

Reynaldo Vega A01114523

Analizador sintáctico, algoritmo CYK

SOFTWARE DESIGN DOCUMENT

21 de octubre de 2020

Contenido

[Introducción 3](#_Toc53427408)

[Propósito 3](#_Toc53427409)

[Gramáticas libres de contexto (GLC) 3](#_Toc53427410)

[Forma Normal de Chomsky (FNCh) 3](#_Toc53427411)

[Algoritmo CYK 4](#_Toc53427412)

[Diseño e implementación 4](#_Toc53427413)

[Algoritmos 4](#_Toc53427414)

[Flujo del programa 4](#_Toc53427415)

[Diagramas de clase 4](#_Toc53427416)

# Introducción

## Propósito

El propósito, del siguiente documento, es explicar la funcionalidad, diseño e implementación que se hizo para lograr la finalidad del proyecto de la segunda unidad de la materia de matemáticas computacionales. El documento abarcará todos aquellos requisitos que se pidieron en el proyecto número dos de la materia, el cual consiste en la implementación de un analizador sintáctico, que utiliza el algoritmo CYK para verificar si una cadena pertenece o no a una Gramática Libre de Contexto (GLC); de igual manera, se abarcarán temas relacionados con la Forma Normal de Chomsky, ya que el algoritmo CYK solo se puede aplicar una vez habiendo convertido una GLC a esta respectiva forma.

## Gramáticas libres de contexto (GLC)

Las Gramáticas Libres de Contexto se pueden definir formalmente como una cuádrupla de la siguiente manera:

*V-* es un conjuntofinito de símbolos no terminales

*T-* es un conjuntofinito de símbolos terminales

*P-* es un conjuntofinito de producciones, donde su forma es de la siguiente manera:

*S-* es el símbolo inicia de la GLC y

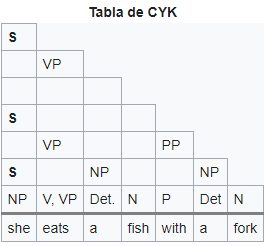
## Forma Normal de Chomsky (FNCh)

Una gramática está en la Forma Normal de Chomsky si todas sus producciones son de alguna de las siguientes formas:

Donde A, B y C forman parte del conjunto de símbolos no terminales y pertenece al conjunto de los símbolos terminales

## Algoritmo CYK

El algoritmo de Cocke Younger Kasami, también conocido como CYK, es un algoritmo de programación dinámica, el cual nos permite definir si una cadena es aceptada o no por una Gramática Libre de Contexto; además, si es aceptada, el mismo algoritmo nos permite obtener una posible solución que lleva al producto de la cadena que se quiere verificar.



# Diseño e implementación

## Algoritmos

A continuación, se explicará detalladamente los algoritmos que se utilizaron para la implementación del proyecto del segundo parcial:

1. Eliminar símbolos no terminales que no generan productos terminales
2. Eliminar símbolos no alcanzables
3. Cerradura de producciones
4. Convertir GLC a FNCh

## Flujo del programa

## Diagramas de clase