SDD – Algoritmo CYK



Francisco Javier Ramos Velasco

A01636425

Matemáticas Computacionales – Algoritmo CYK

Contenido

1 Introducción	2
2 Objetivo	3
3Algortimo doProductions	4
3 Algoritmo para verificación de FNCH	4
4 Algoritmo para transformar a FNCH	5
5 Algoritmo CYK	5
6 Algoritmo para crear árbol de derivación	6
7 Diagrama de flujo	7
8 Dificultades	8

Introducción

Este documento se hizo con el propósito de documentar el segundo proyecto semestral de la materia Matemáticas Computacionales demostrando lo aprendido en el tema de algoritmo de Chomsky; algoritmo CYK y árbol de derivación.

Además de explicar funcionalidad del código implementado (Java) en video de máxima duración 5 min. A continuación se explica cronológicamente cada método del código.

Objetivo

El objetivo de este trabajo es poner en práctica lo aprendido en segundo parcial. Conjunto con esto se debía implementar una solución en código (Java en mi preferencia) donde sus inputs eran gramáticas y una cadena Se espera que el programa te diga si esa cadena es válida para la gramática puesta en el input.

En sí otro objetivo se planteaba aquí, ya que se necesitaba verificar si la gramática estuviera en la Forma Normal de Chomsky (FNCH) en caso de no estar en esa forma normal, se debe implementar un algoritmo el cuál transforme las gramáticas a esa forma, para posteriormente entrar en proceso con el algoritmo CYK y poder verificar si la cadena antes puesta si pertenece a la gramática.

Para finalizar si pertenece deberás sacar un árbol de derivación para que se entienda mejor para el usuario la guía de como la gramática puede sacar dicha cadena.

Algoritmo doProductions

Este algoritmo se encarga de recibir un array de Strings indicando las diferentes gramáticas y separar las producciones de cada uno de los símbolos No Terminales separándolos por cada | (or) que se encuentre.

Por ejemplo : S-> aSb|e

Seria S{aSb, e} (2 producciones).

Este algoritmo regresa un Map<Character, ArrayList<String>> el cual se refiere a cada no terminal con sus diferentes producciones.

Algoritmo verificarChomsky

VerificarChomsky se encarga de revisar cada noTerminal minuciosamente para ver si no rompen ninguna regla para que este en FNCH, como:

- No tener epsilons
- Tener solo una terminal en una producción
- Tener máximo 2 no terminales en una producción
- No tener símbolos inútiles
- No hay que tener producciones unitarias

Regresa un valor booleano el cual nos indica si todo está en orden el cual el programa se pasa directamente al algoritmo CYK si es el caso contrario entonces tocaría pasarnos al algoritmo de Chomsky.

Algoritmo chomsky

El algoritmo chomsky se trata de recibir las gramáticas y convertir a cada una en FNCH por medio de 5 métodos:

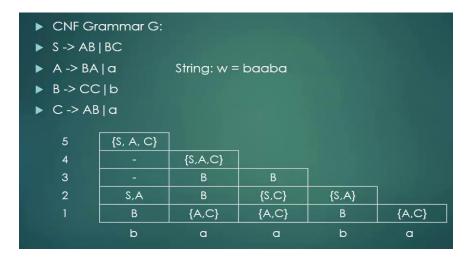
- 1. Eliminar las producciones de cadena vacía
- 2. Eliminar las producciones Unitarias
- 3. Eliminar las Producciones Inútiles
- 4. Sustitución de Terminales
- 5. Sustitución de No Terminales

Con estos pasos (métodos) el HasMap el cual contiene todas las gramáticas estarán modificadas y validadas en la forma normal de Chomsky y se podrá dar paso al algoritmo CYK.

Algoritmo CYK

Este algoritmo inventado por Cocke Younger Kasami, el cual se implentá programación Dinámica, en el método se formó una tabla en forma de escalera, en la cual la fila más alta representa el inicio (símbolo inicial S) y las filas de más abajo se alojan los símbolos terminales.

Para sacar los símbolos de cada casilla debemos de saber si hay una gramática con la cual se puede producir dicha combinación de dos casillas; en las cuales las dos van alternando posición.



Algoritmo arbolDerivación

Este algoritmo se manda a llamar por el algoritmo CYK solo si la cadena ha sido validada por el algoritmo CYK su función es crear un árbol binario el cual su root empieza con el símbolo Inicial S y de ahí se va abriendo pasos a más símbolos terminales o no terminales

Se recorre la tabla pensando que, en cada casilla, las casillas de abajo son sus hijos izquierdos y las que están en diagonal row-1, column+1 se refiere a que son sus hijos derechos. Esa representación sirve mucho para el árbol de derivación.

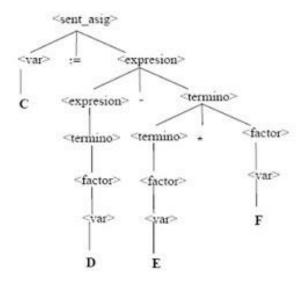
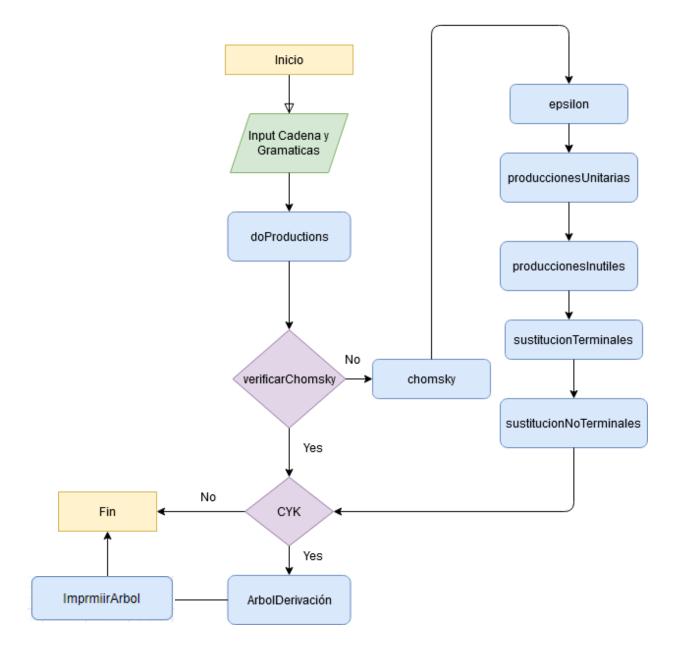


Diagrama de Flujo



Dificultades

El algoritmo imprimirArbol, no se pudo desarrollar óptimamente; imprime el árbol por niveles, pero no quedo tiempo para checar la razón de la falla ya que solo en los primeros niveles daba la impresión correctamente, pero avanzando no aparecían los símbolos o los ponía hasta mero abajo.