

Ejercicios de Sintaxis

Nota: los ejercicios marcados con (*) al principio están sacados del libro de la cátedra los ejercicios marcados con (°) al principio están basados en uno tomado en un final los ejercicios marcados con (^) al principio están basados en uno tomado en un final (reciente)

1. (*) Resuelva el problema de prefijos comunes en la siguiente GIC:

```
S → aTbRba | aTbc
```

T → a | Ta

 $R \rightarrow c \mid cRb$

2. (*) Elimine la recursividad izquierda de la siguiente GIC:

<listaExp> -> <operando> | <listaExp> <operador> <operando>

3. Elimine la recursividad izquierda de la siguiente GIC:

 $E \rightarrow T \mid E+T$

T → F | T*F

 $F \rightarrow 2 \mid 5 \mid 3 \mid (E)$

4. (*) Elimine la recursividad izquierda de la siguiente GIC:

listaExpresiones: expresión |

listaExpresiones; expresión

expresión: término |

expresión + término

término: factor |

término * factor

factor: num |

(expresión)

5. (°) Dada la siguiente GIC encuentre una equivalente LL(1)

S → SbD | ab

D → a | az

6. (°) Obtenga el conjunto Primero(S) de la siguiente GIC:

 $(\{S,M,T\}, \{x, yx\}, \{S \rightarrow xyx, S \rightarrow MT, M \rightarrow yxM, M \rightarrow T, T \rightarrow \epsilon\}, S)$

7. (*) Obtenga los conjuntos Primero para los no terminales de la siguiente GIC:

S -> ABc

 $A \rightarrow a \mid \epsilon$

 $B \rightarrow b \mid \epsilon$

8. Encuentre el conjunto Primero(S) para la siguiente gramática

 $S \rightarrow MeC \mid Bf$

 $M \rightarrow fM \mid \epsilon$

B → aXW

 $C \rightarrow c \mid \epsilon$

9. Encuentre el conjunto Siguente(M) para la siguiente gramática

S → aMbX

M → d

T → f

 $X \rightarrow Xc \mid MT$

10. Encuentre el conjunto Siguiente(D) para la siguiente gramática

S → aDe | DT

 $D \rightarrow bc$

 $T \rightarrow f \mid \epsilon$



```
11. Dada la siguiente GIC
```

Calcule los siguientes conjuntos:

- a) Primero(E)
- b) Primero(E')
- c) Siguiente(E)
- d) Siguiente(E')
- e) Siguiente(T)
- f) Siguiente(T')
- 12. (^) Dada la siguiente GIC calcule Primero(Término)

```
Término → Factor | Término * Factor
Factor → Número | (Término)
Número → 0 | 1
```

- 13. (^) Calcule Primero(S) Siendo S el no terminal SentIteración de la BNF de C, es decir, la gramática de las sentencias de iteración de C
- 14. (^) Analice la siguiente afirmación sobre las GIC LL(1): Dadas las producciones de un no terminal, los conjuntos primeros de los lados derechos deben ser disjuntos. ¿Está de acuerdo con la afirmación? ¿Por qué?
- 15. (^) Tilde todas las afirmaciones verdaderas con respecto a las LL(1):

```
\square La GIC S -> a S | a es LL(1)
```

 \Box La GIC S -> a b S | a es LL(1)

- ☐ Pueden tener producciones recursivas.
- ☐ Pueden tener producciones con épsilon del lado derecho.
- ☐ El conjunto primero de los no terminales deben ser vacíos.
- 16. (^) Dada la siguiente gramática:

```
E \rightarrow E + T \mid T

T \rightarrow T * F \mid F

F \rightarrow 1 \mid 2 \mid 3
```

Indique si las siguientes afirmaciones son verdaderas o falsas y justifique su respuesta

- ☐ Puede ser implementada directamente, es decir, sin aplicar ninguna transformación a la gramática, por un parser descendente recursivo.
- ☐ Puede ser implementada directamente, es decir, sin aplicar ninguna transformación a la gramática, por Yacc/Bison.
- 17. (^) Dada la gramática:

```
sentencia-if:
```

```
if ( expresión ) sentencia
```

if (expresión) sentencia else sentencia

Indique si la gramática es de tipo LL(1). Justifique. Si no lo es, reescríbala para que si lo sea.



18. (^) En la especificación del Lenguaje de C está la siguiente gramática: selection-statement:

```
if ( expression ) statement
if ( expression ) statement else statement
switch ( expression ) statement
```

- a) Indique el conjunto primero de selection-statement
- b) Justifique si la gramática es ambigua.