UNIVERSIDAD DE GRANADA

Escuela Tecnica Superior de Ingenieria
Informatica y Telecomunicaciones

Practicas Modelos de Computación

Grupo B3

Juan Luis Torres Ramos

24 Octubre 2023

Practica 1

Encuentra una gramática libre del contexto para generar cada uno de los siguientes lenguajes:

1. $L = \{a^i b^j \mid i, j \in \mathbb{N},\ i \leq j\}$.
2. $L = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$.
3. $L = \{a^i b^i a^j b^j \mid i, j \in \mathbb{N}\}$.
4. $L = \{a_i b_i \mid i \in \mathbb{N}\} \cup \{b_i a_i \mid i \in \mathbb{N}\}$.
5. $L = \{uu^{-1} \mid u \in \{a, b\}^*\}$.
6. $L = \{a^i b^j c^{i+j} \mid i, j \in \mathbb{N}\}$.

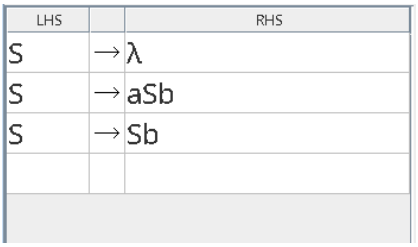
donde $\mathbb{N}$ es el conjunto de los numeros naturales incluyendo el 0

Pasos para resolver el ejercicio:

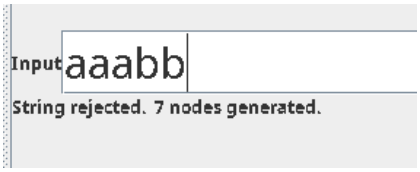
1. Determinar los símbolos terminales y no terminales.
2. Determinar el símbolo inicial.
3. Analizar el lenguaje para determinar qué se pide.
4. Determinar las reglas de producción.
5. Comprobar con JFLAP

A. $L = \{a^i b^j \mid i, j \in \mathbb{N},\ i \leq j\}$**.**

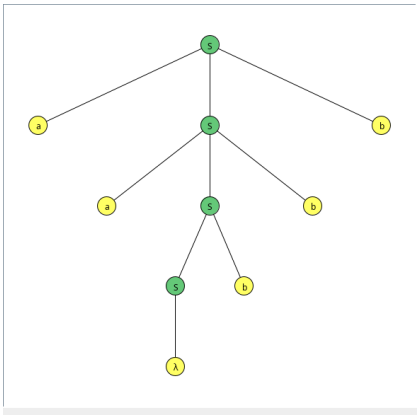
1. Los símbolos terminales serán $\{a, b\}$ y los simbolos no terminales serán S y B.
2. El símbolo inicial será S.
3. Analizar el lenguaje para determinar qué se pide. En este caso, se pide que la cadena tenga un número de a menor o igual que el número de b. Por ejemplo, *aabbb* y *aabb* pertenecen al lenguaje, pero *aab* no.
4. Determino las reglas de producción:
   - $S \to \epsilon$ (genero la cadena vacía).
   - $S \to aSb$.
   - $S \to Sb$.
5. compruebo con JFLAP que la gramática es correcta.



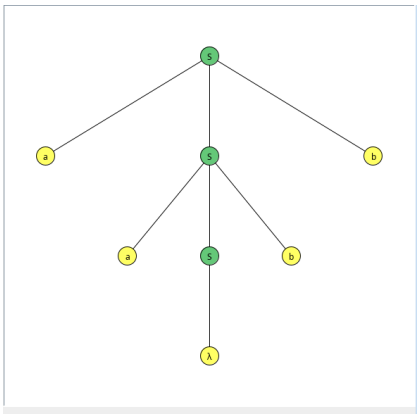
(a) la producción



(b) la cadena *aaabb*



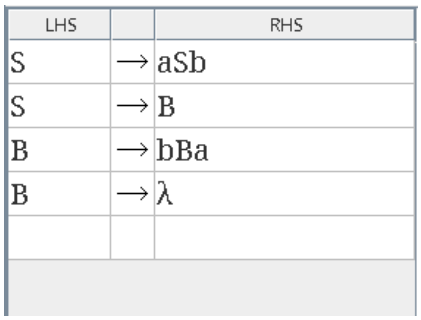
(c) la cadena *aabbb*



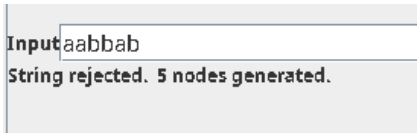
(d) la cadena *aabb*

B. $L = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$**.**

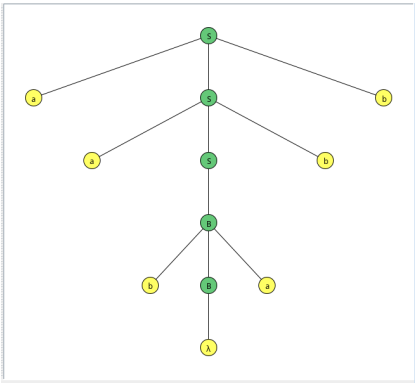
1. Los símbolos terminales serán $\{a, b\}$ y los simbolos no terminales serán S y B.
2. El símbolo inicial será S.
3. El lenguaje nos pide generar una cadena de 4 caracteres donde primero se generen $a^i b^j$ y luego $a^j b^i$, es decir en los extremos un numero caracteres i y en los caracteres del centro un numero de caracteres j. Por ejemplo, *aababb* y *ab* pertenecen al lenguaje, pero *aabbab* no.
4. Determino las reglas de producción:
   - $S \to aSb$ (genero mismo numero de caracteres en los extremos).
   - $S \to B$.
   - $B \to bBa$ (genero mismo numero de caracteres en el centro).
   - $B \to \epsilon$ (genero la cadena vacía).
5. compruebo con JFLAP que la gramática es correcta.



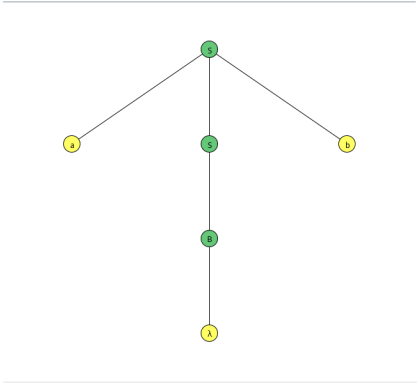
(a) la producción



(b) la cadena *aabbab*



(c) la cadena *aababb*



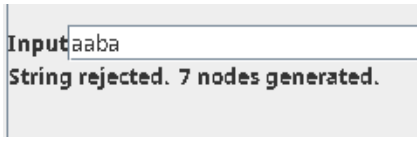
(d) la cadena *ab*

C. $L = \{a^i b^i a^j b^j \mid i, j \in \mathbb{N}\}$**.**

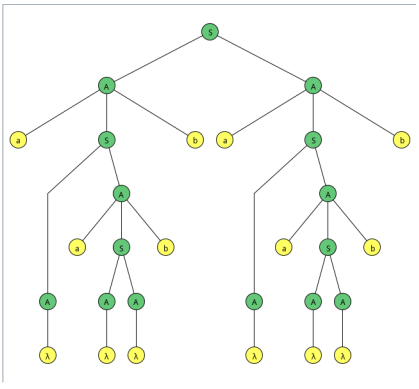
1. Los símbolos terminales serán $\{a, b\}$ y los simbolos no terminales serán S y B.
2. El símbolo inicial será S.
3. El lenguaje nos pide generar cadenas de 4 caracteres de la forma *abab* donde los dos primeros caracteres tengan el mismo nuemoor de caracteres y para los dos ultimos caracteres tambien tengan la misma cantidad.Ejemplos de cadenas serían *aabbaabb* ,*aabbab* pero no acepta *aaba*
4. Determino las reglas de producción:
   - $S \to AA$ (simbolo inicial).
   - $A \to aSb.$ (genero $\{a^i b^i | i \in \mathbb{N}\}$).
   - $A \to \epsilon$ (genero la cadena vacía).
5. compruebo con JFLAP que la gramática es correcta.

LHS		RHS
S	⟶	AA
A	⟶	aSb
A	⟶	λ

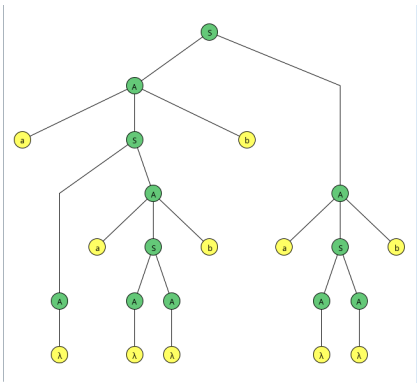
(a) la producción



(b) la cadena *aaba*



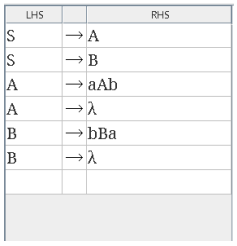
(c) la cadena *aabbaabb*



(d) la cadena *aabbab*

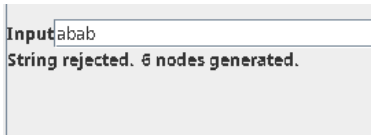
D. $L = \{a_i b_i \,|\, i \in \mathbb{N}\} \cup \{b_i a_i \,|\, i \in \mathbb{N}\}$**.**

1. Los símbolos terminales serán $\{a, b\}$ y los simbolos no terminales serán S , A B.
2. El símbolo inicial será S .
3. Combina dos conjuntos de cadenas: el primero contiene cadenas de la forma $\{a_i b_i \,|\, i \in \mathbb{N}\}$, y el segundo contiene cadenas de la forma $\{b_i a_i \,|\, i \in \mathbb{N}\}$.Las cadenas *aabb bbaa* lo cumplen mientras *abab* no lo cumple Lo resolvemos por partes
4. Determino las reglas de producción:
   - Podemos generar $\{a_i b_i \,|\, i \in \mathbb{N}\}$.
     $A \to aAb$, $A \to \epsilon$.
   - Por otro lado $\{b_i a_i \,|\, i \in \mathbb{N}\}$.
     $B \to bBa$, $B \to \epsilon$.
   - El lenguaje L se puede generar añadiendo .
     $S \to A$, $S \to B$.
5. compruebo con JFLAP que la gramática es correcta.

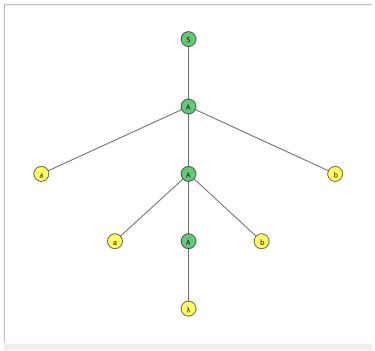


LHS		RHS
S	→	A
S	→	B
A	→	aAb
A	→	λ
B	→	bBa
B	→	λ

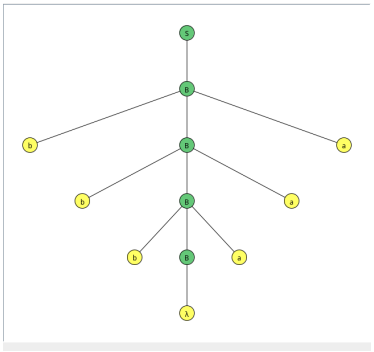
(a) la producción



(b) la cadena *abab*



(c) la cadena *aabb*



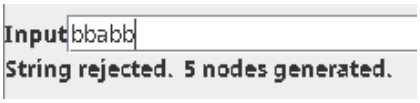
(d) la cadena *bbbaaa*

E. $L = \{uu^{-1} \mid u \in \{a, b\}^*\}$**.**

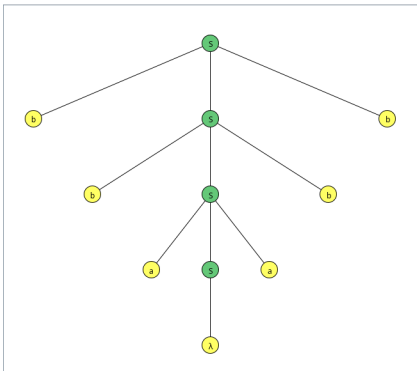
1. Los símbolos terminales serán $\{a, b\}$ y los simbolos no terminales serán S.
2. El símbolo inicial será S.
3. Analizar el lenguaje para determinar qué se pide. En este caso,se pide generar cadenas que son palíndromos formados por caracteres 'a' y 'b'. Cadenas que pertenecen al lenguaje son *abba* y *bbaabb* pero no *bbabb*.
4. Determino las reglas de producción:
   - $S \to \epsilon$ (genero la cadena vacía).
   - $S \to aSa$.
   - $S \to bSb$.
5. compruebo con JFLAP que la gramática es correcta.

LHS		RHS
S	⟶	aSa
S	⟶	bSb
S	⟶	λ

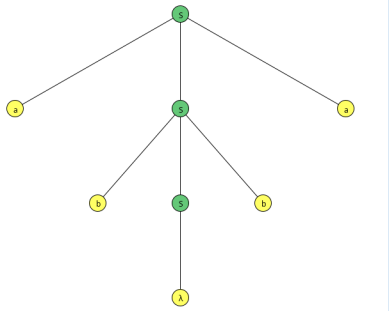
(a) la producción



(b) la cadena *bbab*



(c) la cadena *bbaabb*



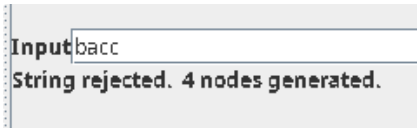
(d) la cadena *abba*

F. $L = \{a^i b^j c^{i+j} \mid i, j \in \mathbb{N}\}$**.**

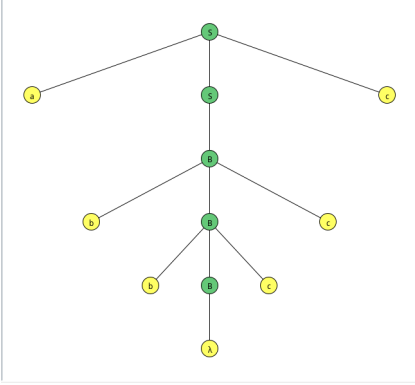
1. Los símbolos terminales serán $\{a, b, c\}$ y los simbolos no terminales serán S.
2. El símbolo inicial será S.
3. En este caso,se pide generar cadenas donde la cantidad de 'a's y 'b's es igual y la cantidad total de 'c's es la suma de las cantidades de 'a' y 'b' . Cadenas que cumplen la gramatica son *abbccc* y *aaabcccc* pero no *bacc*
4. Determino las reglas de producción:
   - $S \to aSc$ (genero la cadena vacía).
   - $S \to B$.
   - $B \to bBc$.
   - $B \to \epsilon$.
5. compruebo con JFLAP que la gramática es correcta.

LHS		RHS
S	⟶	aSc
S	⟶	B
B	⟶	bBc
B	⟶	λ

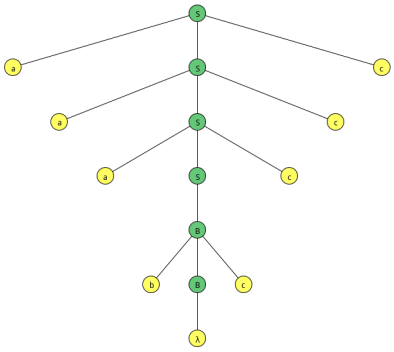
(a) la producción



(b) la cadena *bacc*



(c) la cadena *abbccc*



(d) la cadena *aaabcccc*

Practica 2

Analizadores léxicos, problemas de mineria, trabajo Lex, 2 problema

Tareas a realizar

1. Formar un grupo de trabajo compuesto por una, dos o tres personas.
2. Cada grupo de trabajo debe pensar un problema original de procesamiento de textos. Para la resolución de este problema debe ser apropiado el uso de Lex, o sea, se debe resolver mediante el emparejamiento de cadenas con expresiones regulares y la asociación de acciones a cada emparejamiento.
3. Cada grupo debe resolver el problema propuesto usando Lex. Se deberá realizar una memoria donde se presente una descripción del problema y su solución, además de entregar electrónicamente los ficheros de texto con la implementación de la solución.
4. Esta práctica deberá ser entregada antes del día 31 de Diciembre de 2020. Se entregará a través de la plataforma PRADO en un fichero .zip conteniendo todos los archivos de esta práctica. Sólo es necesario que lo entregue uno de los componentes del grupo.

Pasos para resolver el ejercicio:

1. Descripcion del problema
2. solucion
3. codigo lex

1. Descripcion del Problema

En la definición de este problema, he buscado un enfoque tanto practico como original. Desde el punto de vista de un profesor, podemos establecer, a la hora de evaluar código, una relación directa entre la cantidad/densidad de comentarios presentes en un código fuente y la comprensión del mismo por parte del alumno, por lo que evidencia que le ha dado un mayor esfuerzo y se le podria valorar positivamente. Tambien, por otro lado es más facil evaluar el codigo sin esos comentarios, por lo que vendira bien tener una version del código sin comentarios.Por lo tanto, el problema que se plantea es el siguiente:

Densidad de comentarios y limpieza de comentarios en Codigo Fuente

Tu tarea es desarrollar un programa en Lex que calcule la densidad de comentarios en un código fuente en C. La densidad de comentarios se define como el porcentaje del código total que está ocupado por comentarios. Además, tu programa deberá ser capaz de generar una nueva versión del código eliminando todos los comentarios.

Ejemplo

El alumno ha entregado su ejercicio de C correspondiente de la asignatura, voy a calcular la densidad de comentarios con la siguiente formula:

$$\text{Densidad de comentarios} = \frac{\text{Longitud total de los comentarios}}{\text{Longitud total del código (sin contar los comentarios)}}$$

ademas voy a generar el mismo codigo pero sin texto para evaluarlo aparte. un

ejemplo tanto de entrada como de salida será el siguiente

```c
#include <stdio.h>
#include <stdlib.h>

// funcion de ejemplo
void imprimirMensaje() {
    printf("! Hola, mundo!\n");
}

/*
    comentarios
    multilinea
*/

int main{
    // Llamada a la funcion
    imprimirMensaje();
    return 0;
}

```