

7 EJERCICIOS RESUELTOS

CAP. 1: DEFINICIONES BÁSICAS E INTRODUCCIÓN A LENGUAJES FORMALES

* Ejercicio 1 * (pág.8)

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 9, +, -\}$

* Ejercicio 2 * (pág.8)

012 210

* Ejercicio 3 * (pág.8)

abababcde

* Ejercicio 4 * (pág.9)

$a^{1300}b^{846}a^{257}$

* Ejercicio 5 * (pág.10)

aabbbaabba

*Ejercicio 6 * (pág.11)

Porque c^0 significaría la ausencia de carácter y eso no tiene sentido.

* Ejercicio 7 * (pág.11)

Ambos representan la cadena vacía porque S^0 es la cadena vacía para cualquier cadena S.

* Ejercicio 8 * (pág.11)

$(ab^3)^3 = (abbb)^3 = abbbabbbabbb$ y $((ab)^3)^3 = (ababab)^3 = ababababababababab$

* Ejercicio 10 * (pág.12)

El alfabeto mínimo es $= \{A, r, g, e, n, t, i, a, H, o, l, d, B, s\}$

* Ejercicio 11* (pág.13)

Por ejemplo: $L = \{cccc, ppppp, cpcpc, ppccp\}$

* Ejercicio 12 * (pág.14)

$L = \{b^n / 0 \leq n \leq 8\}$

* Ejercicio 13 * (pág.14)

“El lenguaje de todas las palabras sobre el alfabeto $\{b\}$ que están formadas por la concatenación del carácter b consigo mismo, entre una y ocho veces e incluye a la palabra vacía”.

* Ejercicio 14 * (pág.14)

Por extensión se podría describir aunque sería bastante tedioso. Por comprensión no se puede porque no tenemos los operadores adecuados para hacer esta descripción.

* Ejercicio 15 * (pág.15)

“El lenguaje de todas las primeras 201 palabras sobre el alfabeto $\{a\}$ formado por la concatenación de la letra a consigo misma y donde cada una de ellas tiene un número impar de letras a”.

* Ejercicio 16 * (pág.15)

$L = \{a^{2n} / 0 \leq n \leq 400\}$

* Ejercicio 17 * (pág.16)

a) aba, abba, abbba.

b) "El lenguaje de todas las palabras sobre el alfabeto {a,b} que comienza con una única a seguida de una o varias bes y terminan con exactamente una a".

b') "El lenguaje sobre el alfabeto {a, b} donde todas las palabras tienen exactamente dos aes (una como primer carácter de la palabra y otra como último carácter de la palabra) y en el medio tienen una o más bes."

* Ejercicio 18 * (pág.17)

NO porque el Lenguaje Universal es cerrado bajo concatenación.

*Ejercicio 19 * (pág.18)

palabras reservadas : L Finito

nombres creados por el programador (Identificadores): L Infinito

constantes enteras y reales: L Infinito

caracteres de puntuación: L Finito

operadores aritméticos: L Finito

operadores lógicos: L Finito

declaraciones: L Infinito (no regular)

expresiones: L Infinito (no regular)

sentencias: L Infinito (no regular)

* Ejercicio 20 * (pág.18)

(a)

```
unsigned int LongitudCadena (char *s) {
    unsigned int i;
    for(i=0; s[i]!='\0'; i++);
    return i;
}
```

(b)

Solución 1:

```
int EsCadenaVacía (char *s) {
    if (s[0] == '\0')
        return 1;
    else
        return 0;
}
```

Solución 2:

```
int EsCadenaVacía (char s[]) {
    return s[0]=='\0';
}
```

(c)

```
void ConcatenaDosCadenas(char* s1, const char* s2) {
    unsigned int i,j;
    for (i=0; s1[i]!='\0'; i++);
    for (j=0; s2[j]!='\0'; i++, j++)
        s1[i] = s2[j];
    s1[i]='\0';
}
```

*** Ejercicio 21 * (pág.18)**

Sugerencia: utilice cadenas constantes.

(a)

```
#include<stdio.h>
unsigned int LongitudCadena (char*);
int main (void) {
    char cad1[] = "longitud 11";
    char cad2[] = "";
    char cad3[] = " ";
    printf("La longitud de cad1 es: %u\n", LongitudCadena (cad1));
    printf("La longitud de cad2 es: %u\n", LongitudCadena (cad2));
    printf("La longitud de cad3 es: %u\n", LongitudCadena (cad3));
    return 0;
} /* fin-main */

/* Desarrollo funcion LongitudCadena */
unsigned int LongitudCadena (char *s) {
    unsigned int i;
    for(i=0; s[i]!='\0'; i++);
    return i;
} /* fin-LongitudCadena */
```

(b)

```
#include<stdio.h>
int EsCadenaVacía (char[]);
int main (void) {
    char cad1[] = "no vacía";
    char cad2[] = "";
    if (EsCadenaVacía(cad1)) printf("La cadena 1 es vacía\n");
    else printf("La cadena 1 no es vacía %s\n", cad1);
    if (EsCadenaVacía(cad2)) printf("La cadena 2 es vacía\n");
    else printf("La cadena 2 no es vacía %s\n", cad2);
    return 0;
} /* fin-main*/

/* Desarrollo funcion EsCadenaVacía */
int EsCadenaVacía(char s[]) {
    return s[0]=='\0';
} /* fin-EsCadenaVacía */
```

(c)

```
#include<stdio.h>
void ConcatenaDosCadenas (char*, const char*);
int main (void) {
    char cad1[27+1] = "Primera parte ";
    char cad2[] = "Segunda parte";
    ConcatenaDosCadenas (cad1, cad2);
    printf ("La cadena concatenada es: %s\n", cad1);
    return 0;
} /* fin-main */
```

(continúa en la página siguiente)

```

/* Desarrollo funcion ConcatenaDosCadenas */
void ConcatenaDosCadenas(char* s1, const char* s2) {
    unsigned int i,j;
    for (i=0; s1[i]!='\0'; i++);
    for (j=0; s2[j]!='\0'; i++, j++)
        s1[i] = s2[j];
    s1[i]='\0';
}

```

CAP. 2: GRAMÁTICAS FORMALES Y JERARQUÍA DE CHOMSKY

* Ejercicio 1 * (pág.20)

- a) $S \rightarrow aaT, T \rightarrow \epsilon \Rightarrow aa$
 $S \rightarrow aaT, T \rightarrow b \Rightarrow aab$
 b) No

* Ejercicio 2 * (pág.21)

El lenguaje formal $L = \{a, aa\}$

- $S \rightarrow aT, T \rightarrow \epsilon \Rightarrow a$
 $S \rightarrow aT, T \rightarrow a \Rightarrow aa$

* Ejercicio 3 * (pág.22)

- a) No, porque no hay una producción que permita obtener la última b. $S \rightarrow bQ, Q \rightarrow a, ?$
 b) $\{aa, ab, ba, b\}$

* Ejercicio 4 * (pág.23)

- a) $S \rightarrow aT \mid aQ$
 $Q \rightarrow aT \mid aR$
 $T \rightarrow b$
 $R \rightarrow aT$
 b) $G = (\{S, Q, T, R\}, \{a, b\}, \{S \rightarrow aT \mid aQ, Q \rightarrow aT \mid aR, T \rightarrow b, R \rightarrow aT\}, S)$

* Ejercicio 5 * (pág.23)

- a) $S \rightarrow aR \mid aQ \mid \epsilon$
 $Q \rightarrow aT$
 $T \rightarrow bR$
 $R \rightarrow b$
 b) $G = (\{S, Q, T, R\}, \{a, b\}, \{S \rightarrow aR, S \rightarrow aQ, S \rightarrow \epsilon, Q \rightarrow aT, T \rightarrow bR, R \rightarrow b\}, S)$

* Ejercicio 6 * (pág.24)

- ab $S \rightarrow aT, T \rightarrow b$

* Ejercicio 7 * (pág.24)

- 1º) $S \rightarrow aS, 2º) S \rightarrow aS, 3º) S \rightarrow aT$ y 4º) $T \rightarrow b$ y se genera la palabra: **aaab**

* Ejercicio 8 * (pág.25)

- a) $GR = (\{S\}, \{0,1,2,3,4,5,6,7,8,9\}, \{S \rightarrow 0S, S \rightarrow 1S, S \rightarrow 2S, S \rightarrow 3S, S \rightarrow 4S, S \rightarrow 5S, S \rightarrow 6S, S \rightarrow 7S, S \rightarrow 8S, S \rightarrow 9S, S \rightarrow 0, S \rightarrow 1, S \rightarrow 2, S \rightarrow 3, S \rightarrow 4, S \rightarrow 5, S \rightarrow 6, S \rightarrow 7, S \rightarrow 8, S \rightarrow 9\}, S)$

- b) $GQR = (\{S, N\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{S \rightarrow N, S \rightarrow NS, N \rightarrow 0, N \rightarrow 1, N \rightarrow 2, N \rightarrow 3, N \rightarrow 4, N \rightarrow 5, N \rightarrow 6, N \rightarrow 7, N \rightarrow 8, N \rightarrow 9\}, S)$
- c) GR 20 producciones y la GQR 12 producciones

*Ejercicio 9 * (pág.25)

$S \rightarrow AN$

$A \rightarrow BN$

$B \rightarrow N \mid BN$

$N \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7$

* Ejercicio 10 * (pág.25)

$S \rightarrow 0A, S \rightarrow 1A, S \rightarrow 2A, S \rightarrow 3A, S \rightarrow 4A, S \rightarrow 5A, S \rightarrow 6A, S \rightarrow 7A,$

$A \rightarrow 0B, A \rightarrow 1B, A \rightarrow 2B, A \rightarrow 3B, A \rightarrow 4B, A \rightarrow 5B, A \rightarrow 6B, A \rightarrow 7B,$

$B \rightarrow 0B, B \rightarrow 1B, B \rightarrow 2B, B \rightarrow 3B, B \rightarrow 4B, B \rightarrow 5B, B \rightarrow 6B, B \rightarrow 7B,$

$B \rightarrow 0, B \rightarrow 1, B \rightarrow 2, B \rightarrow 3, B \rightarrow 4, B \rightarrow 5, B \rightarrow 6, B \rightarrow 7$

* Ejercicio 11 * (pág.26)

a) Sí, porque una GR siempre es un caso especial de una GIC.

b) No, porque las producciones de una GR son un subconjunto de las producciones de una GIC.

Por ejemplo: $S \rightarrow abc$ es una producción válida para una GIC pero no lo es para una GR.

* Ejercicio 12 * (pág.26)

a) **a** aplicando la producción $S \rightarrow a$ entonces se genera la palabra **a**

b) **aab** aplicando las producciones: 1º) $S \rightarrow aSb$, 2º) $S \rightarrow a$ entonces se genera la palabra **aab**

* Ejercicio 13 * (pág.26)

$L = \{a^{n+1}b^n \mid n \geq 0\}$

* Ejercicio 14 * (pág.26)

$S \rightarrow aSb \mid b$

* Ejercicio 15 * (pág.26)

$S \rightarrow aaTbQ$

$T \rightarrow aaTb \mid b$

$Q \rightarrow aQ \mid \epsilon$

* Ejercicio 16 * (pág.26)

Si toda GQR puede ser re-escrita mediante una GR y, a su vez, toda GR es un subconjunto de las producciones de una GIC entonces, una GQR es un caso particular de una GIC.

* Ejercicio 17 * (pág.27)

S

ACaB

AaaCB (aplicada $Ca \rightarrow aaC$)

AaaDB (“ $CB \rightarrow DB$)

AaDaB (“ $aD \rightarrow Da$)

ADaaB (“ $aD \rightarrow Da$)

ACaaB (“ $AD \rightarrow AC$)

AaaCaB (“ $Ca \rightarrow aaC$)

AaaaaCB (“ $Ca \rightarrow aaC$)

AaaaaE (“ $CB \rightarrow E$)

AaaaEa (" aE \rightarrow Ea)
 AaaEaa (" aE \rightarrow Ea)
 AaEaaa (" aE \rightarrow Ea)
 AEaaaa (" aE \rightarrow Ea)
 εaaaa (" AE \rightarrow ε)
aaaa

* Ejercicio 18 * (pág.28)

S
 aSb
 aaSbb
 aaaSbbb
 aaaabbbb No es una palabra del LIC
 No hay manera de producir una b más sin una a

* Ejercicio 19 * (pág.30)

$G = (\{S, T\}, \{a, b, c\}, \{S \rightarrow Tb, T \rightarrow aTc, T \rightarrow abc\}, S)$

* Ejercicio 20 * (pág.30)

a) aaabcccb
 S
 Tb
 aTcb
 aaTccb
 aaaTcccb
aaabcccb Es palabra del lenguaje
 b) aabbccb
 S
 Tb
 aTcb
 aaTccb
 ¿? No hay manera de producir dos bes entre la a y la c, por lo tanto, no es palabra del lenguaje
 c) aaabcccbb
 S
 Tb
 aTcb
 aaTccb
 aaaTcccb
 ¿? No hay manera de producir una segunda b después de la última c, por lo tanto, no es una palabra del lenguaje
 d) aaccb
 S
 Tb
 aTcb
 aaTccb
 ¿? No hay manera de no producir la b entre la a y la c, por lo tanto, no es palabra del lenguaje

* Ejercicio 21 * (pág.30)

$GR = (\{S, T\}, \{a, b, c, d, 2, 3, 4, 5, 6\}, \{S \rightarrow a | b | c | d | aT | bT | cT | dT, T \rightarrow aT | bT | cT | dT | 2T | 3T | 4T | 5T | 6T | 2 | 3 | 4 | 5 | 6\}, S)$
 La GQR es más sencilla para ser leída

* Ejercicio 22 * (pág.30)

a) ab23 (Derivación a izquierda)

S

SD

SDD

SLDD

LLDD

aLDD

abDD

ab2D

ab23 es una palabra válida

a') 2a3b (Derivación a izquierda)

S

SD

SDD

SLDD

?? no se puede seguir derivando porque S no produce D, por lo tanto, no es válida

b) ab23 (Derivación a derecha)

S

SD

S3

SD3

S23

SL

Sb23

Lb23

ab23 es una palabra válida

b') 2a3b (Derivación a derecha)

S

SL

Sb

SDb

S3b

SL3b

Sa3b

?? no se puede seguir derivando porque S no produce D, por lo tanto, no es válida

* Ejercicio 23 * (pág.31)

a) S

E;

T;

6; Correcta

b) S

E;

E + T; ¿? No hay manera de producir el terminal + en el extremo derecho.

Por lo tanto, no es correcta

c) S

E;

¿? No hay manera de producir el terminal + en el extremo derecho.

Por lo tanto, no es correcta

d) S
E;
E + T;
E + T + T;
E + T + T + T;
T + T + T + T;
6 + T + T + T;
6 + 6 + T + T;
6 + 6 + 6 + T;
6 + 6 + 6 + 2; Correcta