Universidad de La Laguna

E. T. S. DE INGENIERÍA INFORMÁTICA
Departamento de Estadística, I.O. y Computación
Teoría de Autómatas y Lenguajes Formales

PRACTICA 12: Análisis Sintáctico

12.1. Requisito de codificación

Constantes
En el código fuente nunca deberían aparecer de forma explícita constantes literales.

Defina la constante (`const`) y utilice su identificador en lugar de su valor. Esta regla se aplica tanto a constantes numéricas (del tipo que sea) como de otros tipos (caracteres, strings, etc.).

La única excepción a la regla es que, por ser muy habitual su utilización, permitiremos el uso de las constantes 0 y 1 (es muy frecuente su uso como inicio de cualquier bucle, por ejemplo).

12.2. Introducción

Dadas una gramática independiente del contexto, $G \equiv (V, \Sigma, S, P)$ y una frase $x \in \Sigma^*$, el problema del análisis sintáctico consiste en determinar si la frase pertenece al lenguaje generado por la gramática, es decir, si $x \in L(G)$.

El algoritmo de Cocke-Younger-Kasami (CYK) es un algoritmo de programación dinámica con una complejidad cúbica en el tamaño de la frase a analizar que resuelve este problema. Un pseudocódigo para el algoritmo CYK es el que se presenta en la Figura 12.1

Entrada: una cadena $w = a_1 a_2 \dots a_n$ y una CFG escrita en forma normal de Chomsky.
Salida: el algoritmo determina si $w \in L(G) : w \in L(G) \Leftrightarrow S \in V_{1n}$

```
1  for i := 1 to n do
2     V_i1 := {A|A → a es una producción y el i-ésimo símbolo de w es a}
3  for j := 2 to n do
4     for i := 1 to n - j + 1 do
5        begin
6        V_ij := ∅;
7        for k := 1 to j - 1 do
8           V_ij := V_ij ∪ {A|A → BC es una producción, B → V_ik, C → V_i+k,j-k};
9        end
```

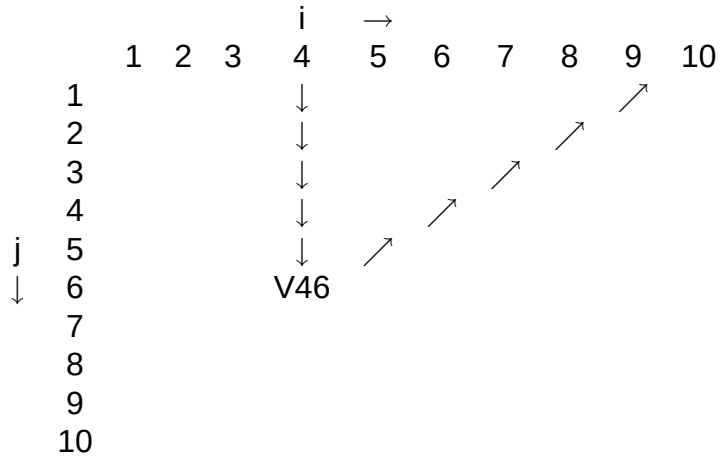
Figura 12.1: El algoritmo CYK

El algoritmo construye los conjuntos V_{ij} que son los conjuntos de símbolos no terminales de la gramática que pueden generar las subcadenas w_{ij} de x siendo i la posición de comienzo de la subcadena y j su longitud: $|w_{ij}| = j$. Es decir, $V_{ij} = \{A \in V | A \Rightarrow^* w_{ij}\}$.

Si $|x| = n$, el conjunto V_{1n} será por tanto el conjunto de símbolos no terminales que pueden derivar la cadena $w_{1n} = x$. Así pues, si el símbolo de arranque de la gramática, S, pertenece a V_{1n}, entonces $x \in L(G)$; en caso contrario, $x \notin L(G)$.

El algoritmo funciona construyendo una matriz triangular superior que en cada posición contiene un conjunto Vij (el índice i recorre las columnas y j las filas de esa matriz). El bucle de las líneas 1 y 2 de la Figura 12.1 rellena la primera fila de la matriz mientras que los bucles anidados que recorren i y j en las líneas 3 y 4 rellenan el resto de posiciones de la matriz.

Para determinar los símbolos no terminales que han de introducirse en V_{ij}, el algoritmo compara las posiciones V_{ik} y V_{i+kj-k}. Este patrón corresponde a moverse en la matriz según el esquema que muestra la Figura 12.1 para el cálculo de V_{46}, es decir, moverse de arriba hacia abajo en la columna i y de forma ascendente y hacia la derecha desde la posición V_{ij} comparando los contenidos de cada par de posiciones de la matriz.



Cuadro 12.1: Valores que intervienen en el cómputo de V_{46}

Por ejemplo, consideremos la gramática siguiente:
$S \rightarrow AB|BC$
$A \rightarrow BA|a$
$B \rightarrow CC|b$
$C \rightarrow AB|a$

y la cadena $w = baaba$ y veamos utilizando la Figura 12.2 cómo se calcula el contenido de la posición V_{24} suponiendo que las tres primeras filas de la matriz han sido previamente calculadas.

Inicialmente, $V_{24} = \emptyset$. Se comparan las posiciones $V_{21} = \{A, C\}$ y $V_{33} = \{B\}$ y se buscan reglas de producción que en la parte derecha contengan la secuencia AB o CB. Encontramos $S \rightarrow AB$ y $C \rightarrow AB$, así que hacemos $V_{24} = V_{24} \cup \{S, C\} = \{S, C\}$. A continuación se consideran las posiciones $V_{22} = \{B\}$ y $V_{42} = \{S, A\}$ y se buscan reglas

			i	→		
		b	a	a	b	a
		1	2	3	4	5
	1	B	A, C	A, C	B	A, C
j	2	S, A	B	S, C	S, A	
↓	3	∅	B	B		
	4	∅	S, A, C			
	5	S, A, C				

Cuadro 12.2: Tabla de análisis CYK

de producción con parte derecha BS o BA. La única regla de esta forma es $A \rightarrow BA$, por lo que $V_{24} = V_{24} \cup \{A\} = \{S, C, A\}$. Finalmente exploramos $V_{23} = \{B\}$ y $V_{51} = \{A, C\}$ y se consideran las producciones con BA o BC en su parte derecha. Hallamos $A \rightarrow BA$ y $S \rightarrow BC$, por lo que $V_{24} = V_{24} \cup \{A, S\} = \{S, C, A\}$. Así pues, la posición V_{24} de la tabla ha de contener los símbolos no terminales $\{S, C, A\}$

12.3. Ficheros de descripción de las gramáticas

Los ficheros de texto que describen gramáticas tienen la extensión `.gra` y en ellos se listan las producciones de la gramática. Los símbolos no terminales de la gramática se representan mediante letras mayúsculas, y los terminales mediante letras minúsculas y caracteres no alfabéticos (`a, b, x, z, +, -, *, %, 0`, etc.)

Para distinguir los elementos de una cadena $\alpha \in (\Sigma \cup V)^*$, es necesario separarlos mediante espacios o tabulaciones. El fichero contendrá en cada línea una serie de producciones de la forma:
A -> $\alpha|\beta|\ldots|\gamma$
correspondientes al símbolo no terminal A.

La cadena vacía (ϵ) se representará mediante el carácter `~`.

Se supondrá que el símbolo de arranque de la gramática será el símbolo no terminal que aparece en la parte izquierda de la primera producción listada.

Cualquier línea que comience con los caracteres `//` será ignorada, puesto que representará un comentario. Los comentarios se utilizarán para anotar en los ficheros características significativas de las gramáticas.

Así para representar la siguiente gramática:
$S->SabT|Ta|\epsilon|b$
$T->\epsilon|b$
en la que $\Sigma = \{a, b\}$, y $V = \{S, T\}$ es el conjunto de símbolos no terminales, siendo S el símbolo de arranque de la gramática, el fichero tendrá el contenido que muestra el Listado 12.1.

```
// Fichero que representa a la gramática:
//
```

```
// S -> SabT | Ta | ~ | b
// T -> ~ | b
//
S -> S a b T  | Ta | ~ | b
T -> ~ | b
```

Listado 12.1: Fichero de representación de una gramática

En la sección de descargas de las páginas de la asignatura encontrará varios ficheros con ejemplos de gramáticas.

12.4. La práctica

El objetivo de esta práctica es diseñar un programa que dada una gramática independiente del contexto escrita en forma normal de Chomsky y una serie de frases, determine si cada una de las frases pertenece o no al lenguaje generado por la gramática.

El programa tomará como entrada un fichero que representa la gramática a considerar y otro fichero conteniendo un conjunto de frases a analizar. Para cada frase del fichero de entradas el programa ha de construir de forma dinámica la tabla de análisis CYK e indicar simplemente si la frase considerada pertenece o no al lenguaje definido por la gramática. Ambos ficheros de entrada se introducirán en la línea de comandos de llamada al programa.

12.5. Kakuy

Kakuy es una aplicación disponible solamente para S.O. Windows que permite trabajar con gramáticas independientes del contexto (se trata de una herramienta similar a ToTALF). En concreto permite visualizar la evolución de un análisis sintáctico mediante el algoritmo CYK.

Si está interesado en Kakuy, puede encontrar un enlace a la aplicación en la sección de enlaces / herramientas del portal web de TALF.