

UNIVERSIDAD DEL VALLE DE GUATEMALA

Teoría de la Computación

Sección 20

Ing. Gabriel Brolo



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Proyecto 2

Jose Santisteban - 21153

Manuel Rodas - 21509

Sebastian Solorzano - 21826

Diseño de aplicación:

- Definición de Objetivos:
 - Implementación del algoritmo de simplificación de gramática.
 - Implementación del algoritmo CYK.

- Investigación y Análisis:

El algoritmo de Cocke-Younger-Kasami (CYK) es un algoritmo de análisis sintáctico utilizado en el procesamiento del lenguaje natural y en la teoría de la gramática formal para determinar si una cadena de entrada puede generarse a partir de una gramática libre de contexto (CFG por sus siglas en inglés). El algoritmo lleva los nombres de sus tres desarrolladores: John Cocke, Daniel Younger y Tadao Kasami.

El algoritmo CYK es especialmente útil para analizar la estructura sintáctica de las cadenas de entrada y determinar si cumplen con las reglas de una gramática específica. Funciona de la siguiente manera:

- Entrada de Datos: El algoritmo toma una cadena de entrada y una gramática libre de contexto como entrada. La gramática se representa generalmente en forma de reglas de producción.
- Creación de la Tabla: Se crea una tabla bidimensional (matriz) en la que las celdas representan subcadenas de la cadena de entrada. Cada celda contendrá una lista de símbolos no terminales que pueden generar la subcadena correspondiente. Inicialmente, las celdas están vacías.
- Llenado de la Tabla: El algoritmo CYK utiliza un enfoque bottom-up para llenar la tabla. Comienza por llenar las celdas correspondientes a las subcadenas de longitud 1 utilizando las reglas de producción de la gramática. Luego, avanza a las subcadenas de longitud 2, 3, y así sucesivamente, combinando las celdas de manera recursiva de acuerdo con las reglas de producción hasta que se complete la tabla.
- Verificación de Aceptación: Si el símbolo inicial de la gramática (generalmente representado como "S") está en la celda superior derecha de la tabla, significa que la cadena de entrada es válida según la gramática. Si la celda superior derecha contiene "S", entonces se puede generar la cadena de entrada utilizando las reglas de la gramática. En caso contrario, la cadena no es válida según la gramática.

El algoritmo CYK es especialmente útil en el análisis sintáctico de lenguaje natural, como en la construcción de analizadores sintácticos para identificar la estructura gramatical de oraciones en un idioma. También se utiliza en la compilación de lenguajes de programación para analizar la sintaxis de programas escritos en lenguajes formales. A través de este algoritmo, es posible determinar si una cadena sigue la estructura gramatical definida por una CFG.

- Desarrollo:
 - Inicio del Programa
 - Se aceptará una gramática CFG en un archivo .txt
 - Se analizarán las producciones y se almacenarán los datos de la gramática en distintas listas
 - Se trasladará el CDG a una cadena de texto legible
 - Luego comenzará el proceso de simplificación con el fin de llegar a la gramática en forma normal de Chomsky
 - Se eliminarán las producciones epsilon
 - Se eliminará cualquier producción unitaria de la gramática resultante.
 - Se eliminará cualquier símbolo inútil de la gramática resultante.
 - Se “traducirá” la gramática resultante a la Forma Normal de Chomsky (CNF)
 - Con el CNF se utiliza el algoritmo CYK para validar la cadena
 - Para el algoritmo se seguirán los pasos descritos previamente en la investigación del algoritmo de CYK
 - Fin del Programa

Discusión:

En este ejercicio, se buscará practicar de forma procedimental la teoría de lenguajes formales. Se experimentará con el algoritmo CYK, un método fundamental para analizar gramáticas libres de contexto (CFG) y determinar si una cadena de entrada puede generarse a partir de una gramática libre de contexto

La gramática que utilizaremos como ejemplo se presenta, pero antes de aplicar el algoritmo CYK, se debe de convertir la gramática original en la Forma Normal de Chomsky, un formato específico que simplifica el proceso de análisis sintáctico. Por lo cual se buscará crear varias funciones que logren este proceso.

El algoritmo CYK se basa en programación dinámica y es esencial para el procesamiento de lenguaje natural y la compilación de lenguajes de programación. A través de esta implementación, se podrá mostrar el resultado completo del algoritmo CYK, presentando la tabla y el árbol correspondientes a la gramática presentada por el usuario.

Algunos problemas presentados en dicho proyecto fueron la conversión de la gramática original a la Forma Normal de Chomsky, ya que, muchas veces se agregaba producciones

vacías y producciones repetidas, esto generaba problema al pasar al algoritmo CYK. Otro de los problemas que se presentaron fue que para la gramática original no aceptaba más de dos símbolos juntos para los no terminales, también se tuvo un problema al generar el parse tree, debido a que no se lograba visualizar correctamente la derivación de la cadena.

Resultados:

Gramática 1:

```
≡ gramatica.txt
1  V: E,A,T,B,F
2  T: +,*,id,(,),epsilon
3  S: E
4  P:
5      E -> T A,
6      A -> + T A | epsilon,
7      T -> F B,
8      B -> * F A | epsilon,
9      F -> ( E ) | id
```

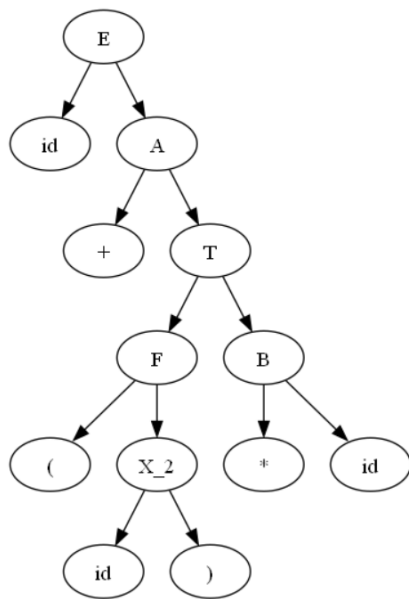
```
Nueva Gramática en Forma Normal de Chomsky:
V: F, E, T, A_0, B_0, F_0, F_1, A, B, X_0, X_1, X_2
T: +, *, id, (, )
S: E
P:
    F -> F_1 X_2 | id
    E -> F_1 X_2 | id | F B | T A
    T -> F_1 X_2 | id | F B
    A_0 -> +
    B_0 -> *
    F_0 -> )
    F_1 -> (
    A -> A_0 X_0 | A_0 T
    B -> B_0 X_1 | B_0 F
    X_0 -> T A
    X_1 -> F A
    X_2 -> E F_0
```

```
Validación de Cadenas:
Tabla CYK
['F', 'E', 'T'] None None None None None None
None ['A\\_0'] None None None None None
None None ['F\\_1'] None None None None
None None None ['F', 'E', 'T'] None None None
None None None None ['F\\_0'] None None
None None None None None ['B\\_0'] None
None None None None None None ['F', 'E', 'T']

Tabla CYK
['F', 'E', 'T'] None None None ['E', 'X\\_0', 'X\\_1'] None ['E', 'X\\_0', 'X\\_1']
None ['A\\_0'] None None ['A'] None ['A']
None None ['F\\_1'] None ['F', 'E', 'T'] None ['E', 'T']
None None None ['F', 'E', 'T'] ['X\\_2'] None None
None None None None ['F\\_0'] None None
None None None None None ['B\\_0'] ['B']
None None None None None None ['F', 'E', 'T']

La cadena pertenece a la gramática y es aceptada.
True

Tiempo de ejecución: 1.3413729667663574 segundos
```



Gramática 2:

```

≡ gramatica2.txt
1  V: S,A,B,C,V,P,N,D
2  T: cooks,drinks,eats,cuts,he,she,in,with,cat,dog,beer,cake,juice,meat,soup,fork,knife,oven,spoon,a,the
3  S: S
4  P:
5      S -> C A,
6      A -> A B,
7      A -> V C,
8      A -> cooks | drinks | eats | cuts,
9      B -> P C,
10     C -> D N,
11     C -> he | she,
12     V -> cooks | drinks | eats | cuts,
13     P -> in | with,
14     N -> cat | dog,
15     N -> beer | cake | juice | meat | soup,
16     N -> fork | knife | oven | spoon,
17     D -> a | the
  
```

Nueva Gramática en Forma Normal de Chomsky:

V: A, C, V, P, N, D, S, B

T: cooks, drinks, eats, cuts, he, she, in, with, cat, dog, beer, cake, juice, meat, soup, fork, knife, oven, spoon, a, the

S: S

P:

A -> cuts | eats | drinks | cooks | V C | A B

C -> she | he | D N

V -> cuts | eats | drinks | cooks

P -> with | in

N -> spoon | oven | knife | fork | soup | meat | juice | cake | beer | dog | cat

D -> the | a

S -> C A

B -> P C

Validación de Cadenas:

Tabla CYK

['D']	None	None	None	None	
None	['N']	None	None	None	
None	None	['A', 'V']	None	None	None
None	None	None	['D']	None	
None	None	None	None	['N']	

Tabla CYK

['D']	['C']	['S']	None	['S']	
None	['N']	None	None	None	
None	None	['A', 'V']	None	['A']	
None	None	None	['D']	['C']	
None	None	None	None	['N']	

La cadena pertenece a la gramática y es aceptada.

True

Tiempo de ejecución: 0.5806913375854492 segundos

