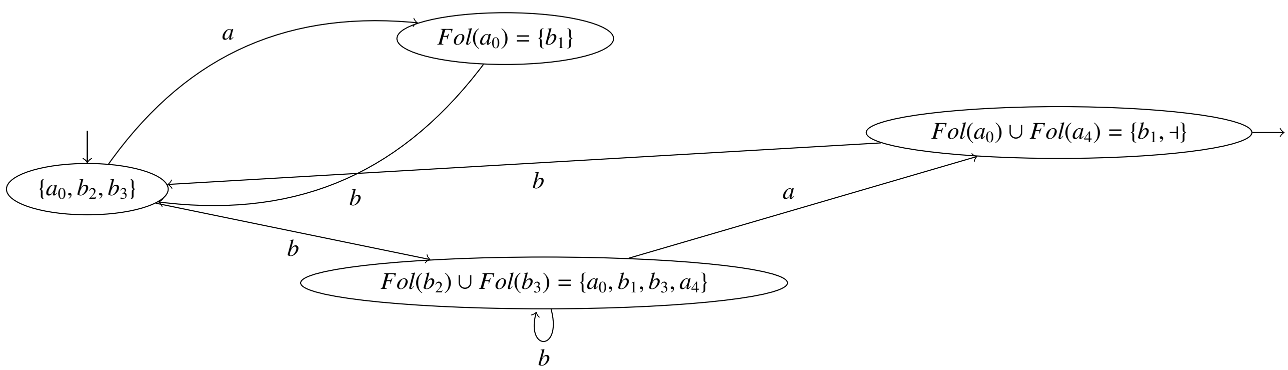
	Fol
a_0	b_1
b_1	a_0, b_2, b_3
b_2	a_0, b_2, b_3
b_3	a_4
a_4	$\dashv$

Calculamos el conjunto $Ini(r'_2 \dashv)$:

$$
\begin{aligned}
Ini(r'_2 \dashv) &= Ini((a_0b_1 \mid b_2)^*b_3a_4 \dashv)\\
&= Ini((a_0b_1 \mid b_2)^*b_3a_4) \cup Null((a_0b_1 \mid b_2)^*b_3a_4)Ini(\dashv)\\
&= Ini((a_0b_1 \mid b_2)^*b_3) \cup Null((a_0b_1 \mid b_2)^*b_3)Ini(a_4) \cup (Null((a_0b_1 \mid b_2)^*b_3) \cap Null(a_4))\{\dashv\}\\
&= Ini((a_0b_1 \mid b_2)^*) \cup Null((a_0b_1 \mid b_2)^*)Ini(b_3) \cup (Null((a_0b_1 \mid b_2)^*) \cap Null(b_3))Ini(a_4) \cup (Null((a_0b_1 \mid b_2)^*b_3) \cap \emptyset)\{\dashv\}\\
&= Ini(a_0b_1 \mid b_2) \cup \{\epsilon\}\{b_3\} \cup (Null((a_0b_1 \mid b_2)^*) \cap \emptyset)Ini(a_4) \cup (\emptyset)\{\dashv\}\\
&= Ini(a_0b_1) \cup Ini(b_2) \cup \{b_3\} \cup (\emptyset)\{a_4\} \cup \emptyset)\\
&= Ini(a_0) \cup Null(a_0)Ini(b_1) \cup \{b_2\} \cup \{b_3\} \cup \emptyset\\
&= \{a_0\} \cup \emptyset Ini(b_1) \cup \{b_2\} \cup \{b_3\}\\
&= \{a_0\} \cup \emptyset \cup \{b_2\} \cup \{b_3\}\\
&= \{a_0, b_2, b_3\}
\end{aligned}
$$

Esto produce el estado inicial $\{a_0, b_2, b_3\}$:

Aplicando el algoritmo de Berry y Sethi se obtiene el siguiente autómata:



3. Autómata de pila no determinista . . . . 25 puntos

Construir un autómata de pila de la siguiente gramática independiente de contexto G_3, utilizando el algoritmo de transformación de gramáticas independientes de contexto en autómatas de pila:

$$\begin{array}{rcl} A & \rightarrow & aB \mid b \\ B & \rightarrow & Cb \\ C & \rightarrow & aD \mid b \\ D & \rightarrow & Cb \end{array}$$

Solución:

	Rule	Movimiento
1	$A \rightarrow aB$	$\texttt{if}\ cc = a \wedge top = A\ \texttt{then}\ pop; push(B); shift$
2	$A \rightarrow b$	$\texttt{if}\ cc = b \wedge top = A\ \texttt{then}\ pop; push(\epsilon); shift$
3	$B \rightarrow Cb$	$\texttt{if}\ top = B\ \texttt{then}\ pop; push(bC)$
4	$C \rightarrow aD$	$\texttt{if}\ cc = a \wedge top = C\ \texttt{then}\ pop; push(D); shift$
5	$C \rightarrow b$	$\texttt{if}\ cc = b \wedge top = C\ \texttt{then}\ pop; push(\epsilon); shift$
6	$D \rightarrow Cb$	$\texttt{if}\ top = D\ \texttt{then}\ pop; push(bC)$
7		$\texttt{if}\ cc = a \wedge top = a\ \texttt{then}\ pop; push(\epsilon); shift$
8		$\texttt{if}\ cc = b \wedge top = b\ \texttt{then}\ pop; push(\epsilon); shift$
9		$\texttt{if}\ cc = \dashv \wedge empty\ \texttt{then}\ accept; halt$

4. ***The crab and berry-sethi strike back*** . . . . 25 puntos

Utilizando el método estructural de Thompson construya el autómata de estado finito no determinista N_4 para la expresión regular r_4:

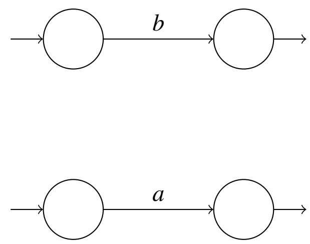
$$r_4 = b.a^*$$

De tal forma que $L(N_4) \equiv L(r_4)$

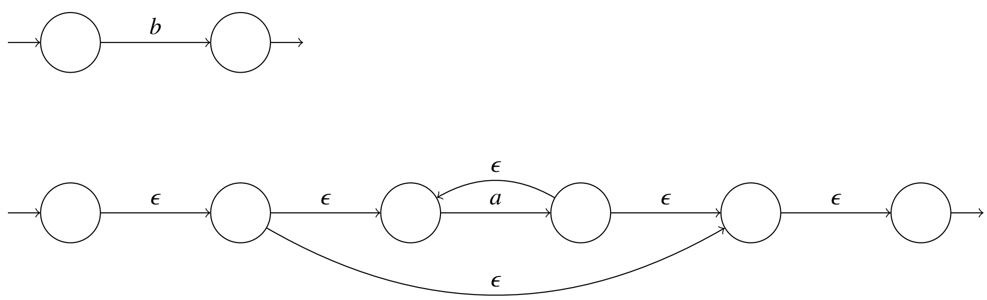
Y utilizando berry-sethi transforme el autómata N_4 en un autómata finito determinista M_4 de tal forma que $L(N_4) \equiv L(M_4)$

Solución:

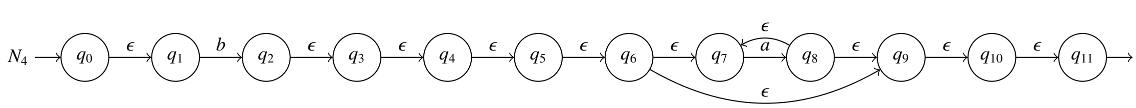
En primer lugar encontramos dos subexpresiones básicas: b y a: Utilizando Thompson construimos los autómatas, correspondientes:



El segundo autómata hace parte de la subexpresion a^*.



Uniendo los dos autómatas a través de la concatenación se obtiene el autómata N_4:



Dado que el autómata es de una expresión regular lineal no es necesario numerar la expresión. Entonces podemos calcular el estado inicial del autómata $Ini(N_4) = \{b\}$.

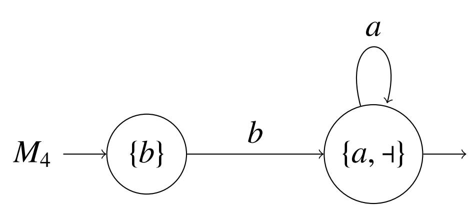
Calculamos el conjunto de los $Dig(N_4) = \{ba, aa\}$.

Calculamos el conjunto de los $Fin(N_4) = \{b, a\}$

Ahora calculamos el conjunto de Fol:

	Fol
b	$a, \dashv$
a	$a, \dashv$

Ahora podemos calcular el autómata M_4 determinista:



ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Cuarto parcial **DIS**
Juan Francisco Cardona McCormick **22 de Mayo de 2015**

Nombre: ______________________________
Código: ______________________________

1. Gramática $LL(1)$ . . . . 20 puntos

La siguiente gramática G_1: ¿Cumple con la condición $LL(1)$?

$$\begin{array}{rcl} X & \rightarrow & Y\, z \mid a \\ Y & \rightarrow & b\, Z \mid \epsilon \\ Z & \rightarrow & \epsilon \end{array}$$

Muestre la correspondiente evaluación de la condición $LL(1)$.

2. Convertir en $LL(1)$ . . . . 20 puntos

La siguiente gramática G_2 no es $LL(1)$ y

$$\begin{array}{rcl} S & \rightarrow & S\ i \mid W\, k \mid K \\ W & \rightarrow & a\, W \mid W\, b \mid b \\ K & \rightarrow & i \end{array}$$

Convertir a una gramática $LL(1)$.

3. Gramática $LR(0)$ . . . . 20 puntos

La siguiente gramática G_3: ¿Cumple con la condición $LR(0)$?

$$\begin{array}{rcl} S_0 & \rightarrow & E\ \dashv \\ E & \rightarrow & E \times B \mid E + B \mid B \\ B & \rightarrow & 0 \mid 1 \end{array}$$

Muestre que cumple con la condición $LR(0)$.

4. Construcción piloto $LR(1)$ . . . . 40 puntos

Construir el piloto $LR(1)$ para la siguiente gramática G_4.

$$\begin{array}{rcl} S_0 & \rightarrow & S\ \dashv \\ S & \rightarrow & A\, A \\ A & \rightarrow & c\, A \mid d \end{array}$$

Entregar el piloto que corresponde a la gramática G_4:

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Transformación de una gramática independiente de contexto a forma de tiempo real**

La gramática independiente de contexto G_1

$$
\begin{array}{rcl}
S & \rightarrow & Ab \mid Ba \mid Cc \\
A & \rightarrow & Aba \mid d \\
B & \rightarrow & Ac \mid e \\
C & \rightarrow & Bc \mid Ca \mid b
\end{array}
$$

Solución:

En primer lugar se observa si existe una derivación por la izquierda inmediata o no inmediata. Es evidente, la derivación a la izquierda inmediata en los no-terminales A y C, pero la derivación por la izquierda no inmediata no existe. Por lo tanto se elimina primero las derivaciones inmediatas por la izquierda:

$$
\begin{array}{rcl}
S & \rightarrow & Ab \mid Ba \mid Cc \\
A & \rightarrow & d \mid dA' \\
A' & \rightarrow & ba \mid baA' \\
B & \rightarrow & Ac \mid e \\
C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\
C' & \rightarrow & a \mid aC'
\end{array}
$$

Ahora a forma de tiempo real. Se observa en los no-terminales S, B y C que estas comienza por no terminales, por lo tanto se debe expandir hasta obtener una versión de tiempo real. Expandiendo A en S se obtiene

$$
\begin{array}{rcl}
S & \rightarrow & db \mid dA'b \mid Ba \mid Cc \\
A & \rightarrow & d \mid dA' \\
A' & \rightarrow & ba \mid baA' \\
B & \rightarrow & Ac \mid e \\
C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\
C' & \rightarrow & a \mid aC'
\end{array}
$$

Luego B en S

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid Aca \mid ea \mid Cc \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & Ac \mid e \\ C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

Nuevamente A en S:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid Cc \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & Ac \mid e \\ C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

Ahora C en S:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid ac\, mid\, aC'c \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & Ac \mid e \\ C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

Ahora A en B:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid ac\, mid\, aC'c \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & dc \mid dA'c \mid e \\ C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

Finalmente, B en C:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid ac \mid aC'c \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & dc \mid dA'c \mid e \\ C & \rightarrow & dcc \mid dA'cc \mid ec \\ & \mid & dccC' \mid dA'ccC' \mid ecC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

(25 %) 2. **Expresiones regulares**
Dado el alfabeto $\Sigma = \{w, x, y, z\}$

a) Defina una expresión regular $r_{2,1}$ cuyo lenguaje $L(r_{2,1})$ genere cadenas que si contiene una x también contiene también una y a la derecha de ésta (no necesariamente al lado).

b) Defina una expresión regular $r_{2,2}$ cuyo lenguaje $L(r_{2,2})$ genere cadena que si contiene una x no puede contener una y al lado derecho de ésta (no necesariamente al lado).

Solución:

1. $r_{2,1} = (w \mid y \mid z)^* \mid (w \mid x \mid y \mid z)^* x (w \mid x \mid y \mid z)^* y (w \mid x \mid y \mid z)^*$
2. $r_{2,2} = (w \mid y \mid z)^* \mid (w \mid y \mid z)^* x (w \mid x \mid z)^*$

(25 %) 3. **Gramáticas independientes de contexto**
Defina la gramática independiente de contexto G_3 para el siguiente lenguaje L_3 con alfabeto $\Sigma = \{x, y\}$:

$$L_3 = \{yx^n y^n x \mid n \geq 1\} \cup \{xy^n x^n y \mid n \geq 1\}$$

Solución:

Una posible solución $G_3 = (V, \Sigma, P, S)$, donde $V = \{A, B, C\}$, $\Sigma = \{x, y\}$, P:

$$\begin{array}{lcl} A & \rightarrow & yBx \mid xCy \\ B & \rightarrow & xy \mid xBy \\ C & \rightarrow & yx \mid yCx \end{array}$$

(25 %) 4. **Transformaciones alfabeticas**

La siguiente gramática G_4 es la conocida gramática de preposiciones de la lógica:

$$\begin{array}{lcl} P & \rightarrow & P \vee A \mid A \\ A & \rightarrow & A \wedge B \mid B \\ B & \rightarrow & \neg C \mid C \\ C & \rightarrow & (P) \mid id \mid V \mid F \end{array}$$

Donde V es verdadero, F es falso y id es un identificador cualquiera.

Si observan bien: la gramática G_4 no puede ser utilizada directamente para construir una expresión de un condición válida en el lenguaje de programación Java™por que sus cadenas son invalidas en él. Pero eso se puede hacer al utilizar una función de transliteración $h(c)$ que transforme las cadenas generadas por el lenguaje G_4 en cadenas válidas para el lenguaje de expresiones de condiciones en el lenguaje de programación Java™.

Implemente dicha función de transformación (o transliteración o homomorfismo alfabetico) $h(c)$.

Solución:

$$\begin{array}{rcl} h(\vee) & = & || \\ h(\wedge) & = & \&\& \\ h(\neg) & = & ! \\ h(V) & = & \text{true} \\ h(F) & = & \text{false} \\ h(id) & = & \text{id} \\ h('(') & = & (\\ h(')') & = &) \end{array}$$

ST0270 Lenguajes formales y compiladores - 031b — Universidad EAFIT
Primer parcial — **DIS**
Juan Francisco Cardona McCormick — **1 de marzo de 2016**

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Transformación de una gramática independiente de contexto a forma de Greibach**

La gramática independiente de contexto $G_1 = (\{S, A, B, C, D, E, F\}, \{a, b, c, d\}, P_1, S)$ donde $P_1 =$:

$$
\begin{array}{lcl}
S & \rightarrow & Aa \mid bB \mid bC \\
A & \rightarrow & Da \mid D \\
B & \rightarrow & Ba \mid b \mid a \mid B \\
C & \rightarrow & a \mid Db \mid bE \\
D & \rightarrow & a \\
E & \rightarrow & bF \\
F & \rightarrow & aF
\end{array}
$$

Convierta en un gramática limpia en forma de Greibach G_1'

Solución:

Se observa que la gramática no está limpia, miremos que esté limpias, si la gramática está definida:

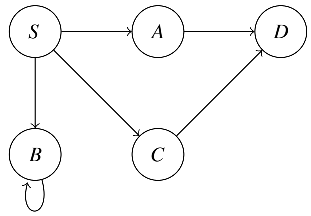
En primer lugar se observa si existe una derivación por la izquierda inmediata o no inmediata. Es evidente, la derivación a la izquierda inmediata en los no-terminales A y C, pero la derivación por la izquierda no inmediata no existe. Por lo tanto se elimina primero las derivaciones inmediatas por la izquierda:

$$
\begin{array}{lcl}
Def_0 & = & \{B, C, D\} \\
Def_1 & = & Def_0 \cup \{A, S\} \\
Def_2 & = & Def_1 \cup \{\}
\end{array}
$$

Se crea un gramática nueva $G_1'' = (\{S, A, B, C, D\}, \{a, b, c, d\}, P_1'', S)$, donde P_1''

$$
\begin{array}{lcl}
S & \rightarrow & Aa \mid bB \mid bC \\
A & \rightarrow & Da \mid D \\
B & \rightarrow & Ba \mid b \mid a \mid B \\
C & \rightarrow & a \mid Db \\
D & \rightarrow & a
\end{array}
$$

Y se observa que los no-terminales son alcanzables:



Pero se observa que existe una circularidad: $B \Rightarrow^{+} B$. Se elimina y se obtiene la gramática $G_1''' = (\{S, A, B, C, D\}, \{a, b, c, d\}, P_1''', S)$ donde P_1'''

$$\begin{array}{rcl} S & \rightarrow & Aa \mid bB \mid bC \\ A & \rightarrow & Da \mid D \\ B & \rightarrow & Ba \mid b \mid a \\ C & \rightarrow & a \mid Db \\ D & \rightarrow & a \end{array}$$

Luego se eliminan las recursividades por la izquierda inmediatas y no inmediatas. Existe una sola recursividad inmediata: $B \Rightarrow Ba$. Se elimina esta recursividad y se obtiene:

$$\begin{array}{rcl} S & \rightarrow & Aa \mid bB \mid bC \\ A & \rightarrow & Da \mid D \\ B & \rightarrow & bB' \mid aB' \mid b \mid a \\ B' & \rightarrow & aB' \mid a \\ C & \rightarrow & a \mid Db \\ D & \rightarrow & a \end{array}$$

Ahora a forma de tiempo real. Se observa que S, A y C comienza por no terminales. Expandimo primero en D en A y en D en A:

$$\begin{array}{rcl} S & \rightarrow & Aa \mid bB \mid bC \\ A & \rightarrow & aa \mid a \\ B & \rightarrow & bB' \mid aB' \mid b \mid a \\ B' & \rightarrow & aB' \mid a \\ C & \rightarrow & a \mid ab \\ D & \rightarrow & a \end{array}$$

Luego A en S y se obtiene:

$$\begin{array}{rcl} S & \rightarrow & aaa \mid aa \mid bB \mid bC \\ A & \rightarrow & aa \mid a \\ B & \rightarrow & bB' \mid aB' \mid b \mid a \\ B' & \rightarrow & aB' \mid a \\ C & \rightarrow & a \mid ab \\ D & \rightarrow & a \end{array}$$

Ya está en forma normal de tiempo real, ahora reemplazamos cada terminal de la diferente de la primera posición en un auxilar no terminal:

$$\begin{array}{rcl} S & \rightarrow & a\langle a\rangle\langle a\rangle \mid a\langle a\rangle \mid bB \mid bC \\ A & \rightarrow & a\langle a\rangle \mid a\langle\epsilon\rangle \\ B & \rightarrow & bB' \mid aB' \mid b \mid a \\ B' & \rightarrow & aB' \mid a \\ C & \rightarrow & a \mid ab \\ D & \rightarrow & a \end{array}$$

Ahora C en S:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid ac\, mid\, aC'c \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & Ac \mid e \\ C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

Ahora A en B:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid ac\, mid\, aC'c \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & dc \mid dA'c \mid e \\ C & \rightarrow & Bc \mid BcC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

Finalmente, B en C:

$$\begin{array}{rcl} S & \rightarrow & db \mid dA'b \mid dca \mid dA'ca \mid ea \mid ac \mid aC'c \\ A & \rightarrow & d \mid dA' \\ A' & \rightarrow & ba \mid baA' \\ B & \rightarrow & dc \mid dA'c \mid e \\ C & \rightarrow & dcc \mid dA'cc \mid ec \\ & \mid & dccC' \mid dA'ccC' \mid ecC' \mid b \mid bC' \\ C' & \rightarrow & a \mid aC' \end{array}$$

(25 %) 2. **Expresiones regulares**
Dado el alfabeto $\Sigma = \{a, b, c, d\}$

a) Defina una expresión regular $r_{2,1}$ cuyo lenguaje $L(r_{2,1})$ genere cadenas que contiene por lo menos una a y por lo menos una b.

b) Defina una expresión regular $r_{2,2}$ cuyo lenguaje $L(r_{2,2})$ genere cadena que tengan a los sumo un par de c consecutivas.

Solución:

1. $r_{2,1} = (w \mid y \mid z)^* \mid (w \mid x \mid y \mid z)^* x (w \mid x \mid y \mid z)^* y (w \mid x \mid y \mid z)^*$
2. $r_{2,2} = (w \mid y \mid z)^* \mid (w \mid y \mid z)^* x (w \mid x \mid z)^*$

(25 %) 3. **Gramáticas independientes de contexto**
Defina la gramática independiente de contexto G_3 correspondiente al "diagrama sintáctico" de la figura:

Solución:
Una posible solución $G_3 = (V, \Sigma, P, S)$, donde $V = \{A, B, C\}$, $\Sigma = \{x, y\}$, P:

$$\begin{array}{rcl} A & \rightarrow & yBx \mid xCy \\ B & \rightarrow & xy \mid xBy \\ C & \rightarrow & yx \mid yCx \end{array}$$

(25 %) 4. **Eliminación de recursividad a la izquierda**
La siguiente gramática $G_4 = (\{A, B, C, D\}, \{0, 1\}, P_4, A)$ donde $P_4 =$:

$$\begin{array}{rcl} A & \rightarrow & B0 \mid 1 \\ B & \rightarrow & A1 \mid B0 \mid 0C \mid 1D \\ C & \rightarrow & A0 \mid B1 \mid 1 \\ D & \rightarrow & C1 \mid 1 \mid 0 \end{array}$$

Solución:

$$\begin{array}{rcl} h(\vee) & = & \mid\mid \\ h(\wedge) & = & \&\& \\ h(\neg) & = & ! \\ h(V) & = & \text{true} \\ h(F) & = & \text{false} \\ h(id) & = & \text{id} \\ h('(') & = & (\\ h(')') & = &) \end{array}$$

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Transformación de una gramática independiente de contexto a forma de Greibach**

La gramática independiente de contexto $G_1 = (\{S, A, B, C\}, \{a, b\}, P_1, S)$ donde P_1 =:

$$\begin{array}{lcl} S & \rightarrow & Aa \mid bA \mid Cb \mid aC \\ A & \rightarrow & aB \mid b \\ B & \rightarrow & Aa \mid a \\ C & \rightarrow & b \end{array}$$

Convierta en un gramática limpia en forma de Greibach G_1'.

(25 %) 2. **Expresiones regulares**

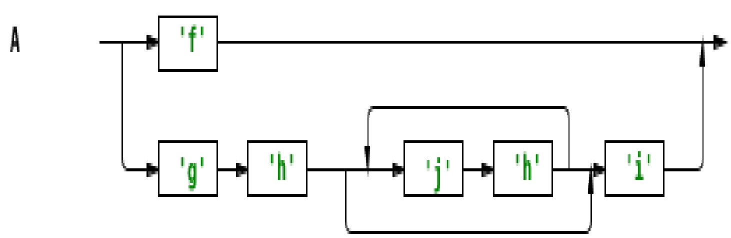
Dado el alfabeto $\Sigma = \{a, b, c, d\}$

a) Defina una expresión regular $r_{2,1}$ cuyo lenguaje $L(r_{2,1})$ genere cadenas que contiene un número par de a.

b) Defina una expresión regular $r_{2,2}$ cuyo lenguaje $L(r_{2,2})$ genere cadena que no terminan con cd.

(25 %) 3. **Gramáticas independientes de contexto**

Defina la gramática independiente de contexto G_3 correspondiente al "diagrama sintáctico" de la figura:



(25 %) 4. **Eliminación de recursividad a la izquierda**

La siguiente gramática $G_4 = (\{A, B, C, D\}, \{0, 1\}, P_4, A)$ donde P_4 =:

$$\begin{array}{lcl} A & \rightarrow & B0 \mid 1 \\ B & \rightarrow & A1 \mid B0 \mid 0C \mid 1D \\ C & \rightarrow & A0 \mid B1 \mid 1 \\ D & \rightarrow & C1 \mid 1 \mid 0 \end{array}$$

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Transformación de una gramática independiente de contexto a forma de Greibach**

La gramática independiente de contexto $G_1 = (\{T, U, V, U, W, Y, Z\}, \{x, y, z\}, P_1, T)$ donde $P_1 =$:

$$\begin{array}{lcl} T & \rightarrow & Ux \mid yV \mid yW \\ U & \rightarrow & Xa \mid X \\ V & \rightarrow & Vx \mid y \mid x \mid V \\ W & \rightarrow & x \mid Xy \mid yY \\ X & \rightarrow & x \\ Y & \rightarrow & yZ \\ Z & \rightarrow & xZ \end{array}$$

Convierta en un gramática limpia en forma de Greibach G'_1.

(25 %) 2. **Expresiones regulares**

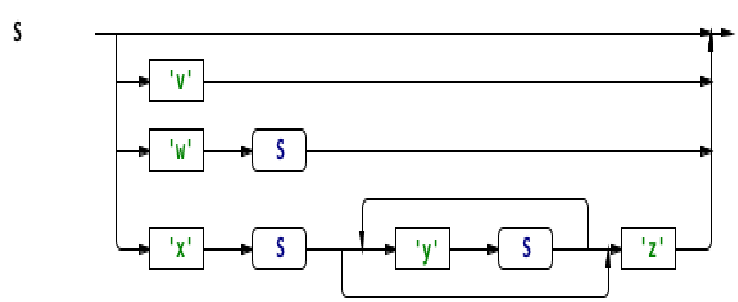
Dado el alfabeto $\Sigma = \{w, x, y, z\}$

a) Defina una expresión regular $r_{2,1}$ cuyo lenguaje $L(r_{2,1})$ genere cadenas que contiene por lo menos una z y una w.

b) Defina una expresión regular $r_{2,2}$ cuyo lenguaje $L(r_{2,2})$ genere cadenas que tengan a lo sumo un par wx consecutivas.

(25 %) 3. **Gramáticas independientes de contexto**

Defina la gramática independiente de contexto G_3 correspondiente al "diagrama sintáctico" de la figura:



(25 %) 4. **Eliminación de recursividad a la izquierda**
La siguiente gramática $G_4 = (\{W, X, Y, Z\}, \{a, b\}, P_4, W)$ donde P_4 =:

$$\begin{array}{rcl} W & \to & Xa \mid b \\ X & \to & Wb \mid Xa \mid aY \mid bZ \\ Y & \to & Wa \mid Xb \mid b \\ Z & \to & Yb \mid b \mid a \end{array}$$

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Transformación de una gramática independiente de contexto a forma de Greibach**

La gramática independiente de contexto $G_1 = (\{W, X, Y, Z\}, \{0, 1\}, P_1, W)$ donde $P_1 =$:

$$\begin{array}{rcl} W & \rightarrow & X0 \mid 1X \mid Z1 \mid 0Z \\ X & \rightarrow & 1Y \mid 0 \\ Y & \rightarrow & X0 \mid 1 \\ Z & \rightarrow & 1 \end{array}$$

Convierta en un gramática limpia en forma de Greibach G'_1.

(25 %) 2. **Expresiones regulares**

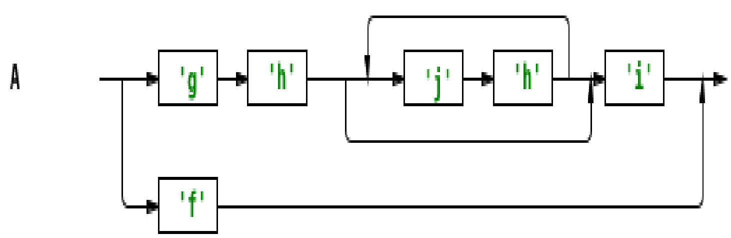
Dado el alfabeto $\Sigma = \{w, x, y, z\}$

a) Defina una expresión regular $r_{2,1}$ cuyo lenguaje $L(r_{2,1})$ genere cadenas que contiene un número par de z.

b) Defina una expresión regular $r_{2,2}$ cuyo lenguaje $L(r_{2,2})$ genere cadena que no terminan con wx.

(25 %) 3. **Gramáticas independientes de contexto**

Defina la gramática independiente de contexto G_3 correspondiente al "diagrama sintáctico" de la figura:



(25 %) 4. **Eliminación de recursividad a la izquierda**

La siguiente gramática $G_4 = (\{W, X, Y, Z\}, \{a, b\}, P_4, W)$ donde $P_4 =$:

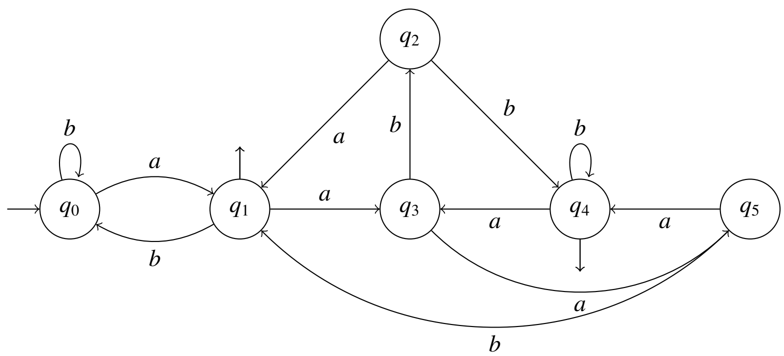
$$\begin{array}{rcl} W & \rightarrow & Xa \mid b \\ X & \rightarrow & Wb \mid Xa \mid aY \mid bZ \\ Y & \rightarrow & Wa \mid Xb \mid b \\ Z & \rightarrow & Yb \mid b \mid a \end{array}$$

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **12 de abril de 2016**

Nombre: ____
Código: ____

1. Minimización de autómatas . . . . 25 puntos

Para el siguiente autómata de estado finito determinista AFD_1 utilice el algoritmo de minimización para encontrar un autómata de estado finito determinista AFD'_1, tal que $L(AFD_1) \equiv L(AFD'_1)$ y que AFD'_1. Es posible que AFD_1 ya sea el mínimo.



2. Berry and Sethi . . . . 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = 10(10 \mid 1)^*$$

Solución:

3. Autómata de pila no determinista . . . . 25 puntos

Construir un autómata de pila de la siguiente gramática independiente de contexto G_3, utilizando el algoritmo de transformación de gramáticas independientes de contexto en autómatas de pila:

$$\begin{array}{lcl} A & \rightarrow & aA \mid bB \mid \epsilon \\ B & \rightarrow & bB \mid cC \mid \epsilon \\ C & \rightarrow & cC \mid dD \mid \epsilon \\ D & \rightarrow & dD \mid \epsilon \end{array}$$

Solución:

	Rule	Movimiento
1	$A \rightarrow aB$	`if` $cc = a \land top = A$ `then` $pop; push(B); shift$
2	$A \rightarrow b$	`if` $cc = b \land top = A$ `then` $pop; push(\epsilon); shift$
3	$B \rightarrow Cb$	`if` $top = B$ `then` $pop; push(bC)$
4	$C \rightarrow aD$	`if` $cc = a \land top = C$ `then` $pop; push(D); shift$
5	$C \rightarrow b$	`if` $cc = b \land top = C$ `then` $pop; push(\epsilon); shift$
6	$D \rightarrow Cb$	`if` $top = D$ `then` $pop; push(bC)$
7		`if` $cc = a \land top = a$ `then` $pop; push(\epsilon); shift$
8		`if` $cc = b \land top = b$ `then` $pop; push(\epsilon); shift$
9		`if` $cc = \dashv \land empty$ `then` $accept; halt$

4. ***The crab and berry-sethi strike back*** .... 25 puntos

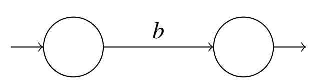
Utilizando el método estructural de Thompson construya el autómata de estado finito no determinista N_4 para la expresión regular r_4:

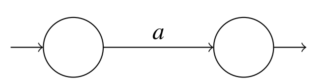
$$r_4 = ab^*a$$

De tal forma que $L(N_4) \equiv L(r_4)$
Y utilizando berry-sethi transforme el autómata N_4 en un autómata finito determinista M_4 de tal forma que $L(N_4) \equiv L(M_4)$

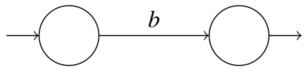
Solución:

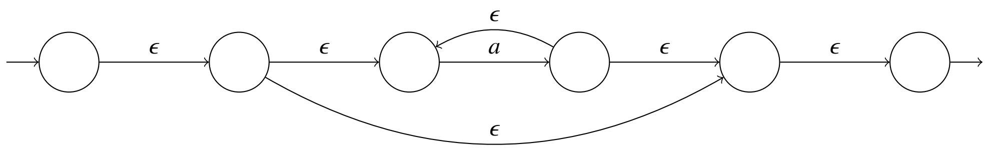
En primer lugar encontramos dos subexpresiones básicas: b y a: Utilizando Thompson construimos los autómatas, correspondientes:



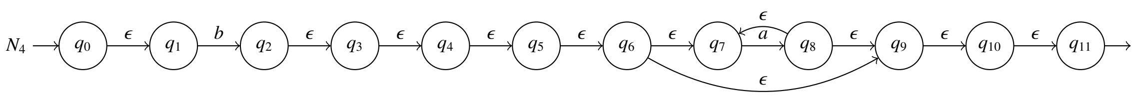


El segundo autómata hace parte de la subexpresion a^*.





Uniendo los dos autómatas a través de la concatenación se obtiene el autómata N_4:



Dado que el autómata es de una expresión regular lineal no es necesario numerar la expresión. Entonces podemos calcular el estado inicial del autómata $Ini(N_4) = \{b\}$.

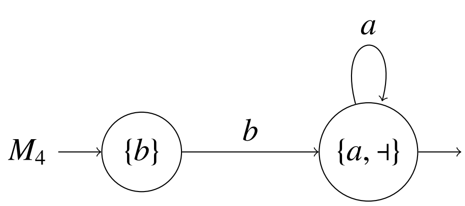
Calculamos el conjunto de los $Dig(N_4) = \{ba, aa\}$.

Calculamos el conjunto de los $Fin(N_4) = \{b, a\}$

Ahora calculamos el conjunto de Fol:

	Fol
b	$a, \dashv$
a	$a, \dashv$

Ahora podemos calcular el autómata M_4 determinista:



Nombre: ______________________________
Código: ______________________________

1. Autómatas de pila .... 25 puntos

Un *autómata contador* es un autómata de pila que tiene sólo dos símbolos en la pila, A, y Z_0 en el que la cadena en la pila es siempre de la forma $A^n Z_0$ para alguna $n \geq 0$. (En otras palabras, el único cambio posible en el contenido de la pila es le del número de A's de la pila). Y en relación con algunos lenguajes de gramáticas independientes de contexto, como $\{0^i 1^i \mid i \geq 0\}$, el autómata de pila evidente que acepta al lenguaje es, de hecho, un autómata de pila contador. Construya un autómata contador que acepte al lenguaje L_1:

$$L_1 = \{x \in \{0,1\}^* \mid |x|_0 = |x|_1\}$$

2. Berry and Sethi .... 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = (ab \mid a)^* ba$$

3. Autómata de estado finito .... 25 puntos

Encontrar el autómata finito determinista que reconozca el siguiente lenguaje:

$$L_3 = \{x \in \Sigma^* \mid \mid x \mid \bmod 3 = 0\}$$

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

4. *The crab and berry-sethi strike back* .... 25 puntos

Utilizando el método estructural de Thompson construya el autómata de estado finito no determinista N_4 para la expresión regular r_4:

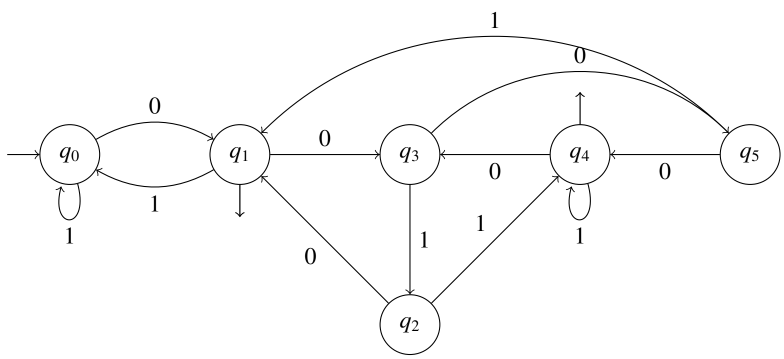
$$r_4 = b^* ba$$

De tal forma que $L(N_4) \equiv L(r_4)$
Y utilizando berry-sethi transforme el autómata N_4 en un autómata finito determinista M_4 de tal forma que $L(N_4) \equiv L(M_4)$

Nombre: ____________________
Código: ____________________

1. Minimización de autómatas .... 25 puntos

Para el siguiente autómata de estado finito determinista AFD_1 utilice el algoritmo de minimización para encontrar un autómata de estado finito determinista AFD'_1, tal que $L(AFD_1) \equiv L(AFD'_1)$ y que AFD'_1. Es posible que AFD_1 ya sea el mínimo.



2. Berry and Sethi .... 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = ba(ba \mid b)^*$$

3. Autómata de pila no determinista .... 25 puntos

Construir un autómata de pila de la siguiente gramática independiente de contexto G_3, utilizando el algoritmo de transformación de gramáticas independientes de contexto en autómatas de pila:

$$\begin{array}{lcl} W & \rightarrow & wW \mid xX \mid \epsilon \\ X & \rightarrow & xX \mid yY \mid \epsilon \\ Y & \rightarrow & yY \mid zZ \mid \epsilon \\ Z & \rightarrow & zZ \mid \epsilon \end{array}$$

4. *The crab and berry-sethi strike back* .... 25 puntos

Utilizando el método estructural de Thompson construya el autómata de estado finito no determinista N_4 para la expresión regular r_4:

$$r_4 = 01^*0$$

De tal forma que $L(N_4) \equiv L(r_4)$

Y utilizando berry-sethi transforme el autómata N_4 en un autómata finito determinista M_4 de tal forma que $L(N_4) \equiv L(M_4)$

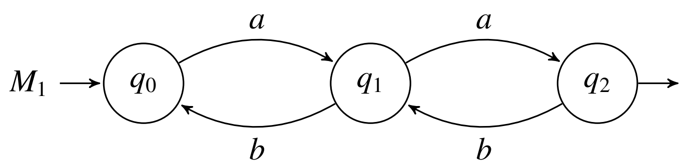
ST0270 Lenguajes formales y compiladores - 031 | **Universidad EAFIT**
Tercer parcial | **DIS**
Juan Francisco Cardona McCormick | **3 de octubre de 2013**

Nombre: ______________________________

Código: ______________________________

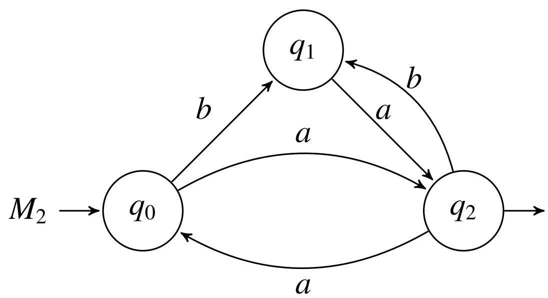
1. De autómata a expresiones regulares . . . . 25 puntos

Obtener la expresión regular definida por el siguiente autómata determinista M_1:



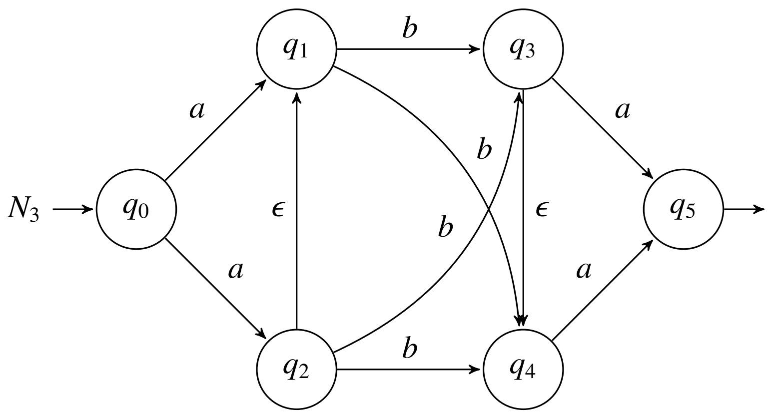
2. Minimizar autómatas . . . . 25 puntos

Minimizar el siguiente autómata M_2:



3. Algoritmo de Berry y Sethi - Aplicado . . . . 25 puntos

Dado el siguiente no determinista y con movimientos espontáneo N_3, definir un autómata determinista M_3, tal que $L(N_3) \equiv L(M_3)$, utilizando el algoritmo de Berry y Sethi:



4. Autómatas de gramáticas lineales por la izquierda . . . . 25 puntos

Calcular el autómata de estado finito para una gramática lineal por la izquierda G_4:

$$\begin{array}{lcl} S & \rightarrow & Ab \mid Sb \\ A & \rightarrow & Aa \mid Ab \mid B \\ B & \rightarrow & Bc \mid Bd \mid \epsilon \end{array}$$

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **17 de Mayo del 2016**

Nombre: ______________________________
Código: ______________________________

1. Algoritmo de reconocimiento determinista $LL(1)$ . . . . 30 puntos

La siguiente gramática G_1 cumple con la condición $LL(1)$:

$$\begin{array}{lcl} S & \rightarrow & (S + F) \\ S & \rightarrow & F \\ F & \rightarrow & a \end{array}$$

a) Construya la red de automátas correspondiente.

b) Verifique que las siguientes cadenas pertenecen o no pertenence utilizando el algoritmo de reconcimiento deterministas y mostrando cada paso.

b.1) $((a + a) + a)$.

b.2) $((a))$.

2. Condición $LL(1)$ . . . . 30 puntos

Verificar que la gramática G_2 cumple con la condición $LL(1)$

$$\begin{array}{lcl} P & \rightarrow & \{P\}P \\ P & \rightarrow & \epsilon \end{array}$$

a) Construya la correspondiente red de autómatas.

b) Calcule los conjunto guías correspondientes y los iniciales y siguientes necesarios para mostrar que la gramática G_2 cumple con la condición $LL(1)$.

3. Obtener una gramática $LL(1)$ . . . . 40 puntos

La siguiente gramática G_3, no cumple con la condición $LL(1)$.

$$\begin{array}{lcl} R & \rightarrow & R\,.\,R \\ R & \rightarrow & R \mid R \\ R & \rightarrow & R^* \\ R & \rightarrow & a \\ R & \rightarrow & b \\ R & \rightarrow & (R) \end{array}$$

La gramática tiene los siguientes operadores: ., | y *. Estos operadores tienen la siguientes precedencias (de mayor a menor): *, | y .. Evidentemente, los paréntesis siempre tiene la mayor precedencia.

Construya utilizando las transformaciones vistas en clase una gramática G_3', tal que $L(G_3) \equiv L(G_3')$ y que G_3' cumple con la condición $LL(1)$. Esta condición no tiene que ser demostrada completamente, solamente mostrando de forma intuitiva las guías necesarias.

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **17 de Mayo del 2016**

Nombre: ______________________________

Código: ______________________________

1. Algoritmo de reconocimiento determinista $LL(1)$ .... 30 puntos

La siguiente gramática G_1 cumple con la condición $LL(1)$:

$$
\begin{array}{lcl}
E & \rightarrow & (E \times V) \\
E & \rightarrow & V \\
V & \rightarrow & i
\end{array}
$$

a) Construya la red de automátas correspondiente.

b) Verifique que las siguientes cadenas pertenecen o no pertenence utilizando el algoritmo de reconcimiento deterministas y mostrando cada paso.

b.1) $((i \times i) \times i)$.

b.2) $((i))$.

2. Condición $LL(1)$ .... 30 puntos

Verificar que la gramática G_2 cumple con la condición $LL(1)$

$$
\begin{array}{lcl}
S & \rightarrow & (S)S \\
S & \rightarrow & \epsilon
\end{array}
$$

a) Construya la correspondiente red de autómatas.

b) Calcule los conjunto guías correspondientes y los iniciales y siguientes necesarios para mostrar que la gramática G_2 cumple con la condición $LL(1)$.

3. Obtener una gramática $LL(1)$ .... 40 puntos

La siguiente gramática G_3, no cumple con la condición $LL(1)$.

$$
\begin{array}{lcl}
S & \rightarrow & S \wedge S \\
S & \rightarrow & S \vee S \\
S & \rightarrow & S\,\dagger \\
S & \rightarrow & \text{true} \\
S & \rightarrow & \text{false} \\
S & \rightarrow & (S)
\end{array}
$$

La gramática tiene los siguientes operadores: $\wedge$, $\vee$ y $\dagger$. Estos operadores tienen la siguientes precedencias (de mayor a menor): $\dagger$, $\wedge$ y $\vee$. Evidentemente, los paréntesis siempre tiene la mayor precedencia.

Construya utilizando las transformaciones vistas en clase una gramática G_3', tal que $L(G_3) \equiv L(G_3')$ y que G_3' cumple con la condición $LL(1)$. Esta condición no tiene que ser demostrada completamente, solamente mostrando de forma intuitiva las guías necesarias.

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Primer parcial **DIS**
Juan Francisco Cardona McCormick **25 de agosto de 2016**

Nombre: ______________________________
Código: ______________________________

Nota: Los puntos (1 y 4) donde resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el método formal, en esos casos no se admiten respuesta sin la correspodiente aplicación formal y desarrollo completo del punto.

(25 %) 1. **Transformación de una gramática independiente de contexto**

La gramática independiente de contexto $G_1 = (\{LL, PA, PS, P\}, \{id, (,),',' \}, P_1, LL)$ donde $P_1 =$:

$$\begin{array}{rcl} LL & \rightarrow & id(PA) \\ PA & \rightarrow & PS \mid \epsilon \\ PS & \rightarrow & PS \ , \ P \mid P \\ P & \rightarrow & id \end{array}$$

Elimine las reglas nulas y las reglas de copia de la gramática G_1 y tranforme la gramática en una sin reglas nulas, ni reglas de copia G'_1.

Solución:

En primer eliminamos las reglas nulas:

Calculamos el conjunto *Null*. En la primera iteración:

$$Null = \{PA\}$$

No se encuentra ningún no-terminal $A \in P_1$ tal que convierta dicho no terminal que genera reglas nulas.

Eliminando las reglas nulas y combinando obtenemos P'_1:

$$\begin{array}{rcl} LL & \rightarrow & id() \mid id(\ PA\ ) \\ PA & \rightarrow & PS \\ PS & \rightarrow & PS \ , \ P \mid P \\ P & \rightarrow & id \end{array}$$

Ya hemos eliminado las reglas nulas, ahora procedemos a eliminar las reglas de copia. Calculamos el conjunto de copia *Copy*.

$$\begin{array}{rcl} Copy(LL) & = & \{LL\} \\ Copy(PA) & = & \{PA\} \\ Copy(PS) & = & \{PS\} \\ Copy(P) & = & \{P\} \end{array}$$

Primera iteración:

$$
\begin{array}{rcl}
Copy(LL) & = & \{LL\} \\
Copy(PA) & = & \{PA,\ PS\} \\
Copy(PS) & = & \{PS,\ P\} \\
Copy(P) & = & \{P\}
\end{array}
$$

Segunda iteración:

$$
\begin{array}{rcl}
Copy(LL) & = & \{LL\} \\
Copy(PA) & = & \{PA,\ PS,\ P\} \\
Copy(PS) & = & \{PS,\ P\} \\
Copy(P) & = & \{P\}
\end{array}
$$

Tercera iteración:

$$
\begin{array}{rcl}
Copy(LL) & = & \{LL\} \\
Copy(PA) & = & \{PA,\ PS,\ P\} \\
Copy(PS) & = & \{PS,\ P\} \\
Copy(P) & = & \{P\}
\end{array}
$$

Ya con el conjunto de copia $Copy$ podemos relcalcular el conjunto de producciones P_1'':

$$
\begin{array}{rcl}
LL & \rightarrow & id() \mid id(\ PA\ ) \\
PA & \rightarrow & PS\ ,\ P \mid id \\
PS & \rightarrow & PS\ ,\ P \mid id \\
P & \rightarrow & id
\end{array}
$$

La gramática generada es limpia.

(25 %) 2. **Lenguaje de paréntesis**
Dado el alfabeto $\Sigma_2 = \{(,), [,]\}$ la gramática G_2:

$$S \rightarrow (S)S \mid [S]S \mid \epsilon$$

Los lenguajes *Dyck* son caracterizados por la siguiente regla de cancelación que examina los parentesis que pueden ser anidados.

$$[] \Rightarrow \epsilon \quad () \Rightarrow \epsilon$$

Implemente una función que reciba una cadena de entrada que contiene carácteres del alfabeto Σ_2 para probar que pertenece a la gramática G_2 utilice una pila para implementar el algoritmo.

Solución:

Una posible solución:

$i \leftarrow 0$
for $Cadena_i \neq 0$ **do**
  if $Cadena_i \equiv$ '[' $\vee$ $Cadena_i \equiv$ '(' **then**
    $stack.push(Cadena_i)$
  else
    $c \leftarrow stack.pop()$
    if ($c \neq$ '[' $\vee$ $Cadena_i \equiv$ ']') $\wedge$ ($c \neq$ '(' $\vee$ $Cadena_i \neq$ ')') **then**
      return $\perp$
    end if
  end if
  $i \leftarrow i + 1$
end for
return $\top$

Donde $\top$ es verdadero y $\perp$ es falso.

(25 %) 3. **Expresiones regulares y lenguajes**
Escribir la expresión regular para el siguiente lenguaje: El conjunto de cadenas de cero y unos que tengan a lo sumo[1] un par de unos consecutivos.

Solución:

Una posible solución es:

$$(0 \mid 1)^* \setminus ((0 \mid 1)^* 11 (0 \mid 1)^*)((0 \mid 1)^* 11 (0 \mid 1)^*)^+$$

La primera expresión genera todas las posibles secuencias de unos y ceros. La segunda expresión elimna de las primera todas las cadenas que tenga más de unos consecutivos.

[1]Según la `rae.es` a los sumo tiene dos significados: 1. loc. adv. A lo más, al mayor grado, número, cantidad, etc, a que puede llegar alguien o algo. 2. loc. adv. Cuando más, si acaso.

(25 %) 4. **Gramática independientes de contexto**

Encuentre las gramática independiente de contexto que generera el siguiente lenguaje $L_4 = \{x^a y^b z^c \mid a = b + c\}$

Solución:

$G_4 = (V_4, \Sigma_4, P_4, S_4)$. Donde $V_4 = \{S, A\}$, $\Sigma_4 = \{x, y, z\}$, $S_4 = \{S\}$ y P_4 es:

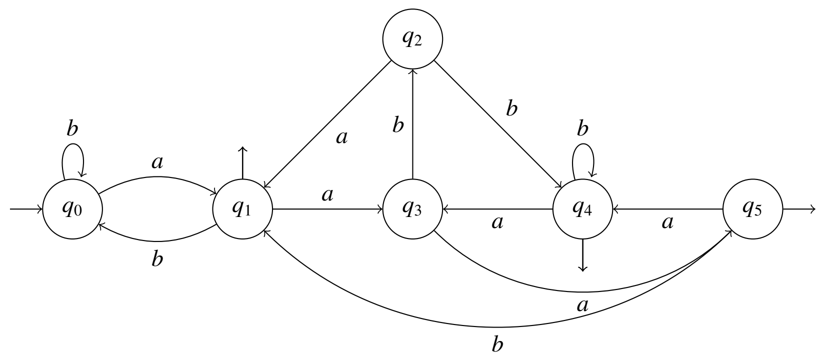
$$\begin{array}{lcl} S & \rightarrow & xSz \mid \epsilon \mid A \\ A & \rightarrow & xAy \mid xy \end{array}$$

ST0270 Lenguajes formales y compiladores - 031 — **Universidad EAFIT**
Segundo parcial — **DIS**
Juan Francisco Cardona McCormick — **29 de septiembre de 2016**

Nombre: ______________________________
Código: ______________________________

1. Minimización de autómatas . . . . 25 puntos

Para el siguiente autómata de estado finito determinista AFD_1 utilice el algoritmo de minimización para encontrar un autómata de estado finito determinista AFD_1', tal que $L(AFD_1) \equiv L(AFD_1')$ y que AFD_1'. Es posible que AFD_1 ya sea el mínimo.



Solución:

Se hace la matriz para evaluar los casos:

$\overrightarrow{q_1}$					
q_2					
q_3					
$\overrightarrow{q_4}$					
$\overrightarrow{q_5}$					
	q_0	$\overrightarrow{q_1}$	q_2	q_3	$\overrightarrow{q_4}$

Se compara los estados finales: $\overrightarrow{q_1}$ y $\overrightarrow{q_4}$ y $\overrightarrow{q_5}$ con cada uno lo que no los son: q_0, q_2 y q_3. Obteniendo que todos ellos son distinguibles.

$\overrightarrow{q_1}$	X				
q_2		X			
q_3		X			
$\overrightarrow{q_4}$	X		X	X	
$\overrightarrow{q_5}$	X		X	X	
	q_0	$\overrightarrow{q_1}$	q_2	q_3	$\overrightarrow{q_4}$

Ahora comparamos q_0 con q_2:

$$
\begin{aligned}
\delta(q_0, a) &= q_1 \\
\delta(q_2, a) &= q_1 \\
\delta(q_0, b) &= q_0 \\
\delta(q_2, b) &= q_0
\end{aligned}
$$

Los resultados q_0 y q_1 ya son distiguibles por lo tanto nos muestran que q_0 y q_2 es distinguible.

$\overrightarrow{q_1}$	X				
q_2	X	X			
q_3		X			
$\overrightarrow{q_4}$	X		X	X	
$\overrightarrow{q_5}$	X		X	X	
	q_0	$\overrightarrow{q_1}$	q_2	q_3	$\overrightarrow{q_4}$

Ahora comparamos con q_0 y q_3 obtenemos:

$$
\begin{aligned}
\delta(q_0, a) &= q_1 \\
\delta(q_3, a) &= q_5 \\
\delta(q_0, b) &= q_1 \\
\delta(q_3, b) &= q_2
\end{aligned}
$$

Mirando los resultados obtenemos q_1 q_2 son distiguibles. Por lo tanto q_0 y q_3 son distiguibles.

$\overrightarrow{q_1}$	X				
q_2	X	X			
q_3	X	X			
$\overrightarrow{q_4}$	X		X	X	
$\overrightarrow{q_5}$	X		X	X	
	q_0	$\overrightarrow{q_1}$	q_2	q_3	$\overrightarrow{q_4}$

Ahora comparamos q_2 y q_3 y se obtiene:

$$
\begin{aligned}
\delta(q_2, a) &= q_1 \\
\delta(q_3, a) &= q_5 \\
\delta(q_2, b) &= q_4 \\
\delta(q_3, b) &= q_2
\end{aligned}
$$

En los resultados obtenemos q_2 y q_4 son distiguibles por lo tanto q_2 y q_3 son distiguibles.

$\overrightarrow{q_1}$	X				
q_2	X	X			
q_3	X	X	X		
$\overrightarrow{q_4}$	X		X	X	
$\overrightarrow{q_5}$	X		X	X	
	q_0	$\overrightarrow{q_1}$	q_2	q_3	$\overrightarrow{q_4}$

Ahora comparamos q_1 y q_4

2. Berry and Sethi . . . . 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = (10 \mid 01)^*01$$

Solución:

3. ***The crab and berry-sethi strike back*** . . . . 25 puntos

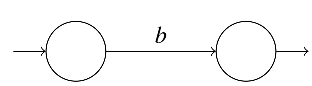
Utilizando el método estructural de Thompson y el método de potencia construir el autómata de estado finito determinista M_3 para la expresión regular r_3:

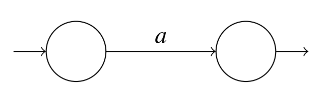
$$r_3 = a^*a$$

De tal forma que $L(M_3) \equiv L(r_3)$.

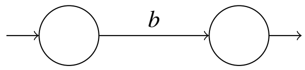
Solución:

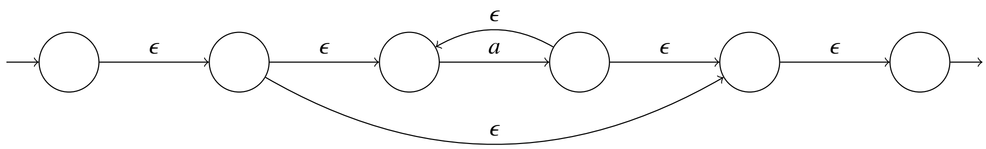
En primer lugar encontramos dos subexpresiones básicas: b y a: Utilizando Thompson construimos los autómatas, correspondientes:



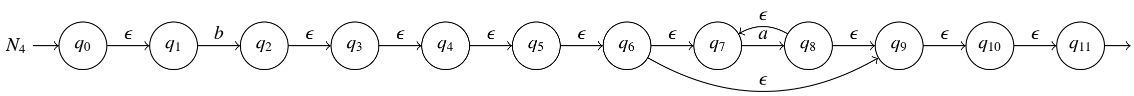


El segundo autómata hace parte de la subexpresion a^*.





Uniendo los dos autómatas a través de la concatenación se obtiene el autómata N_4:



Dado que el autómata es de una expresión regular lineal no es necesario numerar la expresión. Entonces podemos calcular el estado inicial del autómata $Ini(N_4) = \{b\}$.

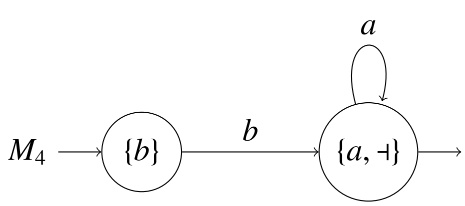
Calculamos el conjunto de los $Dig(N_4) = \{ba, aa\}$.

Calculamos el conjunto de los $Fin(N_4) = \{b, a\}$

Ahora calculamos el conjunto de Fol:

	Fol
b	$a, \dashv$
a	$a, \dashv$

Ahora podemos calcular el autómata M_4 determinista:



ST0270 Lenguajes formales y compiladores - 031 — **Universidad EAFIT**
Tercer parcial — **DIS**
Juan Francisco Cardona McCormick — **3 de Noviembre del 2016**

Nombre: ______________________________
Código: ______________________________

1. Diagrama sintáctico a gramáticas independientes de contexto . . . . 30 puntos

Encuentre la gramática independiente de contexto G_1 correspondiente a los diagrama sintáctico que se ve en la figura 1.

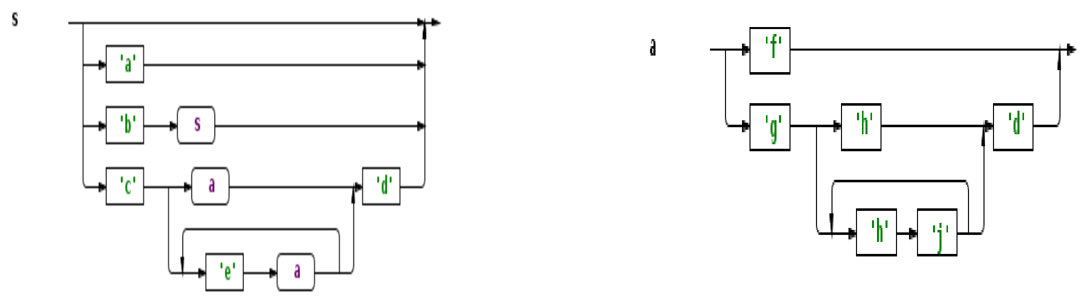


Figura 1: Diagrama sintáctico

2. Autómatas de pila . . . . 35 puntos

Diseñar un autómata de pila A_2 posiblemente no determinista que acepte el siguiente lenguaje:

$$L(A_2) = \{w \in \{0,1\}^* \mid |w|_0 = |w|_1 + 1\}$$

3. Condición $LL(1)$ . . . . 35 puntos

Verificar que la gramática G_3 cumple con la condición $LL(1)$

$$\begin{array}{lcl} S & \rightarrow & zB \mid yA \mid \epsilon \\ A & \rightarrow & zS \mid yAA \\ B & \rightarrow & b \end{array}$$

a) Construya la correspondiente red de autómatas.

b) Calcule los conjunto guías correspondientes y los iniciales y siguientes necesarios para mostrar que la gramática G_2 cumple o no con la condición $LL(1)$.

ST0270 Lenguajes formales y compiladores - 031 | **Universidad EAFIT**
Primer parcial | **DIS**
Juan Francisco Cardona McCormick | **2 de Marzo de 2016**

Nombre: ______________________________

Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Transformación de una gramática independiente de contexto**

Transforme la gramática G_1 en un gramática G_1', de tal forma que la nueva gramática G_1' tiene las siguiente propiedades:

$$\{x \in \Sigma^* \mid S \overset{n}{\Rightarrow} x, S' \overset{m}{\Rightarrow} x, m \ll n\}$$

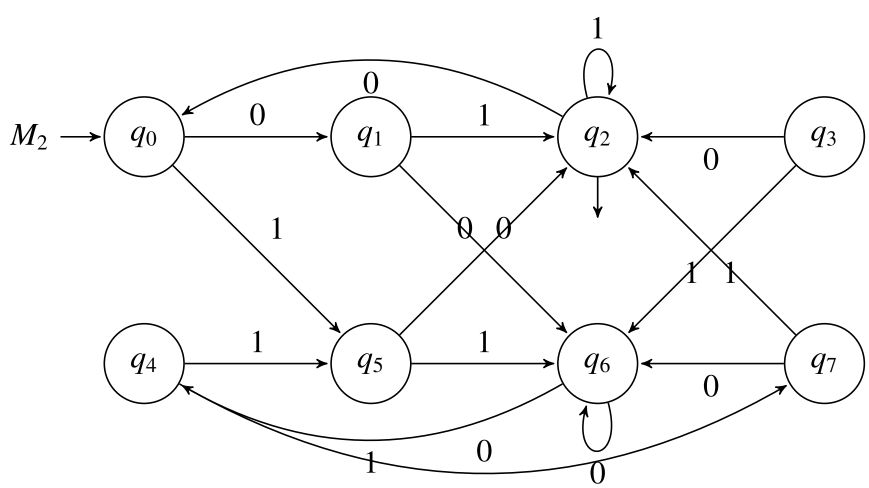
Donde S es el axioma de G_1 y S' es el axioma de G_1'. La siguiente es la definición de la gramática:

$G_1 = (\{S, D, I, E, T, L\}, \{i, c, e, d, b, v, s\}, P_1, S)$ donde $P_1 =$:

$$\begin{array}{rcl} S & \to & DI \mid D \mid I \\ D & \to & ED \mid E \\ I & \to & icI \mid \epsilon \\ E & \to & TL \mid T \\ T & \to & e \mid d \mid b \\ L & \to & vsL \mid vc \end{array}$$

(30 %) 2. **Autómata mínimo**

En la siguiente figura se tiene un autómata de estado finito determinista M_2:



Minimice el autómata M_2 en una nueva versión mínima M_2'.

(20 %) 3. **Gramáticas independientes de contexto extendidas**
Convierta al gramática G_1 en una nueva gramática equivalente G'_3 de tal forma que $L(G_1) \equiv L(G'_3)$ y la nueva gramática G'_3 utilice expresiones regulares en el lado derecho de cada producción.

(20 %) 4. **Gramática independientes de contexto**

Encuentre las gramática independiente de contexto G_4 definida por la expresión regular r_4. Donde $r_4 = ((xz^*) \cup (yw^*))$. De tal forma que $L(G_4) = L(r_4)$.

Nombre: ______________________________
Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Transformación de una gramática independiente de contexto**

Transforme la gramática G_1 en un gramática G_1', de tal forma que la nueva gramática G_1' tiene las siguiente propiedades:

$$\{x \in \Sigma^* \mid S \overset{n}{\Rightarrow} x, S' \overset{m}{\Rightarrow} x, n \ll m\}$$

Donde S es el axioma de G_1 y S' es el axioma de G_1'. La siguiente es la definición de la gramática:

$G_1 = (\{S, C, D, I, E, T, L\}, \{n, i, c, e, v, s, d, b\}, P_1, S)$ donde $P_1 =$:

$$\begin{array}{rcl} S & \rightarrow & CDI \mid C \mid D \mid I \\ C & \rightarrow & nC \mid \epsilon \\ D & \rightarrow & ED \mid E \\ I & \rightarrow & icI \mid i \\ E & \rightarrow & TL \\ T & \rightarrow & e \mid d \mid b \\ L & \rightarrow & vsL \mid vc \end{array}$$

Solución:

Una forma de transformación que produce un árbol más largo es aplicar la transformación de Chomsky o forma normal binaria:

1. *binaria homogénea*: $A \rightarrow BC$, donde $B, C \in V$
2. *terminal con parte única derecha*: $A \rightarrow a$, donde $a \in \Sigma$

Más aun, si la regla vacía está en el lenguaje, existe allí una regla $S \rightarrow \epsilon$ pero la regla vacía no está permitida en la parte derecha.

Entonces, debemos eliminar la reglas nulas del lenguaje. Y solamente permitirlas en el axioma y este no debe estar en el lado derecho. Un renocimiento de la gramática nos muestra que el axioma solamente está en el lado izquierdo. Busquemos que reglas producen la parte nula. Primera iteración reglas que produzca directamente el valor nulo.

$$Null_0 = \{C\}$$

Luego que reglas producen nulo, a partir de estos no-terminales y observamos las reglas homogeneas y nos producen:

$$Null_1 = \{C, S\}$$

Nuevamente aplicamos:

$$Null_2 = \{C, S\}$$

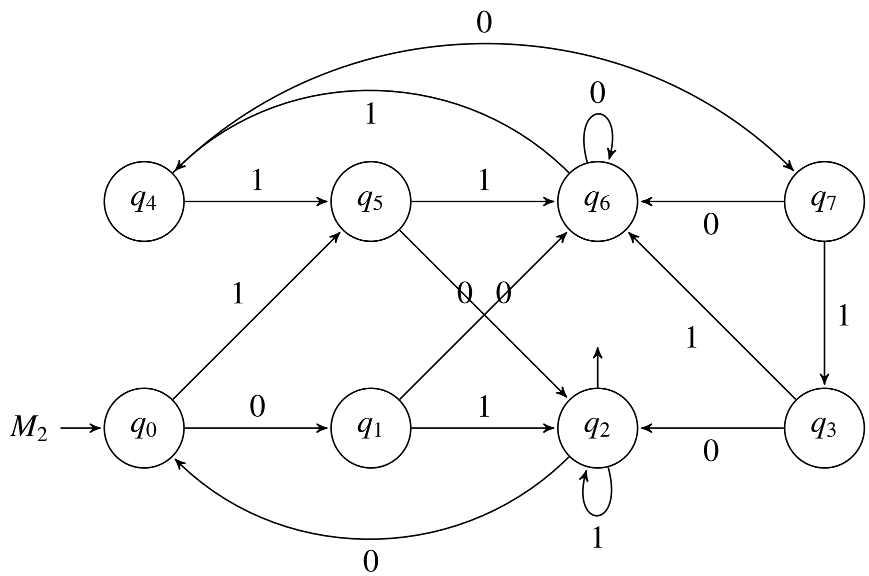
Entonces, nuestro conjunto nulo ya llego a su punto fijo. Eliminamos ahora las reglas nulas.

$$\begin{array}{rcl}
S & \rightarrow & CDI \mid DI \mid C \mid D \mid I \mid \epsilon \\
C & \rightarrow & nC \mid n \\
D & \rightarrow & ED \mid E \\
I & \rightarrow & icI \mid i \\
E & \rightarrow & TL \\
T & \rightarrow & e \mid d \mid b \\
L & \rightarrow & vsL \mid vc
\end{array}$$

Ahora ya podemos transforma en Chosmky:

$$\begin{array}{rcl}
S & \rightarrow & C\langle DI\rangle \mid DI \mid C\langle\epsilon\rangle \mid D\langle\epsilon\rangle \mid I\langle\epsilon\rangle \mid \epsilon \\
\langle DI\rangle & \rightarrow & DI \\
\langle\epsilon\rangle & \rightarrow & \epsilon \\
C & \rightarrow & \langle n\rangle C \mid n \\
\langle n\rangle & \rightarrow & n \\
D & \rightarrow & ED \mid E\langle\epsilon\rangle \\
I & \rightarrow & i\langle cI\rangle \mid i \\
\langle cI\rangle & \rightarrow & \langle c\rangle I \\
\langle c\rangle & \rightarrow & c \\
E & \rightarrow & TL \\
T & \rightarrow & e \mid d \mid b \\
L & \rightarrow & v\langle sL\rangle \mid \langle v\rangle\langle c\rangle \\
\langle sL\rangle & \rightarrow & \langle s\rangle L \\
\langle s\rangle & \rightarrow & s \\
\langle v\rangle & \rightarrow & v \\
\langle c\rangle & \rightarrow & c
\end{array}$$

(30 %) 2. **Autómata mínimo**
En la siguiente figura se tiene un autómata de estado finito determinista M_2:



Minimice el autómata M_2 en una nueva versión mínima M_2'.

Solución:

En primer lugar vamos a calcular que par de estados son distinguibles y cuales son indistinguibles. Observamos que existe un solo estado final q_2. Entonces con los demás este es distiguible:

q_1							
q_2	X	X					
q_3			X				
q_4			X				
q_5			X				
q_6			X				
q_7			X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora vemos q_1 y q_0:

$$\begin{aligned}\delta(q_1,0) &= q_2\\ \delta(q_0,0) &= q_1\end{aligned}$$

q_2 y q_1 son distinguibles por lo tanto q_1 y q_0 también lo son:

q_1	X						
q_2	X	X					
q_3			X				
q_4			X				
q_5			X				
q_6			X				
q_7			X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora vemos q_3 y q_0:

$$\begin{aligned} \delta(q_3, 0) &= q_2 \\ \delta(q_0, 0) &= q_1 \end{aligned}$$

Son también distiguibles:

q_1	X						
q_2	X	X					
q_3	X		X				
q_4			X				
q_5			X				
q_6			X				
q_7			X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora q_4 y q_0:

$$\begin{aligned} \delta(q_4, 0) &= q_7 \\ \delta(q_0, 0) &= q_1 \\ \delta(q_4, 1) &= q_5 \\ \delta(q_0, 1) &= q_5 \end{aligned}$$

No sabemos nada todavía de q_7 y q_1, ya que q_5 por reflexividad es indistiguible, por lo tanto lo dejamos pendiente.

q_1	X						
q_2	X	X					
q_3	X		X				
q_4	P		X				
q_5			X				
q_6			X				
q_7			X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora q_5 y q_0:

$$\begin{array}{rcl} \delta(q_5, 0) & = & q_2 \\ \delta(q_0, 0) & = & q_1 \end{array}$$

q_5 y q_0 son distiguibles:

q_1	X						
q_2	X	X					
q_3	X		X				
q_4	P		X				
q_5	X		X				
q_6			X				
q_7			X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora observamos a q_6 y q_0:

$$\begin{array}{rcl} \delta(q_6, 0) & = & q_6 \\ \delta(q_0, 0) & = & q_1 \\ \delta(q_6, 1) & = & q_4 \\ \delta(q_0, 1) & = & q_5 \end{array}$$

No tenemos nada todavía confirmado entre q_6, q_1 y q_4, q_5:

q_1	X						
q_2	X	X					
q_3	X		X				
q_4	P		X				
q_5	X		X				
q_6	P		X				
q_7			X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos q_7 y q_0:

$$\begin{array}{rcl} \delta(q_7, 0) & = & q_6 \\ \delta(q_0, 0) & = & q_1 \\ \delta(q_7, 1) & = & q_3 \\ \delta(q_0, 1) & = & q_5 \end{array}$$

Igualmente no tenemos nada definitivo sobre los estados: q_6, q_1 y q_3 y q_5 por lo tanto queda pendiente:

q_1	X						
q_2	X	X					
q_3	X		X				
q_4	P		X				
q_5	X		X				
q_6	P		X				
q_7	P		X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Comparemos q_3 y q_1:

$$\begin{array}{rcl} \delta(q_3, 0) & = & q_2 \\ \delta(q_1, 0) & = & q_6 \end{array}$$

Estos estados son distiguibles:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	P		X				
q_5	X		X				
q_6	P		X				
q_7	P		X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos q_4 y q_1:

$$\begin{array}{rcl} \delta(q_4, 0) & = & q_7 \\ \delta(q_1, 0) & = & q_6 \\ \delta(q_4, 1) & = & q_5 \\ \delta(q_1, 1) & = & q_2 \end{array}$$

De esto último q_5 y q_2 son diferenciables. Por lo tanto q_4 y q_1 son diferenciables:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	P	X	X				
q_5	X		X				
q_6	P		X				
q_7	P		X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora revisamos a q_5 y q_1.

$$\begin{aligned}\delta(q_5, 0) &= q_2\\ \delta(q_1, 0) &= q_6\end{aligned}$$

q_2 y q_6 son diferenciables. Por lo tanto q_5 y q_1 son diferenciables:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	P	X	X				
q_5	X	X	X				
q_6	P		X				
q_7	P		X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos q_6 y q_1:

$$\begin{aligned}\delta(q_6, 0) &= q_6\\ \delta(q_1, 0) &= q_6\\ \delta(q_6, 1) &= q_4\\ \delta(q_1, 1) &= q_2\end{aligned}$$

q_4 y q_2 son diferenciables. Por lo tanto q_6 y q_1 son diferenciables. Y también lo son p_6 y p_0; y también lo son p_7 y p_0.

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	P	X	X				
q_5	X	X	X				
q_6	X	X	X				
q_7	X		X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Revisemos: q_7 y q_1. Tenemos:

$$\begin{aligned}\delta(q_7, 0) &= q_6\\ \delta(q_1, 0) &= q_6\\ \delta(q_7, 1) &= q_3\\ \delta(q_1, 1) &= q_2\end{aligned}$$

q_3 y q_2 son diferenciables. Entonces tenemos, q_7 y q_1 son diferenciables. Y por lo tanto también q_4 y q_0 son distinguibles.

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X				
q_5	X	X	X				
q_6	X	X	X				
q_7	X	X	X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos q_4 y q_3.

$$\begin{aligned} \delta(q_4,0) &= q_7 \\ \delta(q_3,0) &= q_2 \end{aligned}$$

q_7 y q_2 son diferenciables por lo tanto también lo son q_4 y q_3.

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X				
q_6	X	X	X				
q_7	X	X	X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos: q_5 y q_3:

$$\begin{aligned} \delta(q_5,0) &= q_2 \\ \delta(q_3,0) &= q_2 \\ \delta(q_5,1) &= q_6 \\ \delta(q_3,1) &= q_6 \end{aligned}$$

Ambos son indistiguibles por lo tanto:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I			
q_6	X	X	X				
q_7	X	X	X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Continuamos con q_6 y q_3:

$$\begin{aligned}\delta(q_6,0) &= q_6\\ \delta(q_3,0) &= q_2\end{aligned}$$

q_6 y q_2 son distiguibles, por lo tanto q_6 y q_3 lo son:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I			
q_6	X	X	X	X			
q_7	X	X	X				
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Se compara q_7 y q_3:

$$\begin{aligned}\delta(q_7,0) &= q_6\\ \delta(q_3,0) &= q_2\end{aligned}$$

q_6 y q_2 son distingibles por lo tanto q_7 y q_3 son distiguibles:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I			
q_6	X	X	X	X			
q_7	X	X	X	X			
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos q_5 y q_4:

$$\begin{aligned}\delta(q_5,0) &= q_2\\ \delta(q_4,0) &= q_7\end{aligned}$$

q_2 y q_7 son distingibles, por lo tanto q_5 y q_4 son distingibles:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I	X		
q_6	X	X	X	X			
q_7	X	X	X	X			
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos q_6 y q_4.

$$\begin{aligned}\delta(q_6,0) &= q_6\\ \delta(q_4,0) &= q_7\\ \delta(q_6,1) &= q_4\\ \delta(q_4,1) &= q_5\end{aligned}$$

q_4 y q_5 son distingibles por lo tanto q_6 y q_4 son distinguibles.

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I	X		
q_6	X	X	X	X	X		
q_7	X	X	X	X			
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora comparamos a q_7 y q_4.

$$\begin{aligned}\delta(q_7,0) &= q_6\\ \delta(q_4,0) &= q_7\\ \delta(q_7,1) &= q_3\\ \delta(q_4,1) &= q_5\end{aligned}$$

q_3 y q_5 son indistingibles, pero no sabemos nada aun d q_6 y q_7 entonces queda pendiente.

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I	X		
q_6	X	X	X	X	X		
q_7	X	X	X	X	P		
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Revisar ahora q_6 y q_5.

$$\begin{aligned}\delta(q_6,0) &= q_6\\ \delta(q_5,0) &= q_2\end{aligned}$$

q_6 y q_2 son distingibles, por lo tanto q_6 y q_5 lo son:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I	X		
q_6	X	X	X	X	X	X	
q_7	X	X	X	X	P		
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Corresponde a q_7 y q_5.

$$\begin{aligned} \delta(q_7, 0) &= q_6 \\ \delta(q_5, 0) &= q_2 \end{aligned}$$

q_6 y q_2 son distingibles. Por lo tanto q_7 y q_5 son distingibles:

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I	X		
q_6	X	X	X	X	X	X	
q_7	X	X	X	X	P	X	
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

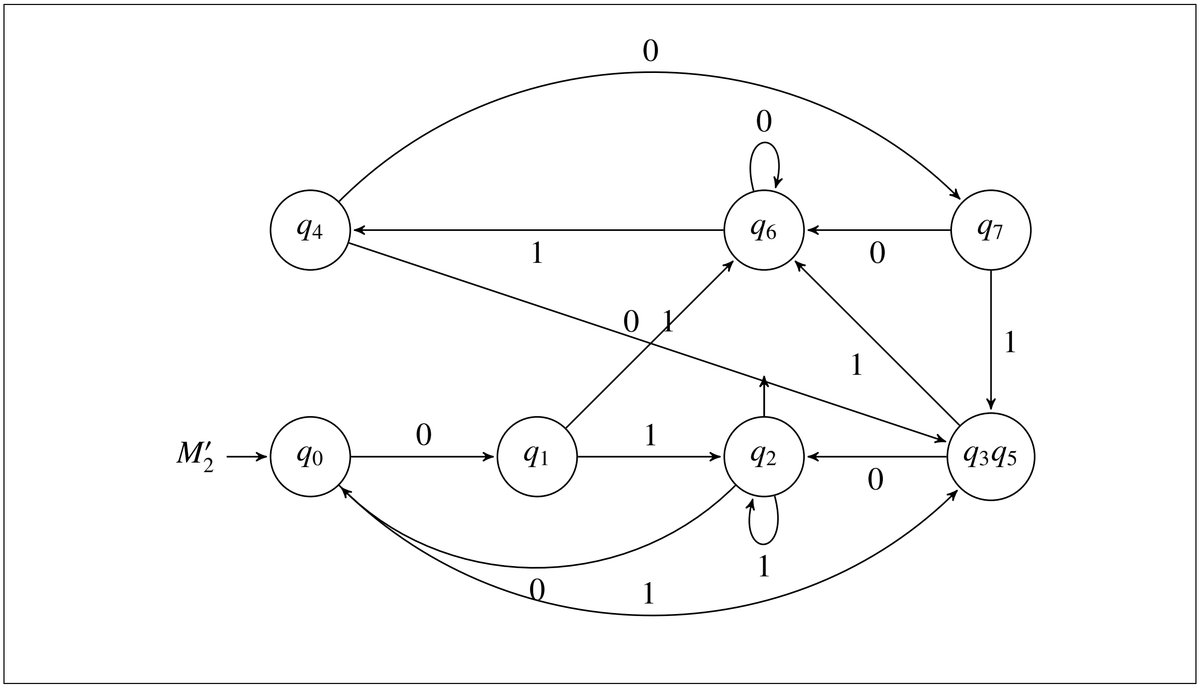
Finalmente, q_7 y q_6

$$\begin{aligned} \delta(q_7, 0) &= q_6 \\ \delta(q_6, 0) &= q_6 \\ \delta(q_7, 1) &= q_3 \\ \delta(q_6, 1) &= q_4 \end{aligned}$$

q_3 y q_4 son distingibles, por lo tanto q_7 y q_6, igualmente q_7 y q_4 lo son.

q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	X	X			
q_5	X	X	X	I	X		
q_6	X	X	X	X	X	X	
q_7	X	X	X	X	X	X	X
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Ahora debemos reconstruir el autómata con un nuevo estado combiando entre q_5 Y q_3


0
0
q_4
q_6
q_7
1
0
0
1
1
1
M'_2
q_0
0
q_1
1
q_2
q_3q_5
0
1
0
1

(20 %) 3. **Gramáticas independientes de contexto extendidas**
Convierta al gramática G_3 en una nueva gramática equivalente G'_3 de tal forma que $L(G_3) \equiv L(G'_3)$ y la nueva gramática G'_3 utilice expresiones regulares en el lado derecho de cada producción.

$G_3 = (\{S, N, D, I, E, T, L\}, \{n, c, i, e, s, v, d, b\}, P_3, S)$ donde P_3 =:

$$\begin{array}{rcl} S & \rightarrow & NDI \mid D \mid I \\ N & \rightarrow & Nn \mid \epsilon \\ D & \rightarrow & DE \mid \epsilon \\ I & \rightarrow & Ici \mid i \\ E & \rightarrow & TLc \\ T & \rightarrow & e \mid d \mid b \\ L & \rightarrow & Lsv \mid v \end{array}$$

Solución:

$$\begin{array}{rcl} S & \rightarrow & NDI \mid D \mid I \\ N & \rightarrow & n^* \\ D & \rightarrow & E^* \\ I & \rightarrow & (ic)^*i \\ E & \rightarrow & TLc \\ T & \rightarrow & e \mid d \mid b \\ L & \rightarrow & v(sv)^* \end{array}$$

(20 %) 4. **Gramática independientes de contexto**

Encuentre las gramática independiente de contexto G_4 definida por la expresión regular r_4. Donde $r_4 = ((x^*y^*) \cup (z^*w^*))$. De tal forma que $L(G_4) = L(r_4)$.

Solución:

$$
\begin{array}{rcl}
E_0 & \rightarrow & E_1 \cup E_6 \\
E_1 & \rightarrow & E_2E_4 \\
E_2 & \rightarrow & E_3E_2 \mid \epsilon \\
E_3 & \rightarrow & x \\
E_4 & \rightarrow & E_5E_4 \mid \epsilon \\
E_5 & \rightarrow & y \\
E_6 & \rightarrow & E_7E_9 \\
E_7 & \rightarrow & E_8E_7 \mid \epsilon \\
E_8 & \rightarrow & z \\
E_9 & \rightarrow & E_{10}E_9 \mid \epsilon \\
E_{10} & \rightarrow & w
\end{array}
$$

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **6 de Abril de 2017**

Nombre: ______________________________
Código: ______________________________

1. Autómata de pila .... 40 puntos

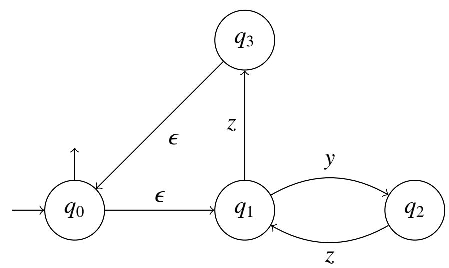
Construya un autómata de pila que reconozca el siguiente lenguaje: $\{a^n b^{n+m} a^m \mid n, m \geq 0\}$.

2. Intersección, negación y complementos de expresiones regulares .... 30 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión intersección de la expresión regular: $a^*a \cap aa^*$.

3. Berry-Sethi strikes back .... 30 puntos

En la figura se observa un autómata finito no-determinista con movimientos espontáneos N construya un autómata finito determinista M tal que $L(N) \equiv L(M)$.

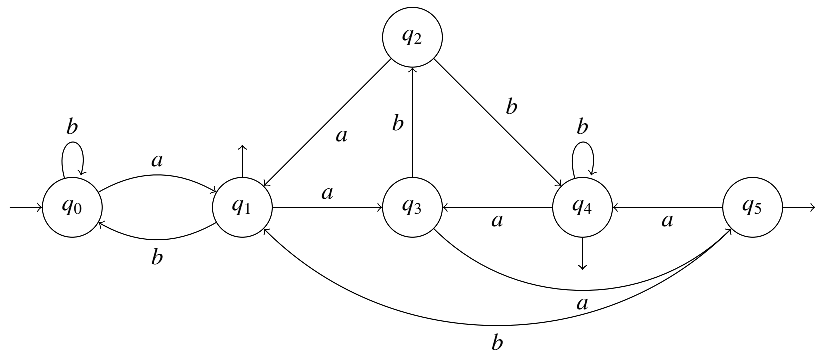


Nombre: ______________________________

Código: ______________________________

1. Autom . . . . 25 puntos

Para el siguiente autómata de estado finito determinista AFD_1 utilice el algoritmo de minimización para encontrar un autómata de estado finito determinista AFD'_1, tal que $L(AFD_1) \equiv L(AFD'_1)$ y que AFD'_1. Es posible que AFD_1 ya sea el mínimo.



2. Berry and Sethi . . . . 25 puntos

Utilizando el algoritmo de Berry and Sethi definir el autómata de estado finito determinista para la expresión regular r_2:

$$r_2 = (10 \mid 01)^*01$$

3. *The crab and berry-sethi strike back* . . . . 25 puntos

Utilizando el método estructural de Thompson y el método de potencia construir el autómata de estado finito determinista M_3 para la expresión regular r_3:

$$r_3 = a^*a$$

De tal forma que $L(M_3) \equiv L(r_3)$.

Nombre: ______________________________

Código: ______________________________

1. Transformar la siguiente gramática en $LL(1)$ . . . . 20 puntos

La siguiente gramática G_1 no cumple con la condición $LL(1)$, transforme la anterior gramática en una gramática G_1' que cumpla con la condición $LL(1)$.

$$
\begin{array}{lcl}
S & \rightarrow & id \\
S & \rightarrow & S + id \\
S & \rightarrow & S - id \\
S & \rightarrow & S \; ; \; [S]
\end{array}
$$

2. Condición $LL(1)$ . . . . 20 puntos

Verificar la condición $LL(1)$ de la gramática G_2:

$$
\begin{array}{rcl}
P & \rightarrow & iEtPP' \mid a \\
P' & \rightarrow & eP \mid \epsilon \\
E & \rightarrow & b
\end{array}
$$

3. Condición $LR(0)$ . . . . 20 puntos

Verificar la condición $LR(0)$ de la gramática G_3 construyendo el piloto correspondiente:

$$
\begin{array}{rcl}
S_0 & \rightarrow & S \dashv \\
S & \rightarrow & AA \\
A & \rightarrow & aA \mid a
\end{array}
$$

4. Condición $LR(1)$ . . . . 40 puntos

Verificar la condición $LR(1)$ de la gramática G_4 construyendo el piloto correspondiente:

$$
\begin{array}{rcl}
S_0 & \rightarrow & S \dashv \\
S & \rightarrow & AA \mid BA \\
A & \rightarrow & aA \mid a \\
B & \rightarrow & bB \mid b
\end{array}
$$

ST0270 Lenguajes formales y compiladores - 032 | **Universidad EAFIT**
Tercer parcial | **DIS**
Juan Francisco Cardona McCormick | **19 de Mayo de 2017**

Nombre: ____________________
Código: ____________________

1. Transformar la siguiente gramática en $LL(1)$ . . . . 20 puntos

La siguiente gramática G_1 no cumple con la condición $LL(1)$, transforme la anterior gramática en una gramática G_1' que cumpla con la condición $LL(1)$.

$$
\begin{array}{lcl}
S & \rightarrow & AB \mid \epsilon \\
A & \rightarrow & Aa \mid a \mid c \\
B & \rightarrow & Bb \mid C \mid b \\
C & \rightarrow & Cv \mid w
\end{array}
$$

2. Condición $LL(1)$ . . . . 20 puntos

Verificar la condición $LL(1)$ de la gramática G_2:

$$
\begin{array}{lcl}
A & \rightarrow & sFeAB \mid w \\
B & \rightarrow & *A \mid \epsilon \\
F & \rightarrow & f
\end{array}
$$

3. Condición $LR(0)$ . . . . 20 puntos

Verificar la condición $LR(0)$ de la gramática G_3 construyendo el piloto correspondiente:

$$
\begin{array}{rcl}
S_0 & \rightarrow & S \dashv \\
S & \rightarrow & [L] \\
S & \rightarrow & a \\
L & \rightarrow & L, E \\
L & \rightarrow & E
\end{array}
$$

4. Condición $LR(1)$ . . . . 40 puntos

Verificar la condición $LR(1)$ de la gramática G_4 construyendo el piloto correspondiente:

$$
\begin{array}{rcl}
S_0 & \rightarrow & S \dashv \\
S & \rightarrow & [L] \\
S & \rightarrow & a \\
L & \rightarrow & L, E \\
L & \rightarrow & E
\end{array}
$$

Nombre: ____________________

Código: ____________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Transformación de una gramática independiente de contexto**

Tranforme la siguiente gramática G_1 en una gramática G_1' de tal forma que la gramática esta libre de reglas de copia.

Donde S es el axioma de G_1 y S' es el axioma de G_1'. La siguiente es la definición de la gramática:

$G_1 = (V_1 \equiv \{E_0, E_1, E_2, E_3\}, \Sigma_1 \equiv \{a, b, c\}, P_1, S_1 \equiv \{E_0\})$ donde $P_1 \equiv$:

$$\begin{array}{lcl} E_0 & \rightarrow & E_0E_1E_2 \mid E_1E_2 \\ E_1 & \rightarrow & aE_1 \mid \epsilon \\ E_2 & \rightarrow & bE_2 \mid\mid E_3 \mid \epsilon \\ E_3 & \rightarrow & cE_3 \mid c \end{array}$$

(24 %) 2. **Expresiones regulares**

Encuentre la expresión regular R_2 cuyo lenguaje tiene la propiedad $L(R_2) \equiv L(G_2)$. Donde $G_2 = (V_2 \equiv \{A, B, C\}, \Sigma_2 \equiv \{a, b, c\}, P_2, S_2 \equiv \{A\})$ y donde $P_2 \equiv$:

$$\begin{array}{lcl} A & \rightarrow & aB \\ A & \rightarrow & a \\ B & \rightarrow & bC \\ B & \rightarrow & b \\ C & \rightarrow & c \\ C & \rightarrow & cB \end{array}$$

(23 %) 3. **Gramática independientes de contexto**

Encuentre la gramática independiente de contexto G_3 que define el siguiente lenguaje:

$$L(G3) \equiv \{w^a x^b y^c \mid a \neq b + c\}$$

(23 %) 4. **Eliminar ambiguedad**

La siguiente gramática G_4 es ambigua construya una gramática G_4', tal que esta no sea ambigua y tenga la siguiente propiedad $L(G_4) \equiv L(G_4')$.

$$\begin{array}{lcl} I & \rightarrow & JK \\ J & \rightarrow & jJ \mid kJ \mid \epsilon \\ K & \rightarrow & kK \mid lK \mid \epsilon \end{array}$$

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Primer parcial **DIS**
Juan Francisco Cardona McCormick **18 de Agosto de 2017**

Nombre: ______________________________
Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Transformación de una gramática independiente de contexto**

Tranforme la siguiente gramática G_1 en una gramática G_1' de tal forma que la gramática esta libre de reglas de copia.

Donde S es el axioma de G_1 y S' es el axioma de G_1'. La siguiente es la definición de la gramática:

$G_1 = (V_1 \equiv \{E_0, E_1, E_2, E_3, E_4\}, \Sigma_1 \equiv \{a, b, c, d\}, P_1, S_1 \equiv \{E_0\})$ donde $P_1 \equiv$:

$$\begin{array}{lcl} E_0 & \rightarrow & E_2E_1E_0 \mid E_2E_1 \\ E_1 & \rightarrow & aE_1 \mid \epsilon \\ E_2 & \rightarrow & bE_2 \mid E_3E_4 \mid \epsilon \\ E_3 & \rightarrow & cE_3 \mid c \mid \epsilon \\ E_4 & \rightarrow & dE_4 \mid \epsilon \end{array}$$

(24 %) 2. **Expresiones regulares**

Encuentre la expresión regular R_2 cuyo lenguaje tiene la propiedad $L(R_2) \equiv L(G_2)$. Donde $G_2 = (V_2 \equiv \{X, Y, Z\}, \Sigma_2 \equiv \{i, j, k\}, P_2, S_2 \equiv \{X\})$ y donde $P_2 \equiv$:

$$\begin{array}{lcl} X & \rightarrow & iY \\ Y & \rightarrow & \epsilon \\ Y & \rightarrow & jZ \\ Z & \rightarrow & \epsilon \\ Z & \rightarrow & kY \end{array}$$

(23 %) 3. **Gramática independientes de contexto**

Encuentre la gramática independiente de contexto G_3 que define el siguiente lenguaje:

$$L(G3) \equiv \{a^x b^y b^z \mid y \neq x + z\}$$

(23 %) 4. **Eliminar ambiguedad**

La siguiente gramática G_4 es ambigua construya una gramática G_4', tal que esta no sea ambigua y tenga la siguiente propiedad $L(G_4) \equiv L(G_4')$.

$$\begin{array}{lcl} D & \rightarrow & DKD \mid DTD \mid a \\ K & \rightarrow & k \mid f \\ T & \rightarrow & l \mid s \end{array}$$

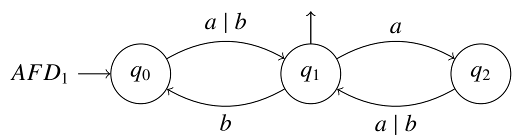
ST0270 Lenguajes formales y compiladores - 031 | **Universidad EAFIT**
Segundo parcial | **DIS**
Juan Francisco Cardona McCormick | **21 de Septiembre de 2017**

Nombre: ____

Código: ____

1. BMC .... 20 puntos

Obtener la expresión regular e_1 para el autómata finito determinista AFD_1, tal que $L(e_1) \equiv L(AFD_1)$.



2. Thompson eats a big crab .... 50 puntos

Se tiene la expresión regular $e_2 = a^*b$. Realizar la siguientes operaciones:

a) Construir el autómata finito no-determinista AFN_2, tal que $L(AFN_2) \equiv L(e_2)$.

b) Construir la gramática lineal GIC_2 por la izquierda a partir del autómata AFN_2, tal que $L(GIC_2) \equiv L(AFN_2)$.

c) Construir el autómata finito no-determinista sin movimientos espontáneos $AFN_2^{sin-\epsilon}$ a partir de AFN_2, utilizando el método de eliminación de reglas de copia, tal que $L(AFN_2) \equiv L(AFN_2^{sin-\epsilon})$

d) Construir el autómata finito determinista AFD_2 a partir de $AFN_2^{sin-\epsilon}$ utilizando la construcción del conjunto de potencia, tal que $L(AFD_2) \equiv L(AFN_2^{sin-\epsilon})$.

3. Berry-Sethi .... 30 puntos

Construir el autómata finito determinista AFD_3 de la expresión regular $e_3 = 1^*(0|1)^+(00|11)^*0^*$. Tal que: $L(AFD_3) \equiv L(e_3)$.

Nombre: ______________________________
Código: ______________________________

1. A minimal crab . . . . 30 puntos

La siguiente gramática independiente de contexto por la *izquierda*, produce un autómata finito determinista AFD_1. Encuentre el autómata minimo $\widehat{AFD}_1$ de forma tal que $L(AFD_1) \equiv L(\widehat{AFD}_1)$.

$$\begin{array}{lcl} A & \rightarrow & B \mid C \\ B & \rightarrow & D0 \mid E0 \\ C & \rightarrow & D1 \mid E1 \\ D & \rightarrow & F0 \\ E & \rightarrow & E1 \\ F & \rightarrow & \epsilon \end{array}$$

2. Thompson eats a big crab . . . . 50 puntos

Se tiene la expresión regular $e_2 = ba^*$. Realizar la siguientes operaciones:

a) Construir el autómata finito no-determinista AFN_2, tal que $L(AFN_2) \equiv L(e_2)$.

b) Construir la gramática lineal GIC_2 por la izquierda a partir del autómata AFN_2, tal que $L(GIC_2) \equiv L(AFN_2)$.

c) Construir el autómata finito no-determinista sin movimientos espontáneos $AFN_2^{sin-\epsilon}$ a partir de AFN_2, utilizando el método de eliminación de reglas de copia, tal que $L(AFN_2) \equiv L(AFN_2^{sin-\epsilon})$

d) Construir el autómata finito determinista AFD_2 a partir de $AFN_2^{sin-\epsilon}$ utilizando la construcción del conjunto de potencia, tal que $L(AFD_2) \equiv L(AFN_2^{sin-\epsilon})$.

3. Berry-Sethi doesn't fear bigger crabs . . . . 20 puntos

Construir el autómata finito determinista AFD_3 de la expresión regular $e_3 = 0^*(01|10)^+1^*$. Tal que: $L(AFD_3) \equiv L(e_3)$.

Nombre: ____________________

Código: ____________________

1. Autómata de pila .... 30 puntos

Implemente un autómata de pila determinista o no determinista que reconoce el siguiente lenguaje L_1

$$L_1 = \{x \in \{j, k, l\}^* \mid |x|_j < |x|_k \vee |x|_j < |x|_l\}$$

Solución:

2. Condición $LR(0)$ . . . . 30 puntos

Verificar la condición $LR(0)$ de la gramática G_2 construyendo el piloto correspondiente:

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & (L) \mid x \\ L & \rightarrow & S \mid L, S \end{array}$$

Solución:

3. Condición $LR(1)$ . . . . 40 puntos

Verificar la condición $LR(1)$ de la gramática G_3 construyendo el piloto correspondiente:

$$
\begin{array}{rcl}
S_0 & \rightarrow & S \dashv \\
S & \rightarrow & L = R\,; \mid R\,; \\
L & \rightarrow & Id \mid *R \\
R & \rightarrow & L
\end{array}
$$

Nombre: ____________________
Código: ____________________

1. Autómata de pila . . . . 30 puntos

Implemente un autómata de pila determinista o no determinista que reconoce el siguiente lenguaje L_1

$$L_1 = \{a^i b^j c^k \mid i, j, k \geq 0 \wedge j = i \vee j = k\}$$

Solución:

2. Condición $LR(0)$ . . . . 30 puntos

Verificar la condición $LR(0)$ de la gramática G_2 construyendo el piloto correspondiente:

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & i\,S\,i \mid j\,S\,j \mid k \end{array}$$

Solución:

3. Condición *LR*(1) . . . . 40 puntos

Verificar la condición $LR(1)$ de la gramática G_3 construyendo el piloto correspondiente:

$$
\begin{array}{rcl}
S_0 & \rightarrow & A \dashv \\
S & \rightarrow & A\,a \mid S\ b \mid B\,c \\
A & \rightarrow & S\ b \mid b \\
B & \rightarrow & S\ c \mid c
\end{array}
$$

Solución:

Nombre: ______________________________

Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Transformación de una gramática independiente de contexto**

Transforme la siguiente gramática G_1 en una gramática G'_1 sin recursividad por la izquierda

Donde S es el axioma de G_1 y S' es el axioma de G'_1. La siguiente es la definición de la gramática:

$G_1 = (V_1 \equiv \{A, B, C, D\}, \Sigma_1 \equiv \{a, b, c, d\}, P_1, S_1 \equiv \{A\})$ donde $P_1 \equiv$:

$$
\begin{array}{rcl}
A & \rightarrow & Aa \mid BC \mid BD \\
B & \rightarrow & Ab \mid Bb \mid \epsilon \\
C & \rightarrow & Cc \mid Bb \mid \epsilon \\
D & \rightarrow & Dd \mid \epsilon
\end{array}
$$

Solución:

(24 %) 2. **Expresiones regulares**

Encuentre la expresión regular R_2 para el alfabeto $\Sigma_2 = \{a, b\}$ cuyo lenguaje tiene la propiedad $L(R_2) = \{x \in \Sigma_2^* \mid x_a \; mod \; 3 \equiv 0\}$. Donde *mod* es la operación residuo.

(23 %) 3. **Gramática independientes de contexto**

Encuentre la gramática independiente de contexto G_3, sobre el alfabeto $\Sigma = \{v, w, x\}$ que define el siguiente lenguaje:

$$L(G3) \equiv \{v^a w^b x^c \mid b = a \lor b = c\}$$

(23 %) 4. **Eliminar ambiguedad**

La siguiente gramática G_4 es ambigua construya una gramática G'_4, tal que esta no sea ambigua y tenga la siguiente propiedad $L(G_4) \equiv L(G'_4)$.

$$
\begin{array}{rcl}
\vdash & \rightarrow & \nabla\sqcup \perp\vdash \triangleleft \vdash \\
 & \mid & \nabla\sqcup \perp\vdash \\
 & \mid & *
\end{array}
$$

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Primer parcial **DIS**
Juan Francisco Cardona McCormick **23 de febrero de 2017**

Nombre: ______________________________
Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Transformación de una gramática independiente de contexto**

Transforme la siguiente gramática G_1 en una gramática G_1' que tiene forma normal Chomsky.

$G_1 = (V_1 \equiv \{A, B, C, D\}, \Sigma_1 \equiv \{a, b, c, d\}, P_1, S_1 \equiv \{A\})$ donde $P_1 \equiv$:

$$\begin{array}{rcl} A & \rightarrow & ABC \mid BCA \mid CDAB \mid a \\ B & \rightarrow & bABC \mid bBCA \mid b \mid \epsilon \\ C & \rightarrow & cABC \mid cBCD \mid c \mid \epsilon \\ D & \rightarrow & Dd \mid dD \end{array}$$

(24 %) 2. **Expresiones regulares**
Encuentre la expresión regular R_2 para el alfabeto $\Sigma_2 = \{y, z\}$ cuyo lenguaje tiene la propiedad $L(R_2) = \{x \in \Sigma_2^* \mid x_z \; mod \; 5 \equiv 0\}$. Donde *mod* es la operación residuo.

(23 %) 3. **Gramática independientes de contexto**

Encuentre la gramática independiente de contexto G_3, sobre el alfabeto $\Sigma = \{y, z\}$ que define el siguiente lenguaje:

$$L(G3) \equiv \{y^a z^b \mid a \leq b \leq 2a\}$$

(23 %) 4. **Eliminar ambiguedad**
La siguiente gramática G_4 construye mensajes de datos sobre el alfabeto $\Sigma_4 = \{A, C, E, R, T\}$

$$\begin{array}{rcl} MESS & \rightarrow & A\,MESS \mid C\,MESS \mid E\,MESS \mid R\,MESS \mid T\,MESS \\ & \mid & A \mid C \mid E \mid R \mid T \end{array}$$

Los símbolos fuentes son traducidos a un lenguaje binario $\Gamma = \{0, 1\}$, utilizando la siguiente sistema de sustitución:

$$\begin{array}{r} A\backslash 01 \\ C\backslash 11 \\ D\backslash 100 \\ R\backslash 101 \\ T\backslash 110 \end{array}$$

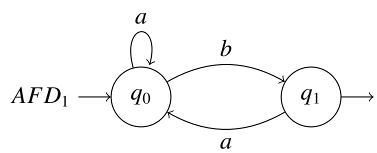
Genera que los mensajes sean ambiguos. Genere una sustitución que produzca un lenguaje no ambiguo. Puede utilizar los digitos necesarios para producir una gramática no ambigua.

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **5 de abril de 2018**

Nombre: ____
Código: ____

1. Obtención de la expresión regular de un autómata de estado finito . . . . 30 puntos

En la siguiente figura se tiene un AFD_1.



Calcule lo siguiente:

a) (15) Utilizando identidad de Arden calcule la expresión regular e_1 para AFD_1. De tal forma que $L(AFD_1) \equiv L(e_1)$.

b) (15) Utilizando BMC calcule la expresión regular e_1' para AFD_1. De tal forma que $L(AFD_1) \equiv L(e_1')$.

Solución:

a) En primer lugar hay que convertir el automata en una gramática.

$$\begin{array}{rcl} q_0 & \rightarrow & aq_0 \mid bq_1 \\ q_1 & \rightarrow & aq_0 \mid \epsilon \end{array}$$

Reescribamos la gramática para facilitar:

$$\begin{array}{rcl} A & \rightarrow & aA \mid bB \\ B & \rightarrow & aA \mid \epsilon \end{array}$$

Convertimos a ecuaciones de lenguajes:

$$\begin{array}{rcl} L_A & \rightarrow & aL_A \mid bL_B \\ L_B & \rightarrow & aL_A \mid \epsilon \end{array}$$

Sustituimos L_B en L_A y se obtiene:

$$L_A \rightarrow aL_A \mid b(aL_A \mid \epsilon)$$

Obtenemos:

$$L_A \rightarrow aL_A \mid baL_A \mid b$$

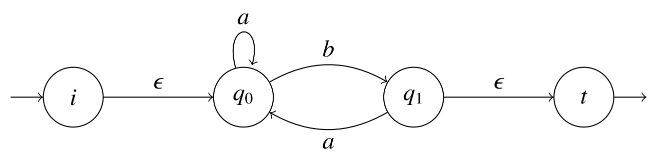
Factorizamos:

$$L_A \rightarrow (a \mid ba)L_A \mid b$$

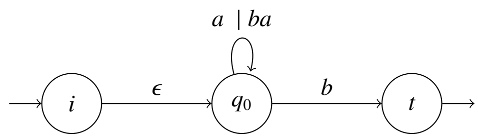
Aplicando Arden se obtiene:

$$L_A \equiv (a \mid ba)^*b$$

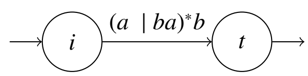
b) Aplicando BMC, construimos un nuevo automáta.



Eliminemos q_1.

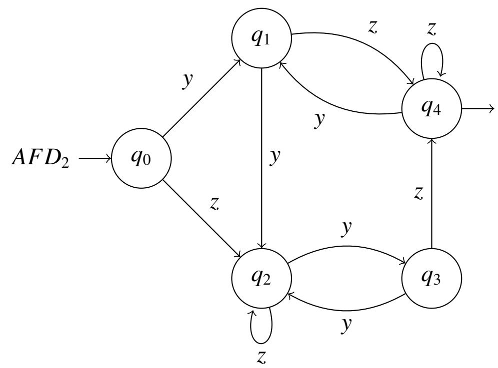


Eliminemos q_0.



2. Minimiza a littler crab . . . . 40 puntos

En la figura se tiene un AFD_2. Encuentre la versión mínima AFD'_2 de tal forma que se cumpla la siguiente ecuación: $L(AFD_2) \equiv L(AFD'_2)$:



Solución:

Realizamos la minimización haciendo la matriz:

q_1				
q_2				
q_3				
q_4				
	q_0	q_1	q_2	q_3

Se compara los estados finales con los que no son: q_4 que es final con el resto:

q_1				
q_2				
q_3				
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

$$\begin{aligned}\delta(q_0, y) &= q_1\\ \delta(q_1, y) &= q_2\end{aligned} \qquad \begin{aligned}\delta(q_0, z) &= q_2\\ \delta(q_1, z) &= q_4\end{aligned}$$

Observamos que q_2 y q_4 son distinguibles obtenemos que q_1 y q_0 son distinguibles.

q_1	X			
q_2				
q_3				
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Analizamos ahora q_2 y q_0:

$$\begin{aligned}\delta(q_0, y) &= q_1 \\ \delta(q_2, y) &= q_3\end{aligned} \qquad \begin{aligned}\delta(q_0, z) &= q_2 \\ \delta(q_2, z) &= q_2\end{aligned}$$

Todavía no tenemos información relevante para determinar si son distiguibles o indistiguibles. Postponemos hasta saber de q_1 y q_3.

Comparemos q_1 y q_2:

$$\begin{aligned}\delta(q_1, y) &= q_2 \\ \delta(q_2, y) &= q_3\end{aligned} \qquad \begin{aligned}\delta(q_1, z) &= q_4 \\ \delta(q_2, z) &= q_2\end{aligned}$$

Se observa que q_4 y q_2 son distiguibles por lo tanto q_1 y q_2 son distiguibles:

q_1	X			
q_2		X		
q_3				
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Ahora comparamos $q3$ y q_0:

$$\begin{aligned}\delta(q_0, y) &= q_1 \\ \delta(q_3, y) &= q_2\end{aligned} \qquad \begin{aligned}\delta(q_0, z) &= q_2 \\ \delta(q_3, z) &= q_4\end{aligned}$$

Se observa q_2 y q_4 son distiguibles por lo tanto q_0 y q_3 son distiguibles.

q_1	X			
q_2		X		
q_3	X			
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Ahora vamos a revisar q_1 y q_3.

$$\begin{aligned}\delta(q_1, y) &= q_2 \\ \delta(q_3, y) &= q_2\end{aligned} \qquad \begin{aligned}\delta(q_1, z) &= q_4 \\ \delta(q_3, z) &= q_4\end{aligned}$$

Tienen un comportamiento similar por lo tanto son indistiguibles:

q_1	X			
q_2		X		
q_3	X	I		
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Y también q_0 y q_2

q_1	X			
q_2	I	X		
q_3	X	I		
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Ahora, examinamos q_2 y q_3:

$$\begin{aligned} \delta(q_2, y) &= q_3 \\ \delta(q_3, y) &= q_2 \end{aligned}$$

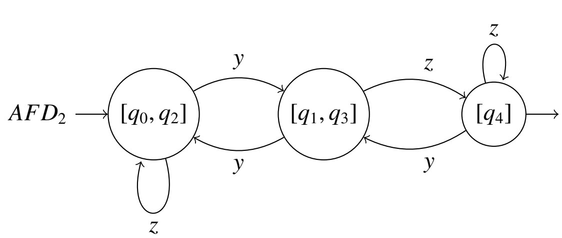
$$\begin{aligned} \delta(q_2, z) &= q_2 \\ \delta(q_3, z) &= q_4 \end{aligned}$$

Son distiguibles:

q_1	X			
q_2	I	X		
q_3	X	I	X	
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Ahora q_0 y q_2 son el mismo estado y q_1 y q_3 son igualmente el mismo estado.

Ahora construimos el autómata:



3. Berry-Sethi . . . . 30 puntos

Construir el autómata finito determinista AFD_3 de la expresión regular $e_3 = (a \mid b \mid c)^+(ab \mid bc)^*c^*$. Tal que: $L(AFD_3) \equiv L(e_3)$.

Solución:

Enumeralos la expresión:

$$e'_3 = (a_1 \mid b_2 \mid c_3)^+(a_4b_5 \mid b_6c_7)^*c_8^*$$

Calculamos los conjuntos:

$$\begin{array}{rcl}
Ini(e'_3 \dashv) & = & \{a_1, b_2, c_3\} \\
Fin(e'_3) & = & \{a_1, b_2, c_3, b_5, c_7, c_8\} \\
Dig(e'_3) & = & \{a_1a_1, a_1b_2, a_1c_3, a_1a_4, a_1b_6, a_1c_8, \\
 & & b_2a_1, b_2b_2, b_2c_3, b_2a_4, b_2b_6, b_2c_8, \\
 & & c_3a_1, c_3b_2, c_3c_3, c_3a_4, c_3b_6, c_3c_8, \\
 & & a_4b_5, \\
 & & b_5a_4, b_5b_6, b_5c_8, \\
 & & b_6c_7, \\
 & & c_7a_4, c_7b_6, c_7c_8, \\
 & & c_8c_8\} \\
Fol(a_1) & = & \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \\
Fol(b_2) & = & \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \\
Fol(c_3) & = & \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \\
Fol(a_4) & = & \{b_5\} \\
Fol(b_5) & = & \{a_4, b_6, c_8, \dashv\} \\
Fol(b_6) & = & \{c_7\} \\
Fol(c_7) & = & \{a_4, b_6, c_8, \dashv\} \\
Fol(c_8) & = & \{c_8, \dashv\}
\end{array}$$

Estado inicial: $\{a_1, b_2, c_3\} \equiv q_0$.

Con el terminal a:

$$Fol(a_1) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \equiv q_1$$

Con el terminal b:

$$Fol(b_2) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \equiv q_1$$

Con el terminal c:

$$Fol(c_3) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \equiv q_1$$

A partil del estado $q_1 \equiv \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\}$.

Con el terminal a:

$$Fol(a_1) \cup Fol(a_4) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \cup \{b_5\} = \{a_1, b_2, c_3, a_4, b_5, b_6, c_8, \dashv\} \equiv q_2$$

Con el terminal b:

$$Fol(b_2) \cup Fol(b_6) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \cup \{c_7\} = \{a_1, b_2, c_3, a_4, b_6, c_7, c_8, \dashv\} \equiv q_3$$

Con el terminal c:

$$Fol(c_3) \cup Fol(c_8) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \cup \{c_8, \dashv\} = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \equiv q_1$$

A partir del estado $q_2 \equiv \{a_1, b_2, c_3, a_4, b_5, b_6, c_8, \dashv\}$:
Con el terminal a:

$$Fol(a_1) \cup Fol(a_4) = q_2$$

Con el terminal b:

$$Fol(b_2) \cup Fol(b5) \cup Fol(b_6) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \cup \{a_4, b_6, c_8, \dashv\} \cup \{c_7\} = \{a_1, b_2, c_3, a_4, b_6, c_7, c_8, \dashv\} \equiv q_3$$

Con el terminal c:

$$Fol(c_3) \cup Fol(c_8) = q_1$$

A partir de estado $q_3 \equiv \{a_1, b_2, c_3, a_4, b_6, c_7, c_8, \dashv\}$
Con el terminal a:

$$Fol(a_1) \cup Fol(a_4) = q_2$$

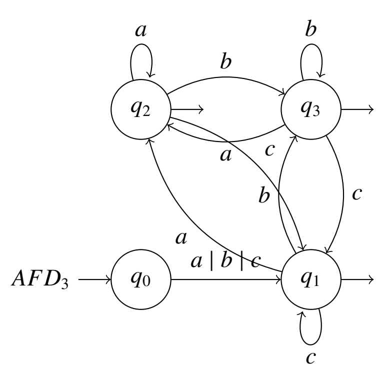
Con el terminal b:

$$Fol(b_2) \cup Fol(b_6) = q_3$$

Con el terminal c:

$$Fol(c_3) \cup Fol(c_7) \cup (c_8) = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \cup \{a_4, b_6, c_8, \dashv\} \cup \{c_8, \dashv\} = \{a_1, b_2, c_3, a_4, b_6, c_8, \dashv\} \equiv q_1$$

El autómata final queda así:



ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **6 de abril de 2018**

Nombre: ______________________________
Código: ______________________________

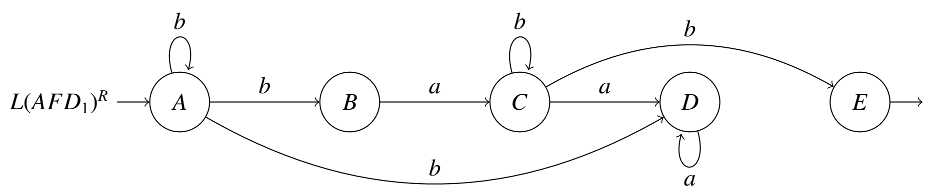
1. A minimal crab .... 40 puntos

La siguiente gramática independiente de contexto por la *izquierda*, produce un autómata finito determinista AFD_1. Encuentre el autómata minimo $\widehat{AFD}_1$ de forma tal que $L(AFD_1) \equiv L(\widehat{AFD}_1)$.

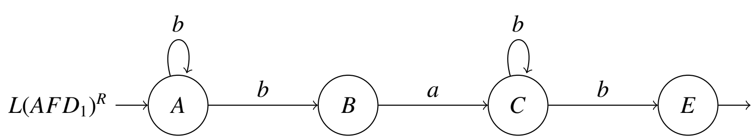
$$
\begin{array}{lcl}
A & \rightarrow & Ab \mid Db \mid Bb \\
B & \rightarrow & Ca \\
C & \rightarrow & Cb \mid Eb \mid Da \\
D & \rightarrow & Da \\
E & \rightarrow & \epsilon
\end{array}
$$

Solución:

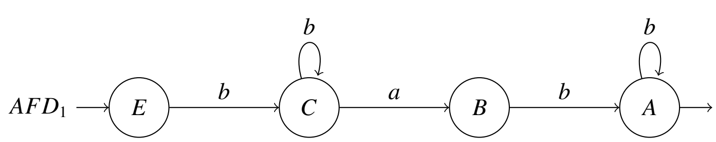
En primer lugar construimos el autómata que reconoce el lenguaje invertido: $L(AFD_1)^R$



El automáta invertido no se encuentra limpio. Al limpiarlo obtenemos:



Ahora invertimos el autómata:



Encontramos el autómata mínimo:

Realizamos la minimización haciendo la matriz:

C			
B			
A			
	E	C	B

Todo lo que es final A con lo que no lo es es distinguible:

C			
B			
A	X	X	X
	E	C	B

Ahora el estado C y E:

$$\begin{aligned}\delta(E,a) &= \perp \\ \delta(C,a) &= B\end{aligned} \qquad \begin{aligned}\delta(E,b) &= C \\ \delta(C,b) &= C\end{aligned}$$

Son distiguibles. Por lo tanto:

C	X		
B			
A	X	X	X
	E	C	B

Ahora, comparamos B y E

$$\begin{aligned}\delta(E,a) &= \perp \\ \delta(B,a) &= \perp\end{aligned} \qquad \begin{aligned}\delta(E,b) &= C \\ \delta(B,b) &= A\end{aligned}$$

A y C son distiguibles por lo tanto: E y C son distingibles:

C	X		
B	X		
A	X	X	X
	E	C	B

Ahora comparamos: B y C:

$$\begin{aligned}\delta(B,a) &= \perp \\ \delta(C,a) &= B\end{aligned} \qquad \begin{aligned}\delta(B,b) &= A \\ \delta(C,b) &= B\end{aligned}$$

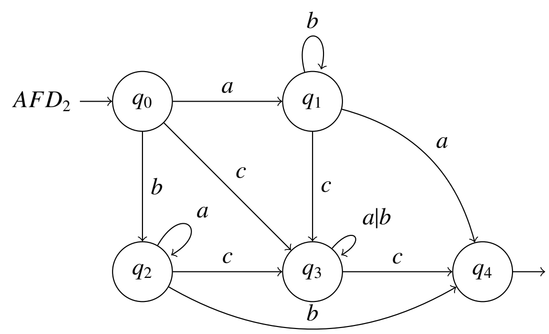
B y A son distiguibles. Por lo tanto B y C lo son:

C	X		
B	X	X	
A	X	X	X
	E	C	B

El autómata ya es mínimo.

2. BMC .... 30 puntos

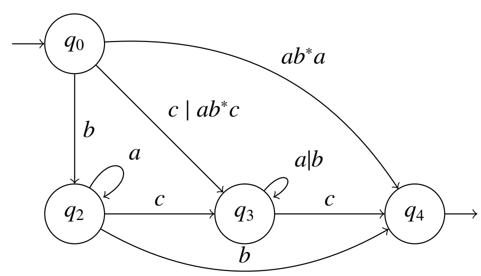
Utilizando el algoritmo de BMC convierta el autómata AFD_2 que se encuentra en la figura siguiente en su correspodiente expresión regular e_2 de forma que se cumpla la siguiente condición $L(AFD_2) \equiv L(e_2)$



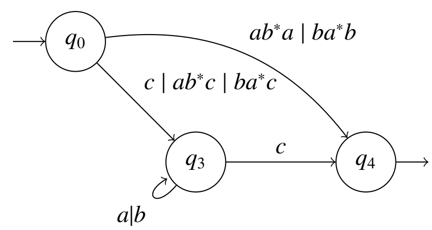
Solución:

El estado inicial q_0 es único y no tiene arcos que lleguen a él. El estado final q_4 es único y no tiene estados que salgan de él. Por lo tanto no hay que añadir ningún estado inicial y final.

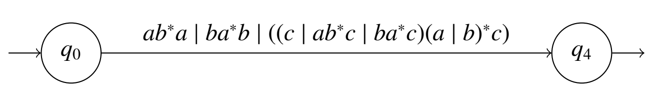
Vamos a eliminar el estado q_1. Hay dos caminos $q_0 \rightarrow q_1 \rightarrow q_4$ y $q_0 \rightarrow q_1 \rightarrow q_3$.



Si se elimina q_2 hay dos caminos $q_0 \rightarrow q_2 \rightarrow q_3$ y $q_0 \rightarrow q_2 \rightarrow q_4$.



Eliminamos q_3.



3. GMY constructs accessible sets .... 30 puntos

Construir el autómata finito determinista AFD_3 de la expresión regular $e_3 = a^*(ab|ba)^+a^*$. Tal que: $L(AFD_3) \equiv L(e_3)$ y el automáta es determinista, utilizando el método *GMY*.

Solución:

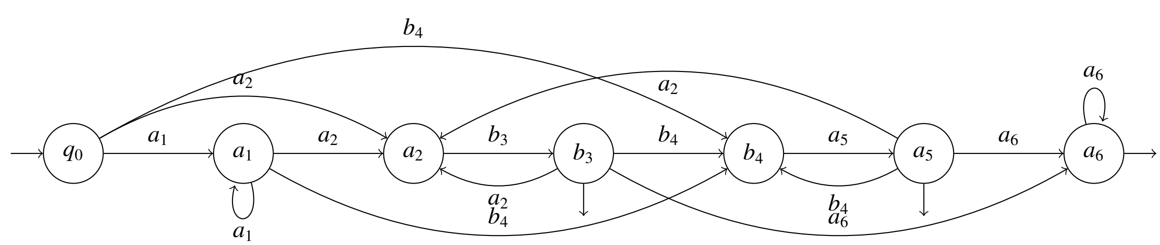
Enumeramos la expresión e_3:

$$e'_3 = a_1^*(a_2b_3 \mid b_4a_5)^+a_6^*$$

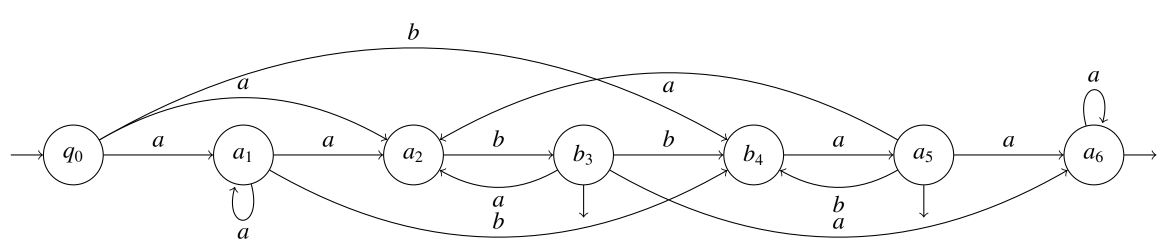
Calculamos los conjuntos:

$$\begin{aligned}
Ini(e'_3 \dashv) &= \{a_1, a_2, b_4\} \\
Fin(e'_3) &= \{b_3, a_5, a_6\} \\
Dig(e'_3) &= \{a_1a_1, a_1a_2, a_1b_4, \\
&\quad a_2b_3, \\
&\quad b_3a_2, b_3b_4, b_3a_6, \\
&\quad b_4a_5, \\
&\quad a_5a_2, a_5b_4, a_5a_6, \\
&\quad a_6a_6\}
\end{aligned}$$

Ahora construimos el autómata utilizando el método GMY:



Eliminamos los números en los arcos:



Ahora construimos el conjunto potencia:

Desde el estado $[q_0]$.
$\delta([q_0], a) = \delta(q_0, a) = \{a_1, a_2\} = [a_1, a_2]$.
$\delta([q_0], b) = \delta(q_0, b) = \{b_4\} = [b_4]$.

Desde el estado $[a_1, a_2]$.
$\delta(a_1, a) \cup \delta(a_2, a) = \{a_1, a_2\} \cup \emptyset = [a_1, a_2]$.
$\delta(a_1, a) \cup \delta(a_2, b) = \{b_4\} \cup \{b_3\} = [b_3, b_4]$.

Desde el estado $[b_4]$.
$\delta(b_4, a) = \{a_5\} = [a_5]$.
$\delta(b_4, b) = \emptyset$.

Desde el estado $[b_3, b_4]$.
$\delta(b_3, a) \cup \delta(b_4, a) = \{a_2, a_6\} \cup \{a_5\} = [a_2, a_5, a_6]$.
$\delta(b_3, b) \cup \delta(b_4, b) = \{b_4\} \cup \emptyset$.

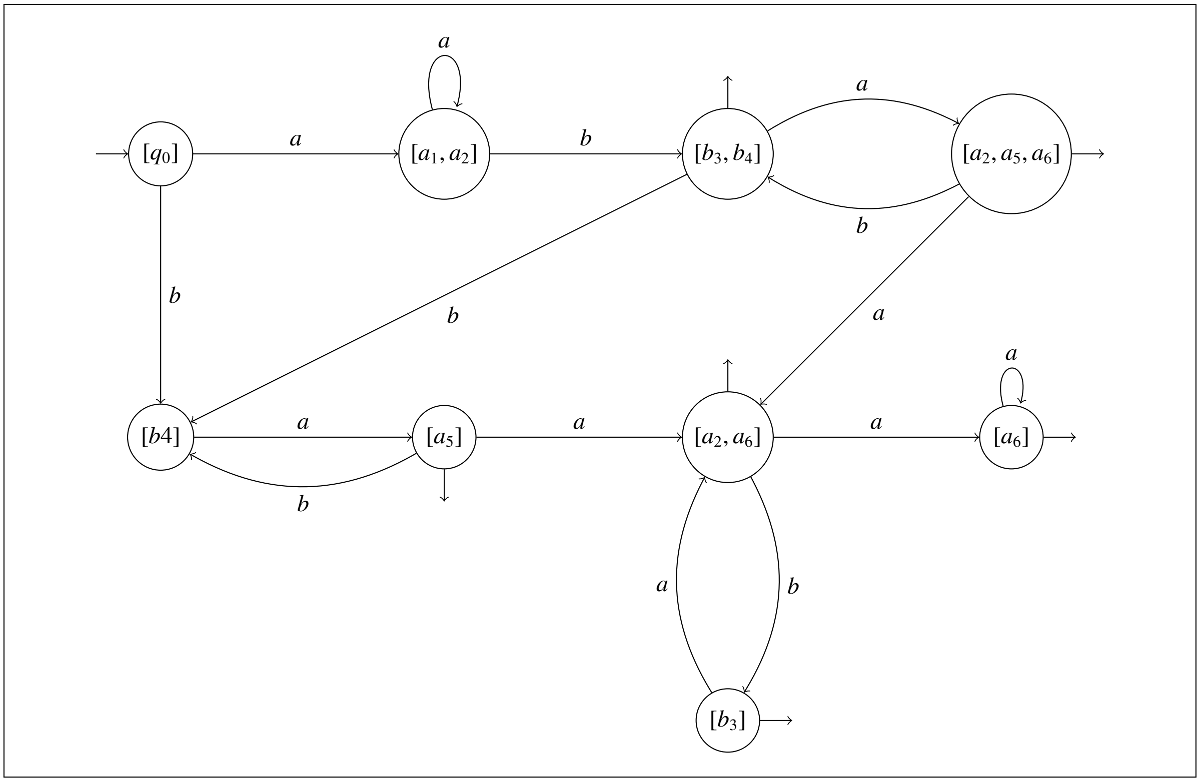
Desde el estado $[a_5]$.
$\delta(a_5, a) = \{a_2, a_6\} = [a_2, a_6]$.
$\delta(a_5, b) = \{b_4\} = [b_4]$.

Desde el estado $[a_2, a_5, a_6]$.
$\delta(a_2, a) \cup \delta(a_5, a) \cup \delta(a_6, a) = \emptyset \cup \{a_2, a_6\} \cup \{a_6\} = [a_2, a_6]$.
$\delta(a_2, b) \cup \delta(a_5, b) \cup \delta(a_6, b) = \{b_3\} \cup \{b_4\} \cup \emptyset = [b_3, b_4]$.

Desde el estado $[a_2, a_6]$.
$\delta(a_2, a) \cup \delta(a_6, a) = \emptyset \cup \{a_6\} = [a_6]$.
$\delta(a_2, b) \cup \delta(a_6, b) = \{b_3\} \cup \emptyset = [b_3]$.

Desde el estado $[a_6]$.
$\delta(a_6, a) = \{a_6\} = [a_6]$.
$\delta(a_6, b) = \emptyset$.

Desde el estado $[b_3]$.
$\delta(b_3, a) = \{a_2, a_6\} = [a_2, a_6]$.
$\delta(b_3, b) = \{b_4\} = [b_4]$.


[q0]
[a1, a2]
[b3, b4]
[a2, a5, a6]
[b4]
[a5]
[a2, a6]
[a6]
[b3]
a
b

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **17 de mayo de 2018**

Nombre: ______________________________
Código: ______________________________

1. Autómata de pila .... 20 puntos

Implemente un autómata de pila determinista o no determinista que reconoce el siguiente lenguaje L_1

$$L_1 = \{a^n x \mid n \geq 0,\ x \in \{a, b\}^* \wedge |x| \leq n\}$$

2. Condición $LR(1)$ .... 20 puntos

La siguiente gramática no cumple con la condición $LR(1)$ de la gramática G_2, construya un piloto parcial que muestre que uno estado no cumple con dicha condición.

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & AC \mid BD \\ A & \rightarrow & a \\ B & \rightarrow & ab \\ C & \rightarrow & c \mid bC \\ D & \rightarrow & d \end{array}$$

3. Condición $LR(1)$ .... 60 puntos

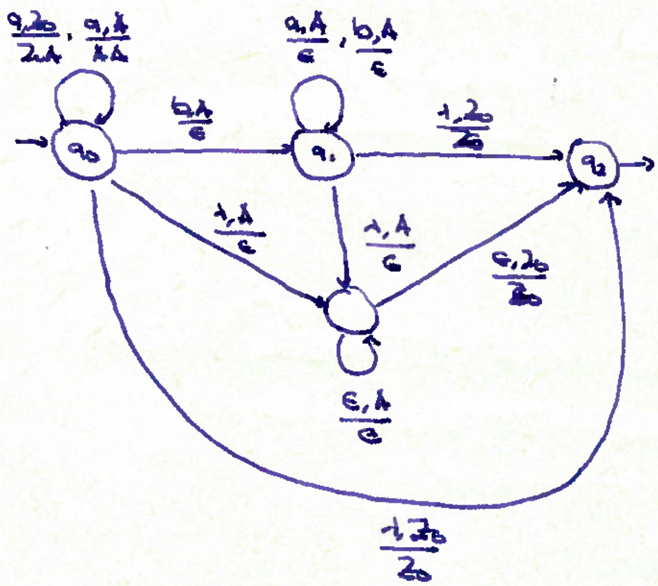
Verificar la condición $LR(1)$ (45 Puntos) de la gramática G_3 construyendo el piloto correspondiente y verificando que la la cadena $a0b$ (15 Puntos) pertenece a la gramática.

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & aAb \mid aBc \\ A & \rightarrow & 0 \\ B & \rightarrow & 1 \end{array}$$

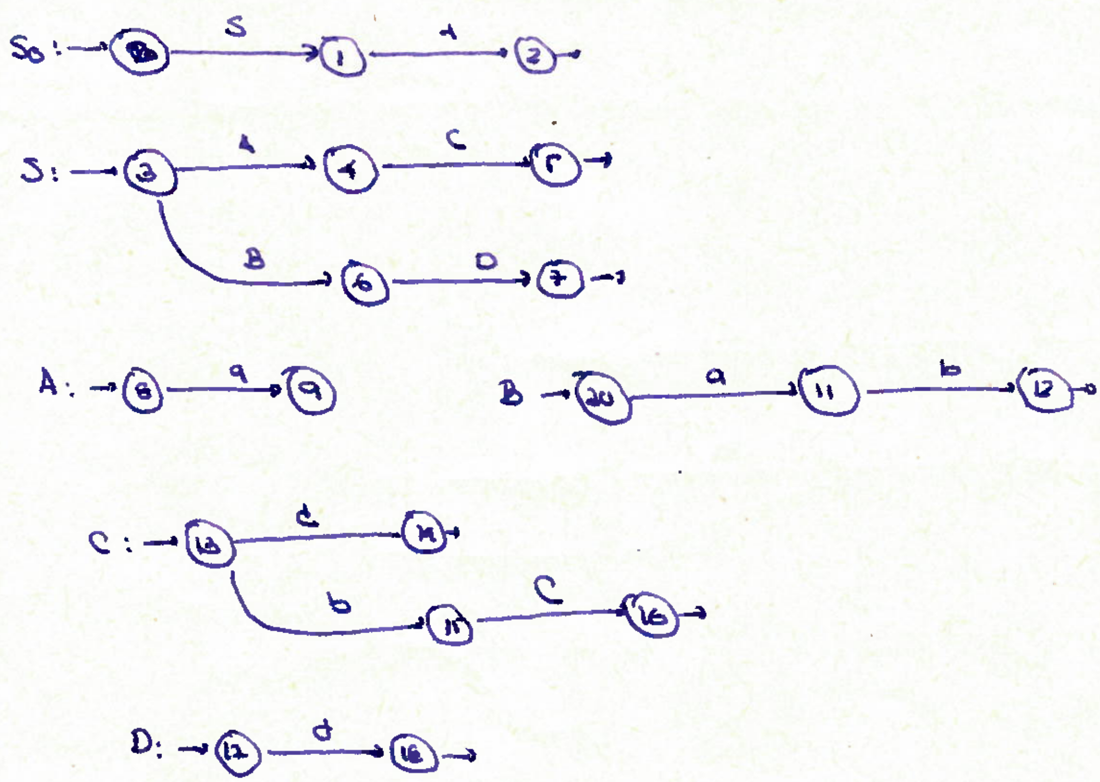
310230
2016-1-031
Parcial C3

1. Autómata de pila.

Una posible versión:



2. Condición LR(1)



$closure_1(\{<0,\dashv>\}) = \{<0,\dashv>, <3,\dashv>, <8,\{c,b\}>, <10,d>\}$

I_0

0	$S_0 \to \bullet S\dashv$	$\dashv$
3	$S \to \bullet AC$	$\dashv$
	$S \to \bullet BD$	
8	$A \to \bullet a$	$\{c,b\}$
10	$B \to \bullet ab$	d

$I_0 \xrightarrow{a} I_1$

I_1

9	$A \to a\bullet$	$\{c,b\}$
11	$B \to a\bullet b$	d

$closure_1(\delta(I_0,a)) = closure_1(<\delta(8,a),\{c,b\}>) \cup closure_1(<\delta(10,a),d>)$

$= closure_1(<9,\{c,b\}>) \cup closure_1(<11,d>)$

$= \{<9,\{c,b\}>\} \cup \{<11,d>\}$

$= \{<9,\{c,b\}>, <11,d>\}$

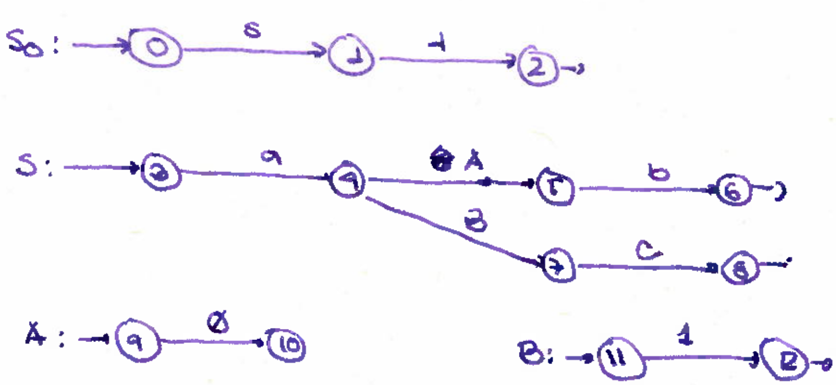
$<9,\{c,b\}>$ es un candidato a reducción

$<11,d>$ es un candidato a desplazamiento

9, $\{c,b\}$ no son disjuntos por lo tanto no cumple con la condición

11, b

3.



$\Sigma = \{a, b, \emptyset, d\} \cup \{\dashv\}$ $V = \{S, A, B\}$

$I_0 = closure(\langle 0, \dashv\rangle) = \{\langle 0,\dashv\rangle, \langle 3,\dashv\rangle\}$

I_0

$X = a$

$closure(\delta(I_0, a)) = closure(\langle \delta(3,a), \dashv\rangle) = closure(\langle 4,\dashv\rangle) = \{\langle 4,\dashv\rangle, \langle 9,b\rangle, \langle 11,c\rangle\} = I_1$

$X = S$

$closure(\delta(I_0, S)) = closure(\langle \delta(0,S), \dashv\rangle) = closure(\langle 1,\dashv\rangle) = \{\langle 1,\dashv\rangle\} = I_2$

I_1

$X = \emptyset$

$closure(\delta(I_1, \emptyset)) = closure(\langle \delta(9,\emptyset), b\rangle) = closure(\langle 10,b\rangle) = \{\langle 10,b\rangle\} = I_3$

$X = d$

$closure(\delta(I_1, d)) = closure(\langle \delta(11,d), c\rangle) = closure(\langle 12,c\rangle) = \{\langle 12,c\rangle\} = I_4$

$X = A$

$closure(\delta(I_1, A)) = closure(\langle \delta(4,A), \dashv\rangle) = closure(\langle 5,\dashv\rangle) = \{\langle 5,\dashv\rangle\} = I_5$

$X = B$

$closure(\delta(I_1, B)) = closure(\langle \delta(4,B), \dashv\rangle) = closure(\langle 7,\dashv\rangle) = \{\langle 7,\dashv\rangle\} = I_6$

I_2

$X = \dashv$

$closure(\delta(I_2, \dashv)) = closure(\langle \delta(1,\dashv), \dashv\rangle) = closure(\langle 2,\dashv\rangle) = \{\langle 2,\dashv\rangle\} = I_7$

I_5

$X = b$

$closure(\delta(I_5, b)) = closure(\langle \delta(5,b), \dashv\rangle) = closure(\langle 6,\dashv\rangle) = \{\langle 6,\dashv\rangle\} = I_8$

I_6

$X = c$

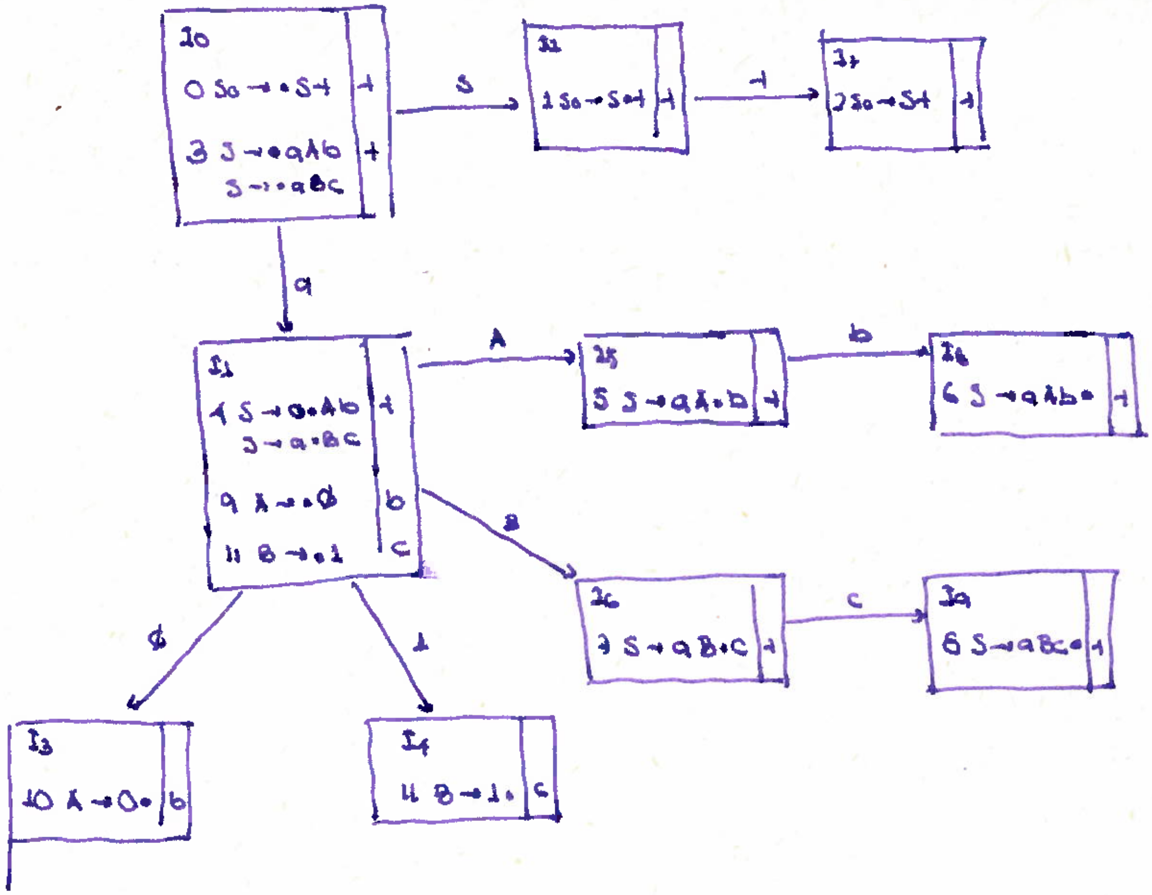
$closure(\delta(I_6, c)) = closure(\langle \delta(7,c), \dashv\rangle) = closure(\langle 8,\dashv\rangle) = \{\langle 8,\dashv\rangle\} = I_9$

Conflicto Reducción-Desplazamiento

Algún estado hace reducción y desplazamiento.

Conflicto Reducción-Reducción

Los estados de reducción son únicos.



Verificando que aØb pertenece a la gramática.

Pila	Cadena	Movimiento
I_0	$a Ø b \dashv$	Desp.
$I_0 a I_1$	$Ø b \dashv$	Desp
$I_0 a I_1 Ø I_3$	$b \dashv$	Red
$I_0 a I_1 A I_5$	$b \dashv$	Desp
$I_0 a I_1 A I_5 b I_6$	$\dashv$	Red
$I_0 S I_1$	$\dashv$	Desp
$I_0 S I_1 \dashv I_7$	ϵ	Red
$I_0 S_0$		Aceptar y parar.

ST0270 Lenguajes formales y compiladores - 032 — **Universidad EAFIT**
Tercer parcial — **DIS**
Juan Francisco Cardona McCormick — **18 de mayo de 2018**

Nombre: ______________________________

Código: ______________________________

1. Autómata de pila . . . . 20 puntos

Implemente un autómata de pila determinista o no determinista que reconoce el siguiente lenguaje L_1

$$L_1 = \{ww^R \mid w \in \{0,1\}^*\}$$

2. Condición $LR(1)$ . . . . 20 puntos

Verificar la condición $LR(1)$ de la gramática G_2 construyendo el piloto correspondiente:

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & Abb \mid Bbc \\ A & \rightarrow & aA \mid a \\ B & \rightarrow & aB \mid a \end{array}$$

3. Condición $LR(1)$ . . . . 60 puntos

Verificar la condición $LR(1)$ de la gramática G_3 construyendo el piloto correspondiente:

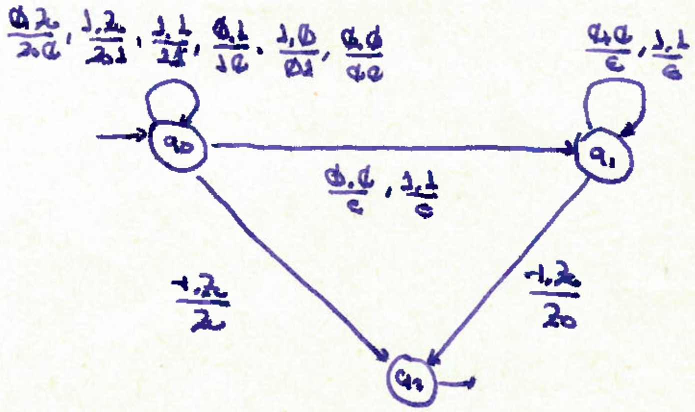
$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & A \mid B \\ A & \rightarrow & aAb \mid 1 \\ B & \rightarrow & aBbb \mid 0 \end{array}$$

ST0270

2018-1-032

Parcial-03 (solución)

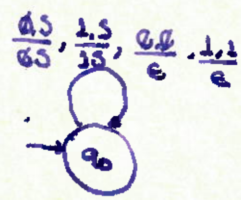
1. Autómata de pila



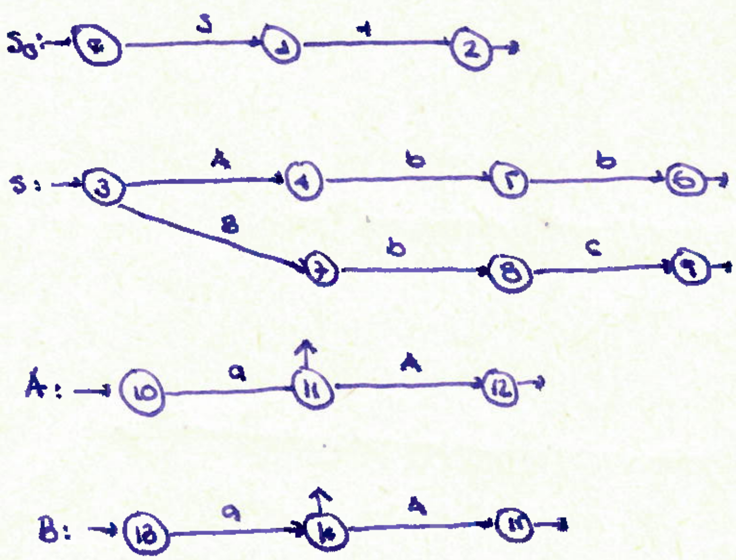
Una posible versión

* partir de la gramática

$S \to 0S0$
$\mid 1S1$
$\mid \epsilon$



2. Condición LR(1)



$I_0 = \text{closure}_1(\langle 0,\dashv\rangle) = \{\langle 0,\dashv\rangle, \langle 3,\dashv\rangle, \langle 10,b\rangle, \langle 13,b\rangle\}$ No hay ningún conflicto

I_0

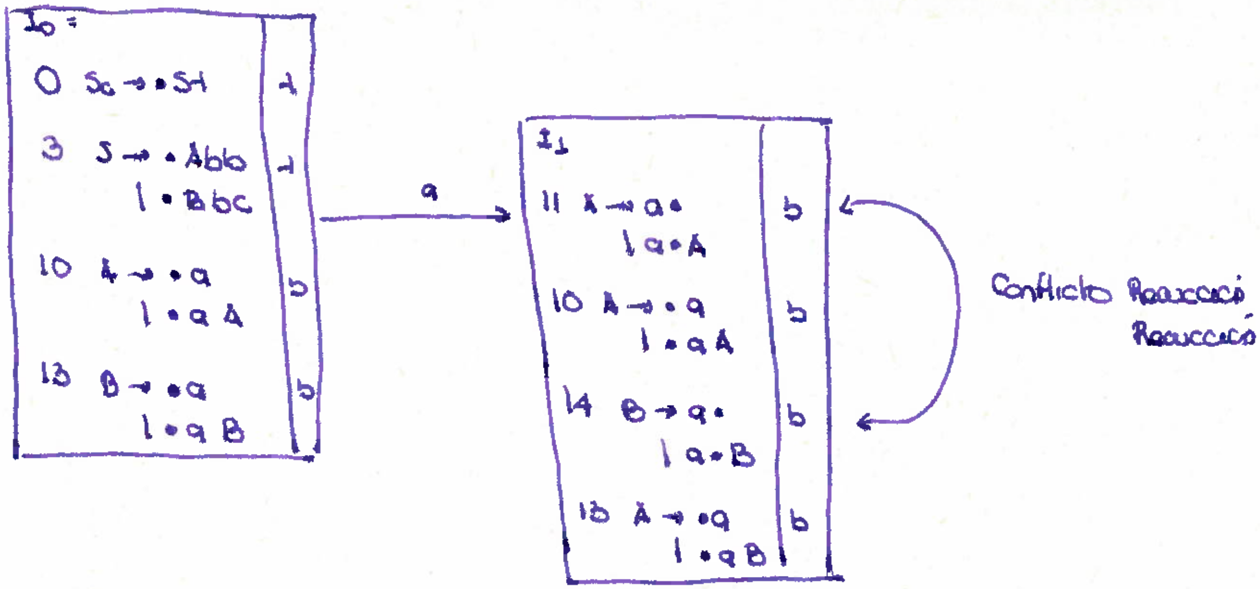
$X: a$

$\text{closure}_1(\delta(I_0,a)) = \text{closure}_1(\langle\delta(10,a),b\rangle) \cup \text{closure}_1(\langle\delta(13,a),b\rangle)$

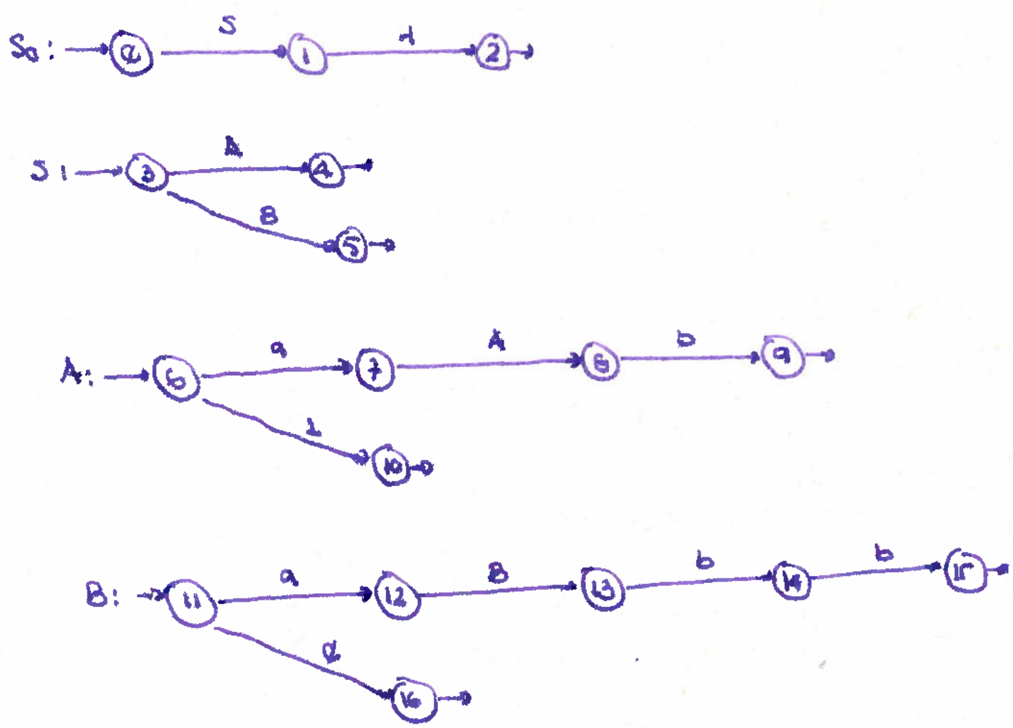
$= \text{closure}_1(\langle 11,b\rangle) \cup \text{closure}(\langle 14,b\rangle)$

$= \{\langle 11,b\rangle, \langle 10,b\rangle\} \cup \{\langle 14,b\rangle, \langle 13,b\rangle\}$

$= \{\langle 11,b\rangle, \langle 10,b\rangle, \langle 14,b\rangle, \langle 13,b\rangle\}$



3. Condición LR(1)



$\Sigma = \{a, b, \perp, \emptyset\} \cup \{\dashv\}$ $\quad V = \{S, A, B\}$

$I_0 = closure_1(\langle 0, \dashv\rangle) = \{\langle 0,\dashv\rangle, \langle 3,\dashv\rangle, \langle 6,\dashv\rangle, \langle 11,\dashv\rangle\}$

I_0

$X = a$

$closure_1(\delta(I_0, a)) = closure_1(\langle\delta(6,a),\dashv\rangle) \cup closure_1(\langle\delta(11,a),\dashv\rangle)$

$= closure_1(\langle 7,\dashv\rangle) \cup closure_1(\langle 12,\dashv\rangle)$

$= \{\langle 7,\dashv\rangle, \langle 6,b\rangle\} \cup \{\langle 12,\dashv\rangle, \langle 11,b\rangle\} = \{\langle 7,\dashv\rangle, \langle 6,b\rangle, \langle 12,\dashv\rangle, \langle 11,b\rangle\} = I_1$

$X = \perp$

$closure_1(\delta(I_0,\perp)) = closure_1(\langle\delta(6,\perp),\dashv\rangle) = closure_1(\langle 10,\dashv\rangle) = \{\langle 10,\dashv\rangle\} = I_2$

$X = \emptyset$

$closure_1(\delta(I_0,\emptyset)) = closure_1(\langle\delta(11,\emptyset),\dashv\rangle) = closure_1(\langle 16,\dashv\rangle) = \{\langle 16,\dashv\rangle\} = I_3$

$X = S$

$closure_1(\delta(I_0,S)) = closure_1(\langle\delta(0,S),\dashv\rangle) = closure_1(\langle 1,\dashv\rangle) = \{\langle 1,\dashv\rangle\} = I_4$

$X = A$

$closure_1(\delta(I_0,A)) = closure_1(\langle\delta(3,A),\dashv\rangle) = closure_1(\langle 4,\dashv\rangle) = \{\langle 4,\dashv\rangle\} = I_5$

$X = B$

$closure_1(\delta(I_0,B)) = closure_1(\langle\delta(2,B),\dashv\rangle) = closure_1(\langle 5,\dashv\rangle) = \{\langle 5,\dashv\rangle\} = I_6$

I_1

$X = a$

$closure_1(\delta(I_1,a)) = closure_1(\langle\delta(6,a),b\rangle) \cup closure_1(\langle\delta(11,a),b\rangle) = closure_1(\langle 7,b\rangle) \cup closure_1(\langle 12,b\rangle)$

$= \{\langle 7,b\rangle, \langle 6,b\rangle, \langle 12,b\rangle, \langle 11,b\rangle\} = \{\langle 7,b\rangle, \langle 6,b\rangle, \langle 12,b\rangle, \langle 11,b\rangle\} = I_7$

$X = \perp$

$closure_1(\delta(I_1,\perp)) = closure_1(\langle\delta(6,\perp),b\rangle) = closure_1(\langle 10,b\rangle) = \{\langle 10,b\rangle\} = I_8$

$X = \emptyset$

$closure_1(\delta(I_1,\emptyset)) = closure_1(\langle\delta(11,\emptyset),b\rangle) = closure_1(\langle 16,b\rangle) = \{\langle 16,b\rangle\} = I_9$

$X = A$

$closure_1(\delta(I_1,A)) = closure_1(\langle\delta(7,A),\dashv\rangle) = closure_1(\langle 8,\dashv\rangle) = \{\langle 8,\dashv\rangle\} = I_{10}$

$X = B$

$closure_1(\delta(I_1,B)) = closure_1(\langle\delta(12,B),\dashv\rangle) = closure_1(\langle 13,\dashv\rangle) = \{\langle 13,\dashv\rangle\} = I_{11}$

I_4

$X = \dashv$

$closure_1(\delta(I_4,\dashv)) = closure_1(\langle\delta(1,\dashv),\dashv\rangle) = closure_1(\langle 2,\dashv\rangle) = I_{12}$

I_7

$X = a$

$closure_1(\delta(I_7,a)) = closure_1(\langle\delta(6,a),b\rangle) \cup closure_1(\langle\delta(11,a),b\rangle) = closure_1(\langle 7,b\rangle) \cup closure_1(\langle 12,b\rangle)$

$= I_7$

$X = \perp$

$closure(\delta(I_7,\perp)) = closure(\langle\delta(6,b),b\rangle) = closure_1(\langle 10,b\rangle) = I_8$

$X = \emptyset$

$closure_1(\delta(I_7,\emptyset)) = closure_1(\langle\delta(11,b),b\rangle) = closure_1(\langle 16,b\rangle) = I_9$

I_7

$X=A$

$closure_1(\delta(I_7,A)) = closure_1(<\delta(7,A),b>) = closure_1(<8,b>) = \{<8,b>\} = I_{12}$

$X=B$

$closure_1(\delta(I_7,B)) = closure_1(<\delta(12,B),b>) = closure_1(<13,b>) = \{<13,b>\} = I_{13}$

I_{10}

$X=b$

$closure_1(\delta(I_{10},b)) = closure_1(<\delta(8,b),+>) = closure_1(<9,+>) = \{<9,+>\} = I_{14}$

I_{11}

$X=b$

$closure_1(\delta(I_{11},b)) = closure_1(<\delta(13,b),+>) = closure_1(<14,+>) = \{<14,+>\} = I_{15}$

I_{12}

$X=b$

$closure_1(\delta(I_{12},b)) = closure_1(<\delta(8,b),b>) = closure_1(<9,b>) = \{<9,b>\} = I_{16}$

I_{13}

$X=b$

$closure_1(\delta(I_{13},b)) = closure_1(<\delta(13,b),b>) = closure_1(<14,b>) = \{<14,b>\} = I_{17}$

I_{15}

$X=b$

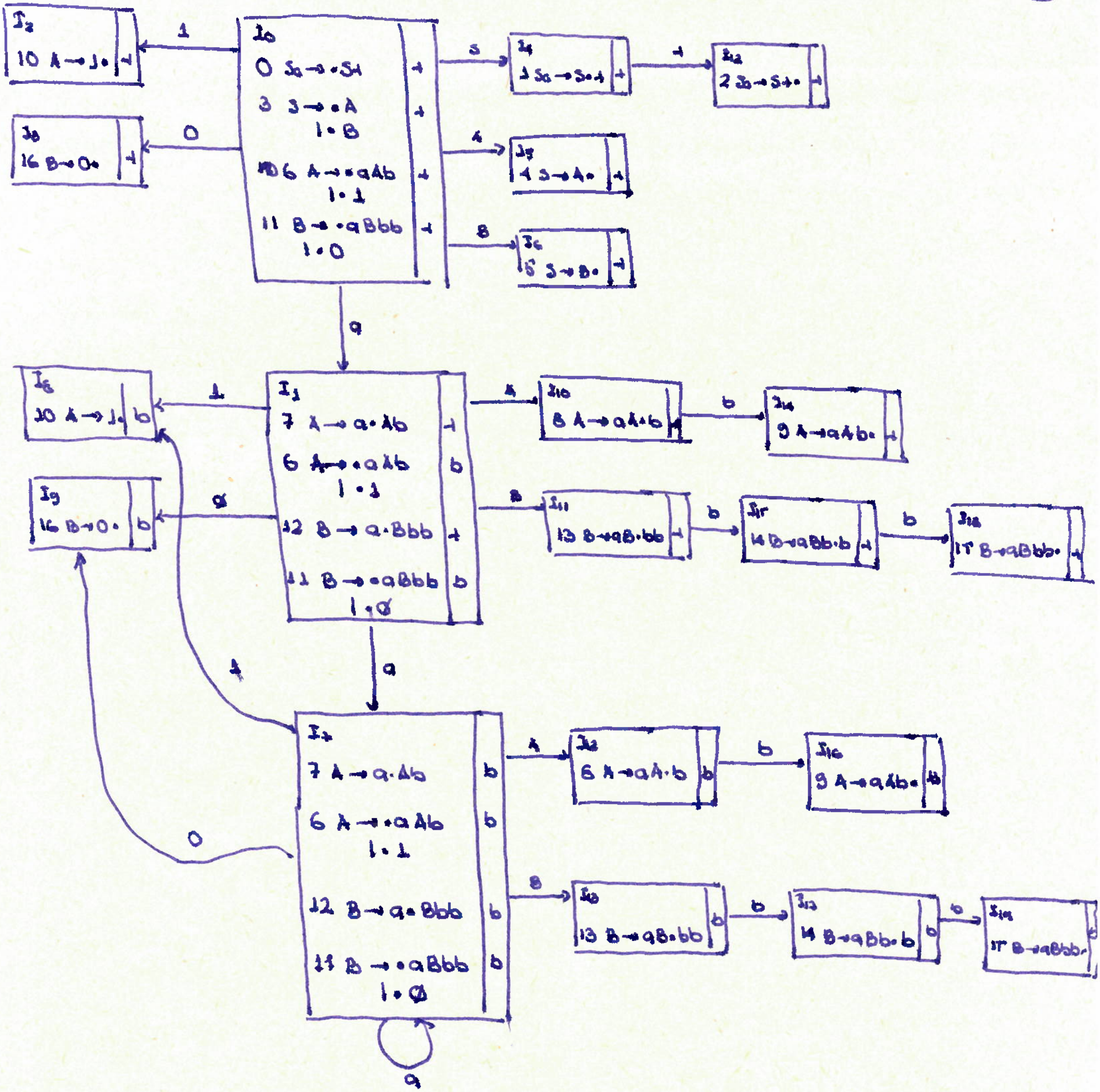
$closure_1(\delta(I_{15},b)) = closure_1(<\delta(14,b),+>) = closure_1(<15,\overset{+}{b}>) = \{<15,\overset{+}{b}>\} = I_{18}$

I_{17}

$X=b$

$closure_1(\delta(I_{17},b)) = closure_1(<\delta(14,b),b>) = closure_1(<15,b>) = \{<15,b>\} = I_{19}$

* No hay conflicto desplazamiento y reducción, puesto que no hay macro-estados con ambos.
* No hay conflicto reducción ~~despla~~ - reducción, puesto que no hay macro-estado con más de un ~~[illegible]~~ candidato a reducción.


I0
0 S0 → •S⊣ ⊣
3 S → •A ⊣
| •B
6 A → •aAb ⊣
| •1
11 B → •aBbb ⊣
| •0
I2
10 A → 1• ⊣
I3
16 B → 0• ⊣
I4
1 S0 → S•⊣ ⊣
I12
2 S0 → S⊣• ⊣
I5
4 S → A• ⊣
I6
5 S → B• ⊣
I1
7 A → a•Ab ⊣
6 A → •aAb b
| •1
12 B → a•Bbb ⊣
11 B → •aBbb b
| •0
I8
10 A → 1• b
I9
16 B → 0• b
I10
8 A → aA•b ⊣
I14
9 A → aAb• ⊣
I11
13 B → aB•bb ⊣
I15
14 B → aBb•b ⊣
I18
15 B → aBbb• ⊣
I7
7 A → a•Ab b
6 A → •aAb b
| •1
12 B → a•Bbb b
11 B → •aBbb b
| •0
I13
8 A → aA•b b
I16
9 A → aAb• b
I13
13 B → aB•bb b
I17
14 B → aBb•b b
I19
15 B → aBbb• b
1
0
S
⊣
A
B
a
b

Nombre: ______________________________

Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(20 %) 1. **Expresiones regulares ambiguas**

La siguiente expresión regular R_1 es ambigua:

$$R_1 = (jk)^*(j \mid k)^*$$

Demuestrélo formalmente con un caso donde se presenta la ambiguedad:

Solución:

Se renombra y se obtiene una nueva expresión regular R_1:

$$R_1' = (j_1k_2)^*(j_3 \mid k_4)^*$$

Escogemos generar una cadena:

$$\begin{aligned} (j_1k_2)^*(j_3 \mid k_4)^* &\Rightarrow (j_1k_2)^* \\ &\Rightarrow (j_1k_2)^1 \\ &\Rightarrow j_1k_2 \end{aligned}$$

Otra posible generación:

$$\begin{aligned} (j_1k_2)^*(j_3 \mid k_4)^* &\Rightarrow (j_3 \mid k_4)^* \\ &\Rightarrow (j_3 \mid k_4)^2 \\ &\Rightarrow (j_3 \mid k_4)(j_3 \mid k_4) \\ &\Rightarrow j_3(j_3 \mid k_4) \\ &\Rightarrow j_3k_4 \end{aligned}$$

Al borrar los carácteres de la primera derivación y la segunda se obtiene la misma cadena jk.

Por lo tanto tenemos una cadena ambigua por lo tanto la expresión regular R_1 es ambigua.

(30 %) 2. **Ambigüedad en gramáticas independientes de contexto**
La siguiente gramática independiente de contexto G_2 es una gramática ambigua:

$$G_2 = (V = \{S, A, B\}, \Sigma = \{a, b\}, P = \{S \rightarrow AB, A \rightarrow abA \mid \epsilon, B \rightarrow aB \mid bB \mid \epsilon\}, S)$$

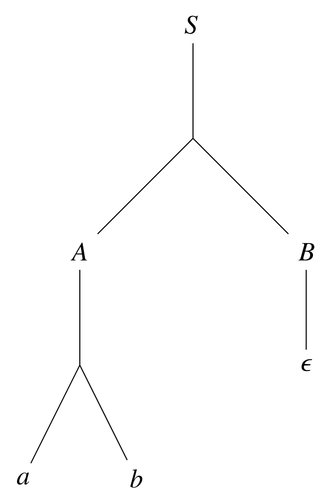
a) Muestre que G_2 es ambigua.

b) Defina una nueva gramática G'_2 tal que $L(G_2) \equiv L(G'_2)$ y $L(G'_2)$ no es ambiguo.

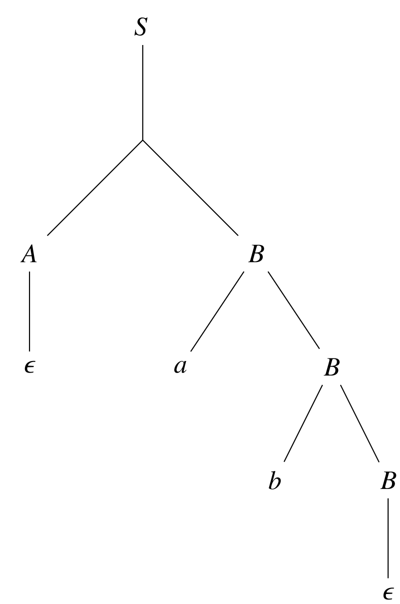
Solución:

a) La cadena ab se puede generar por la siguiente derivación:

$$\begin{aligned} S &\Rightarrow AB \\ &\Rightarrow abB \\ &\Rightarrow ab\epsilon \equiv ab \end{aligned}$$



$$\begin{aligned} S &\Rightarrow AB \\ &\Rightarrow \epsilon B \equiv B \\ &\Rightarrow aB \\ &\Rightarrow abB \\ &\Rightarrow ab\epsilon \equiv ab \end{aligned}$$



b) Se puede observar que la segunda producción obtiene el mismo lenguaje esperado.

$$G' = (V = \{S, B\}, \Sigma = \{a, b\}, P = \{S \rightarrow B, B \rightarrow aB \mid bB \mid \epsilon\}, S)$$

(30 %) 3. **Gramáticas en forma normal Chomsky**

La G_3 no se encuentra en forma normal Chomsky, transforme en una nueva gramática G'_3 que $L(G_3) \equiv L(G'_3)$, donde G'_3 tiene forma normal Chomsky.

$$G_3 = (V = \{S, A, B, C\}, \Sigma = \{a, b, c\}, P = \{S \rightarrow ABC, A \rightarrow aAA \mid \epsilon, B \rightarrow bBB \mid \epsilon, C \rightarrow cCC \mid \epsilon\}, S)$$

Solución:

Calculamos el conjunto $Null = \{A, B, C, D\}$:

Y eliminamos los nulos las producciones queda:

$$\begin{array}{rcl} S & \rightarrow & ABC \mid \epsilon \mid AB \mid BC \mid AC \mid A \mid B \mid C \\ A & \rightarrow & aAA \mid aA \mid a \\ B & \rightarrow & bBB \mid bB \mid b \\ C & \rightarrow & bCC \mid cC \mid c \end{array}$$

Todavía no esta en forma normal Chomsky:

$$\begin{array}{rcl} S & \rightarrow & A\langle BC\rangle \mid \epsilon \mid AB \mid BC \mid AC \mid A\langle\epsilon\rangle \mid B\langle\epsilon\rangle \mid C\langle\epsilon\rangle \\ \langle\epsilon\rangle & \rightarrow & \epsilon \\ \langle BC\rangle & \rightarrow & BC \\ A & \rightarrow & \langle a\rangle\langle AA\rangle \mid \langle a\rangle A \mid a \\ \langle AA\rangle & \rightarrow & AA \\ B & \rightarrow & \langle b\rangle\langle BB\rangle \mid \langle b\rangle B \mid b \\ \langle BB\rangle & \rightarrow & BB \\ C & \rightarrow & \langle b\rangle\langle CC\rangle \mid \langle c\rangle C \mid c \\ \langle CC\rangle & \rightarrow & CC \\ \langle a\rangle & \rightarrow & a \\ \langle b\rangle & \rightarrow & b \\ \langle c\rangle & \rightarrow & c \end{array}$$

(20 %) 4. **Expresiones regulares**
Encuentre la expresión regular R_4 para el alfabeto $\Sigma_4 = \{i, j\}$ cuyo lenguaje esta definido:

$$L(R_4) \;=\; \{i^n j^m \;\mid\; n \geq 2, m \geq 1\}$$

Solución:
Una posible solución:

$$R_4 = ii^+ j j^*$$

Nombre: ______________________________
Código: ______________________________

Nota: El parcial tiene dos puntos, seleccione uno de los dos puntos y resuélvalo.

1. Working with a little crab . . . . 100 puntos

La siguiente es una gramática lineal por la izquierda GLI_1.

$$
\begin{array}{lcl}
S & \rightarrow & C \mid D \\
C & \rightarrow & Ca \mid Cb \mid B \mid Bc \\
D & \rightarrow & Da \mid Db \mid B \mid Bc \\
B & \rightarrow & Ba \mid Bb \mid Ac \mid A \\
A & \rightarrow & Aa \mid Ab \mid \epsilon
\end{array}
$$

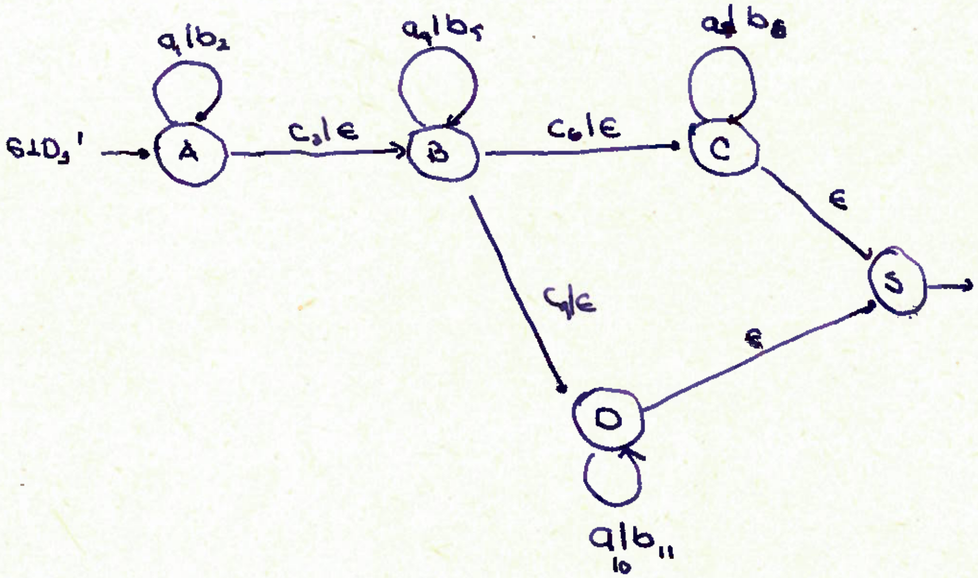
a) (33 puntos) Obtener la gramática lineal por la derecha GLD_1 tal que $L(GLI_1) \equiv L(GLD_1)$.

b) (33 puntos) Obtener un autómata no determinista eliminando los movimientos espontaneos por medio del GMY $N_1^{\neg\epsilon}$, tal que $L(GLD_1) \equiv L(N_1^{\neg\epsilon})$

c) (34 puntos) A partir del autómata $N_1^{\neg\epsilon}$ construir un autómata determinista por medio del cálculo del conjunto de potencia D_1, tal que $L(GLI_1) \equiv L(D_1)$.

Solución:

ST0270 - 2016-2-081-Parcial-02-Punto-01.



$S \to C \mid D$

$C \to Ca \mid Cb \mid B \mid Bc$

$D \to Da \mid Db \mid B \mid Bc$

$B \to Ba \mid Bb \mid Ac \mid A$

$A \to Aa \mid Ab \mid \epsilon$

$S \to C \mid D$

$C \to aC \mid bC \mid B \mid cB$

$D \to aD \mid bD \mid B \mid cB$

$B \to aB \mid bB \mid cA \mid A$

$A \to aA \mid bA \mid \epsilon$

Gramática por la derecha

$A \to aA \mid bA \mid cB \mid B$

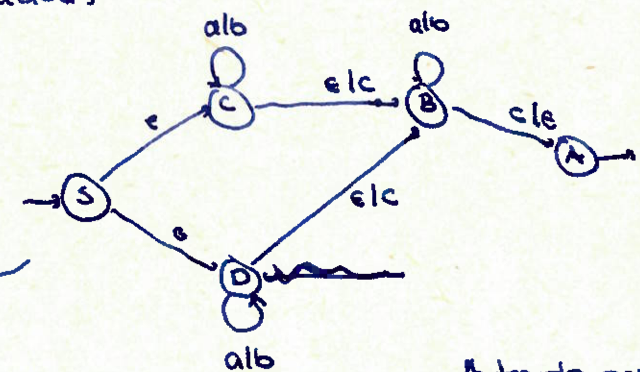
$B \to aB \mid bB \mid cC \mid cD \mid C \mid D$

$C \to aC \mid bC \mid S$

$D \to aD \mid bD \mid S$

$S \to \epsilon$

Cambiando el orden



Automata original.

Reto 1. <u>b.4c</u>

$Ini(G \downarrow D_3') = \{a_1, b_2, c_3, a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}, \cancel{a}\}$

$Fin(G \downarrow D_3') = \{a_1, b_2, c_3, a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}\}$

$Dis(G \downarrow D_3') = \{a_1a_1, a_1b_2, a_1c_3, a_1a_4, a_1b_5, a_1c_6, a_1a_7, a_1b_8, a_1c_9, a_1a_{10}, a_1b_{11},$

$b_2a_1, b_2b_2, b_2c_3, b_2a_4, b_2b_5, b_2c_6, b_2a_7, b_2b_8, b_2c_9, b_2a_{10}, b_2b_{11},$

$\cancel{c_3a_1}, \cancel{c_3b_2}, \cancel{c_3c_3}, c_3a_4, c_3b_5, c_3c_6, c_3a_7, c_3b_8, c_3c_9, c_3a_{10}, c_3b_{11},$

$\cancel{a_4a_1}, \cancel{a_4b_2}, \cancel{a_4c_3}, a_4a_4, a_4b_5, a_4c_6, a_4a_7, a_4b_8, a_4c_9, a_4a_{10}, a_4b_{11},$

$\cancel{b_5a_1}, \cancel{b_5b_2}, \cancel{b_5c_3}, b_5a_4, b_5b_5, b_5c_6, b_5a_7, b_5b_8, b_5c_9, b_5a_{10}, b_5b_{11},$

$c_6a_7, c_6b_8,$

$\cancel{c_9a_{10}, c_9b_{11}}, \quad a_7a_7, a_7b_8,$

$b_8a_7, b_8b_8,$

$c_9a_{10}, c_9b_{11}, \quad a_{10}a_{10}, a_{10}b_{11}, b_{11}a_{10}, b_{11}b_{11}\}$

$Fol(a_1) = \{a_1, b_2, c_3, a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}, \dashv\}$

$Fol(b_2) = \{a_1, b_2, c_3, a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}, \dashv\}$

$Fol(c_3) = \{a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}, \dashv\}$

$Fol(a_4) = \{a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}, \dashv\}$

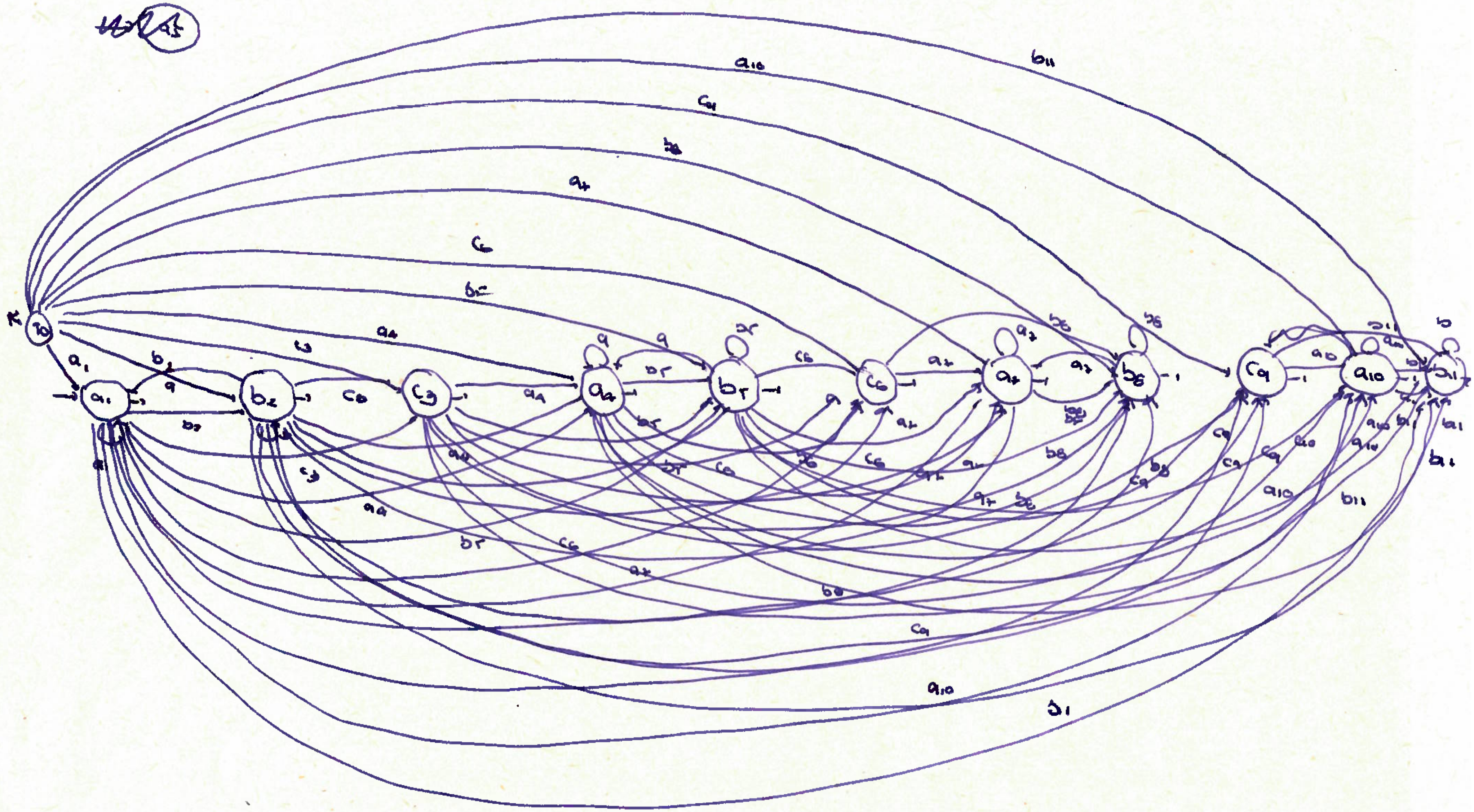
$Fol(b_5) = \{a_4, b_5, c_6, a_7, b_8, c_9, a_{10}, b_{11}, \dashv\}$

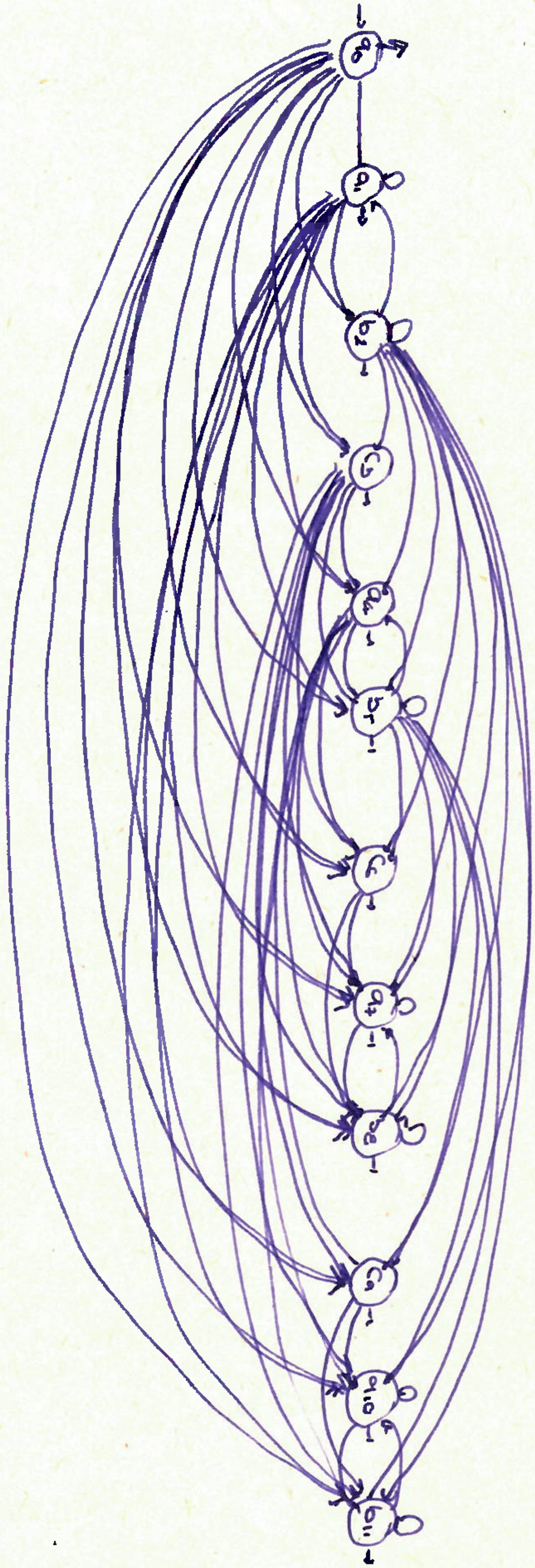
$Fol(c_6) = \{a_7, b_8, \dashv\}$

$\cancel{Fol(c_6)} =$

$Fol(a_7) = \{a_7, b_8, \dashv\}$

$Fol($





$\delta(q_0,a) = \{a_1\} \cup \{a_4\} \cup \{a_7\} \cup \{a_{10}\} = \{a_1, a_4, a_7, a_{10}\}$

$\delta(q_0,b) = \{b_2\} \cup \{b_5\} \cup \{b_8\} \cup \{b_{11}\} = \{b_2, b_5, b_8, b_{11}\}$

$\delta(q_0,c) = \{c_3\} \cup \{c_6\} \cup \{c_9\} = \{c_3, c_6, c_9\}$

$\delta(\{a_1,a_4,a_7,a_{10}\},a) = \{a_1,a_4,a_7,a_{10}\} \cup \{a_4,a_7,a_{10}\} \cup \{a_7\} \cup \{a_{10}\} = \{a_1,a_4,a_7,a_{10}\}$

$\delta(\{a_1,a_4,a_7,a_{10}\},b) = \{b_2,b_5,b_8,b_{11}\} \cup \{b_5,b_8,b_{11}\} \cup \{b_8\} \cup \{b_{11}\} = \{b_2,b_5,b_8,b_{11}\}$

$\cancel{\delta(\{a_1,a_4,a_7,a_{10}\}},$

$\delta(\{a_1,a_4,a_7,a_{10}\},c) = \{c_3,c_6,c_9\} \cup \{c_6,c_9\} \cup \phi \cup \phi = \{c_3,c_6,c_9\}$

$\delta(\{b_2,b_5,b_8,b_{11}\},a) = \{a_1,a_4,a_7,a_{10}\} \cup \{a_4,a_7,a_{10}\} \cup \{a_7\} \cup \{a_{10}\} = \{a_1,a_4,a_7,a_{10}\}$

$\delta(\{b_2,b_5,b_8,b_{11}\},b) = \{b_2,b_5,b_8,b_{11}\} \cup \{b_5,b_8,b_{11}\} \cup \{b_8\} \cup \{b_{11}\} = \{b_2,b_5,b_8,b_{11}\}$

$\delta(\{b_2,b_5,b_8,b_{11}\},c) = \{c_3,c_6,c_9\} \cup \{c_6,c_9\} = \{c_3,c_6,c_9\}$

$\delta(\{c_3,c_6,c_9\},a) = \{a_4,a_7,a_{10}\} \cup \{a_7\} \cup \{a_{10}\} = \{a_4,a_7,a_{10}\}$

$\delta(\{c_3,c_6,c_9\},b) = \{b_5,b_8,b_{11}\} \cup \{b_8\} \cup \{b_{11}\} = \{b_5,b_8,b_{11}\}$

$\delta(\{c_3,c_6,c_9\},c) = \{c_6,c_9\} \cup \phi \cup \phi = \{c_6,c_9\}$

$\delta(\{a_4, a_7, a_{10}\}, a) = \{a_4, a_7, a_{10}\} \cup \{a_7\} \cup \{a_{10}\} = \{a_4, a_7, a_{10}\}$

$\delta(\{a_4, a_7, a_{10}\}, b) = \{b_5, b_8, b_{11}\} \cup \{b_8\} \cup \{b_{11}\} = \{b_5, b_8, b_{11}\}$

$\delta(\{a_4, a_7, a_{10}\}, c) = \{c_6, c_9\} \cup \emptyset \cup \emptyset = \{c_6, c_9\}$

$\delta(\{b_5, b_8, b_{11}\}, a) = \{a_4, a_7, a_{10}\} \cup \{a_7\} \cup \{a_{10}\} = \{a_4, a_7, a_{10}\}$

$\delta(\{b_5, b_8, b_{11}\}, b) = \{b_5, b_8, b_{11}\} \cup \{b_8\} \cup \{b_{11}\} = \{b_5, b_8, b_{11}\}$

$\delta(\{b_5, b_8, b_{11}\}, c) = \{c_6, c_9\} \cup \emptyset \cup \emptyset = \{c_6, c_9\}$

$\delta(\{c_6, c_9\}, a) = \{a_7\} \cup \{a_{10}\} = \{a_7, a_{10}\}$

$\delta(\{c_6, c_9\}, b) = \{b_8\} \cup \{b_{11}\} = \{b_8, b_{11}\}$

$\delta(\{c_6, c_9\}, c\} = \emptyset \cup \emptyset$

$\delta(\{a_7, a_{10}\}, a) = \{a_7\} \cup \{a_{10}\} = \{a_7, a_{10}\}$

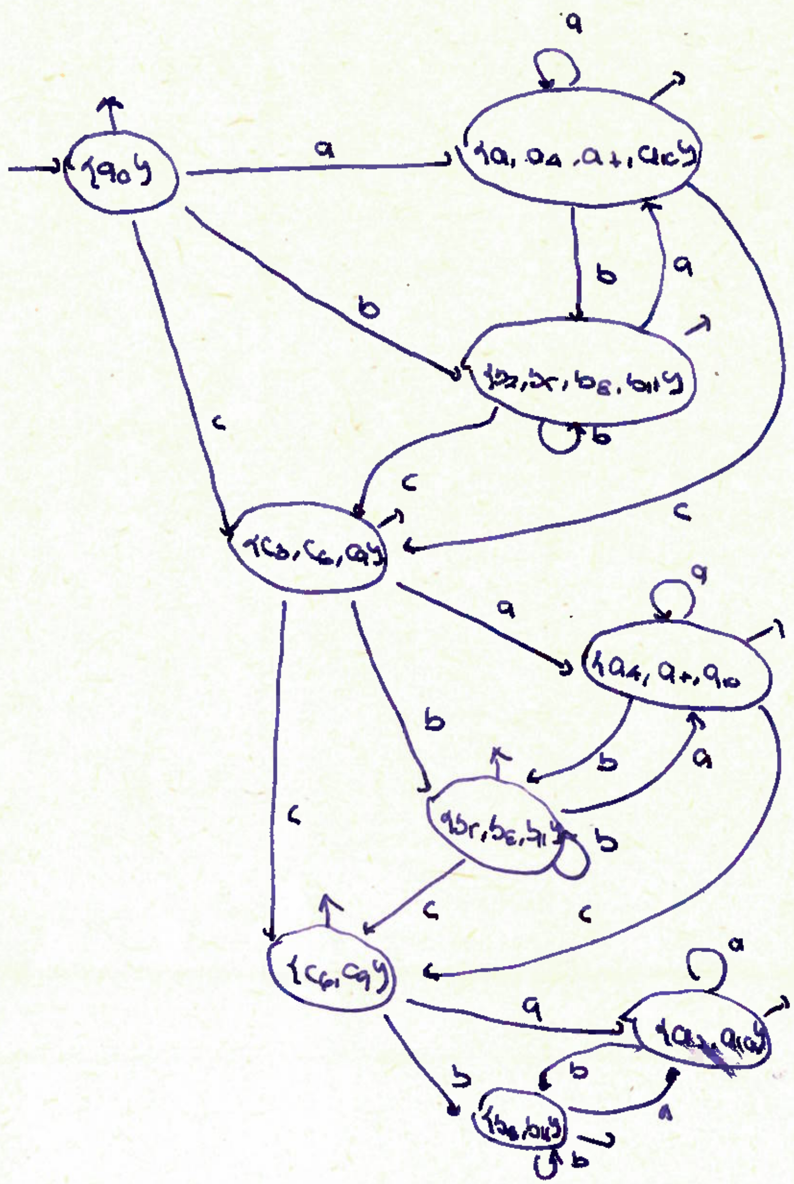
$\delta(\{a_7, a_{10}\}, b) = \{b_8\} \cup \{b_{11}\} = \{b_8, b_{11}\}$

$\delta(\{a_7, a_{10}\}, c) = \phi \cup \phi$

$\delta(\{b_8, b_{11}\}, a) = \{a_7\} \cup \{a_{10}\} = \{a_7, a_{10}\}$

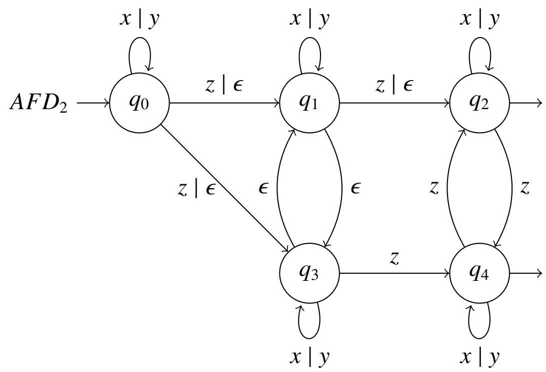
$\delta(\{b_8, b_{11}\}, b) = \{b_8\} \cup \{b_{11}\} = \{b_8, b_{11}\}$

$\delta(\{b_8, b_{11}\}, c) = \phi \cup \phi$


{q0}
{c6, q9}
{b8, b11}

2. Minimal Berry-Sethi with a little crab . . . . 100 puntos

En la figura se tiene un $AFND_2$.

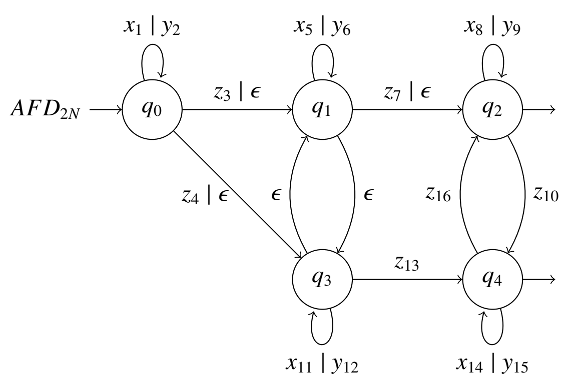


a) (70 puntos) Encuentre un autómata finito determinista AFD'_2 de tal forma que se cumpla la siguiente ecuación: $L(AFND_2) \equiv L(AFD'_2)$ utilizando Berry-Sethi.

b) (30 puntos) partir de autómata $L(AFD'_2)$ Obtenga una versión mímina AFD''_2 tal que $L(AFD'_2) \equiv L(AFD''_2)$.

Solución:

a) Construimos un autómata númerado:



A partir de este autómata construimos los conjutos: Ini, Fin, Dig, Fol.

$$Init(AFD_{2N} \dashv) = \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}$$

$$Fin(AFD_{2N}) = \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, x_{14}, y_{15}, z_{16}\}$$

$$
\begin{aligned}
Dig(AFD_{2N}) \;=\; \{ \;& x_1x_1, x_1y_2, x_1z_3, x_1z_4, x_1x_5, x_1y_6, x_1z_7, x_1x_8, x_1y_9, x_1z_{10}, x_1x_{11}, x_1y_{12}, x_1z_{13},\\
& y_2x_1, y_2y_2, y_2z_3, y_2z_4, y_2x_5, y_2y_6, y_2z_7, y_2x_8, y_2y_9, y_2z_{10}, y_2x_{11}, y_2y_{12}, y_2z_{13},\\
& z_3x_5, z_3y_6, z_3z_7, z_3x_8, z_3y_9, z_3z_{10}, z_3x_{11}, z_3y_{12}, z_3z_{13},\\
& z_4x_5, z_4y_6, z_4z_7, z_4x_8, z_4y_9, z_4z_{10}, z_4x_{11}, z_4y_{12}, z_4z_{13},\\
& x_5x_5, x_5y_6, x_5z_7, x_5x_8, x_5y_9, x_5z_{10}, x_5x_{11}, x_5y_{12}, x_5z_{13},\\
& y_6y_6, y_6y_6, y_6z_7, y_6x_8, y_6y_9, y_6z_{10}, y_6x_{11}, y_6y_{12}, y_6z_{13},\\
& z_7x_8, z_7y_9, z_7z_{10},\\
& x_8x_8, x_8y_9, x_8z_{10},\\
& y_9x_8, y_9y_9, y_9z_{10},\\
& z_{10}x_{14}, z_{10}y_{15}, z_{10}z_{16},\\
& x_{11}x_{11}, x_{11}x_{11}, x_{11}z_7, x_{11}x_8, x_{11}y_9, x_{11}z_{10}, x_{11}x_{11}, x_{11}y_{12}, x_{11}z_{13},\\
& y_{12}y_{12}, y_{12}y_{12}, y_{12}z_7, y_{12}x_8, y_{12}y_9, y_{12}z_{10}, y_{12}x_{11}, y_{12}y_{12}, y_{12}z_{13},\\
& z_{13}x_{14}, z_{13}y_{15}, z_{13}z_{16},\\
& x_{14}x_{14}, x_{14}y_{15}, x_{14}z_{16},\\
& y_{15}x_{14}, y_{15}y_{15}, y_{15}z_{16},\\
& z_{16}x_8, z_{16}y_9, z_{16}z_{10}\\
& \}
\end{aligned}
$$

Del anterior conjunto $Dig(AFD_{2N})$ computamos el conjunto Fol:

$$
\begin{aligned}
Fol(x_1) &= \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(y_2) &= \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(z_3) &= \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(z_4) &= \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(x_5) &= \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(y_6) &= \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(z_7) &= \{x_8, y_9, z_{10}, \dashv\}\\
Fol(x_8) &= \{x_8, y_9, z_{10}, \dashv\}\\
Fol(y_9) &= \{x_8, y_9, z_{10}, \dashv\}\\
Fol(z_{10}) &= \{x_{14}, y_{15}, z_{16}, \dashv\}\\
Fol(x_{11}) &= \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(y_{12}) &= \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}\\
Fol(z_{13}) &= \{x_{14}, y_{15}, z_{16}, \dashv\}\\
Fol(x_{14}) &= \{x_{14}, y_{15}, z_{16}, \dashv\}\\
Fol(y_{15}) &= \{x_{14}, y_{15}, z_{16}, \dashv\}\\
Fol(z_{16}) &= \{x_8, y_9, z_{10}, \dashv\}
\end{aligned}
$$

Ahora ya podemos construir el autómata. Comenzamos con el estado inicial:

$$Init(AFD_{2N} \dashv) = \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\} \equiv q_0$$

Calculamos $\delta(q_0, x) = Fol(x_1) \cup Fol(x_5) \cup Fol(x_8) \cup Fol(x_{11})$ obtenemos $\delta(q_0, x) = \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}$. Es un estado ya conocido q_0, por lo tanto $\delta(q_0, x) = q_0$

Calculamos $\delta(q_0, y) = Fol(y_2) \cup Fol(y_6) \cup Fol(y_9) \cup Fol(y_{12})$ obtenemos $\delta(q_0, y) = \{x_1, y_2, z_3, z_4, x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, \dashv\}$. Es un estado ya conocido, por lo tanto $\delta(q_0, y) = q_0$.

Calculamos $\delta(q_0, z) = F(z_3) \cup Fol(z_4) \cup Fol(z_7) \cup Fol(z_{10}) \cup Fol(z_{13})$ obtenemos $\delta(q_0, z) = \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un estado nuevo por lo tanto $\delta(q_0, z) = q_1$.

Calculamos $\delta(q_1, x) = Fol(x_5) \cup Fol(x_8) \cup Fol(x_{11}) \cup Fol(x_{14})$ obtenemos $\delta(q_1, x) = \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un estado ya conocido, por lo tanto $\delta(q_1, x) = q_1$.

Calculamos $\delta(q_1, y) = Fol(y_6) \cup Fol(y_9) \cup Fol(y_{12}) \cup Fol(y_{15})$ obtenemos $\delta(q_1, y) = \{x_5, y_6, z_7, x_8, y_9, z_{10}, x_{11}, y_{12}, z_{13}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un estado ya conocido, por lo tanto $\delta(q_1, y) = q_1$.

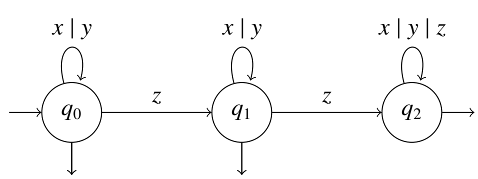
Calculamos $\delta(q_1, z) = Fol(z_7) \cup Fol(z_{10}) \cup Fol(z_{13}) \cup Fol(z_{16})$ obtenemos $\delta(q_1, z) = \{x_8, y_9, z_{10}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un nuevo, por lo tanto $\delta(q_1, z) = q_2$.

Calculamos $\delta(q_2, x) = Fol(x_8) \cup Fol(x_{14})$ obtenemos $\delta(q_2, x) = \{x_8, y_9, z_{10}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un estado ya conocido, por lo tanto $\delta(q_2, x) = q_2$.

Calculamos $\delta(q_2, y) = Fol(y_9) \cup Fol(y_{15})$ obtenemos $\delta(q_2, y) = \{x_8, y_9, z_{10}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un estado ya conocido, por lo tanto $\delta(q_2, y) = q_2$.

Calculamos $\delta(q_2, z) = Fol(z_{10}) \cup Fol(z_{16})$ obtenemos $\delta(q_2, z) = \{x_8, y_9, z_{10}, x_{14}, y_{15}, z_{16}, \dashv\}$. Es un estado ya conocido, por lo tanto $\delta(q_2, z) = q_2$.

El autómata final:



b) Minimización:

Se hace la matriz para evaluar los casos:

$\overrightarrow{q_1}$		
$\overrightarrow{q_2}$		
	$\overrightarrow{q_0}$	$\overrightarrow{q_1}$

Todos los estados son finales por lo tanto no se puede definir nada de ellos.

Ahora comparamos q_1 con q_0:

$$
\begin{aligned}
\delta(q_1, x) &= q_1 \\
\delta(q_0, x) &= q_0 \\
\delta(q_1, y) &= q_1 \\
\delta(q_0, y) &= q_0 \\
\delta(q_1, z) &= q_2 \\
\delta(q_0, z) &= q_1
\end{aligned}
$$

Con los dos primeros terminales x, y ambos estados son de aceptación, lo que podríamos inferir que son indistinguibles, pero queda pendiente el caso con z que no sabemos todavía nada entre q_2 y q_1.

Ahora comparamos q_2 y q_0.

$$
\begin{aligned}
\delta(q_2, x) &= q_2 \\
\delta(q_0, x) &= q_0 \\
\delta(q_2, y) &= q_2 \\
\delta(q_0, y) &= q_0 \\
\delta(q_2, z) &= q_2 \\
\delta(q_0, z) &= q_1
\end{aligned}
$$

Con los dos primeros terminales x, y ambos estados son de aceptación, lo que podríamos inferir que son indistinguibles, pero queda pendiente el caso con z que no sabemos todavía nada entre q_2 y q_1.

Vamos finalmente con los estados q_2 y q_1:

$$
\begin{aligned}
\delta(q_2, x) &= q_2 \\
\delta(q_1, x) &= q_1 \\
\delta(q_2, y) &= q_2 \\
\delta(q_1, y) &= q_1 \\
\delta(q_2, z) &= q_2 \\
\delta(q_1, z) &= q_2
\end{aligned}
$$

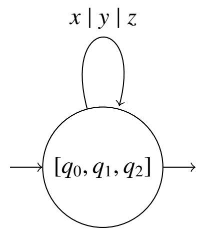
Con los dos primeros terminales x, y ambos estados son de aceptación, lo que podríamos inferir que son indistinguibles, y con el terminal z se llegan a un estado indistiguible puesto que es el mismo estado q_2 este estado siempre es indistiguible. Por lo tanto q_2 y q_1 son indistiguibles:

$\overrightarrow{q_1}$		
$\overrightarrow{q_2}$		I
	$\overrightarrow{q_0}$	$\overrightarrow{q_1}$

Y tanto los anteriores esperaban por resolver si q_2 y q_1 eran indistiguibles obtenemos:

$\overrightarrow{q_1}$	I	
$\overrightarrow{q_2}$	I	I
	$\overrightarrow{q_0}$	$\overrightarrow{q_1}$

Por lo tanto los estados q_0, q_1 y q_2 forma una clase $[q_0, q_1, q_2]$ y automáta queda de la siguiente forma:



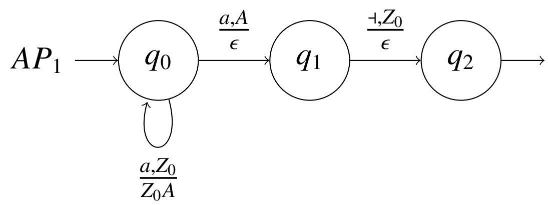
ST0270 Lenguajes formales y compiladores - 031 — **Universidad EAFIT**
Tercer parcial — **DIS**
Juan Francisco Cardona McCormick — **25 de octubre de 2018**

Nombre: ____

Código: ____

1. Autómata de pila y gramáticas .... 20 puntos

El automáta de pila AP_1 representa una gramática independiente de contexto. Utilizando el algoritmo de conversión de una autómata de pila a una gramática independiente de contexto obtener la gramática independiente de contexto CGI_1 de tal forma $L(CGI_1) \equiv L(AP_1)$.



2. Condición $LR(0)$ .... 30 puntos

Determinar si la gramática G_2 siguiente cumple con la condición $LR(0)$ de la gramática G_2, construya un piloto correspondiente y valide los macro estado para establecer que cumple con la condición $LR(0)$ y adicionalmente válida las siguientes cadenas si el piloto es $LR(0)$: $yaay \dashv$ y $zbay \dashv$

$$
\begin{array}{rcl}
S_0 & \rightarrow & S \dashv \\
S & \rightarrow & A\,y \mid B\,z \\
A & \rightarrow & A\,a \mid y \\
B & \rightarrow & B\,b \mid z
\end{array}
$$

3. Condición $LR(1)$ .... 50 puntos

Verificar la condición $LR(1)$ de la gramática G_3 construyendo el piloto correspondiente.

$$
\begin{array}{rcl}
S_0 & \rightarrow & E \dashv \\
E & \rightarrow & T\ ((\text{'+'} \mid \text{'-'})\ T)^* \\
T & \rightarrow & S\ ((\text{'*'} \mid \text{'/'})\ S)^* \\
S & \rightarrow & N\ [\ \text{'S'} \mid \text{'P'} \mid \text{'M'}\ ] \\
N & \rightarrow & [\text{'}-\text{'}]\ F \\
F & \rightarrow & \text{ID} \mid \text{NUM} \mid \text{'R'} \mid \text{'C'} \\
 & \mid & \text{'let' ID '='}\ E\ \text{'in'}\ E\ \text{'end'} \\
 & \mid & \text{'('}\ E\ \text{')'}
\end{array}
$$

$\langle q_0, z_0, q_2 \rangle \Rightarrow a \langle q_0, A, q_1 \rangle \langle q_1, z_0, q_2 \rangle$

$\Rightarrow a\ a \langle q_1, z_0, q_2 \rangle$

$\Rightarrow a\ a \dashv$

$\langle q_0, z_0, q_2 \rangle$

$\langle q_0, z_0, q_0 \rangle \to a \langle q_0, A, q_0 \rangle \langle q_0, z_0, q_0 \rangle$

$\langle q_0, z_0, q_0 \rangle \to a \underline{\langle q_0, A, q_1 \rangle} \langle q_1, z_0, q_0 \rangle$

$\langle q_0, z_0, q_0 \rangle \to a \langle q_0, A, q_2 \rangle \langle q_2, z_0, q_0 \rangle$

$\langle q_0, z_0, q_1 \rangle \to a \langle q_0, A, q_0 \rangle \langle q_0, z_0, q_1 \rangle$

$\langle q_0, z_0, q_1 \rangle \to a \underline{\langle q_0, A, q_1 \rangle} \langle q_1, z_0, q_1 \rangle$

$\langle q_0, z_0, q_1 \rangle \to a \langle q_0, A, q_2 \rangle \langle q_2, z_0, q_1 \rangle$

$\langle q_0, z_0, q_2 \rangle \to a \langle q_0, A, q_0 \rangle \langle q_0, z_0, q_2 \rangle$

$\langle q_0, z_0, q_2 \rangle \to a \underline{\langle q_0, A, q_1 \rangle} \underline{\langle q_1, z_0, q_2 \rangle}$

$\langle q_0, z_0, q_2 \rangle \to a \langle q_0, A, q_2 \rangle \langle q_2, z_0, q_2 \rangle$

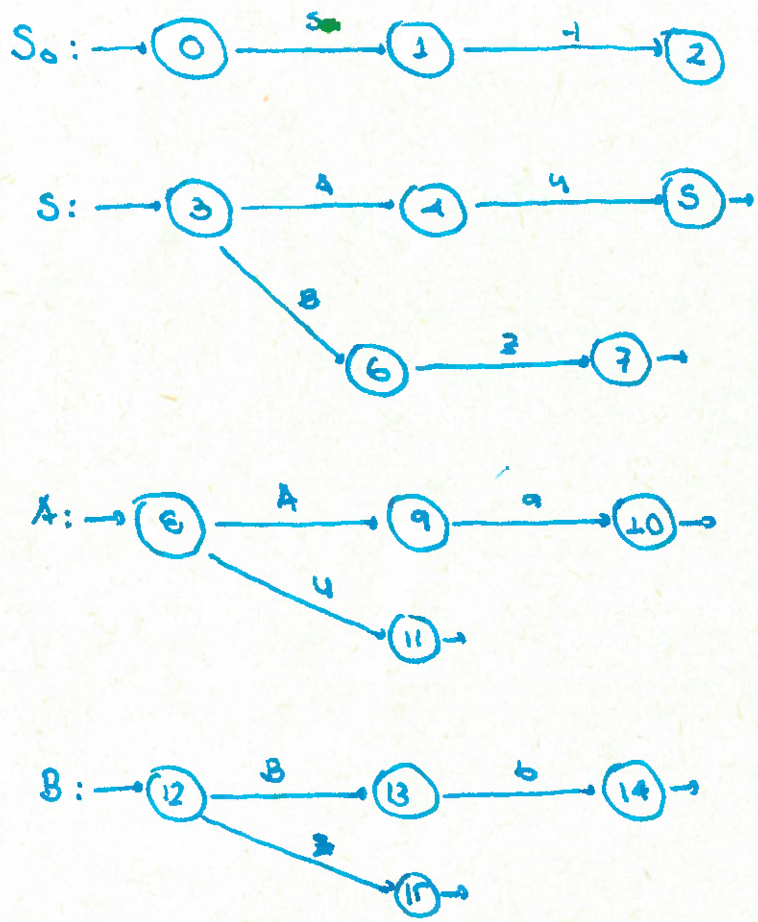
$\underline{\langle q_0, A, q_1 \rangle} \to a$

$\underline{\langle q_1, z_0, q_2 \rangle} \to \dashv$

$\langle q_0, z_0, q_2 \rangle \to a \langle q_0, A, q_1 \rangle \langle q_1, z_0, q_2 \rangle$

$\langle q_0, A, q_1 \rangle \to a$

$\langle q_1, z_0, q_2 \rangle \to \dashv$



$S_0 \to S\dashv$

$S \to Ay$
$\quad \mid Bz$

$A \to Aa$
$\quad \mid y$

$B \to Bb$
$\quad \mid z$

$S_0 \Rightarrow S\dashv$	$S_0 \Rightarrow S\dashv$
$\Rightarrow Ay\dashv$	$\Rightarrow Bz\dashv$
$\Rightarrow Aay\dashv$	$\Rightarrow Bbz\dashv$
$\Rightarrow Aaay\dashv$	$\Rightarrow Bbbz\dashv$
$\Rightarrow yaay\dashv$	$\Rightarrow Bbbbz\dashv$
	$\Rightarrow zbbbz\dashv.$

$zbay\dashv$

$closure(S) = \{0, 3, 8, 12\} = I_0$

I_0

$\vartheta(I_0, S) = closure(\delta(0,S)) = closure(1) = \{1\} = I_1$

$\vartheta(I_0, X) = closure(\delta(3,A) \cup \delta(8,A)) = closure(4) \cup closure(9) = \{4,9\} = I_2$

$\vartheta(I_0, B) = closure(\delta(3,B) \cup \delta(12,B)) = closure(6) \cup closure(13) = \{6,13\} = I_3$

$\vartheta(I_0, y) = closure(\delta(8,y)) = closure(11) = \{11\} = I_4$

$\vartheta(I_0, z) = closure(\delta(12,z)) = closure(15) = \{15\} = I_5$

I_1

$\vartheta(I_1, +) = closure(\delta(1,+)) = closure(2) = \{2\} = I_6$

I_2

$\vartheta(I_2, a) = closure(\delta(9,a)) = closure(10) = \{10\} = I_7$

$\vartheta(I_2, y) = closure(\delta(4,y)) = closure(5) = \{5\} = I_8$

I_3

$\vartheta(I_3, b) = closure(\delta(13,b)) = closure(14) = \{14\} = I_9$

$\vartheta(I_3, z) = closure(\delta(6,z)) = closure(7) = \{7\} = I_{10}$

I_4 es de terminación (reducción) no hay siguiente

I_5 es de terminación (reducción) no hay siguiente

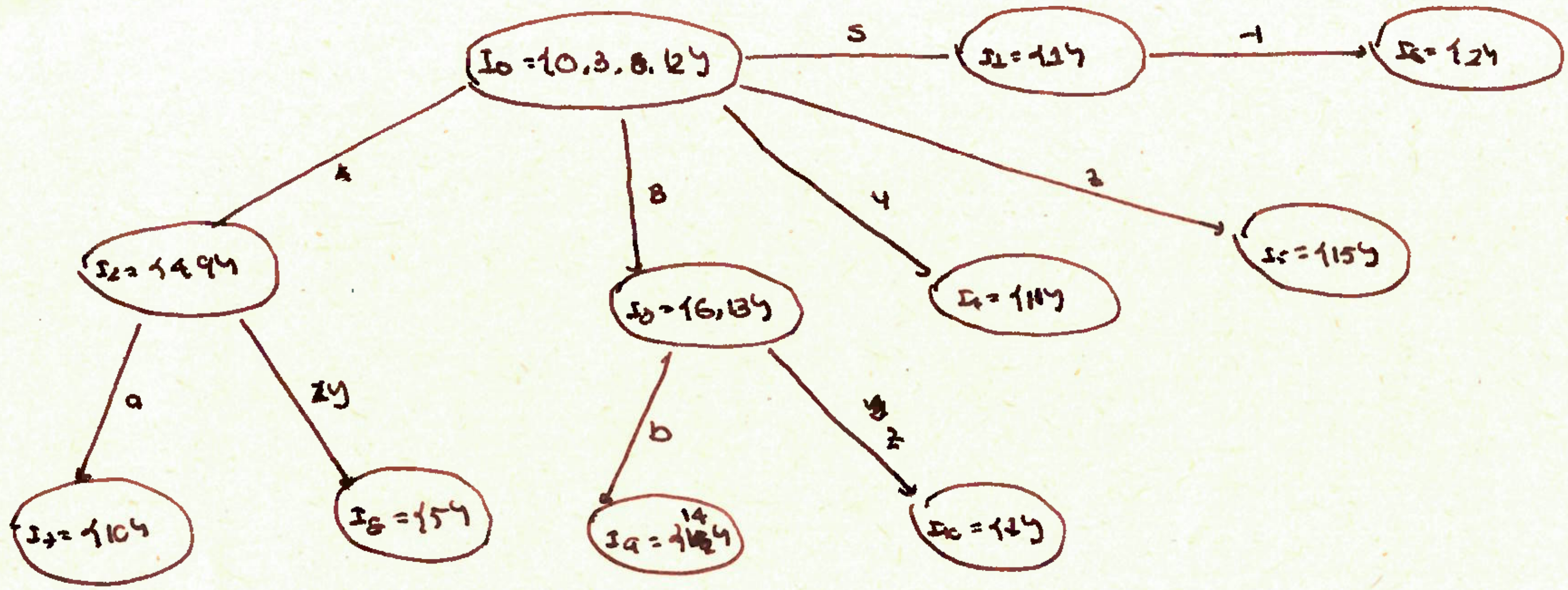
I_6 es de terminación (reducción) no hay siguiente

I_7 es de terminación (reducción) no hay siguiente

I_8 es de terminación (reducción) no hay siguiente

I_9 es de terminación (reducción) no hay siguiente

I_{10} idem.



Estados de desplazamiento I_0, I_1, I_2, I_3 no son mixtos

de reducción $I_4, I_5, I_6, I_7, I_8, I_9, I_{10}$ tienen una sola reducción

Reduccion (I_4) = $A \to y$

Reduccion (I_5) = $B \to z$

Reduccion (I_6) = $S_0 \to S ⊣$

Reduccion (I_7) = $A \to Aa$

Reducción (I_8) = $S \to Ay$

Reducción (I_9) = $B \to Bb$

Reduccion (I_{10}) = $S \to Bz$

Pila	Cadena	Mov.
I_0	uay+	D
$I_0 u I_4$	ay+	R
$I_0 A I_2$	ay+	D
$I_0 A I_2 a I_7$	y+	R
$I_0 A I_2$	ay+	D
$I_0 A I_2 a I_7$	y+	R
$I_0 A I_2$	y+	D
$I_0 A I_2 y I_6$	+	R
$I_0 S I_1$	+	D
$I_0 S I_1 + I_6$		R
$I_0 S_0$		Aceptar y parar

Pila	Cadena	Mov
I_0	2bay⊣	D
$I_0 2 I_5$	bay⊣	R
$I_0 B I_3$	ay⊣	No hay movimiento, Rechazar la cadena.

Nombre: ______________________________

Código: ______________________________

1. Autómata de pila .... 20 puntos

Implemente un autómata de pila determinista o no determinista que reconoce el siguiente lenguaje L_1

$$L_1 = \{xa^n \mid n \geq 0,\ x \in \{a, b\}^* \wedge |x| \equiv n\}$$

2. Condición $LR(1)$ .... 20 puntos

La siguiente gramática no cumple con la condición $LR(1)$ de la gramática G_2, construya un piloto parcial que muestre que uno estado no cumple con dicha condición.

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & WY \mid XZ \\ W & \rightarrow & w \\ X & \rightarrow & wx \\ Y & \rightarrow & y \mid yY \\ Z & \rightarrow & z \end{array}$$

3. Condición $LR(1)$ .... 60 puntos

Verificar la condición $LR(1)$ (45 Puntos) de la gramática G_3 construyendo el piloto correspondiente y verificando que la la cadena $0a1$ (15 Puntos) pertenece a la gramática.

$$\begin{array}{rcl} S_0 & \rightarrow & S \dashv \\ S & \rightarrow & 0W1 \mid 0X2 \\ W & \rightarrow & a \\ X & \rightarrow & b \end{array}$$

Nombre: ______________________________
Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(40 %) 1. **Expresiones regulares y gramáticas independientes de contexto**

Encuentre las expresiones regulares y las gramáticas independientes de contexto correspondiente.

a) La expresión regular $r_1 = a^*b \cup cd^*$ produce el lenguaje $L(r_1)$. Escribir una gramática independiente de contexto G_1 cuyo lenguaje cumple la siguiente condición $L(G_1) \equiv L(r_1)$.

b) La gramática independiente de contexto G_2 que se muestra a continuación es una gramática que produce un lenguaje regular r_2 que tiene como propiedad $L(G_2) \equiv L(r_2)$.

$$
\begin{aligned}
G_2 \quad = \quad & (V = \{S, W, W', W'', X, Y, Y', Y''\}, \\
& \Sigma = \{w, x, y\}, \\
& P = \{S \rightarrow WX \mid Y, W \rightarrow W'W'' \mid w, W' \rightarrow w, W'' \rightarrow W'W \\
& X \rightarrow x, \\
& Y \rightarrow Y'Y'' \mid y, Y' \rightarrow y, Y'' \rightarrow Y'Y\}, \\
& S)
\end{aligned}
$$

Solución:

a) Una posible solución:

$$G_1 = (V = \{S, A, B, C, D\}, \Sigma = \{a, b, c, d\}, P = \{S \rightarrow AB \mid CD, A \rightarrow aA \mid \epsilon, B \rightarrow b, C \rightarrow c, D \rightarrow dD \mid \epsilon\}, S)$$

b) Una posible solución:

$$r_2 = (ww)^*wx \mid (yy)^*y$$

(30 %) 2. **Ambigüedad en gramáticas independientes de contexto**
La siguiente gramática independiente de contexto G_3 es una gramática ambigua:

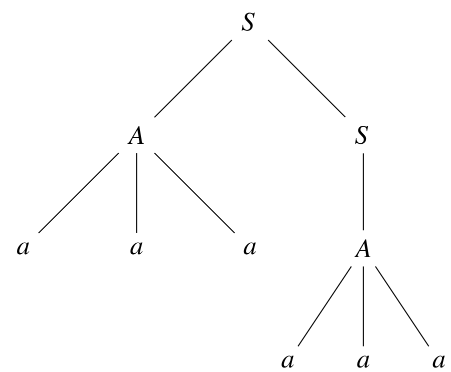
$$G_3 = (V = \{S, A, B, C\}, \Sigma = \{a, b\}, P = \{S \to AS \mid BS \mid CS \mid A \mid B \mid C,\ A \to aaa, B \to aa, C \to ab\},\ S)$$

a) Muestre que G_3 es ambigua.

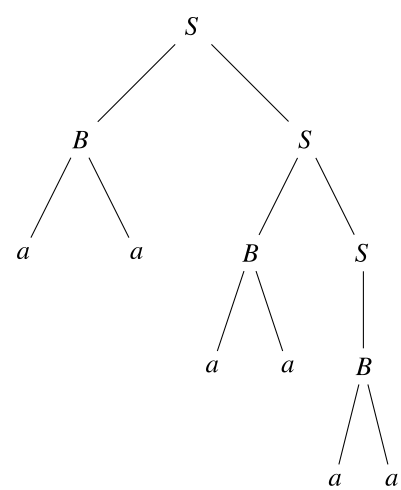
b) Defina una nueva gramática G_3' tal que $L(G_3')$ no es ambigua.

Solución:

a) La cadena $aaaaaa$ se puede generar por la siguiente derivación:



$$\begin{aligned} S &\Rightarrow AS \\ &\Rightarrow aaaS \\ &\Rightarrow aaaA \\ &\Rightarrow aaaaaa \end{aligned}$$



$$\begin{aligned} S &\Rightarrow BS \\ &\Rightarrow aaS \\ &\Rightarrow aaBS \\ &\Rightarrow aaaaS \\ &\Rightarrow aaaaB \\ &\Rightarrow aaaaaa \end{aligned}$$

b) Una posible solución:

$$G_3' = (V = \{S, A, B, C\}, \Sigma = \{a, b\}, P = \{S \to AS \mid BS \mid CS \mid A \mid B \mid C, A \to aa, B \to bb, C \to ab\},\ S)$$

(30 %) 3. **Transformación de gramáticas independientes de contexto**

La G_4 tiene regla de copias, transforme en una nueva gramática G'_4 que $L(G_4) \equiv L(G'_4)$, donde G'_4 tiene no tiene reglas de copia.

$$G_4 = (V = \{S, A, B, C, D\}, \Sigma = \{a, b, c, d\}, P = \{S \to A \mid B \mid C, A \to Aa \mid D \mid c, B \to Bb \mid \epsilon, C \to ccC \mid c \mid \epsilon, D \to d\}, S)$$

Solución:

Calculamos el conjunto $Null_0 = \{B, C\}$.

$$Null_1 = Null_0 \cup \{S\} = \{B, C, S\}$$

Calculando el siguiente conjunto:

$$Null_2 = Null_1 \cup \{\} = \{B, C, S\}$$

Ahora obtenemos $Null = \{B, C, S\}$.

Creamos una gramática G''_4

$$\begin{array}{rcl} S & \to & A \mid B \mid C \mid \epsilon \\ A & \to & Aa \mid D \mid c \\ B & \to & Bb \mid b \\ C & \to & ccC \mid cc \mid c \\ D & \to & d \end{array}$$

Calculamos un conjunto de copia:

$$\begin{array}{rcl} Copy(S) & = & \{S, A, B, C, D\} \\ Copy(A) & = & \{A, D\} \\ Copy(B) & = & \{B\} \\ Copy(C) & = & \{C\} \\ Copy(D) & = & \{D\} \end{array}$$

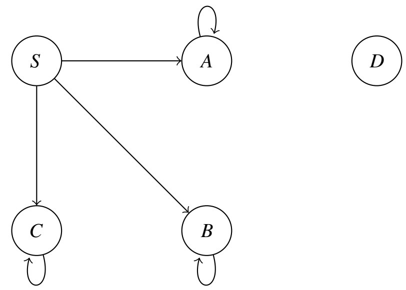
Define una gramática G'''_4

$$\begin{array}{rcl} S & \to & Aa \mid c \mid Bb \mid b \mid ccC \mid cc \mid c \mid d \mid \epsilon \\ A & \to & Aa \mid c \mid d \\ B & \to & Bb \mid b \\ C & \to & ccC \mid cc \mid c \\ D & \to & d \end{array}$$

La gramática anterior no es limpia:

El conjunto $Def_0 = \{S, A, B, C, D\} \equiv V$.

Calculamos el grafo de produce:



Es claro que el no-terminal D no es alcanzable, la gramática G_4':

$$
\begin{array}{rcl}
S & \rightarrow & Aa \mid c \mid Bb \mid b \mid ccC \mid cc \mid c \mid d \mid \epsilon \\
A & \rightarrow & Aa \mid c \mid d \\
B & \rightarrow & Bb \mid b \\
C & \rightarrow & ccC \mid cc \mid c
\end{array}
$$

Nombre: ______________________________

Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(20 %) 1. **Expresiones regulares ambiguas**
La expresión $r_1 = ((ab)^*|(aba^*)^*)$ es una expresión regular ambigua. Demuestre formalmente lo afirmado en el parágrafo anterior.

Solución:

Obtenemos una nueva expresión regular numerada $r_1'1 = ((a_1b_2)^*|(a_3b_4a_5^*)^*)$

Vamos a obtener una cadena:

$$\begin{aligned} ((a_1b_2)^*|(a_3b_4a_5^*)^*) &\Rightarrow (a_1b_2)^* \\ &\Rightarrow (a_1b_2)^1 \equiv a_1b_2 \end{aligned}$$

Vamos a obtener otra cadena por derivaciones diferentes:

$$\begin{aligned} ((a_1b_2)^*|(a_3b_4a_5^*)^*) &\Rightarrow (a_3b_4a_5^*)^* \\ &\Rightarrow (a_3b_4a_5^*)^1 \equiv a_3b_4a_5^* \\ &\Rightarrow a_3b_4\epsilon \equiv a_3b_4 \end{aligned}$$

Tenemos dos cadenas aparente diferentes: a_1b_2 y a_3b_4 al borrar los número obtenemos: ab y ab. Por lo tanto son la misma cadena obtenidas por dos derivaciones diferentes la expresión regular es ambigua.

(40 %) 2. **Expresiones regulares y gramáticas independientes de contexto**

Encuentre las expresiones regulares y las gramáticas independientes de contexto correspondiente.

a) La expresión regular $r_2 = ab^*a \cup cd^*c$ produce el lenguaje $L(r_2)$. Escribir una gramática independiente de contexto G_2 cuyo lenguaje cumple la siguiente condición $L(G_2) \equiv L(r_2)$.

b) La gramática independiente de contexto G_3 que se muestra a continuación es una gramática que produce un lenguaje regular r_2 que tiene como propiedad $L(G_3) \equiv L(r_3)$.

$$\begin{aligned} G_3 &= (V = \{S, A, B, C, D\}, \\ &\quad \Sigma = \{a, b, c, d\}, \\ &\quad P = \{S \rightarrow AB \mid CD, A \rightarrow aA \mid \epsilon, B \rightarrow b, C \rightarrow c, D \rightarrow dD \mid \epsilon\} \\ &\quad S\,) \end{aligned}$$

Solución:

a) Una posible solución:

$$G_2 = (V = \{S, B, D\}, \Sigma = \{a, b, c, d\}, P = \{S \rightarrow aBa \mid cDc, B \rightarrow bB \mid \epsilon, D \rightarrow dD \mid \epsilon\}, S)$$

b) Una posible solución:

$$r_3 = a^*b \mid cd^*$$

(20 %) 3. **Ambigüedad en gramáticas independientes de contexto**
La siguiente gramática independiente de contexto G_3 es una gramática ambigua:

$$G_4 = (V = \{A, B, C\}, \Sigma = \{a, b, c, d\}, P = \{A \to dBAA \mid dBA \mid C,\ B \to b, C \to a \mid c\},\ A)$$

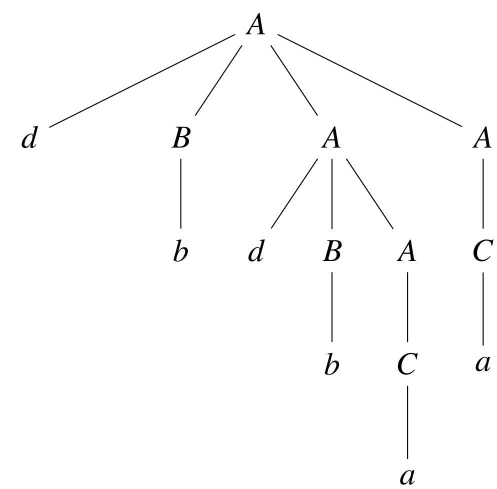
a) Muestre que G_4 es ambigua.

b) Defina una nueva gramática G'_4 tal que $L(G'_4)$ no es ambigua y $L(G_4) \equiv L(G'_4)$.

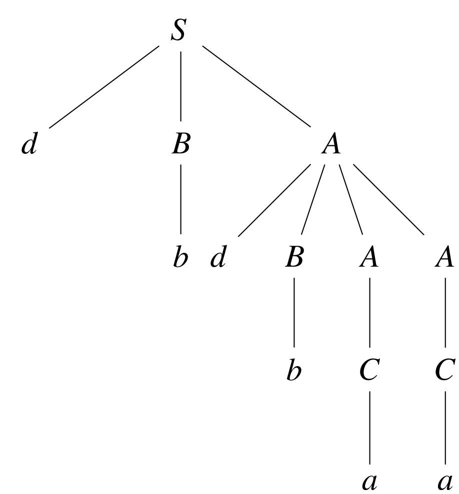
Solución:

a) La cadena *dbdbaa* se puede generar por la siguiente derivación:

$$\begin{aligned} A &\Rightarrow dBAA \\ &\Rightarrow dbAA \\ &\Rightarrow dbdBAA \\ &\Rightarrow dbdbAA \\ &\Rightarrow dbdbCA \\ &\Rightarrow dbdbaA \\ &\Rightarrow dbdbaC \\ &\Rightarrow dbdbaa \end{aligned}$$



$$\begin{aligned} A &\Rightarrow dBA \\ &\Rightarrow dbA \\ &\Rightarrow dbdBAA \\ &\Rightarrow dbdbAA \\ &\Rightarrow dbdbCA \\ &\Rightarrow dbdbaA \\ &\Rightarrow dbdbaC \\ &\Rightarrow dbdbaa \end{aligned}$$



b) Una posible solución: Como esta gramática sufre el mismo problema del *colgante* se hace una solución similar.

$$
\begin{aligned}
G'_4 \;=\; & (V = \{A, A', A'', B, C\}, \\
& \Sigma = \{a, b, d\}, \\
& P = \{A \rightarrow A' \;\mid\; A'', A' \rightarrow dBA'A' \;\mid\; C,\; A'' \rightarrow dBA'A'' \;\mid\; dbA,\; B \rightarrow b, C \rightarrow a\}, \\
& A)
\end{aligned}
$$

(20 %) 4. **Gramática limpia**

La G_5 no es una gramática limpia, por favor limpie la gramática y produzca una gramática nueva G'_5 que $L(G_5) \equiv L(G'_5)$, donde G'_5 esta limpia.

$$\begin{array}{rcl} A & \rightarrow & BC \mid BD \\ B & \rightarrow & Bb \mid \epsilon \\ C & \rightarrow & cC \mid dC \\ D & \rightarrow & ddD \mid E \\ E & \rightarrow & eeeD \mid e \\ F & \rightarrow & FG \mid f \\ G & \rightarrow & hG \mid h \end{array}$$

Solución:

Calculamos el conjunto $Def_0 = \{B, E, F, G\}$:

Siguiente iteración:

$$Def_1 = Def_0 \cup \{D\} = \{B, E, F, G, D\}$$

Siguiente iteración:

$$Def_2 = Def_1 \cup \{A\} = \{B, E, F, G, D, A\}$$

Siguiente iteración:

$$Def_3 = Def_2 \cup \{\} = \{B, E, F, G, D, A\}$$

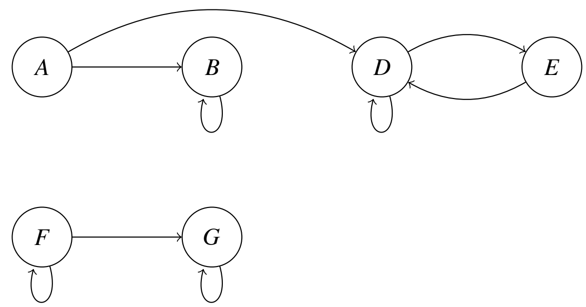
Llegamos al punto fijo.

Encontramos que no están definidos: $NoDef = V \setminus Def = \{C\}$.

Eliminamos el no-terminal C y obtenemos la gramática G''_5

$$\begin{array}{rcl} A & \rightarrow & BD \\ B & \rightarrow & Bb \mid \epsilon \\ D & \rightarrow & ddD \mid E \\ E & \rightarrow & eeeD \mid e \\ F & \rightarrow & FG \mid f \\ G & \rightarrow & hG \mid h \end{array}$$

Ahora creamos el grafo de produce:



Se observa que los no-terminales F y G no son alcanzables desde el axioma entonces los eliminamos de la gramática y obtenemos la gramática G_5':

$$
\begin{array}{rcl}
A & \rightarrow & BD \\
B & \rightarrow & Bb \mid \epsilon \\
D & \rightarrow & ddD \mid E \\
E & \rightarrow & eeeD \mid e
\end{array}
$$

Nombre: ____________________
Código: ____________________

Nota: El parcial tiene dos puntos, seleccione uno de los dos puntos y resuélvalo.

1. Crabs, crabs everywhere . . . . 100 puntos

La siguiente es una gramática lineal por la izquierda GLI_1.

$$\begin{array}{lcl} A & \rightarrow & Ab \mid Ba \mid Ca \\ B & \rightarrow & Bb \mid \epsilon \\ C & \rightarrow & Cb \mid Aa \end{array}$$

a) (30 puntos) Obtener la gramática lineal por la derecha GLD_1 tal que $L(GLI_1) \equiv L(GLD_1)$.

b) (20 puntos) Obtener un autómata M_1 correspondiente a la gramática GLD_1 de tal forma que el lenguague que reconoce $L(M_1) \equiv L(GLD_1)$.

c) (40 puntos) Obtener un autómata M_1' equivalente a M_1 de forma que M_1 sea un autómata mínimo y que lenguaje que reconocen ambos sea el mismo $L(M_1) \equiv$.

d) (10 puntos) Obtener la gramática líneal por la derecha del autómata mínimo M_1'.

Solución:

a) Ver el paso siguiente y del autómata allí obtenido:

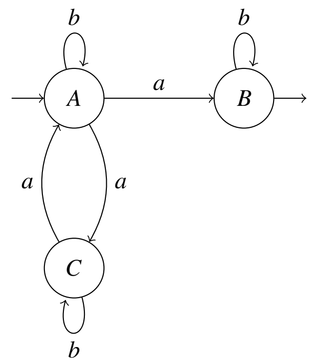
Obtenemos la gramática derecha:

$$\begin{array}{lcl} B & \rightarrow & bB \mid aA \\ A & \rightarrow & bA \mid aC \mid \epsilon \\ C & \rightarrow & bC \mid aA \end{array}$$

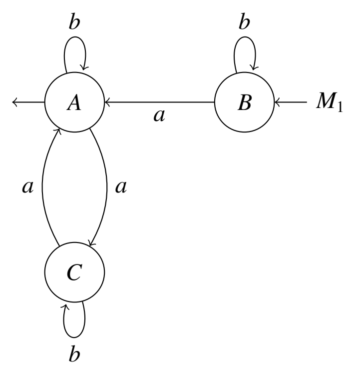
b) Invertimos la gramática GLD_1:

$$\begin{array}{lcl} A & \rightarrow & bA \mid aB \mid aC \\ B & \rightarrow & bB \mid \epsilon \\ C & \rightarrow & bC \mid aA \end{array}$$

Obtenemos su autómata:



Invertimos el autómata y obtenemos el autómata M_1:



c) Minimización:

$\overrightarrow{A}$		
C		
	B	A

$\overrightarrow{A}$ es un estado de finalización con los que son de finalización:

$\overrightarrow{A}$	X	
C		X
	B	$\overrightarrow{A}$

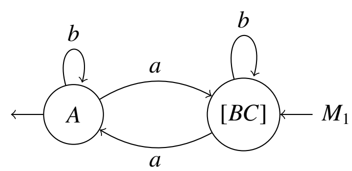
Ahora examinamos los estados C y B:

$$\begin{aligned} \delta(C,a) &= A \\ \delta(B,a) &= A \\ \delta(C,b) &= C \\ \delta(B,b) &= B \end{aligned}$$

En la primera transición llegamos a estados que ambos son finales A y en la segunda ambos no lo son. Estos cumplen con la definición. Por lo tanto son indistinguibles.

$\overrightarrow{A}$	X	
C	I	X
	B	$\overrightarrow{A}$

Entonces reconstruimos el autómata así:



d) Gramática:

$$[BC] \rightarrow a[A] \mid b[BC]$$

$$[A] \rightarrow a[BC] \mid b[A] \mid \epsilon$$

2. A crab history .... 100 puntos

La siguiente es una gramática lineal por la izquierda GLI_2.

$$\begin{array}{rcl} A & \rightarrow & Ba \mid Bb \mid Ab \mid Ca \mid Da \\ B & \rightarrow & \epsilon \\ C & \rightarrow & Cb \mid Db \mid Aa \\ D & \rightarrow & Db \mid Cb \mid Aa \end{array}$$

a) (30 puntos) Obtener la gramática lineal por la izquierda GLD_2 tal que $L(GLI_2) \equiv L(GLD_2)$.

b) (20 puntos) Obtener el autómata no determinista N_2 correspondiente a la gramática GLD_2 de tal forma que el lenguaje que reconoce $L(M_2) \equiv L(GLD_2)$.

c) (40 puntos) Obtener el autómata determinista M_2 de tal forma $L(N_2) \equiv L(M_2)$.

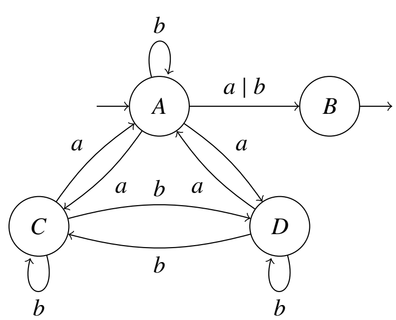
d) (10 puntos) La gramática línea por la derecha GLI'_2 del automáta determinista M_2.

Solución:

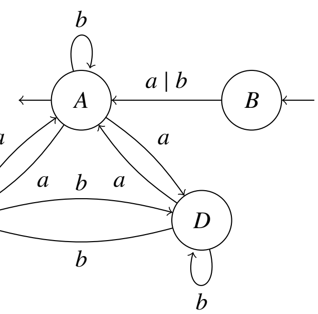
a) Del autómata obtenido del punto siguiente se obtiene la gramática por la derecha:

$$\begin{array}{rcl} B & \rightarrow & aA \mid bA \\ A & \rightarrow & aC \mid aD \mid bA \mid \epsilon \\ C & \rightarrow & aA \mid bC \mid bD \\ D & \rightarrow & aA \mid bC \mid bD \end{array}$$

b) Invertimos la gramática GLD_2:



Ahora invertimos este autómata.



c) Eliminación del no-determinismo:

$$\begin{aligned} \delta'([B], a) &= \delta(B, a) \equiv \{A\} \equiv [A] \\ \delta'([B], b) &= \delta(B, b) \equiv \{A\} \equiv [A] \end{aligned}$$

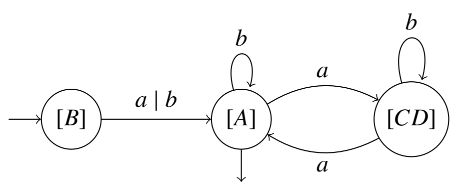
Ahora tenemos un nuevo estado $[A]$ y analizamos:

$$\begin{aligned} \delta'([A], a) &= \delta(A, a) \equiv \{C, D\} \equiv [C, D] \\ \delta'([A], b) &= \delta(A, b) \equiv \{A\} \equiv [A] \end{aligned}$$

Ahora tenemos un nuevo estado $[C, D]$ y analizamos:

$$\begin{aligned} \delta'([C, D], a) &= \delta(C, a) \cup \delta(D, a) \equiv \{A\} \cup \{A\} \equiv [A] \\ \delta'([C, D], b) &= \delta(C, b) \cup \delta(D, b) \equiv \{C\} \cup \{D\} \equiv [C, D] \end{aligned}$$

Y no se ha añadido ningún nuevo estado. Ahora podemos construir el nuevo autómata:



Construimos la gramática:

$$[B] \rightarrow a[A] \mid b[A]$$

$$[A] \rightarrow b[A] \mid b[C, D] \mid \epsilon$$

$$[C, D] \rightarrow a[A] \mid b[C, D]$$

Nombre: ______________________________
Código: ______________________________

Nota: El parcial tiene dos puntos, seleccione uno de los dos puntos y resuélvalo.

1. Mami, I don't like crabs . . . . 100 puntos

La siguiente es una gramática en forma normal Chomsky G_1.

$$\begin{array}{rcl} I & \rightarrow & IM \mid JL \mid KL \mid 0 \\ J & \rightarrow & JM \mid 1 \\ K & \rightarrow & KM \mid IL \\ L & \rightarrow & 0 \\ M & \rightarrow & 1 \end{array}$$

Esta gramática G_1 fue obtenida de una gramática GLI_1. La transformación se aplicaron dos transformación previas. La primera la eliminación de los nulos, obteniendo un conjunto $Null = \{J\}$, se eliminaron los nulos de las gramáticas. La segunda transformación es la transformación Chomsky en la que se añadieron dos no-terminales auxiliares: L, M.

a) (30 puntos) Obtener la gramática lineal por la izquierda original GLI_1. (Pistas: deshaga las transformaciones en orden inverso, primer deshaga la transformación de Chomsky, y luego la del nulo, en el paso de la eliminación del nulo tenga en cuenta el conjunto de los nulos, produzca una gramática estrictamente lineal por la izquierda).

b) (30 puntos) Obtener la gramática lineal por la derecha GLD_1 tal que $L(GLI_1) \equiv L(GLD_1)$.

c) (30 puntos) Obtener un autómata M_1' equivalente a M_1 de forma que M_1' sea un autómata mínimo y que lenguaje que reconocen ambos sea el mismo $L(M_1) \equiv L(M_1')$.

d) (10 puntos) Obtener la gramática líneal por la derecha del autómata mínimo M_1'.

Solución:

a) Sustituyendo los no-terminales L y M en las primeras tres producciones obtenemos:

$$\begin{array}{rcl} I & \rightarrow & I1 \mid J0 \mid K0 \mid 0 \\ J & \rightarrow & J1 \mid 1 \\ K & \rightarrow & K1 \mid I0 \\ L & \rightarrow & 0 \\ M & \rightarrow & 1 \end{array}$$

Al observar la anterior gramática podemos darnos cuenta que no se encuentra limpia, obtenemos la siguiente:

$$\begin{array}{rcl} I & \rightarrow & I1 \mid J0 \mid K0 \mid 0 \\ J & \rightarrow & J1 \mid 1 \\ K & \rightarrow & K1 \mid I0 \end{array}$$

Ahora deshacemos la eliminación de lo nulos. En primer lugar el único no-terminal que deriva directamente el nulo es J, por lo tanto la transformación implicó que cualquier parte derecha de una producción era sustituida la J. En la primera producción se observa ello y se puede rescribir así

$$\begin{array}{rcl} I & \rightarrow & I1 \mid J0 \mid K0 \mid J0 \\ J & \rightarrow & J1 \mid 1 \\ K & \rightarrow & K1 \mid I0 \end{array}$$

Pero la producción esta repetida por lo tanto se elimina. En la siguiente producción pasa algo similar y se deja el nulo.

$$I \rightarrow I1 \mid J0 \mid K0 \mid J0$$

Se elimina la reglas repetidas.

$$I \rightarrow I1 \mid J0 \mid K0$$

Ahora en J se rescribe la producción igualmente y se obtiene:

$$J \rightarrow J1 \mid J1$$

Se tiene repetida y recordamos que el conjunto $J \in Null$ añadimos la regla nula.

$$J \rightarrow J1 \mid \epsilon$$

Por lo tanto la gramática líneal por la izquierda obtenida es la siguiente:

$$\begin{array}{rcl} I & \rightarrow & I1 \mid J0 \mid K0 \\ J & \rightarrow & J1 \mid \epsilon \\ K & \rightarrow & K1 \mid I0 \end{array}$$

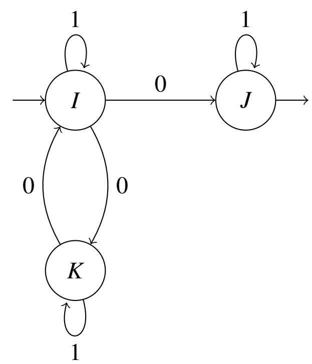
b) Ver el paso siguiente del autómata allí obtenido. Obtenemos la gramática derecha:

$$\begin{array}{rcl} J & \rightarrow & 1J \mid 0I \\ I & \rightarrow & 1I \mid 0K \mid \epsilon \\ K & \rightarrow & 1K \mid 0A \end{array}$$

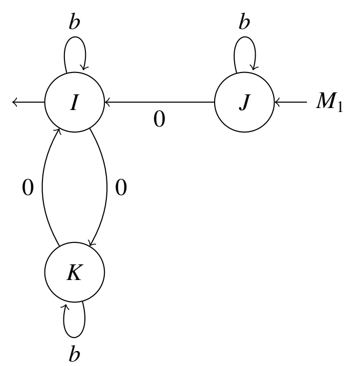
c) Invertimos la gramática obtenida en el primer punto.

$$\begin{array}{rcl} I & \rightarrow & 1A \mid 0J \mid 0K \\ J & \rightarrow & 1J \mid \epsilon \\ K & \rightarrow & 1C \mid 0I \end{array}$$

Obtenemos su autómata:



Invertimos el autómata:



d) Minimización:

$\overrightarrow{I}$		
K		
	J	$\overrightarrow{i}$

$\overrightarrow{A}$ es un estado de finalización con los que no son de finalización:

$\overrightarrow{I}$	X	
K		X
	J	$\overrightarrow{I}$

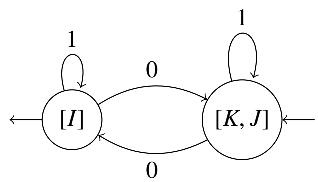
Ahora examinamos los estados K y J:

$$
\begin{aligned}
\delta(K,0) &= I \\
\delta(J,0) &= I \\
\delta(K,1) &= K \\
\delta(J,1) &= J
\end{aligned}
$$

En la primera transición llegamos a estados que ambos son finales I y en la segunda ambos no lo son. Estos cumplen con la definición. Por lo tanto son indistinguibles.

$\overrightarrow{I}$	X	
K	I	X
	J	$\overrightarrow{I}$

Entonces reconstruimos el autómata así:



e) Ahora construimos la gramática:

$$[K, J] \rightarrow 0[I] \mid 1[K, J]$$

$$[I] \rightarrow 1[I] \mid 0[K, J] \mid \epsilon$$

2. Crab foot, again! .... 100 puntos

La siguiente es una gramática en forma normal Chomsky G_2.

$$\begin{array}{rcl} I & \rightarrow & 0 \mid 1 \mid IN \mid KM \mid LM \\ K & \rightarrow & KN \mid LN \mid IM \\ L & \rightarrow & LN \mid KN \mid IM \\ M & \rightarrow & 0 \\ N & \rightarrow & 1 \end{array}$$

Esta gramática G_2 fue obtenida de una gramática GLI_2. La transformación se aplicaron dos transformación previas. La primera la eliminación de los nulos, obteniendo un conjunto $Null = \{J\}$, se eliminaron los nulos de las gramáticas. La segunda transformación es la transformación Chomsky en la que se añadieron dos no-terminales auxiliares: M, N.

a) (30 puntos) Obtener la gramática lineal por la izquierda original GLI_2. (Pistas: deshaga las transformaciones en orden inverso, primer deshaga la transformación de Chomsky, y luego la del nulo, en el paso de la eliminación del nulo tenga en cuenta el conjunto de los nulos, produzca una gramática estrictamente lineal por la izquierda).

b) (30 puntos) Obtener la gramática lineal por la derecha GLD_2 tal que $L(GLI_2) \equiv L(GLD_2)$.

c) (30 puntos) Obtener el autómata determinista M_2 de tal forma $L(GLD_2) \equiv L(M_2)$.

d) (10 puntos) La gramática línea por la derecha GLD'_2 del automáta determinista M_2.

Solución:

a) En primer lugar deshacemos la transformación de Chomsky, sustituyendo M y N. En las demás partes de la gramática:

$$\begin{array}{rcl} I & \rightarrow & 0 \mid 1 \mid I1 \mid K0 \mid L0 \\ K & \rightarrow & K1 \mid L1 \mid I0 \\ L & \rightarrow & L1 \mid K1 \mid I0 \\ M & \rightarrow & 0 \\ N & \rightarrow & 1 \end{array}$$

Se observa que los dos no-terminales M y N no son alcanzables, se eliminan de la gramática.

$$\begin{array}{rcl} I & \rightarrow & 0 \mid 1 \mid I1 \mid K0 \mid L0 \\ K & \rightarrow & K1 \mid L1 \mid I0 \\ L & \rightarrow & L1 \mid K1 \mid I0 \end{array}$$

Ahora identificamos que el único no-terminal que produce el nulo es J entonces lo añadimos.

$$\begin{array}{rcl} I & \rightarrow & 0 \mid 1 \mid I1 \mid K0 \mid L0 \\ J & \rightarrow & \epsilon \\ K & \rightarrow & K1 \mid L1 \mid I0 \\ L & \rightarrow & L1 \mid K1 \mid I0 \end{array}$$

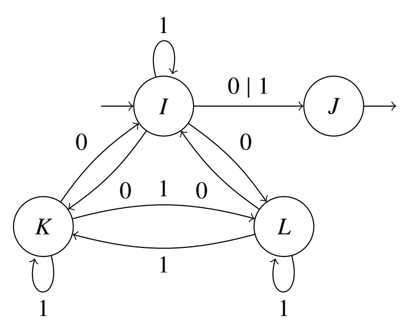
Esta gramática todavía no es una gramática estrictamente unilineal por la izquierda, por lo tanto observamos que el no-terminal I tiene dos reglas terminales, esta reglas son producidas por el nulo J. Producimos la gramática GLI_2:

$$\begin{array}{rcl} I & \rightarrow & J0 \mid J1 \mid I1 \mid K0 \mid L0 \\ J & \rightarrow & \epsilon \\ K & \rightarrow & K1 \mid L1 \mid I0 \\ L & \rightarrow & L1 \mid K1 \mid I0 \end{array}$$

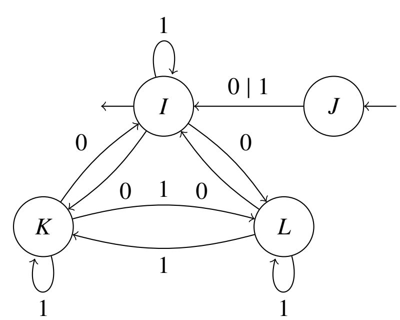
b) Invertimos la gramática GLI_2:

$$\begin{array}{rcl} I & \rightarrow & 0J \mid 1J \mid 1I \mid 0K \mid 0L \\ J & \rightarrow & \epsilon \\ K & \rightarrow & 1K \mid 1L \mid 0I \\ L & \rightarrow & 1L \mid 1K \mid 0I \end{array}$$

Obtenemos su autómata:



Ahora invertimos este autómata.



Y obtenemos la gramática GLD_2:

$$\begin{array}{rcl} J & \rightarrow & 0I \mid 1I \\ I & \rightarrow & 1I \mid 0K \mid OL \mid \epsilon \\ K & \rightarrow & 0I \mid 1K \mid 1L \\ L & \rightarrow & 0I \mid 1K \mid 1L \end{array}$$

c) Eliminación del no-determinismo:

$$\begin{aligned}\delta'([J],0) &= \delta(J,0) \equiv \{I\} \equiv [I]\\ \delta'([J],1) &= \delta(J,1) \equiv \{I\} \equiv [I]\end{aligned}$$

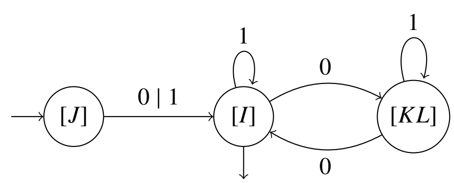
Ahora tenemos un nuevo estado $[I]$ y analizamos:

$$\begin{aligned}\delta'([I],0) &= \delta(I,0) \equiv \{K,L\} \equiv [K,L]\\ \delta'([I],1) &= \delta(I,1) \equiv \{I\} \equiv [I]\end{aligned}$$

Ahora tenemos un nuevo estado $[K,L]$ y analizamos:

$$\begin{aligned}\delta'([K,L],0) &= \delta(K,0) \cup \delta(L,0) \equiv \{I\} \cup \{I\} \equiv [I]\\ \delta'([K,L],1) &= \delta(K,1) \cup \delta(L,1) \equiv \{K\} \cup \{L\} \equiv [K,L]\end{aligned}$$

Y no se ha añadido ningún nuevo estado. Ahora podemos construir el nuevo autómata:



d) Construimos la gramática:

$$[J] \rightarrow 0[I] \mid 1[I]$$

$$[I] \rightarrow 1[I] \mid 1[K,L] \mid \epsilon$$

$$[K,L] \rightarrow 0[I] \mid 1[K,L]$$

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **2 de mayo de 2019**

Nombre: ____________________
Código: ____________________

1. Thompson eres tan predecible .... 20 puntos

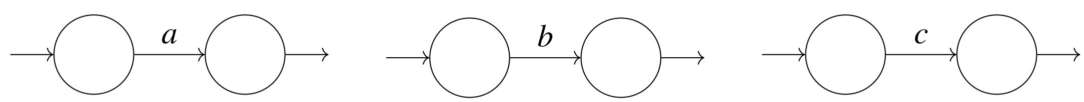
Dada la expresión regular $r_1 = (a \mid b)c^*$. Obtener:

a) (**15 puntos**) El autómata no determinista (N_1) utilizando el método estructural de Thompson, tal que $L(r_1) \equiv L(N_1)$.

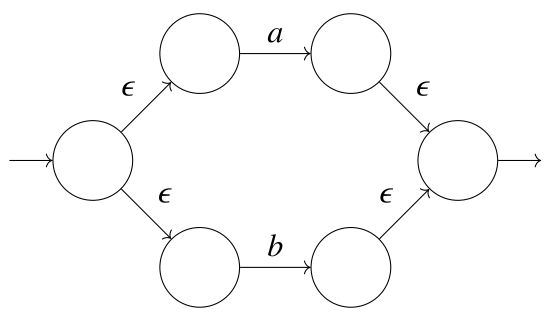
b) (**5 puntos**). Construir la gramática por la derecha (N_1) del autómata N_1.

Solución:

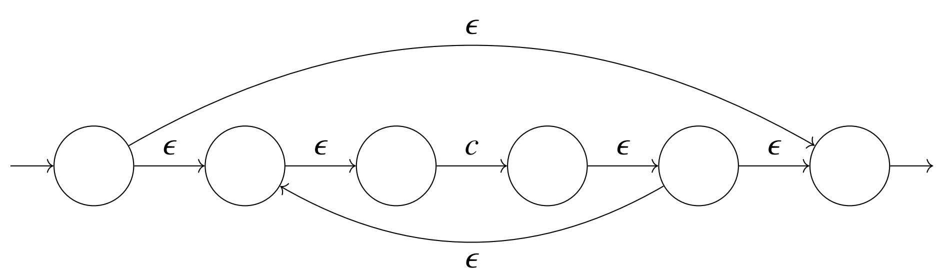
a) Construimos los autómatas correspondientes a las versiones más simples, para a, b y c:



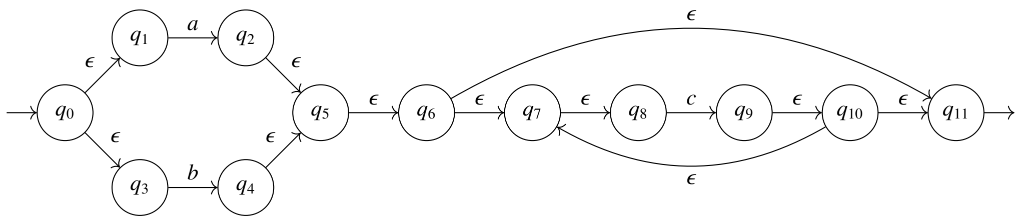
Luego construimos el autómata para la subexpresión: $a \mid b$



Hacemos la expresion c^*:



Ahora conectamos las dos subexpresiones: $a \mid b$ y c^* en la expresion $r_1 = (a \mid b)c^*$.



b) Ahora obtenemos la gramática correspondiente:

$$\begin{array}{rcl}
q_0 & \rightarrow & q_1 \mid q_3 \\
q_1 & \rightarrow & aq_2 \\
q_2 & \rightarrow & q_5 \\
q_3 & \rightarrow & bq_3 \\
q_4 & \rightarrow & q_5 \\
q_5 & \rightarrow & q_6 \\
q_6 & \rightarrow & q_7 \mid q_{11} \\
q_7 & \rightarrow & q_8 \\
q_8 & \rightarrow & cq_9 \\
q_9 & \rightarrow & q_{10} \\
q_{10} & \rightarrow & q_7 \mid q_{11}
\end{array}$$

2. GMY tu nombre me suena a pirámide . . . . 20 puntos

Dada la expresión regular $r_2 = (a^*(a \mid b)b^*)$. Obtener utilizando el método GMY el autómata finito no-determinista N_2 tal que $L(r_2) \equiv L(N_2)$. **Nota:** Los conjuntos pueden ser hallados de forma informal.

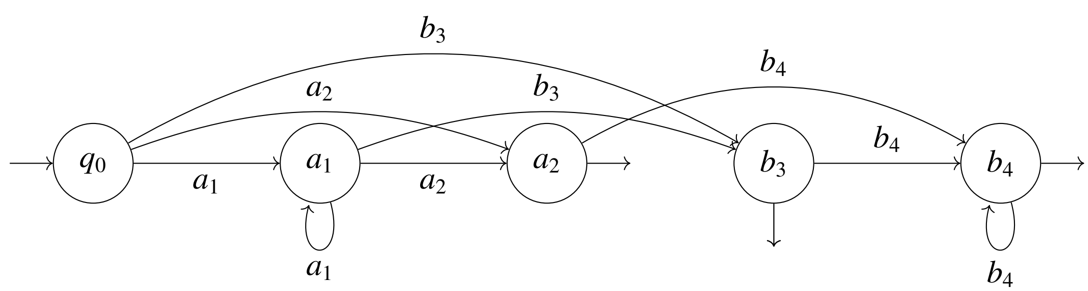
Solución:

En primer lugar la expresión regular no es lineal y la convertimos en lineal $r_2' = (a_1^*(a_2 \mid b_3)b_4^*)$.

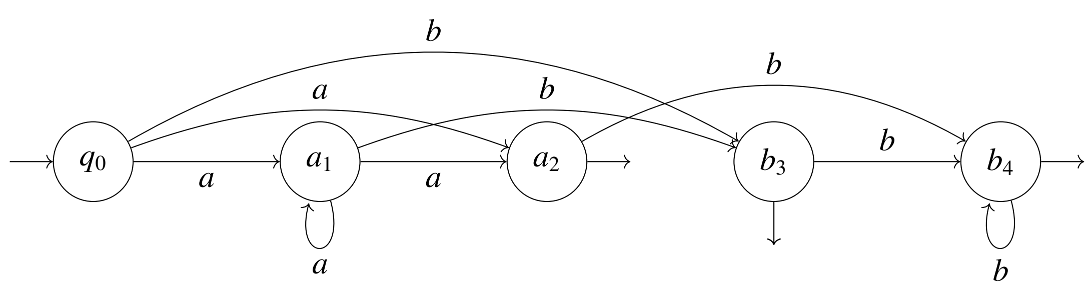
Ahora computamos los conjuntos:

$$\begin{aligned} Init(r_2') &= \{a_1, a_2, b_3\} \\ Fin(r_2') &= \{a_2, b_3, b_4\} \\ Dig(r_2') &= \{a_1a_1, a_1a_2, a_1b_3, a_2b_4, b_3b_4, b_4b_4\} \end{aligned}$$

Con esto conjunto podemos construir el autómata utilizando el procedimiento de GMY.

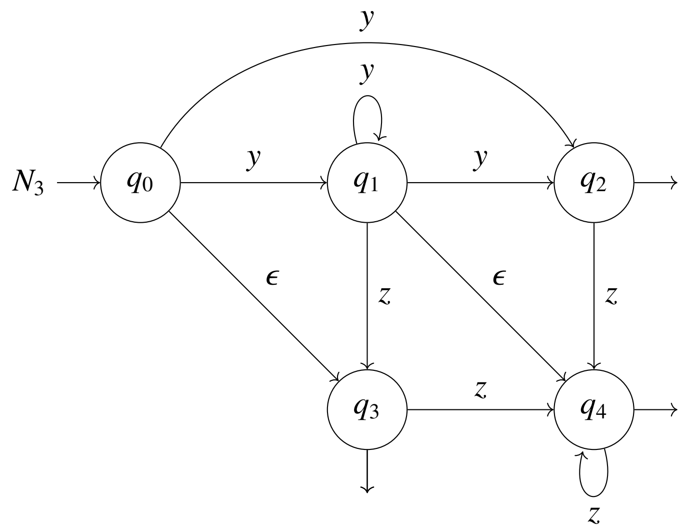


El último paso es borrar los índices de las transiciones.



3. Berry-Sethi sirve para todo .... 30 puntos

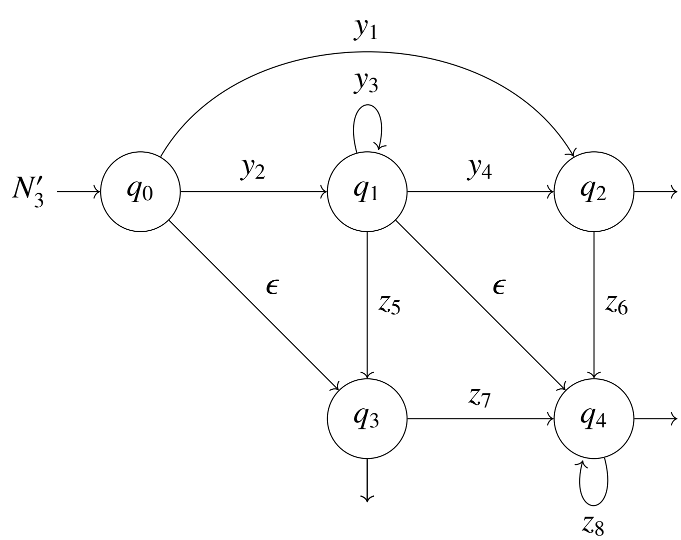
En la siguiente figura se representa un autómata finito no-determinista (N_3):



Utilizando el método de Berry-Sethi obtener un autómata finito determinista M_3 tal que $L(M_3) \equiv L(N_3)$. **Nota:** los conjuntos puede ser hallados de forma informal.

Solución:

Enumeramos las transiciones del autómata y se obtiene:



Ahora computamos los conjuntos $Init(N'_3 \dashv)$, $Fin(N'_3)$, $Fol(a)\ \forall a \in \Sigma$:

$$
\begin{array}{rcl}
Init(N_3' \dashv) & = & \{y_1, y_2, z_7, \dashv\} \\
Fin(N_3') & = & \{y_1, y_2, y_3, y_4, z_5, z_6, z_7, z_8\} \\
Fol(y_1) & = & \{z_6, \dashv\} \\
Fol(y_2) & = & \{y_3, y_4, z_5, z_8, \dashv\} \\
Fol(y_3) & = & \{y_3, y_4, z_5, z_8, \dashv\} \\
Fol(y_4) & = & \{z_6, \dashv\} \\
Fol(z_5) & = & \{z_7, \dashv\} \\
Fol(z_6) & = & \{z_8, \dashv\} \\
Fol(z_7) & = & \{z_8, \dashv\} \\
Fol(z_8) & = & \{z_8, \dashv\}
\end{array}
$$

Tomando como el estado de inicio $\{y_1, y_2, z_7, \dashv\}$. Iniciamos a construir el autómata.

$$
\begin{array}{rcl}
\delta(\{y_1, y_2, z_7, \dashv\}, y) & = & Fol(y_1) \cup Fol(y_2) \\
& = & \{z_6, \dashv\} \cup \{y_3, y_4, z_5, z_8, \dashv\} \\
& = & \{y_3, y_4, z_5, z_6, z_8, \dashv\}
\end{array}
$$

Este es un nuevo estado.

$$
\begin{array}{rcl}
\delta(\{y_1, y_2, z_7, \dashv\}, z) & = & Fol(z_7) \\
& = & \{z_8, \dashv\}
\end{array}
$$

Este es un nuevo estado. Tomamos el estado $\{y_3, y_4, z_5, z_6, z_8, \dashv\}$ y vemos que nuevos estados produce.

$$
\begin{array}{rcl}
\delta(\{y_3, y_4, z_5, z_6, z_8, \dashv\}, y) & = & Fol(y_3) \cup Fol(y_4) \\
& = & \{y_3, y_4, z_5, z_8, \dashv\} \cup \{z_6, \dashv\} \\
& = & \{y_3, y_4, z_5, z_6, z_8, \dashv\}
\end{array}
$$

Este ya es un estado conocido.

$$
\begin{array}{rcl}
\delta(\{y_3, y_4, z_5, z_6, z_8, \dashv\}, z) & = & Fol(z_5) \cup Fol(z_6) \cup Fol(z_8) \\
& = & \{z_7, \dashv\} \cup \{z_8, \dashv\} \cup \{z_8, \dashv\} \\
& = & \{z_7, z_8, \dashv\}
\end{array}
$$

Este es un nuevo estado

Ahora tomamos el estado $\{z_8, \dashv\}$.

$$
\begin{array}{rcl}
\delta(\{z_8, \dashv\}, z) & = & Fol(z_8) \\
& = & \{z_8, \dashv\}
\end{array}
$$

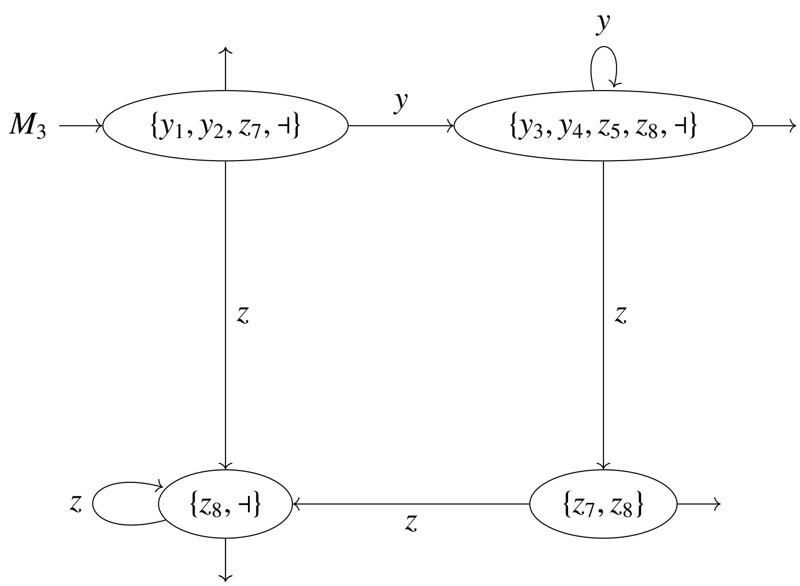
Es un estado conocido.

Ahora tomamos el estado $\{z_7, z_8, \dashv\}$.

$$\begin{aligned}\delta(\{z_7, z_8, \dashv\}, z) &= Fol(z_7) \cup Fol(z_8)\\ &= \{z_8, \dashv\} \cup \{z_8, \dashv\}\\ &= \{z_8, \dashv\}\end{aligned}$$

Un estado conocido. No se han generado más estado por lo tanto obtenemos el siguiente autómata de estado finito.

El autómata M_3 correspondiente:



4. Los autómatas no olvidan . . . . 30 puntos

Obtener el autómata de pila ADP_4 que reconozca cadenas del lenguaje L_4. Tal que $L(ADP_4) \equiv L_4$. Donde:

$$L_4 = \{x \in \{a, b, c\}^* \mid \; |x|_a < |x|_b\}$$

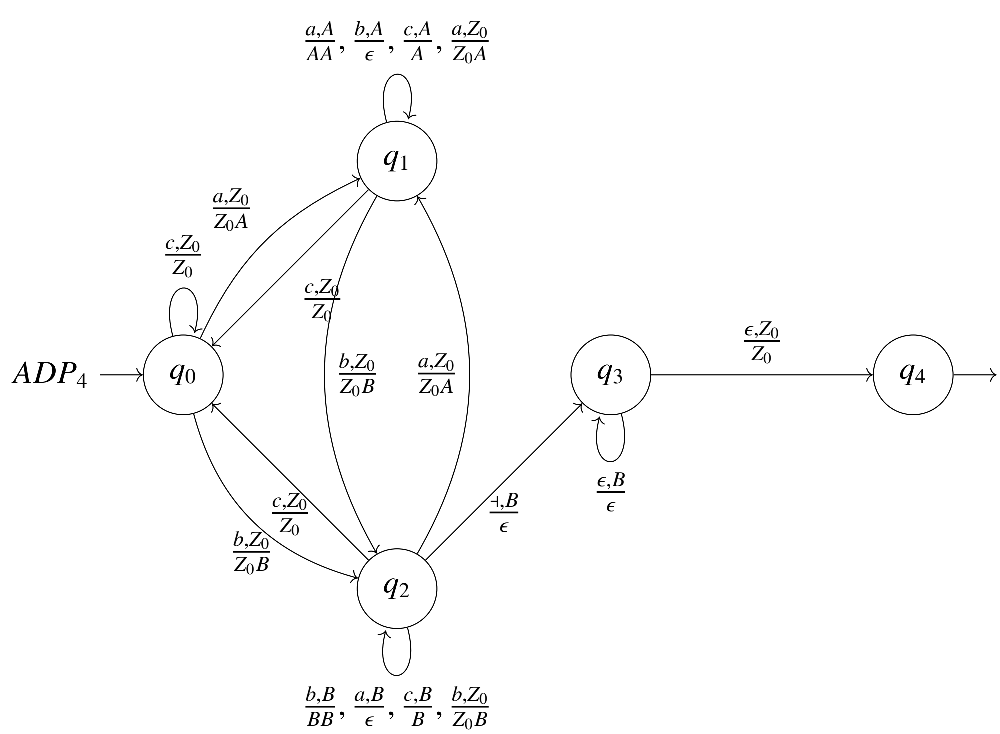
El autómata de pila ADP_4 debe vacía completamente la pila únicamente con las cadenas válidas.

a) (**21 puntos**): Obtener el autómata de pila (ADP_4). Tal que $L(ADP_4) \equiv L_4$.

b) (**9 puntos:**): Probar si dichas cadenas se aceptan o se rechazan por su implementación del autómata: *abac*, *acbb*, *cbab*.

Solución:

a) Para construir el autómata en debemos tener en cuenta que el autómata debe garantizar que ha contado más bs que as. Entonces, vamos a tener un estado inicial q_0 donde tenemos el mismo número de as y de bs. En un momento determinado puede llegar una a que hace que se aumente el número de as, por lo tanto esas las vamos a contar en el estado q_1, este nos indicara que tendremos más as que b. En dicho estado (q_1) podemos segur contando más as, pero también puede surgir bs que decrementan el conteo de as. Hasta que las b sean superiores en cuyo caso pasará al estado q_2 donde se cuentas más bs que as.



b) 1) Reconocer la cadena *abac*:

$$\begin{array}{rcl} \langle q_0, abac \dashv, Z_0\rangle & \longmapsto & \langle q_1, bac \dashv, Z_0A\rangle \\ & \longmapsto & \langle q_1, ac \dashv, Z_0\rangle \\ & \longmapsto & \langle q_1, c \dashv, Z_0A\rangle \\ & \longmapsto & \langle q_1, \dashv, Z_0A\rangle \\ & \longmapsto & \textbf{Error: } \text{No hay movimiento} \end{array}$$

La cadena *abac* es *rechazada*.

2) Reconocer la cadena *acbb*:

$$\begin{array}{rcl} \langle q_0, acbb \dashv, Z_0\rangle & \longmapsto & \langle q_1, cbb \dashv, Z_0A\rangle \\ & \longmapsto & \langle q_1, bb \dashv, Z_0A\rangle \\ & \longmapsto & \langle q_1, b \dashv, Z_0\rangle \\ & \longmapsto & \langle q_2, \dashv, Z_0B\rangle \\ & \longmapsto & \langle q_3, \epsilon, Z_0\rangle \\ & \longmapsto & \langle q_4, \epsilon, Z_0\rangle \end{array}$$

La cadena *acbb* es *aceptada*.

3) Reconocer la cadena *cbab*:

$$\begin{array}{rcl} \langle q_0, cbab \dashv, Z_0\rangle & \longmapsto & \langle q_0, bab \dashv, Z_0\rangle \\ & \longmapsto & \langle q_2, ab \dashv, Z_0B\rangle \\ & \longmapsto & \langle q_2, b \dashv, Z_0\rangle \\ & \longmapsto & \langle q_2, \dashv, Z_0B\rangle \\ & \longmapsto & \langle q_3, \epsilon, Z_0\rangle \\ & \longmapsto & \langle q_4, \epsilon, Z_0\rangle \end{array}$$

La cadena *cbab* es *aceptada*.

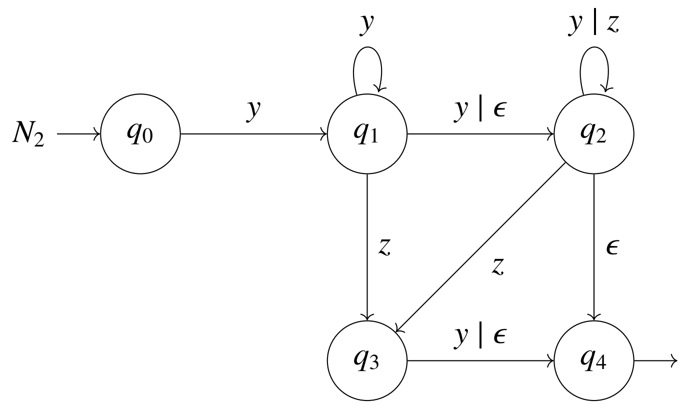
Nombre: ______________________________
Código: ______________________________

1. Ladies and Gentelmen: Berry and Sethi .... 20 puntos

Dada la expresión regular $r_1 = yy^*zy$. Obtener a través de Berry-Sethi el autómata finito determinista M_1, tal que $L(r_1) \equiv L(M_1)$.

2. Berry-Sethi ven en nuestra ayuda contra Thompson y sus potencias .... 30 puntos

En la siguiente figura se representa un autómata finito no-determinista (N_2):



Utilizando el método de Berry-Sethi obtener un autómata finito determinista M_2 tal que $L(M_2) \equiv L(N_2)$. **Nota:** los conjuntos puede ser hallados de forma informal.

3. $ASF \subseteq ADP$ .... 20 puntos

Obtener un autómata de pila, posiblemente no determinista, que M_3 que acepte el lenguaje de la expresión r_1. De tal forma que $L(M_3) \equiv L(r_1)$ y el número de estados $\left|Q_{M_3}\right| = 2$.

Si se observa el punto1 1 el autómata aquí obtenido tiene 4 estados. La memoria permite que se puedan economizar estados y esto permite que los valores en la pila represente situaciones especiales.

4. Un conjunto de cosas dispuestas sobre otras .... 30 puntos

Obtener un autómata de pila determinista, $ADPD_4$ de tal forma $L(ADPD_4) \equiv L_4$. Donde:

$$L_4 = \{w \in \{x, y, z\}^* \mid |w|_x \neq |w|_z\}$$

El autómata de pila $ADPD_4$ debe dejar la pila vacía antes de aceptar una cadena.

1. (**21 puntos**): Obtener el autómata de pila ($ADPD_4$). Tal que $L(ADPD_4) \equiv L_4$.
2. (**9 puntos:**): Probar si dichas cadenas se aceptan o se rechazan por su implementación del autómata: $xzyx$, $zyzx$, $zxxz$.

1. yy^*zy. $r_1 = yy^*zy$ $r_1' = y_1 y_2^* z_3 y_4$

$Ini(r_1') = \{y_1\}$

$Fin(r_1') = \{y_4\}$

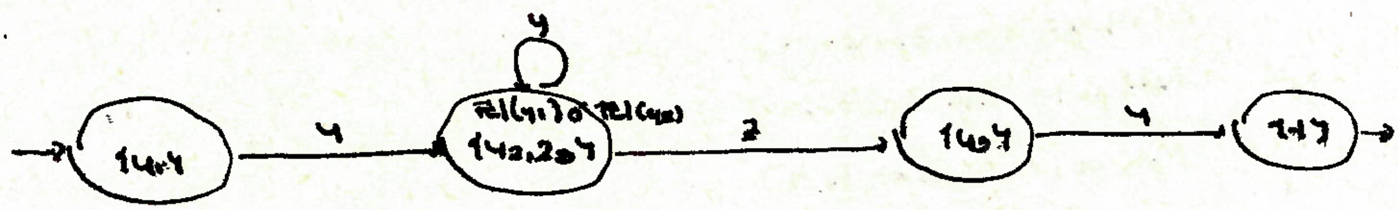
$Dig(r_1') = \{ y_1y_2, y_1z_3,$
$y_2y_2, y_2z_3,$
$z_3y_4\}$

$Fol(y_1) = \{y_2, z_3\}$

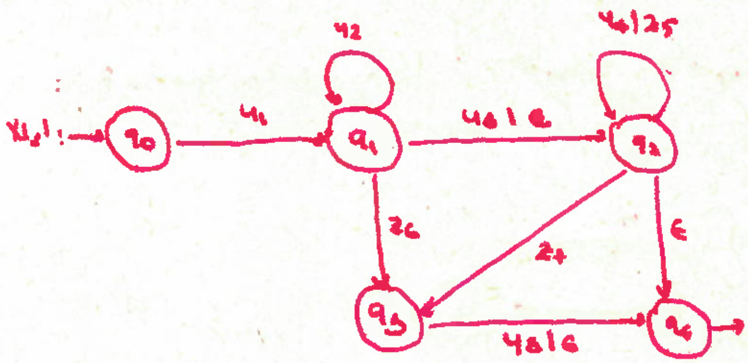
$Fol(y_2) = \{y_2, z_3\}$

$Fol(z_3) = \{y_4\}$

$Fol(y_4) = \{\}$



7.



$Ini(y^! \dashv) = \{y_1\}$

$Fin(y^!) = \{y_1, y_2, y_3, y_4, z_5, z_6, z_7, y_8\}$

$Fol(y_1) = \{y_2, y_3, y_4, z_5,$ ~~[illegible]~~ $z_6, z_7, \dashv\}$

$Fol(y_2) = \{y_2, y_3, y_4, z_5, z_6, z_7, \dashv\}$

$Fol(y_3) = \{y_4, z_5, z_7, \dashv\}$

$Fol(y_4) = \{y_4, z_5, z_7, \dashv\}$

$Fol(z_5) = \{y_4, z_5, z_7, \dashv\}$

$Fol(z_6) = \{y_8, \dashv\}$

$Fol(z_7) = \{y_8, \dashv\}$

$Fol(y_8) = \{\dashv\}$

$q_0 = \{y_1\}$

$\delta(q_0, y) = Fol(y_1) = \{y_2, y_3, y_4, z_5, z_6, z_7, \dashv\} = q_1$

$\delta(q_1, y) = Fol(y_2) \cup Fol(y_3) \cup Fol(y_4) = \{y_2, y_3, y_4, z_5, z_6, z_7, \dashv\} \cup \{y_4, z_5, z_7, \dashv\}$
$\cup \{y_4, z_5, z_7, \dashv\} = \{y_2, y_3, y_4, z_5, z_6, z_7, \dashv\}$
$= q_1$

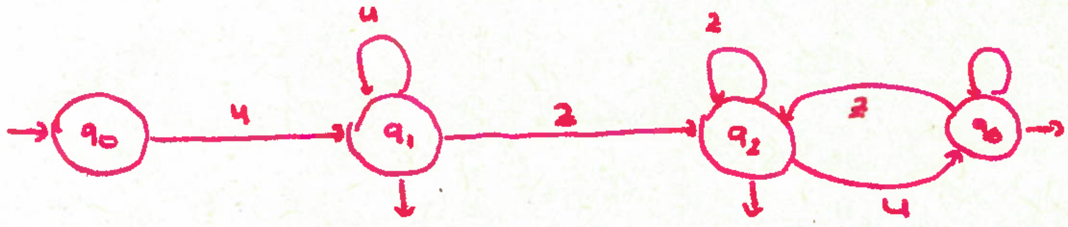
$\delta(q_1, z) = Fol(z_5) \cup Fol(z_6) \cup Fol(z_7) = \{y_4, z_5, z_7, \dashv\} \cup \{y_8, \dashv\} \cup \{y_8, \dashv\}$
$= \{y_4, z_5, z_7, y_8, \dashv\} = q_2$

$\delta(q_2, y) = Fol(y_4) \cup Fol(y_8) = \{y_4, z_5, z_7, \dashv\} \cup \{\dashv\} = \{y_4, z_5, z_7, \dashv\} = q_3$

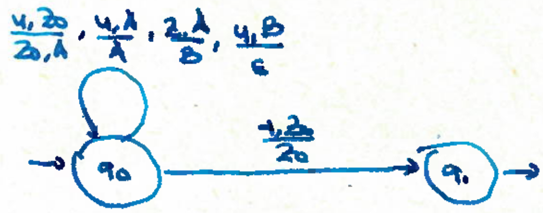
$\delta(q_2, z) = Fol(z_5) \cup Fol(z_7) = \{y_4, z_5, z_7, \dashv\} \cup \{y_8, \dashv\} = \{y_4, z_5, z_7, y_8, \dashv\} = q_2$

$\delta(q_3, u) = Fol(u_6) = \{u_6, 2_5, 2_7, \dashv\} = q_3$

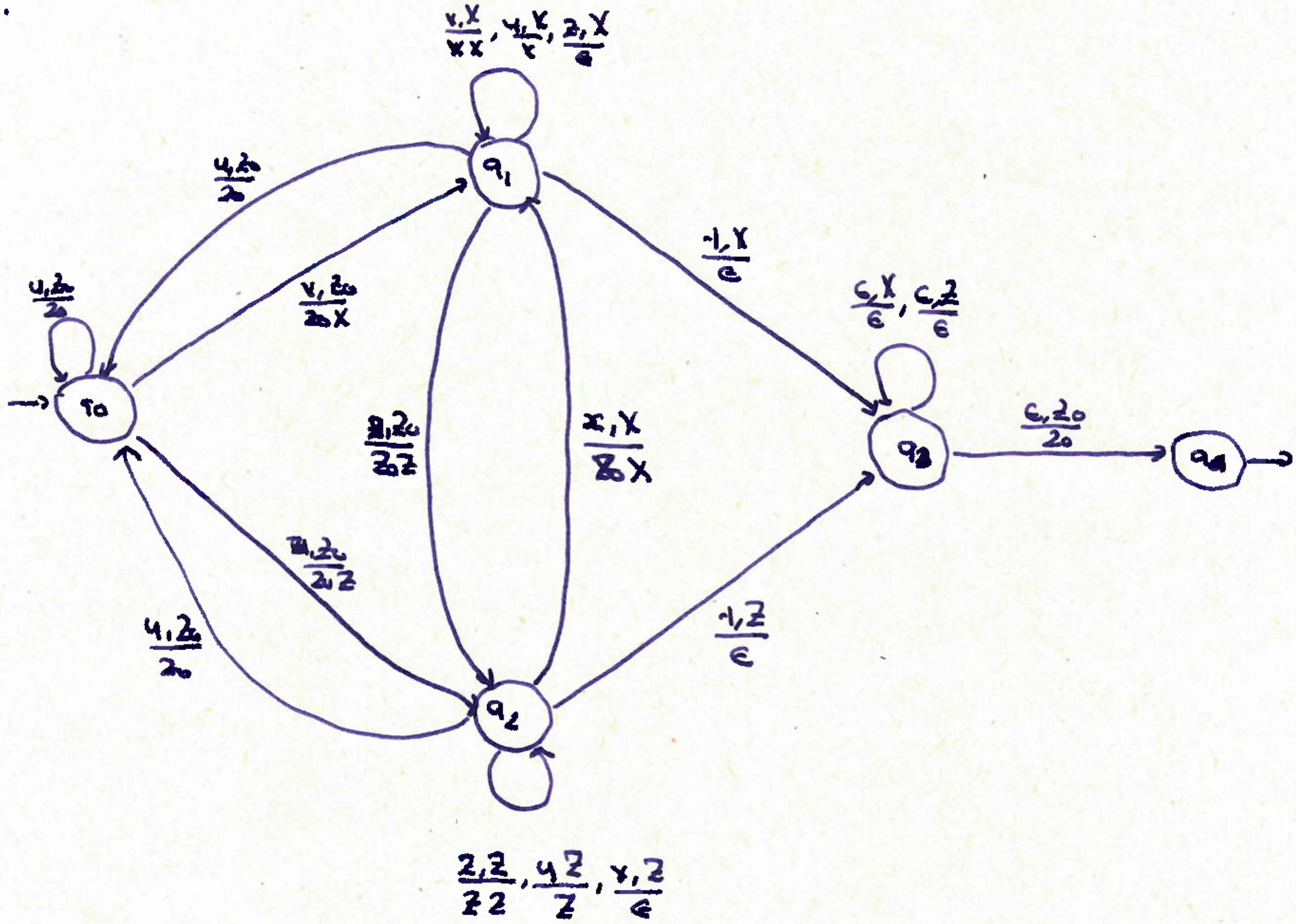
$\delta(q_3, 2) = Fol(2_5) \cup Fol(2_7) = \{u_6, 2_5, 2_7, \dashv\} \cup \{u_8, \dashv\} = \{u_6, 2_5, 2_7, u_8, \dashv\} = q_2$



3.



4.



a) $xzyx$

$\langle q_0, xzyx\dashv, Z_0\rangle \longmapsto \langle q_1, zyx\dashv, Z_0X\rangle$

$\longmapsto \langle q_1, yx\dashv, Z_0\rangle$

$\longmapsto \langle q_0, x\dashv, Z_0\rangle$

$\longmapsto \langle q_1, \dashv, Z_0X\rangle$

$\longmapsto \langle q_3, \epsilon, Z_0\rangle$

$\longmapsto \langle q_4, \epsilon, Z_0\rangle$ Aceptar.

b) $zyzx$

$\langle q_0, zyzx\dashv, Z_0\rangle \longmapsto \langle q_2, yzx\dashv, Z_0Z\rangle$

$\longmapsto \langle q_2, zx\dashv, Z_0Z\rangle$

$\longmapsto \langle q_2, x\dashv, Z_0ZZ\rangle$

$\longmapsto \langle q_2, \dashv, Z_0Z\rangle$

$\longmapsto \langle q_3, \epsilon, Z_0\rangle$

$\longmapsto \langle q_4, \epsilon, Z_0\rangle$ Aceptar.

c) $zxxz$

$\langle q_0, zxxz\dashv, Z_0\rangle \longmapsto \langle q_2, xxz\dashv, Z_0Z\rangle$

$\longmapsto \langle q_2, xz\dashv, Z_0\rangle$

$\longmapsto \langle q_1, z\dashv, Z_0X\rangle$

$\longmapsto \langle q_1, \dashv, Z_0\rangle$

No hay movimientos posibles.

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Primer parcial **DIS**
Juan Francisco Cardona McCormick **15 de agosto de 2019**

Nombre: ____________________
Código: ____________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Expresiones regulares extendidas**

Dado el siguiente alfabeto $\Sigma = \{a, b\}$, construya una expresión regular r_1 que permite cualquier cadena de Σ^* de tal forma que las cadenas no pueden comenzar *ab*, ni terminar con *ba*. Por ejemplo las siguientes cadenas son válida: *a*, *b*, *aab*, *baa*, *bbaa*. Las siguiente cadenas no son válidas *aba*, *abba*, *ababa*, *abbba*.

(20 %) 2. **Transformación de gramáticas - Eliminación de los nulos**

La siguiente gramática G_2, tiene reglas nulas, encuentre la gramática G_2' de tal forma que el $L(G_2) \equiv L(G_2')$ y sean estructuralmente equivalentes.

$$G_2 = (V = \{S, A, B\},\ \Sigma = \{x, y, z\}, P = \{S \rightarrow AzB,\ A \rightarrow\ xA \mid \epsilon,\ B \rightarrow\ yB \mid \epsilon\},\ S)$$

(30 %) 3. **Gramáticas independiente de contexto**

La siguiente gramática independiente de contexto G_3 es una gramática ambigua:

$$G_3 = (V = \{S, A, B, C\}, \Sigma = \{a, b, e, f, i, s\}, P = \{S \rightarrow A \mid B \mid C,\ A \rightarrow aie, B \rightarrow fbSSf, C \rightarrow\ SsS\},\ S)$$

a) Muestre que G_3 es ambigua.

b) Defina una nueva gramática G_3' tal que $L(G_3')$ no es ambigua y que $L(G_3) \equiv L(G_3')$

(20 %) 4. **Gramática independiente de contexto y expresiones regulares**

La siguiente gramática G_4 genera un lenguaje regular, pese a que la gramática independiente de contexto no es regular en sí misma, encuentre una expresión regular r_4 de tal forma que $L(G_4) \equiv L(r_4)$.

$$G_4 = (V = \{S\},\ \Sigma = \{a, b\}, P = \{S \rightarrow SSS \mid a \mid ab\}, S)$$

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Primer parcial **DIS**
Juan Francisco Cardona McCormick **16 de agosto de 2019**

Nombre: ______________________________
Código: ______________________________

Nota: Todo los puntos se resuelven aplicando una definición formal deben ser desarrollados completamente utilizando el correspodiente método.

(30 %) 1. **Expresiones regulares extendidas**
Dado el siguiente alfabeto $\Sigma = \{w, x\}$, construya una expresión regular r_1 que permite cualquier cadena de Σ^* de tal forma que las cadenas tener las cadenas xx: ϵ, x, xw, wx, ww, wxw, $\ldots$. Las siguiente cadenas no son válidas xx, wxx, xxw, $\ldots$.

(20 %) 2. **Expresiones regulares ambiguas**
La expresión regular $r_2 = (a+)^*a(+a)^*$:

a) Probar que la expresión regular anterior es ambigua.

b) Reescribir la expresión regular r_2' de tal forma $L(r_2) \equiv L(r_2')$ y que r_2' no es ambigua.

(30 %) 3. **Gramáticas independiente de contexto**
La siguiente gramática independiente de contexto G_3 es una gramática ambigua:

$$G_3 = (V = \{S, A, B, C\}, \Sigma = \{a, b\}, P = \{S \to ABC,\ A \to aA \mid \epsilon, B \to bB \mid \epsilon, C \to A\},\ S)$$

a) Muestre que G_3 es ambigua.

b) Defina una nueva gramática G_3' tal que $L(G_3')$ no es ambigua y que $L(G_3) \equiv L(G_3')$

(20 %) 4. **Gramática independiente de contexto y expresiones regulares**

La siguiente gramática G_4 genera un lenguaje regular, pese a que la gramática independiente de contexto no es regular en sí misma, encuentre una expresión regular r_4 de tal forma que $L(G_4) \equiv L(r_4)$.

$$G_4 = (V = \{S, X, Y\},\ \Sigma = \{x, y\}, P = \{S \to XxyY,\ X \to xX \mid yX \mid \epsilon, Y \to Yxy \mid Yy \mid xy \mid y\}, S)$$

Nombre: ______________________________
Código: ______________________________

Nota: El parcial tiene dos puntos, seleccione uno de los dos puntos y resuélvalo.

1. (sic)"Achiquitar el auto q'da, si se pudo" .... 100 puntos

De la siguiente gramática lineal por la derecha GLD_1.

$$
\begin{array}{rcl}
q_0 & \rightarrow & aq_3 \mid bq_5 \\
q_1 & \rightarrow & aq_0 \mid bq_6 \\
q_2 & \rightarrow & aq_1 \mid bq_3 \\
q_3 & \rightarrow & aq_5 \mid bq_4 \\
q_4 & \rightarrow & aq_6 \mid bq_4 \\
q_5 & \rightarrow & aq_2 \mid bq_5 \mid \epsilon \\
q_6 & \rightarrow & bq_6 \mid aq_2 \mid \epsilon
\end{array}
$$

a) (20 puntos) Obtener el autómata determininista M_1 tal que $L(GLD_1) \equiv L(M_1)$.

b) (60 puntos) Obtener un autómata determinista M_1' que sea el autómata mínimo de M_1 tal que $L(M_1') \equiv L(M_1)$.

c) (20 puntos) Utilizando a BMC obtener la expresión regular r_1 tal que $L(r_1) \equiv L(M_1')$.

2. (sic)”Marconik me la montó” .... 100 puntos

La siguiente es una gramática lineal por la izquierda GLI_2.

$$
\begin{array}{rcl}
S & \rightarrow & A \mid F \\
A & \rightarrow & \epsilon \mid Ca \\
B & \rightarrow & Aa \\
C & \rightarrow & Ca \mid Bb \\
D & \rightarrow & A \mid Bd \mid Fb \\
E & \rightarrow & Eb \mid Db \\
F & \rightarrow & Eb
\end{array}
$$

a) (30 puntos) Obtener la gramática lineal por la izquierda GLD_2 tal que $L(GLI_2) \equiv L(GLD_2)$.

b) (20 puntos) Obtener el autómata no determinista N_2 correspondiente a la gramática GLD_2 de tal forma que el lenguaje que reconoce $L(M_2) \equiv L(GLD_2)$.

c) (40 puntos) Obtener el autómata determinista M_2 de tal forma $L(N_2) \equiv L(M_2)$.

d) (10 puntos) La gramática línea por la derecha GLI'_2 del automáta determinista M_2.

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Segundo parcial **DIS**
Juan Francisco Cardona McCormick **20 de septiembre de 2019**

Nombre: ____________________

Código: ____________________

Nota: El parcial tiene dos puntos, seleccione uno de los dos puntos y resuélvalo.

1. Escabeche de cangrejo . . . . 100 puntos

La siguiente gramática estrictamente lineal por la izquierda GLI_1

$$\begin{array}{lcl} A & \rightarrow & B \mid Cb \\ B & \rightarrow & Bb \mid E \\ C & \rightarrow & Da \mid E \\ D & \rightarrow & Da \mid Cb \\ E & \rightarrow & \epsilon \end{array}$$

a) (40 puntos) Obtener el autómata posiblemente no determinista correspondiente N_1 tal que $L(GLI_1) \equiv L(N_1)$.

b) (10 puntos) Obtener la gramática lineal por la derecha GLD_1 tal que $L(GLI_1) \equiv L(GLD_1)$.

c) (50 puntos) Obtener un autómata M_1 determinista de N_1 tal que $L(M_1) \equiv L(N_1)$.

2. Lenguas de cangrejo al vino . . . . 100 puntos

La siguiente gramática estrictamente línea por la izquierda GLI_2:

$$
\begin{array}{rcl}
A & \rightarrow & B \mid D \mid E \\
B & \rightarrow & Ba \mid Fa \mid Da \\
C & \rightarrow & Bb \mid Fb \mid Db \\
D & \rightarrow & Ca \mid Ga \mid Ea \\
E & \rightarrow & Eb \mid Gb \mid Cb \\
F & \rightarrow & Ha \\
G & \rightarrow & Hb \\
H & \rightarrow & \epsilon
\end{array}
$$

a) (30 puntos) Obtener el autómata determinista M_2 tal que $L(M_2) \equiv L(GLI_2)$.

b) (10 puntos) Obtener la gramática línea por la derecha GLD_2 de M_2 tal que $L(M_2)$ *equiv* $L(GLD_2)$.

c) (50 puntos) Obtener un autómata determinista M'_2 que sea el autómata mínimo de M_2 tal que $L(M'_2) \equiv L(M_2)$.

d) (20 puntos) Utilizando a BMC obtener la expresión regular r_2 tal que $L(r_2) \equiv L(M'_2)$.

ST0270 Lenguajes formales y compiladores - 031 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **29 de Octubre de 2019**

Nombre: ____

Código: ____

1. Three friends: Thompson, Berry and Sethi, go to Divercity .... 25 puntos

Dada la expresión regular $r_1 = a^*a$. Obtener:

a) (**10 puntos**) El autómata no determinista (N_1) utilizando el método estructural de Thompson de tal forma que $L(r_1) \equiv L(N_1)$.

b) (**15 puntos**). Transformar el autómata no determinista (N_1) en un autómata determinista M_1 por medio del algoritmo de Berry-Sethi de tal forma que $L(N_1) \equiv L(M_1)$

2. Divercity was invaded by crabs .... 30 puntos

Dada las expresiones regulares $r_2' = a^*b$, $r_2'' = a^*b^*$ del alfabeto $\Sigma = \{a, b\}$, encontrar la expresión. $r_2''' = r_2' \setminus r_2''$. Utilizando cualquiera de los métodos vistos: Thompson, GMY, Berry-Sethi, cómputo del conjunto de potencias y eliminación de reglas de copia:

3. A Crab's grammar .... 20 puntos

Dada la gramática independiente de contexto:

$$G_3 = (V = \{S, A, B\}, \Sigma = \{w, x, y, z\}, P = \{S \to wAx \mid yBz, A \to wSx \mid \epsilon, B \to ySz \mid yz\}, S)$$

Obtener:

a) (**15 puntos**) El autómata de pila posiblemente no determinista N_3 utilizando el método de transformación de gramática independiente de contexto a un autómata de pila. De tal forma $L(G_3) \equiv L(N_3)$.

b) (**5 puntos**). Utilizando el anterior autómata de pila N_3 probar que la cadena $wwyzxx \in L(G_3)$.

4. A Crab zombie attacks Divercity .... 25 puntos

Obtener el autómata de pila posiblemente N_4 no determinista que reconoce el siguiente lenguaje:

$$L_4 \equiv \{0^i1^j2^k \mid j = i \text{ ó } j = k\}$$

Tal que $L(N_4) \equiv L_4$.

ST0270 Lenguajes formales y compiladores - 032 **Universidad EAFIT**
Tercer parcial **DIS**
Juan Francisco Cardona McCormick **25 de Octubre de 2019**

Nombre: ____

Código: ____

1. The watchcrabs .... 25 puntos

Dada la expresión regular $r_1 = b^*ab$. Obtener:

a) (**10 puntos**) El autómata no determinista (N_1) utilizando el método estructural de Thompson de tal forma que $L(r_1) \equiv L(N_1)$.

b) (**15 puntos**). Transformar el autómata no determinista (N_1) en un autómata determinista M_1 por medio del algoritmo de Berry-Sethi de tal forma que $L(N_1) \equiv L(M_1)$

2. Crab's invasion .... 30 puntos

Dada la expresión regular $r_2 = ba^*$ del alfabeto $\Sigma = \{a, b\}$, encontrar la expresión. $r_2' = \neg r_2$ de tal forma que $L(r_2) \cap L(r_2') = \emptyset$. Utilizando cualquiera de los métodos vistos: Thompson, GMY, Berry-Sethi, cómputo del conjunto de potencias y eliminación de reglas de copia:

3. Crab: Covenant .... 20 puntos

Dada la gramática independiente de contexto:

$$G_3 = (V = \{S, A, B\}, \Sigma = \{a, b, c, d\}, P = \{S \rightarrow aAb \mid cBd, A \rightarrow aSb \mid ab, B \rightarrow cSd \mid \epsilon\}, S)$$

Obtener:

a) (**15 puntos**) El autómata de pila posiblemente no determinista N_3 utilizando el método de transformación de gramática independiente de contexto a un autómata de pila. De tal forma $L(G_3) \equiv L(N_3)$.

b) (**5 puntos**). Utilizando el anterior autómata de pila N_3 probar que la cadena $aacdbb \in L(G_3)$.

4. The crab speech .... 25 puntos

Obtener el autómata de pila posiblemente N_4 no determinista que reconoce el siguiente lenguaje:

$$L_4 \equiv \{0^n x \mid n \geq 0, x \in \{0, 1\}^* \text{ y } |x| \leq n\}$$

Tal que $L(N_4) \equiv L_4$.