Introducción a la Informática Teórica
Tarea 3
"¡Opa Chomsky Style!"

Hernán Vargas Leighton
201073009-3

16 de mayo 2014

Respuestas

Gramáticas

1. Digamos $G = (\Sigma, N, P, S)$ la gramática de libre contexto con:

$$\Sigma = \{a, b\} \qquad N = \{S, A, B\} \qquad P = \left\{ \begin{array}{c} S \to A|B \\ A \to aSa|aa|a \\ B \to bSb|bb|b \end{array} \right\}$$

   Con S símbolo de partida.
   Luego G genera los palíndromos.

2. Suponiendo w cualquier expresión que puede ser encerrada por los paréntesis, además digamos que el string vacío ϵ es un string con paréntesis equilibrados, tenemos que $G = (\Sigma, N, P, S)$ con:

$$\Sigma = \Big\{(,), \{,\}, [,], w\Big\} \qquad N = \{S, A, B, C\} \qquad P = \left\{ \begin{array}{c} S \to A|B|C|SS|w|\epsilon \\ A \to (S) \\ B \to [S] \\ C \to \{S\} \end{array} \right\}$$

   Será la gramática que acepta cualquier expresión con paréntesis equilibrados.
   NOTA: w puede ser reemplazado por cualquier alfabeto, por ejemplo, si decimos $w = x|y|z$ simplemente reemplazamos w en Σ y en S y generamos strings con paréntesis equilibrados de la forma: $((x)y[z(x)]), \ldots$

3. Los cuatro niveles de la jerarquía de Chomsky son:

   - **Tipo 0:**
     - Lenguaje recursivamente enumerable.
     - Gramática sin restricciones.
     - Autómata: Máquina de Turing.
   - **Tipo 1:**
     - Lenguaje sensible al contexto.
     - Gramática: $\alpha \to \beta$ tal que $|\alpha| \leq |\beta|$
     - Autómata: Linealmente acotado.
   - **Tipo 2:**
     - Lenguaje de contexto libre.
     - Gramática: Tipo 1 y al lado izquierdo solo un no terminal: $A \to B$ con $A \in N, B \in (N \cup \Sigma)^*$
     - Autómata con pila (PDA)

- **Tipo 3:**
  - Lenguaje Regular.
  - Gramática: Tipo 2 y al lado derecho a lo más un no terminal $A \to \alpha, A \to \alpha B$ con $A, B \in N, \alpha \in \Sigma^*$
  - Autómata finito.

4. Gramática para las operaciones aritméticas:

   - Digamos $G = \{\Sigma, N, P, S\}$ con:

     $$\Sigma = \{(,), +, -, \cdot, a, b\} \qquad N = \{S\} \qquad P = \left\{S \to SS|(S)|S \cdot S|S + S|S - S|a|b\right\}$$

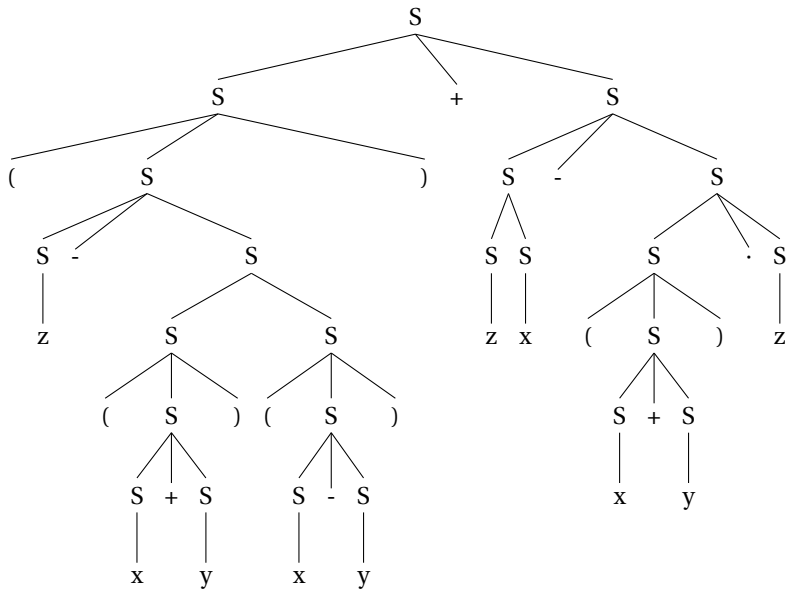
     Con S símbolo de partida. G representará las operaciones aritméticas.

   - Se puede, nos basta con cambiar el alfabeto Σ para reemplazar a, b por x, y, z y hacer lo mismo en G:
     Ahora tenemos $G = \{\Sigma, N, P, S\}$ con:

     $$\Sigma = \{(,), +, -, \cdot, x, y, z\} \qquad N = \{S\} \qquad P = \left\{S \to SS|(S)|S \cdot S|S + S|S - S|x|y|z\right\}$$

     NOTA: Se considera que el cambio en el alfabeto no afecta a los caracteres propios de una operación aritmética $((,), +, -, \cdot)$.

   - El árbol de derivación para $(z - (x + y)(x - y)) + zx - (x + y) \cdot z$ será:



   - La gramática es ambigua ya que existe más de una forma de hacer la derivación de extrema izquierda:

     $$S \to S + S \to (S) + S \to (S - S) + S \ldots \quad (1)$$
     $$S \to S - S \to S + S - S \to (S) + S - S \to (S - S) + S - S \ldots \quad (2)$$

     (1) y (2) son diferentes formas de derivación de extrema izquierda que nos llevan al mismo resultado (el árbol). Es más evidente (y menos costoso de escribir) para un string $x + y - z$, tenemos:

     $$S \to S + S \to x + S \to x + S - S \to x + y - S \to x + y - z$$
     $$S \to S - S \to S + S - S \to x + S - S \to x + y - S \to x + y - z$$

- La gramática cumple con no tener ϵ ni producciones unitarias ni producciones que no participen en la derivación del lenguaje $\mathscr{L}(G)$, por lo tanto nos basta con escribirla de la forma $A \to \alpha$, $A \to BC$, entonces:

  $S \to x|y|z$
  $A \to ($
  $B \to)$
  $C \to +$
  $D \to -$
  $E \to \cdot$
  $F \to AS$
  $G \to SC$
  $H \to SD$
  $I \to SE$
  $S \to SS|FB|GS|HS|IS$

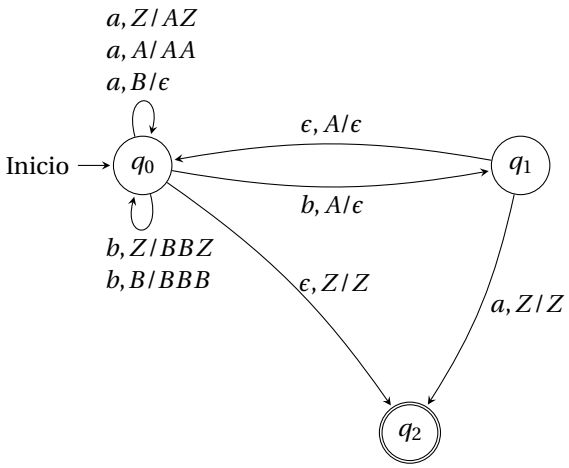
  Será la gramática en forma normal de Chomsky.

PDA

1. El PDA que acepta $\mathscr{L} = w \in \Sigma$ tal que el string w tiene el doble de $a's$ que $b's$ es $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ con:

   $Q = \{q_0, q_1, q_2\}$
   $\Sigma = \{a, b\}$
   $\Gamma = \{Z, A, B\}$
   δ = Función de transición
   $q_0 = q_0$
   $F = \{q_2\}$

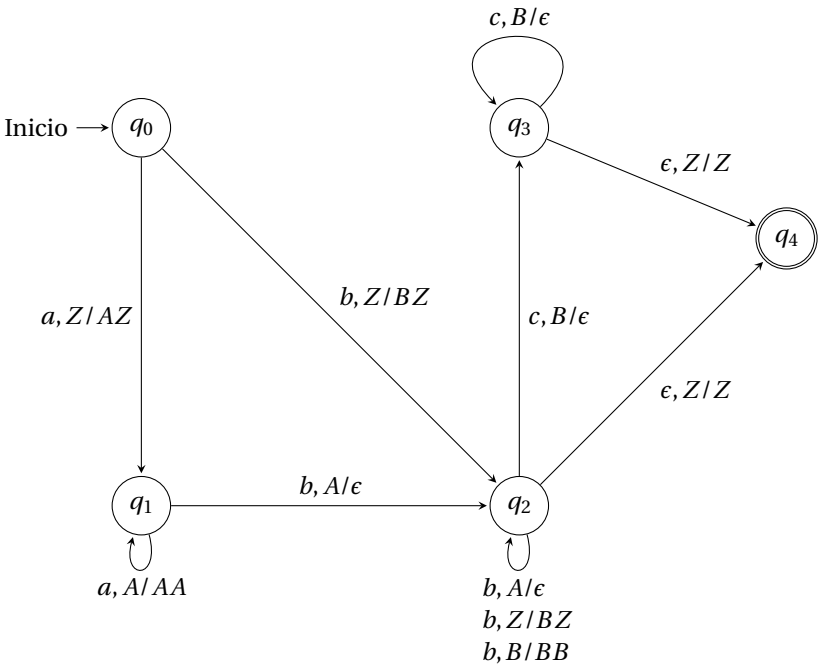
   Luego:



2. Digamos $\mathscr{L}(M) = \{a^x b^y c^z : |a| + |c| = |b|\}$. Creamos el PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ tal que:

   $Q = \{q_0, q_1, q_2, q_3\}$
   $\Sigma = \{a, b\}$
   $\Gamma = \{Z, A, B\}$
   δ = Función de transición
   $q_0 = q_0$
   $z_0 = Z$
   $F = \{q_4\}$

   Entonces:



Lema de Bombeo

1. Creo que cuando queremos demostrar con el lema del bombeo que un lenguaje no es regular el error más frecuente tiene relación con elegir un string no adecuado o hacer mal la división $\alpha\beta\gamma$ y no poder demostrar para toda división que el lema no se cumple.
   En general estos errores están relacionados con hacer mal la negación del lema y, debido a ello, no poder probar la contradicción.

2. Supongamos el lenguaje $\mathscr{L} = \{0^i 1^j : \gcd(i, j) = 1\}$ regular, entonces cumple con el lema del bombeo:

   - Digamos $N \in \mathbb{N}_0$ constante del lema.
   - Digamos $w = 0^i 1^j \in \mathscr{L}$ con $0 < i < j$, además cumple con $|w| = i + j \geq N$ y $\gcd(i, j) = 1$.
   - Digamos $\alpha = 0^p \wedge \beta = 0^{i-p} \wedge \gamma = 1^j$ será toda partición que cumple con $|\alpha\beta| = p + i - p = i \leq N \wedge |\beta| \geq 1$
   - Al bombear vemos que tenemos una cantidad de ceros igual a $p + k(i - p)$, por lo que con $k = \frac{j-p}{i-p}$ tendríamos j ceros y, por lo tanto, la misma cantidad de ceros que unos, así $\gcd(j, j) = j \neq 1$

Autómatas y expresiones regulares

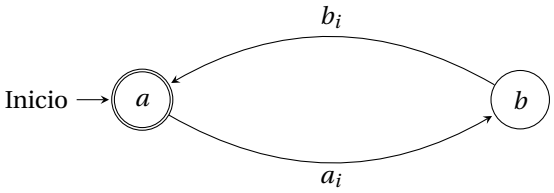
1. El lenguaje $\mathscr{L} = \{a^n b^n : n = 2\}$ es regular ya que n es constante y igual a 2 por lo tanto el lenguaje no es infinito y puede ser escrito como $\mathscr{L} = aabbb$.

2. Para los lenguajes regulares $\mathscr{A}$ y $\mathscr{B}$ digamos Σ_A, Σ_B los alfabetos respectivos, además digamos $\Sigma = \Sigma_A \cup \Sigma_B$ la unión de ambos alfabetos, buscamos mostrar que la operación BMBMCHQBM produce lenguajes regulares.

   - **Demostración usando un autómata:** Digamos $\alpha = a_i \forall\, i \in [1,k]$, $\beta = b_i \forall\, i \in [1,k]$, tenemos:

     $\Sigma = \{\alpha, \beta\}$ $\quad Q = \{a, b\}$ $\quad q_0 = a$ $\quad F = \{a\}$

     Entonces $M = (\Sigma, Q, \delta, q_o, F)$ será:

   - **Demostración usando propiedades de clausura:** Logramos la intercalación duplicando cada símbolo con todas las alternativas posibles, antes y después de él, según corresponda, así definimos las sustituciones:

     $S_1(a) = a \cdot \Sigma$
     $S_2(a) = \Sigma \cdot a$

     Con $\Sigma = \bigcup_{a \in \Sigma} \{A\}$
     Luego debemos hacer la intercepción y el string resultante será justamente la intercalación pues es el único que se repite. Entonces:

     $\text{BMBMCHQBM}(\mathscr{A}, \mathscr{B}) = S_1(\mathscr{A}) \cap S_2(\mathscr{B})$