

Teoría de la Computación 2022

Proyecto 2

31.octubre.2022

Este proyecto consta de dos partes, una en el tema de gramáticas, y otra en el tema de funciones y cálculo lambda (similar al de la otra sección). La entrega del trabajo es en equipos.

1 Gramáticas

Investigar sobre el algoritmo CYK (Cocke-Younger-Kasami) para realizar el *parsing* de una gramática CFG.

Esta parte del proyecto consiste en implementar el algoritmo CKY para determinar si una frase simple en el idioma inglés es parte de un lenguaje generado por una gramática.

La gramática que usaremos como ejemplo es la siguiente:

```
S  → NP VP
VP → VP PP
VP → V NP
VP → cooks | drinks | eats | cuts
PP → P NP
NP → Det N
NP → he | she
V  → cooks | drinks | eats | cuts
P  → in | with
N  → cat | dog
N  → beer | cake | juice | meat | soup
N  → fork | knife | oven | spoon
Det → a | the
```

Objetivos

- Implementación del algoritmo CYK.

Especificaciones

a) Entrada:

- Solamente se ingresará textualmente una expresión sentencia (frase) en el idioma inglés w . Por ejemplo, se ingresa la expresión $w = \text{She eats a cake with a fork}$; $w = \text{The cat drinks the beer}$.
- Indique en su reporte ejemplos de frases en el lenguaje, y ejemplos que no estén en el lenguaje.

b) Salida:

- un SÍ, si la expresión w pertenece al lenguaje descrito por la gramática, o un NO en caso contrario.
- También se debe indicar el tiempo que se tarda el algoritmo en realizar dicha validación.
- Construir el *parse tree* de la expresión w . (Para ello, basta extender el algoritmo CYK, e ir guardando los nodos de la tabla dinámica, en lugar sólo del valor booleano).

Nota: El algoritmo CYK requiere que la gramática a usar esté en la Forma Normal de Chomsky. No se olviden de convertir la gramática a su Forma Normal de Chomsky, antes de implementar el algoritmo.

El algoritmo requiere el uso de *programación dinámica*. Deberán investigar en qué consiste y cómo se implementa.

Las frases aceptadas son sintácticamente correctas (nadie garantiza que sean semánticamente correctas).

2 Funciones Lambda

Investigar sobre el cálculo lambda y las funciones lambda (funciones anónimas). Esta parte del proyecto consiste en desarrollar un programa en Python en el que se definan las siguientes expresiones usando funciones lambda (anónimas):

- a) `cero(f, x)`
- b) `uno(f, x)`
- c) `dos(f, x)`
- d) `tres(f, x)`
- e) `sucesor(n, f, x)`
- f) `suma(a, b, f, x)`
- g) `multiplicación(a, b, f, x)`
- h) `potencia(a, b, f, x)`

Luego de las definiciones incluya ejemplos de uso de sus funciones sucesor, suma, multiplicación y potencia. Estas funciones deben operar sobre números también definidos como funciones lambda, tales como los requeridos aquí (`cero`, `uno`, `dos`, `tres`) o cualquier otro número que desee (tendría que agregar la definición de dicho número). Como los “números lambda” deben recibir una función para aplicar y un argumento sobre el cual aplicarla (i.e., `f` y `x`), defina también las siguientes funciones y pruebe sus ejemplos con cada una de ellas:

```
alpha = lambda x: x + 1
beta  = lambda x: 2 * x
```

Despliegue los resultados de sus ejemplos y explique las diferencias entre usar *alpha* y *beta*.
¿Con cuál de éstas y con qué argumento se obtienen los resultados esperados de las operaciones definidas?

Ponderación

El proyecto en total tiene un valor de 20 puntos, distribuidos de la siguiente forma:

| Característica | Ponderación |
|---|-------------|
| Implementación de la parte dinámica (algoritmo) | 2 |
| Implementación del algoritmo CYK (algoritmo) | 5 |
| Construcción del Parse Tree (algoritmo) | 3 |
| Construcción de funciones (a)-(e) | 3 |
| Construcción de funciones (f)-(h) | 5 |
| Informe con ejemplos | 2 |
| Total | 20 |

Debe entregarse documentación para tener derecho a nota. La documentación debe de abarcar:

- Diseño de la aplicación (el método de modelado a discreción del desarrollador o desarrolladora).
- Discusión (obstáculos encontrados, recomendaciones, etc.).
- Ejemplos y pruebas realizadas.

Referencias

- https://en.wikipedia.org/wiki/CYK_algorithm
- https://en.wikipedia.org/wiki/Phrase_structure_rules
- <https://www.geeksforgeeks.org/cyk-algorithm-for-context-free-grammar/>
- <https://www.geeksforgeeks.org/cocke-younger-kasami-cyk-algorithm/>
- <https://www.xarg.org/tools/cyk-algorithm/>
- <https://web.cs.ucdavis.edu/~rogaway/classes/120/winter12/CYK.pdf>
- https://www.w3schools.com/python/python_lambda.asp
- <https://riptutorial.com/es/python/example/16723/lambda-recursiva-utilizando-variable-asignada>
- <https://riptutorial.com/es/python/example/3226/expresion-condicional--o--el-operador-ternario-->
- <https://realpython.com/python-lambda/>
- <https://www.geeksforgeeks.org/nested-lambda-function-in-python/#:~:text=In%20Python%2C%20anonymous%20function%20means,is%20called%20Nested%20Lambda%20Function.>
- <https://www.learnbyexample.org/python-lambda-function/>