Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3725 - Traductores e Interpretadores
Septiembre-Diciembre 2006

Examen Parcial II

(25 puntos)

Carnet:

Nombre:

1. **(6 puntos)** Sea L el lenguaje de prefijos de la cadena $abaabaaabaaaab \ldots ba^n ba^{n+1}b \ldots$ que terminan en b. Use el Lema de Bombeo de Lenguajes Libres de Contexto para demostrar que L no es Libre de Contexto.

   Supongamos que L es Libre de Contexto, y que tenemos el k apropiado según el Lema de Bombeo. Consideremos la palabra $z = abaab \ldots ba^k b$ que está en L y de acuerdo con el Lema de Bombeo podemos encontrar una descomposición $z = uvwxy$ con $|vx| \geq 1$ y $|vwx| \leq k$, y además $\forall i \geq 0$ se cumple $uv^iwx^iy \in L$. Precisamente la primera condición nos dice que v y x no pueden ser vacíos simultáneamente. Supongamos que $v \neq \lambda$ y observemos las descomposiciones de z:

   - Es posible que v no tenga ninguna ocurrencia de b. En ese caso v sólo tiene a y está entre dos b consecutivas dentro de la cadena, es decir
   $$\ldots ba^n ba^i v a^j ba^{n+2} b \ldots$$
   teniendo $i + |v| + j = n + 1$. Si se bombea v, entonces el número de a en el segmento será diferente a $n + 1$ y por tanto $uv^iwx^iy \notin L$, contradiciendo el Lema de Bombeo.
   - Es posible que v tenga dos o más ocurrencias de b. En ese caso dentro de v habrá una subcadena de la forma
   $$ba^n b$$
   Si se bombea v, entonces aparecerán dos o más subcadenas de esa forma y entonces $uv^iwx^iy \notin L$, contradiciendo el Lema de Bombeo.
   - Es posible que v tenga exactamente una ocurrencia de b. En ese caso $v = a^i ba^j$ y z tendría la forma
   $$\ldots ba^{n-1} ba^{n-i} v a^{n+1-j} ba^{n+2} \ldots$$
   Si se bombea v se produce
   $$\ldots ba^{n-1} ba^{n-i} a^i ba^j a^i ba^j a^{n+1-j} ba^{n+2} \ldots$$
   y al simplificar
   $$\ldots ba^{n-1} ba^n ba^{j+1} ba^{n+1} ba^{n+2} \notin L$$
   contradiciendo el Lema de Bombeo.

   Las tres posibles particiones contradicen el Lema de Bombeo. El mismo resultado se obtiene si se considera el caso $x = \lambda$. En consecuencia, L no es Libre de Contexto.

2. Sea la gramática $G = (\{A, B, C\}, \{\mathbf{a}, \mathbf{c}, \mathbf{f}, \mathbf{o}, \mathbf{v}, \mathbf{y}\}, P, A)$ cuyo conjunto de producciones P se define como

$$\begin{array}{rcl} A & \to & A\mathbf{o}B \\ A & \to & B \\ B & \to & B\mathbf{y}C \\ B & \to & C \\ C & \to & \mathbf{v} \\ C & \to & \mathbf{f} \\ C & \to & \mathbf{a}A\mathbf{c} \end{array}$$

a) **(5 puntos)** Construya un ADP extendido que tenga solamente un estado y que acepte palabras del lenguaje de la gramática. El ADP debe aceptar palabras por estado final y pila vacía simultáneamente. Justifique el funcionamiento del ADP propuesto.

El ADP extendido $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ definido como

$$\begin{array}{rcl} Q & = & \{q\} \\ \Sigma & = & \{\mathbf{a}, \mathbf{c}, \mathbf{f}, \mathbf{o}, \mathbf{v}, \mathbf{y}\} \\ \Gamma & = & \{\mathbf{a}, \mathbf{c}, \mathbf{f}, \mathbf{o}, \mathbf{v}, \mathbf{y}, A, B, C\} \\ q_0 & = & q \\ F & = & \{q\} \end{array}$$

y con δ definida

$$\begin{array}{rcl} \delta(q, \lambda, \lambda) & = & \{[q, A]\} \\ \delta(q, \lambda, A) & = & \{[q, A\mathbf{o}B], [q, B]\} \\ \delta(q, \lambda, B) & = & \{[q, B\mathbf{y}C], [q, C]\} \\ \delta(q, \lambda, C) & = & \{[q, \mathbf{v}], [q, \mathbf{f}], [q, \mathbf{a}A\mathbf{c}]\} \\ \delta(q, x, x) & = & \{[q, \lambda]\} \forall x \in \Sigma \end{array}$$

satisface los requerimientos del problema pues simula las derivaciones más izquierdas a partir de las reglas de la gramática.

b) **(2 puntos)** Demuestre el funcionamiento del PDA procesando la palabra **avofcyv**.

Uno de los posibles cómputos del PDA es

$$\begin{array}{rcl} [q, \mathbf{avofcyv}, \lambda] & \vdash & [q, \mathbf{avofcyv}, A] \\ & \vdash & [q, \mathbf{avofcyv}, B] \\ & \vdash & [q, \mathbf{avofcyv}, B\mathbf{y}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{a}A\mathbf{cy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{a}A\mathbf{o}B\mathbf{cy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{a}B\mathbf{o}B\mathbf{cy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{a}C\mathbf{o}B\mathbf{cy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{avo}B\mathbf{cy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{avo}C\mathbf{cy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{avofcy}C] \\ & \vdash & [q, \mathbf{avofcyv}, \mathbf{avofcyv}] \\ & \vdash & [q, \mathbf{vofcyv}, \mathbf{vofcyv}] \\ & \vdash^* & [q, \lambda, \lambda] \end{array}$$

c) **(7 puntos)** Modifique la gramática hasta que sea $LL(1)$ y demuestre formalmente tal propiedad en la gramática resultante.

1) La gramática no es $LL(1)$ porque tiene recursión izquierda. Después de eliminarla, la gramática queda

$$\begin{array}{rcl} A & \to & BA' \\ A' & \to & \mathbf{o}BA' \\ A' & \to & \lambda \\ B & \to & CB' \\ B' & \to & \mathbf{y}CB' \\ B' & \to & \lambda \\ C & \to & \mathbf{v} \\ C & \to & \mathbf{f} \\ C & \to & \mathbf{a}A\mathbf{c} \end{array}$$

2) Calculamos el $FIRST$ y $FOLLOW$ de cada símbolo no terminal

$$\begin{array}{rcl} FIRST(A) & = & \{\mathbf{a}, \mathbf{f}, \mathbf{v}\} \\ FIRST(A') & = & \{\mathbf{o}, \lambda\} \\ FIRST(B) & = & \{\mathbf{a}, \mathbf{f}, \mathbf{v}\} \\ FIRST(B') & = & \{\mathbf{y}, \lambda\} \\ FIRST(C) & = & \{\mathbf{a}, \mathbf{f}, \mathbf{v}\} \\ FOLLOW(A) & = & \{\mathbf{c}, \$\} \\ FOLLOW(A') & = & \{\mathbf{c}, \$\} \\ FOLLOW(B) & = & \{\mathbf{c}, \mathbf{o}, \$\} \\ FOLLOW(B') & = & \{\mathbf{c}, \mathbf{o}, \$\} \\ FOLLOW(C) & = & \{\mathbf{c}, \mathbf{o}, \mathbf{y}, \$\} \end{array}$$

3) Calculamos los Conjuntos de Lookahead $LA_1(A \to \alpha) = FIRST(FIRST(\alpha)FOLLOW(A))$ para las producciones:

$$\begin{array}{rcl} LA_1(A \to BA') & = & \{\mathbf{a}, \mathbf{f}, \mathbf{v}\} \\ LA_1(A' \to \mathbf{o}BA') & = & \{\mathbf{o}\} \\ LA_1(A' \to \lambda) & = & \{\mathbf{c}, \$\} \\ LA_1(B \to CB') & = & \{\mathbf{a}, \mathbf{f}, \mathbf{v}\} \\ LA_1(B' \to \mathbf{y}CB') & = & \{\mathbf{y}\} \\ LA_1(B' \to \lambda) & = & \{\mathbf{c}, \mathbf{o}, \$\} \\ LA_1(C \to \mathbf{v}) & = & \{\mathbf{v}\} \\ LA_1(C \to \mathbf{f}) & = & \{\mathbf{f}\} \\ LA_1(C \to \mathbf{a}A\mathbf{c}) & = & \{\mathbf{a}\} \end{array}$$

Puede observarse que $\forall A \to \alpha|\beta \in P, \alpha \neq \beta$ se cumple $LA_1(A \to \alpha) \cap LA_1(A \to \beta) = \emptyset$ por tanto la gramática es $LL(1)$.

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

	a	**c**	**f**	**o**	**v**	**y**	$
A	$A \to BA'$		$A \to BA'$		$A \to BA'$		
A'		$A' \to \lambda$		$A' \to \mathbf{o}BA'$			$A' \to \lambda$
B	$B \to CB'$		$B \to CB'$		$B \to CB'$		
B'		$B' \to \lambda$		$B' \to \lambda$		$B' \to \mathbf{y}CB'$	$B' \to \lambda$
C	$C \to \mathbf{a}A\mathbf{c}$		$C \to \mathbf{f}$		$C \to \mathbf{v}$		

e) **(3 puntos)** Utilice el reconocedor para encontrar la derivación más izquierda de la palabra **aafovcyvc**.

Pila	Entrada	Regla
$A\$$	**aafovcyvc**$	$A \to BA'$
$BA'\$$	**aafovcyvc**$	$B \to CB'$
$CB'A'\$$	**aafovcyvc**$	$C \to \mathbf{a}A\mathbf{c}$
$\mathbf{a}A\mathbf{c}B'A'\$$	**aafovcyvc**$	
$A\mathbf{c}B'A'\$$	**afovcyvc**$	$A \to BA'$
$BA'\mathbf{c}B'A'\$$	**afovcyvc**$	$B \to CB'$
$CB'A'\mathbf{c}B'A'\$$	**afovcyvc**$	$C \to \mathbf{a}A\mathbf{c}$
$\mathbf{a}A\mathbf{c}B'A'\mathbf{c}B'A'\$$	**afovcyvc**$	
$A\mathbf{c}B'A'\mathbf{c}B'A'\$$	**fovcyvc**$	$A \to BA'$
$BA'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**fovcyvc**$	$B \to CB'$
$CB'A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**fovcyvc**$	$C \to \mathbf{f}$
$\mathbf{f}B'A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**fovcyvc**$	
$B'A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**ovcyvc**$	$B' \to \lambda$
$A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**ovcyvc**$	$A' \to \mathbf{o}BA'$
$\mathbf{o}BA'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**ovcyvc**$	
$BA'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**vcyvc**$	$B \to CB'$
$CB'A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**vcyvc**$	$C \to \mathbf{v}$
$\mathbf{v}B'A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**vcyvc**$	
$B'A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**cyvc**$	$B' \to \lambda$
$A'\mathbf{c}B'A'\mathbf{c}B'A'\$$	**cyvc**$	$A' \to \lambda$
$\mathbf{c}B'A'\mathbf{c}B'A'\$$	**cyvc**$	
$B'A'\mathbf{c}B'A'\$$	**yvc**$	$B' \to \mathbf{y}CB'$
$\mathbf{y}CB'A'\mathbf{c}B'A'\$$	**yvc**$	
$CB'A'\mathbf{c}B'A'\$$	**vc**$	$C \to \mathbf{v}$
$\mathbf{v}B'A'\mathbf{c}B'A'\$$	**vc**$	
$B'A'\mathbf{c}B'A'\$$	**c**$	$B' \to \lambda$
$A'\mathbf{c}B'A'\$$	**c**$	$A' \to \lambda$
$\mathbf{c}B'A'\$$	**c**$	
$B'A'\$$	$	$B' \to \lambda$
$A'\$$	$	$A' \to \lambda$
$\$$	$	

Y la derivación más izquierda resulta

$$
\begin{aligned}
A &\Rightarrow BA' \\
&\Rightarrow CB'A' \\
&\Rightarrow \mathbf{a}A\mathbf{c}B'A' \\
&\Rightarrow \mathbf{a}BA'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{a}CB'A'\mathbf{c}B'A'
\end{aligned}
$$

$$
\begin{aligned}
&\Rightarrow \mathbf{aa}A\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aa}BA'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aa}CB'A'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aaf}B'A'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aaf}A'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafo}BA'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafo}CB'A'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafov}B'A'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafov}A'\mathbf{c}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafovc}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafovcy}CB'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafovcyv}B'A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafovcyv}A'\mathbf{c}B'A' \\
&\Rightarrow \mathbf{aafovcyvc}B'A' \\
&\Rightarrow \mathbf{aafovcyvc}A' \\
&\Rightarrow \mathbf{aafovcyvc}
\end{aligned}
$$