Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3725 - Traductores e Interpretadores
Abril-Julio 2011

Examen Parcial II

(25 puntos)

Carnet:

Nombre:

1. **(4 puntos)** Considere el lenguaje $L = \{ww \mid w \in \{a,b\}^*\}$. Use el Lema de Bombeo de Lenguajes Regulares para demostrar que L no es regular.

   Asumo que L es Lenguaje Regular, entonces existe $M = (Q, \Sigma, \delta, q_0, F)$ con $|Q| = k$ que acepta palabras de L. Por el Lema de Bombeo para Lenguajes Regulares sabemos que $\forall z \in L$ con $|z| \geq k$ siempre se puede escribir $z = uvw$ tal que $|uv| \leq k, |v| > 0$ y luego $\forall i \geq 0$ se cumple $uv^iw \in L$.

   Consideremos la palabra $w = a^kba^kb \in L$, entonces cualquier partición de w que cumpla con las condiciones del Lema de Bombeo debe tener necesariamente $uv = a^j$ con $1 \leq j \leq k$, más aún $v = a^p$ con $p \geq 1$. Así para *cualquier* partición que escojamos, si se bombea v dos o más veces, el primer segmento de a va a aumentar de longitud mientras el segundo segmento de a permanece intacto, con lo que la primera mitad de la palabra resultante será necesariamente diferente a la segunda mitad y en consecuencia $uv^iw \notin L$ contradiciendo el Lema de Bombeo. Esa contradicción es consecuencia de haber asumido que L era en efecto Lenguaje Regular, por tanto no puede serlo.

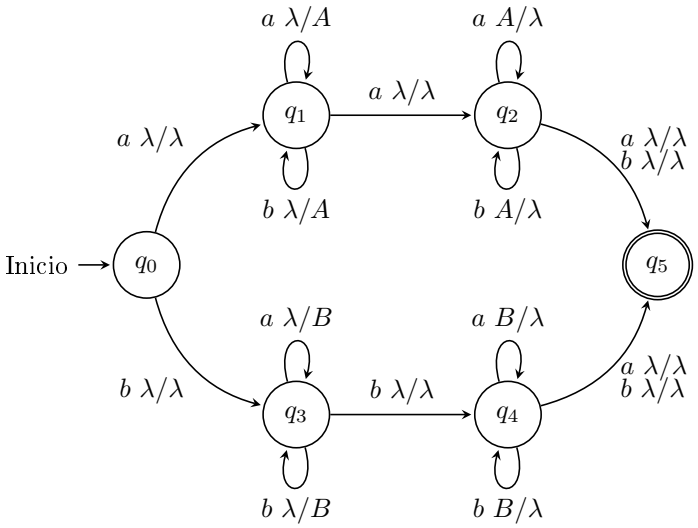
2. Para cada uno de los siguientes lenguajes demuestre si es libre de contexto o no. Para demostrar que un lenguaje es libre de contexto puede construir una Gramática Libre de Contexto que lo genere o construir un PDA que lo acepte (no es necesario explicar la lógica de la gramática ni del PDA, basta que funcione). Para demostrar que un lenguaje no es libre de contexto, aplique el Lema de Bombeo para Lenguajes Libres de Contexto con toda la formalidad del caso. Debe usar un método diferente para cada lenguaje; usar más de un método para el mismo lenguaje invalida la pregunta.

   (a) **(4 puntos)** $L = \{w = cucv \mid u, v \in \{a,b\}^*, c \in \{a,b\}, |u| = |v| - 1, |w| = 2k + 1, k > 0\}$
   El lenguaje corresponde a las palabras de longitud impar, tales que el símbolo inicial y el símbolo en medio sean idénticos. El lenguaje L puede ser generado con la gramática

$$\begin{aligned} S &\rightarrow \mathbf{a}A\mathbf{a} \mid \mathbf{a}A\mathbf{b} \mid \mathbf{b}B\mathbf{a} \mid \mathbf{b}B\mathbf{b} \\ A &\rightarrow CAC \mid \mathbf{a} \\ B &\rightarrow CBC \mid \mathbf{b} \\ C &\rightarrow \mathbf{a} \mid \mathbf{b} \end{aligned}$$

   En esta gramática el no-terminal A corresponde a generar palabras de longitud impar con **a** en medio, mientras que el no-terminal B corresponde a generar palabras con **b** en medio, en ambos casos con al menos un símbolo. Así, el no-terminal inicial S corresponde a generar palabras de longitud impar tales que según el símbolo inicial (**a** o **b**), se use el no-terminal A o B correspondiente para garantizar la presencia del mismo símbolo en medio. Note que como la palabra debe tener longitud impar, en S tenemos que considerar dos casos por cada símbolo inicial, uno para cada posible símbolo final.

El lenguaje L también puede ser generado con el PDA de la figura



(b) **(4 puntos)** $L = \{a^i b^j c^k \,|\, 0 < i < j < k < mi\}$

Asumamos que L es libre de contexto y sea k el especificado en el Lema de Bombeo para Lenguajes Libres de Contexto. Sea $z = a^k b^{k+1} c^{k+2}$, como $z \in L$ y $|z| > k$ entonces de acuerdo con el Lema de Bombeo para Lenguajes Libres de Contexto, z tiene al menos una forma de escribirse $uvwxy$ que satisface $|vwx| \leq k$, $|v| + |x| > 0$ y luego $\forall i \geq 0$ se cumple $uv^i wx^i y \in L$.

Como $|vwx| \leq k$, entonces la subcadena vwx puede tener las siguientes formas:

- a^i o b^i. En ambos casos, si se bombea uv^2wx^2y entonces habrá más a que b o más b que c, respectivamente, por lo que la palabra resultante no pertenece a L.
- c^i. Si se bombea uv^0wx^0y entonces habrá a lo sumo tantas c como b, por lo que la palabra resultante no pertenece a L.
- $a^i b^j$. Si se bombea uv^2wx^2y entonces habrá al menos tantas a como b, o bien al menos tantas b como c, por lo que la palabra resultante no pertenece a L.
- $b^i c^j$. Si se bombea uv^0wx^0y entonces habrá a lo sumo tantas b como a, por lo que la palabra resultante no pertenece a L.

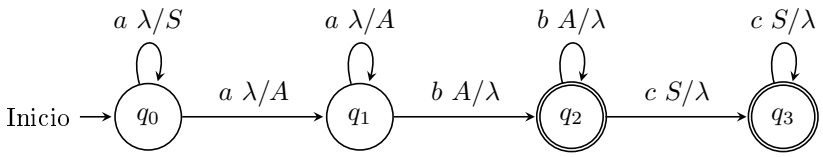
Como todas las particiones posibles para la palabra contradicen el Lema de Bombeo para Lenguajes Libres de Contexto, se concluye que L no puede ser Lenguaje Libre de Contexto.

(c) **(4 puntos)** $L = \{a^{i+j} b^i c^j \,|\, i > 0,\, j \geq 0\}$

Notamos que $a^{i+j} b^i c^j = a^{j+i} b^i c^j = a^j a^i b^i c^j$, de modo que el lenguaje L puede ser generado con la gramática

$$\begin{aligned} S &\rightarrow \mathbf{a}S\mathbf{c} \,|\, A \\ A &\rightarrow \mathbf{a}A\mathbf{b} \,|\, \mathbf{ab} \end{aligned}$$

El lenguaje L también puede ser generado con el PDA de la figura



3. Sea la gramática $G = (\{S, L\}, \{\mathbf{a}, \mathbf{c}, \mathbf{x}, \mathbf{s}\}, P, S)$ cuyo conjunto de producciones P se define como

$$\begin{aligned} S &\to \mathbf{x} \\ S &\to \mathbf{a}L\mathbf{c} \\ L &\to L\mathbf{s}S \\ L &\to S \end{aligned}$$

(a) **(4 puntos)** Modifique la gramática hasta que sea *fuertemente* $LL(1)$ y demuestre formalmente tal propiedad en la gramática resultante.

i. La gramática no es $LL(1)$ porque el símbolo inicial es recursivo y además tiene recursión izquierda. Introducimos un nuevo símbolo inicial S', agregamos la marca de final de entrada y, después de eliminar la recursión izquierda, la gramática queda

$$\begin{aligned} S' &\to S\$ \\ S &\to \mathbf{a}L\mathbf{c} \\ S &\to \mathbf{x} \\ L &\to SA \\ A &\to \lambda \\ A &\to \mathbf{s}SA \end{aligned}$$

ii. Calculamos el $FIRST$ y el $FOLLOW$ para cada símbolo no terminal

$$\begin{aligned} FIRST(S') &= \{\mathbf{a}, \mathbf{x}\} \\ FIRST(S) &= \{\mathbf{a}, \mathbf{x}\} \\ FIRST(L) &= \{\mathbf{a}, \mathbf{x}\} \\ FIRST(A) &= \{\lambda, \mathbf{s}\} \\ FOLLOW(S') &= \{\$\} \\ FOLLOW(S) &= \{\mathbf{c}, \mathbf{s}, \$\} \\ FOLLOW(L) &= \{\mathbf{c}\} \\ FOLLOW(A) &= \{\mathbf{c}\} \end{aligned}$$

iii. Calculamos los Conjuntos de Lookahead $LA(A \to \alpha) = FIRST(FIRST(\alpha) \cdot FOLLOW(A))$ para las producciones

$$\begin{aligned} LA(S' \to S\$) = FIRST(\{\mathbf{a}, \mathbf{x}\} \cdot \{\$\}) &= \{\mathbf{a}, \mathbf{x}\} \\ LA(S \to \mathbf{a}L\mathbf{c}) = FIRST(\{\mathbf{a}\} \cdot \{\mathbf{c}, \mathbf{s}, \$\}) &= \{\mathbf{a}\} \\ LA(S \to \mathbf{x}) = FIRST(\{\mathbf{x}\} \cdot \{\mathbf{c}, \mathbf{s}, \$\}) &= \{\mathbf{x}\} \\ LA(L \to SA) = FIRST(\{\mathbf{a}, \mathbf{x}\} \cdot \{\mathbf{c}\}) &= \{\mathbf{a}, \mathbf{x}\} \\ LA(A \to \lambda) = FIRST(\{\lambda\} \cdot \{\mathbf{c}\}) &= \{\mathbf{c}\} \\ LA(A \to \mathbf{s}SA) = FIRST(\{\mathbf{s}\} \cdot \{\mathbf{c}\}) &= \{\mathbf{s}\} \end{aligned}$$

Puede observarse que $\forall A \to \alpha | \beta \in P, \alpha \neq \beta$ se cumple $LA(A \to \alpha) \cap LA(A \to \beta) = \emptyset$, por lo tanto la gramática es fuertemente $LL(1)$.

(b) **(2 puntos)** Construya la tabla de análisis para un reconocedor predictivo no recursivo utilizando el algoritmo presentado en clase.

	a	**c**	**s**	**x**	$
S'	$S' \to S\$$			$S' \to S\$$	
S	$S \to \mathbf{a}L\mathbf{c}$			$S \to \mathbf{x}$	
L	$L \to SA$			$L \to SA$	
A		$A \to \lambda$		$A \to \mathbf{s}SA$	

(a) **(3 puntos)** Use el reconocedor para encontrar la derivación más izquierda de la palabra **aaxsxcsxsaxcc**.

Pila	Entrada	Acción
$S'\$$	**aaxsxcsxsaxcc**$	$S' \to S\$$
$S\$$	**aaxsxcsxsaxcc**$	$S \to \mathbf{a}L\mathbf{c}$
$\mathbf{a}L\mathbf{c}\$$	**aaxsxcsxsaxcc**$	Consumir **a**
$L\mathbf{c}\$$	**axsxcsxsaxcc**$	$L \to SA$
$SA\mathbf{c}\$$	**axsxcsxsaxcc**$	$S \to \mathbf{a}L\mathbf{c}$
$\mathbf{a}L\mathbf{c}A\mathbf{c}\$$	**axsxcsxsaxcc**$	Consumir **a**
$L\mathbf{c}A\mathbf{c}\$$	**xsxcsxsaxcc**$	$L \to SA$
$SA\mathbf{c}A\mathbf{c}\$$	**xsxcsxsaxcc**$	$S \to \mathbf{x}$
$\mathbf{x}A\mathbf{c}A\mathbf{c}\$$	**xsxcsxsaxcc**$	Consumir **x**
$A\mathbf{c}A\mathbf{c}\$$	**sxcsxsaxcc**$	$A \to \mathbf{s}SA$
$\mathbf{s}SA\mathbf{c}A\mathbf{c}\$$	**sxcsxsaxcc**$	Consumir **s**
$SA\mathbf{c}A\mathbf{c}\$$	**xcsxsaxcc**$	$S \to \mathbf{x}$
$\mathbf{x}A\mathbf{c}A\mathbf{c}\$$	**xcsxsaxcc**$	Consumir **x**
$A\mathbf{c}A\mathbf{c}\$$	**csxsaxcc**$	$A \to \lambda$
$\mathbf{c}A\mathbf{c}\$$	**csxsaxcc**$	Consumir **c**
$A\mathbf{c}\$$	**sxsaxcc**$	$A \to \mathbf{s}SA$
$\mathbf{s}SA\mathbf{c}\$$	**sxsaxcc**$	Consumir **s**
$SA\mathbf{c}\$$	**xsaxcc**$	$S \to \mathbf{x}$
$\mathbf{x}A\mathbf{c}\$$	**xsaxcc**$	Consumir **x**
$A\mathbf{c}\$$	**saxcc**$	$A \to \mathbf{s}SA$
$\mathbf{s}SA\mathbf{c}\$$	**saxcc**$	Consumir **s**
$SA\mathbf{c}\$$	**axcc**$	$S \to \mathbf{a}L\mathbf{c}$
$\mathbf{a}L\mathbf{c}A\mathbf{c}\$$	**axcc**$	Consumir **a**
$L\mathbf{c}A\mathbf{c}\$$	**xcc**$	$L \to SA$
$SA\mathbf{c}A\mathbf{c}\$$	**xcc**$	$S \to \mathbf{x}$
$\mathbf{x}A\mathbf{c}A\mathbf{c}\$$	**xcc**$	Consumir **x**
$A\mathbf{c}A\mathbf{c}\$$	**cc**$	$A \to \lambda$
$\mathbf{c}A\mathbf{c}\$$	**cc**$	Consumir **c**
$A\mathbf{c}\$$	**c**$	$A \to \lambda$
$\mathbf{c}\$$	**c**$	Consumir **c**
$\$$	$\$$	**Acepta**

Por lo tanto, la derivación más izquierda para la palabra será

$$
\begin{aligned}
S' &\Rightarrow \underline{S} \\
&\Rightarrow \mathbf{a}\underline{L}\mathbf{c} \\
&\Rightarrow \mathbf{a}\underline{S}A\mathbf{c} \\
&\Rightarrow \mathbf{aa}\underline{L}\mathbf{c}A\mathbf{c} \\
&\Rightarrow \mathbf{aa}\underline{S}A\mathbf{c}A\mathbf{c} \\
&\Rightarrow \mathbf{aax}\underline{A}\mathbf{c}A\mathbf{c} \\
&\Rightarrow \mathbf{aaxs}\underline{S}A\mathbf{c}A\mathbf{c} \\
&\Rightarrow \mathbf{aaxsx}\underline{A}\mathbf{c}A\mathbf{c} \\
&\Rightarrow \mathbf{aaxsxc}\underline{A}\mathbf{c} \\
&\Rightarrow \mathbf{aaxsxcs}\underline{SA}\mathbf{c} \\
&\Rightarrow \mathbf{aaxsxcsx}\underline{A}\mathbf{c} \\
&\Rightarrow \mathbf{aaxsxcsxs}\underline{S}A\mathbf{c}
\end{aligned}
$$

$$
\begin{aligned}
&\Rightarrow \quad \mathbf{aaxsxcsxsa}\underline{L}\mathbf{c}A\mathbf{c}\\
&\Rightarrow \quad \mathbf{aaxsxcsxsa}\underline{S}A\mathbf{c}A\mathbf{c}\\
&\Rightarrow \quad \mathbf{aaxsxcsxsax}\underline{A}\mathbf{c}A\mathbf{c}\\
&\Rightarrow \quad \mathbf{aaxsxcsxsaxc}\underline{A}\mathbf{c}\\
&\Rightarrow \quad \mathbf{aaxsxcsxsaxcc}
\end{aligned}
$$