

# FORMATOS DE LOS FICHEROS PARA EL MÓDULO DE ANÁLISIS SINTÁCTICO DE LA PRÁCTICA DE PROCESADORES DE LENGUAJES

## 1. Formato del fichero de gramática de tipo 2

Deberá proporcionarse un fichero que contenga la gramática utilizada para construir el Analizador Sintáctico de la Práctica. La gramática debe ser de tipo 2 (independiente del contexto) y deberá estar escrita siguiendo el siguiente formato.

El lenguaje diferencia mayúsculas de minúsculas. En general, los delimitadores (espacios en blanco, tabuladores y saltos de línea) se utilizan para separar los distintos elementos del fichero, salvo que se indique lo contrario.

Una gramática consta de cuatro componentes principales:

- Conjunto de símbolos terminales.
- Conjunto de símbolos no terminales o metanociones.
- Un axioma o símbolo inicial.
- Un conjunto de producciones o reglas.

Estos cuatro elementos deberán aparecer reflejados en el fichero de la gramática.

Cada conjunto será identificado mediante una palabra clave específica, que serán *Terminales*, *NoTerminales*, *Axioma* y *Producciones*. Estas palabras no son reservadas y pueden ser utilizadas para representar otros símbolos.

Los conjuntos de símbolos terminales, no terminales y el axioma deben aparecer antes que el conjunto de producciones.

La representación de los diferentes conjuntos será:

- **Conjuntos de símbolos terminales:** Se indicará con la palabra *Terminales*, seguida de un símbolo igual (=) y un símbolo de llave abierta ( { ). A continuación aparecerá la lista de símbolos terminales separados por delimitadores. La finalización del conjunto de símbolos terminales se considerará cuando aparezca un símbolo de llave cerrada ( } ) seguido de un salto de línea. Por tanto, si se va a utilizar el símbolo de llave cerrada ( } ) como parte del conjunto, no se puede poner como último símbolo de una línea del conjunto.
- **Conjuntos de símbolos no terminales:** Se indicará con la palabra *NoTerminales*, seguida de un símbolo igual (=) y un símbolo de llave abierta ( { ). A continuación aparecerá la lista de símbolos no terminales separados por delimitadores. Para finalizar el conjunto, se utilizará un símbolo de llave cerrada ( } ). No puede aparecer en este conjunto un símbolo que aparezca en el conjunto de símbolos terminales.
- **Axioma:** Se indicará con la palabra *Axioma*, seguida de un símbolo igual (=) y el símbolo no terminal que se utilizará como axioma. Este símbolo debe estar contenido en el conjunto de símbolos no terminales.
- **Producciones:** Se indicará con la palabra *Producciones*, seguida de un símbolo igual (=) y un símbolo de llave abierta ( { ). A partir de la siguiente línea se escribirán las diferentes reglas, una por línea (pudiendo aparecer líneas en blanco). Para finalizar el conjunto de producciones, se pondrá un símbolo de llave cerrada ( } ) en una línea nueva.

El formato de los símbolos será (las letras minúsculas y las mayúsculas son diferentes):

- **Símbolos no terminales:** Combinación de letras, números y símbolos de subrayado ( \_ ).



- **Símbolos terminales:** Combinación de letras, números o símbolos, excepto delimitadores. Además, para representar la cadena vacía ( $\lambda$ ) se utiliza la palabra reservada *Lambda*<sup>1</sup>.

Cada regla estará formada por un antecedente (que, para que la gramática sea tipo 2, deberá estar formado únicamente por un símbolo no terminal), seguido por uno o varios delimitadores (que no sean saltos de línea). Luego aparecerá una flecha ( -> ) seguida de uno o varios delimitadores y después aparecerá el consecuente de la regla, que estará formado por uno o varios símbolos terminales o no terminales separados por espacios en blanco o tabuladores.

El fichero podrá contener comentarios colocados entre los distintos conjuntos, nunca dentro de ellos, aunque sí se podrán escribir comentarios detrás de cada regla del conjunto de producciones, en la misma línea que éstas. Los comentarios se iniciarán con los símbolos `////` y terminarán en el final de la línea.

Por tanto, la estructura del fichero de la gramática sería como la siguiente:

```

Terminales = { term1 term2 term3 ..... termN }
NoTerminales = { nterm1 nterm2 nterm3 ..... ntermM }
    //// Comentario entre conjuntos
Axioma = ntermX
Producciones = {      //// Los ConsXY son terminales o no terminales (o lambda)
nterm1 -> Cons1A Cons1B ... Cons1I    //// Comentario
nterm2 -> Cons2A Cons2B ... Cons2J    //// Comentario
. . .
ntermM -> ConsMA ConsMB ... ConsMK    //// Comentario
}

```

## 2. Formato del fichero de parse

El formato del fichero donde se almacenará el *parse* generado por el Analizador de la Práctica es muy sencillo: únicamente contendrá dos elementos, el primero será una palabra (que puede estar formada por una única letra mayúscula o minúscula) que empiece por *A* si se ha generado utilizando un analizador sintáctico ascendente o una palabra que empiece por *D* si se ha generado utilizando un analizador sintáctico descendente. En segundo lugar (precedido de al menos un delimitador), deberá aparecer un conjunto de números que representarán las reglas del *parse*. Estos números estarán separados por delimitadores (espacios en blanco, tabuladores y saltos de línea) y se dará por concluido el fichero del *parse* cuando se lea el final del fichero.

Por tanto, un ejemplo de la estructura del fichero del *parse* sería:

```
ascendente 2 21 1 4 7 11 27 8
```

Es muy importante hacer notar que los números indicados en el fichero del *parse*, deberán corresponder exactamente con el orden de las reglas indicadas en el fichero de la gramática. De esta manera, en el ejemplo previo, si el primer número del *parse* es el 2, quiere decir que corresponde con la segunda regla del fichero de la gramática (en el ejemplo de antes, nterm2).

<sup>1</sup> En la versión Linux de VASt es necesario declarar la palabra *Lambda* como un símbolo dentro del conjunto de terminales. En la versión Windows no hay que hacerlo.

