Teoría de Lenguajes

2do. Parcial – Curso 2005

Ejercicio 1.-

sent-if ::= "**if**" "(" expr ")" sent ["**else**" sent] "**end if**"

a.1)Clasifique los símbolos según sean terminales, no terminales y del metalenguaje, en una tabla del estilo:

Simb. Terminales	Simb. No Terminales	Simb. del Metalenguaje
If else endif ( )	Expr sent sent-if	[ ] ::= “

a.2)
terminal IF, ELSE, PARCURDER, PARCURIZQ ;
nonterminal sent-if, expr, sent ;
start programa;

programa ::= .....

sent-if ::= IF PARCURIZQ expr PARCURDER sent ENDIF
| IF PARCURIZQ expr PARCURDER sent ELSE sent ENDIF
;

expr ::= ....
Sent ::= ......

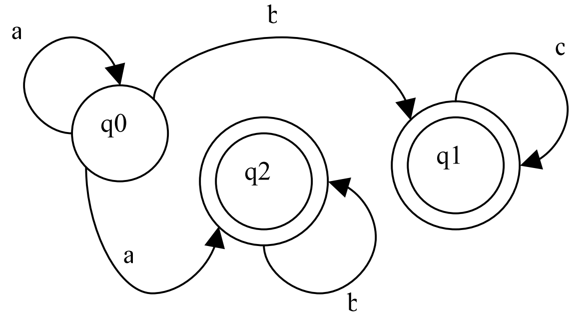
a.3) III
La gramática no presentará conflictos. La ambigüedad de la sentencia if se anula con el ENDIF.

a. Opción 4.

Cup utiliza un reconocimiento bottom-up, parte de las hojas del arbol que construye, y termina en la raiz.
Se imprime lo que esta al costado de la sentencia luego que se terminaron de reconocer cada uno de los “subárboles” de la misma.

Ejercicio 2.-

Sea L_2 el lenguaje reconocido por el autómata que se muestra a continuación.



a) Escriba una gramática lineal derecha que genere L_2

Solución:
Una forma fácil para generar una gramática lineal derecha a partir de un AFD es asociar una variable a cada estado, y escribir las producciones tal que cada variable genere todas las tiras que me llevan del estado asociado a todos los estados finales a los que se puede llegar desde el estado asociado.[1]

Gramática Lineal Dercha[2]	Analogía con las Ecs. características
$S \rightarrow aS \mid bT \mid aU$	$x_0 \rightarrow ax_0 \mid bx_1 \mid ax_2$
$T \rightarrow cT \mid \varepsilon$	$x_1 \rightarrow cx_1 \mid \varepsilon$
$U \rightarrow bU \mid \varepsilon$	$x_2 \rightarrow bx_2 \mid \varepsilon$

Como dese S se generan todas las tiras que van desde q0 a los estados finales, S es la variables inicial desde la cual genero todas las tiras del lenguaje.

Elimino las producciones-ε

$S \rightarrow aS \mid bT \mid aU \mid b \mid a$
$T \rightarrow cT \mid c$
$U \rightarrow bU \mid b$

Como no tiene producciones-ε, ni producciones unitarias, todas las variables son positivas y alcanzables, entonces esta simplificada. Y además sigue siendo una gramática lineal derecha.

b) Escriba una gramática lineal izquierda que genere L_2

Solución:
En forma similar a lo anterior, para generar una gramática lineal izquierda a partir de un AFD se asocia una variable a cada estado, y se escriben las producciones tal cada variable genere todas las tiras que me llevan del estado inicial al estado asociado a la variable.[3]

Gramática Lineal Izquierda[4]	Analogía con las Ecs. para hallar las E.R. de las clases de R_M
$S \rightarrow Sa \mid \varepsilon$	$y_0 \rightarrow y_0a \mid \varepsilon$
$T \rightarrow Sb \mid Tc$	$y_1 \rightarrow y_0b \mid y_1c$
$U \rightarrow Sa \mid Ub$	$y_2 \rightarrow y_0a \mid y_2b$

S no puede ser la variable inicial porque desde S solo se genera a*. En realidad el lenguaje esta dado por las tiras generadas por T y U, dado que T genera todas las tiras que llevan de qo a q1 y U las que llevan de q0 a q2. Entonces faltaría agregar una regla más: S_0 -> T | U, siendo S0 la variable inicial y no S.

Elimino las producciones-ε

[1] En forma similar a la asociación de las xi a los qi en las ecuaciones características en la demostración del teorema "Análisis de Kleene"

[2] Es una grmática lineal derecha porque todas su producciones son de la forma A->wB con $A,B \in V$ y $w \in T^*$

[3] En forma similar a la asociación de las yi a los qi en las ecuaciones vistas en el método para hallar las epresiones regulares de las clases de R_M.

[4] Es una grmática lineal izquierda porque todas su producciones son de la forma A->Bw con $A,B \in V$ y $w \in T^*$

S_0 -> T | U
S → Sa | a
T → Sb | Tc | b
U → Sa | Ub | a

Elimino las unitarias

S_0 -> Sb | Tc | b | Sa | Ub | a
S → Sa | a
T → Sb | Tc | b
U → Sa | Ub | a

Como no tiene producciones-ε, ni producciones unitarias, todas las variables son positivas y alcanzables entonces esta simplificada. Y además sigue siendo una gramática lineal izquierda.

c) Escriba una gramática de acuerdo a la Jerarquía de Chomsky para el lenguaje $L_2 . L_2$.
Nota: Las tres gramáticas **deben** estar simplificadas.

Como L_2 es regular, L_2. L_2 es regular porque los lenguajes regulares son cerrados bajo la concatenación.
Entonces según la jerarquía de Chomsky corresponde hacer una gramática regular (lineal izquierda o derecha)

Teniendo presenta como la grámtica de la parte a que genera L_2:
S → aS | bT | aU | b | a
T → cT | c
U → bU | b

Puedo generar una gramática que genera L_2. L_2 de la siguiente manera:
S → aS | bT | aU | bS’ | aS’
T → cT | cS’
U → bU | bS’
S’ → aS’ | bT’ | aU’ | b | a
T’ → cT’ | c
U’ → bU’ | b

Que es regular (es lineal derecha) y esta simplificada (no tiene producciones-ε, ni producciones unitarias, y todas sus variables son útiles)

Ejercicio 3.-

Genero CaAaAaF (una A menos que la cantidad de a)
S -> CS'F
S' -> aAS' | a

Muevo las A a la derecha y las transformo en B
porque van a "volver"
Queda CaaaBBF
Aa -> aA
AF -> B
AB -> BB

```
# Muevo la B a la izquierda y duplico las a
# Cuando llega a la C, se transforman en D
# Queda CDDaXXaXXaXXF
aB -> BaX
XB -> BX
CB -> CD
DB -> DD

# Muevo la D a la derecha y duplico las X y las a
# Cuando llega al final desaparece la D
# Queda CaYYXYYXYYaYYXYYXYYaYYXYYXYYF
Da -> aYD
DX -> XYD
DY -> YD
DF -> F

# Muevo la C a la derecha y genero la tira
# Queda aaaaaaaaaaaaaaaaaaaaaaaaaaa
Ca -> aC
CX -> aC
CY -> aC
CF -> eps
```

Ejercicio 4.-

a)

Es NO determinista. Alcanza con ver las que existen 2 transiciones leyendo a y con Zo en el tope.

b) S -> SaSbS | SbSaS | aS | ε

Ejercicio 5.-

a) Enuncie el Pumping Lemma para lenguajes libres de contexto.
b) Considere el lenguaje L_5 formado por las tiras de la forma $a^pb^q\#b^pa^q$, con p,q ≥0. Sea la tira $z=a^Nb\#ba^N$. Suponga que z es la tira elegida para demostrar que L_5 no es libre de contexto utilizando el contrarrecíproco del PL. ¿Es apropiada esta tira? En caso afirmativo, realice la demostración utilizándola. En caso contrario, dé todas las descomposiciones de z tal que, para todo i, $z_i \in L_5$.

Nota: Las gramáticas y los autómatas deben corresponderse con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.

$z=a^Nb\#b^Na$, la tira no es apropiada ya que la descomposición $u=a^{N-1}$ $v=a$ $w=b\#$ $x=b$ $y=b^{N-1}a$, cumple que $z_i=a^{N-1}a^ib\#b^ib^{N-1} \in L_5$ para todo i. (suponiendo N>=4)

si N=3 entonces la tira serviría
$z=a^3b\#b^3a$

si N=2 también serivría
$z=a^2b\#b^2a$

si N=1 también serivría
$z=ab\#ba$

como solo se sabe que N>=1, y para N>=4 la z no sirve, entonces la z planteada finalmente no sirve.

Ahora doy todas las descomposiciones tales que z pertenece para todo i
$u = a^k$, $v = a^j$, $w = a^{n-j-k}b\#b^l$, $x = b^j$, $y=b^{N-j-l}a$
$z_i = a^ka^{j*i}a^{n-j-k}b\#b^lb^{j*i}b^{n-j-l}a = a^{N-j*(i-1)}b\#b^{N-j*(i-1)}a \in L$ para todo i.