Teoría de la Computación 2
Solución del Parcial 1

25 de febrero de 2008 Prof: Rodrigo López

1. **[Punto de recuperación del quiz]** Sean $G_1 = (N_1, T_1, P_1, \Sigma_1)$ y $G_2 = (N_2, T_2, P_3, \Sigma_2)$ dos gramáticas tipo 2 (independientes del contexto). Describa cómo construir, a partir de G_1 y G_2, una gramática G, independiente del contexto, tal que $L(G) = L(G_1) \cup L(G_2)$.

   Solución: Se define una nueva gramática $G = (N_1 \cup N_2, T_1 \cup T_2, P, \Sigma)$ que contiene:

   - Todos los símbolos (terminales y no terminales) de las dos gramáticas originales.
   - Un nuevo símbolo distinguido Σ.
   - Un nuevo conjunto de producciones $P = P_1 \cup P_2 \cup \{\Sigma \to \Sigma_1, \Sigma \to \Sigma_2\}$

   Ahora, si $\omega \in L(G)$, su derivación es de la forma $\Sigma \Rightarrow \Sigma_1 \overset{*}{\Rightarrow} \omega$ (en cuyo caso también era derivable en G_1) o es de la forma $\Sigma \Rightarrow \Sigma_2 \overset{*}{\Rightarrow} \omega$ (en cuyo caso también era derivable en G_2)

2. Marque la opción correcta. Si G es una gramática tipo 2 y ambigua entonces existe una cadena $\alpha \in L(G)$ para la cual:

   ◯ Hay exactamente una derivación por la derecha y exactamente una derivación por la izquierda.

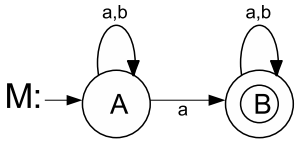
   ◯ No hay derivaciones por la izquierda pero hay 2 o más derivaciones por la derecha.

   ◯ No hay derivaciones por la derecha pero hay 2 o más derivaciones por la izquierda.

   ⊗ Hay más de una derivación por la izquierda.

   ◯ Hay exactamente dos derivaciones por la izquierda y exactamente dos derivaciones por la derecha.

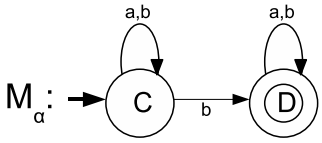
3. Considere el Aceptador de Estados Finitos (no determinístico) M y la expresión regular α de la figura siguiente.

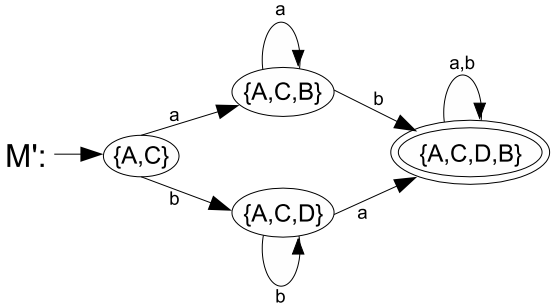


$$\alpha = (a \cup b)^* b (a \cup b)^*$$

   Llamaremos L_α al lenguaje definido por α y, como ha sido habitual, $L(M)$ será el lenguaje aceptado por M.

   a) Calcule una expresión regular para el lenguaje $L_\alpha \cap L(M)$
   Solución: Se puede usar el siguiente método.
   - Construir un aceptador M_α para el lenguaje L_α.
   - Apoyarse en el anterior y en M para construir un aceptador M' para el lenguaje $L_\alpha \cap L(M)$.
   - Calcular la expresión regular correspondiente.





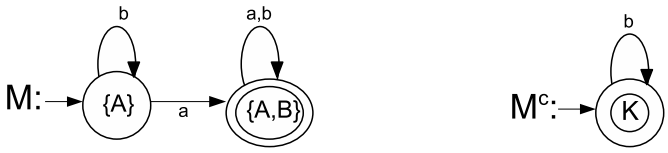
Se podrían plantear las ecuaciones simultáneas de conjuntos para calcular la expresión regular correspondiente a $L(M')$ pero es posible deducirla por simple inspección. En efecto, en M' solamente hay dos "recorridos" que conducen del estado inicial al final. Al primero corresponde la expresión $aa^*b(a \cup b)^*$ y al segundo corresponde la expresión $bb^*a(a \cup b)^*$. Es decir,

$$\boxed{L(M') = L_\alpha \cap L(M) = aa^*b(a \cup b)^* \cup bb^*a(a \cup b)^* = (aa^*b \cup bb^*a)(a \cup b)^*}$$

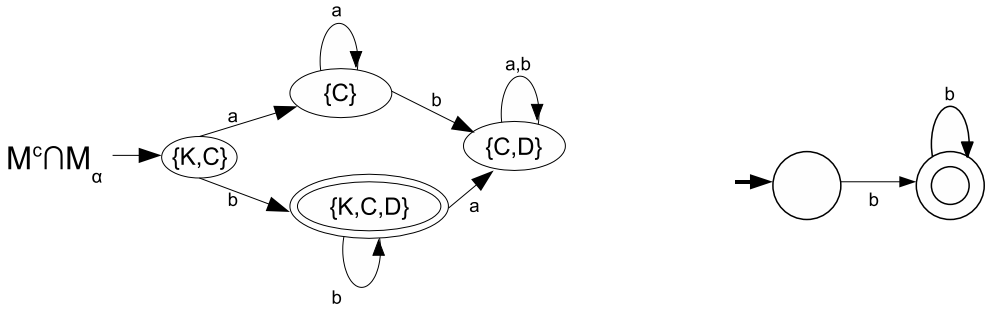
b) Calcule una expresión regular para el lenguaje $L_\alpha - L(M) = \{\omega \in (a \cup b)^* \mid \omega \in L_\alpha \text{ y } \omega \notin L(M)\}$

Solución: El lenguaje pedido corresponde a $L_\alpha \cap L(M)^c$ luego se puede,

- Calcular el aceptador para $L(M)^c$. Note que no basta con intercambiar estados finales por no finales porque M es no determinístico. La figura siguiente muestra la versión determinística de M y luego el aceptador M^c que acepta el lenguaje $L(M)^c$.



- Calcular la "intersección" de M^c con M_α. En la figura siguiente aparece la máquina calculada con el procedimiento estándar de intersección y al lado le versión reducida en la que se han eliminado (por simple inspección) los caminos que conducen a un estado de trampa.



- Calcular la expresión regular correspondiente al último aceptador.

$$\boxed{L_\alpha - L(M) = bb^*}$$

4. Sea $R \subseteq V^*$ un conjunto regular sobre un alfabeto V y sea α una cadena fija de V^*. Se define *la derivada de R con respecto a α* como el conjunto $R_\alpha = \{\beta \mid \alpha\beta \in R\}$. Demuestre que R_α es regular. Ayuda: Describa cómo, a partir de un aceptador para R, se puede construir un aceptador para R_α.

Solución: Sea $M = (Q, V, f, F, q_I)$ un aceptador de estados finitos determinístico tal que $R = L(M)$. Se estimula a M con la cadena α y se ve a qué estado se llega. Como M es determinístico ese estado es único. Ahora definimos una nueva máquina que usa ese estado como estado inicial. Las cadenas aceptadas a partir de ese estado son justamente las β tales que $\alpha\beta$ eran aceptadas por el aceptador original.

Más formalmente, definimos $M_\alpha = (Q, V, f, F, q_\alpha)$ en donde $q_I \stackrel{\alpha}{\Rightarrow} q_\alpha$.

5. Describa un algoritmo que permita decidir si un lenguaje regular es finito o infinito (puede usar cualquiera de los algoritmos discutidos en clase).

Solución: Si el lenguaje es regular debe existir un aceptador de estados finitos determinístico $M = (Q, S, f, F, q_I)$ que lo acepte. Suponga que hay n estados en Q. El algoritmo propuesto es el siguiente:

a) Construya el conjunto $J = \{\omega \in S^* \text{ tales que } n \leq \mid \omega \mid < 2n\}$.

b) Estimule a M con cada una de las cadenas de J y ahora,

- Si alguna cadena es aceptada entonces el lenguaje es infinito.
- Si ninguna cadena es aceptada entonces el lenguaje es finito.

Explicación: Si una cadena $\omega \in J$ es aceptada, en su sucesión admisible de estados tiene que haber un estado q que se repite, de modo que debe ser posible escribir $\omega = \alpha\phi\psi$ con $\phi \neq \lambda$ y en M

$$q_I \overset{\alpha}{\Rightarrow} q \overset{\phi}{\Rightarrow} q \overset{\psi}{\Rightarrow} q' \in F$$

Es decir, la cadena ϕ puede ser "bombeada" de manera que el conjunto de cadenas de la forma $\alpha\phi^*\psi$ también debe ser aceptado por M y ese conjunto es infinito.

6. Demuestre que el lenguaje $P = \{a^p \mid p \text{ es primo}\}$ no es regular. Ayuda: recuerde que un entero n no es primo cuando puede expresarse como el producto de dos enteros $n = q \times r$ y además $q, r > 1$.

   Solución: Suponga que P es regular. Es decir, existe un aceptador de estados finitos determinístico $M = (Q, \{a\}, f, F, q_I)$ tal que $P = L(M)$. Suponga además que hay n estados en Q. Como el conjunto de primos es infinito, siempre es posible escoger un primo $p \geq n$. Si se estimula a M con la cadena a^p, en la sucesión admisible de estados para a^p tiene que haber un estado q que se repite; es decir, debe ser posible escribir a^p en la forma $a^s a^t a^u$ con $p = s + t + u,\ \ t \geq 1$ y además, en M

$$q_I \overset{a^s}{\Rightarrow} q \overset{a^t}{\Rightarrow} q \overset{a^u}{\Rightarrow} q' \in F$$

   Es decir, la subcadena a^t puede ser "bombeada" para obtener otras cadenas aceptadas por M y en consecuencia todo el conjunto

$$X = \{a^s (a^t)^* a^u \ \mid \ p = s + t + u,\ \ t \geq 1\} \subseteq L(M)$$

   también debe ser aceptado por M. En particular, a^t podría bombearse $s+u$ veces de modo que la longitud total de la cadena sería $s + t(s + u) + u = (s + u)(t + 1)$ que evidentemente no es primo y la cadena correspondiente no podría estar en P. Es decir, P no puede ser regular.