Proyecto de Autómatas y Lenguajes

Curso 2018-2019

Práctica 3: ANÁLISIS SINTÁCTICO

Fecha de entrega según calendario publicado en moodle

Objetivo de la práctica:

El objetivo de la práctica es la codificación de un analizador sintáctico mediante el uso del lenguaje de programación C y la herramienta de ayuda a la generación de analizadores sintácticos ***bison***.
Este analizador será utilizado por el compilador del lenguaje *Ómicron* que va a desarrollarse durante el curso.
Junto con el analizador sintáctico se entregará un programa escrito en C para probarlo.

Para ello debe seguir las indicaciones que se describen a continuación.

Desarrollo de la práctica:

1. Codificación de una especificación para *bison*

El estudiante deberá escribir un fichero con nombre ***omicron.y*** que sea una especificación correcta para la entrada de la herramienta ***bison***.

Cuando se compile el fichero ***omicron.y*** pueden aparecer mensajes de conflictos. Estos conflictos se refieren a los que encuentra el autómata a pila generado por la herramienta ***bison***. Para obtener información acerca de ellos, se debe compilar el fichero ***omicron.y*** de la siguiente manera:

bison -d -y -v omicron.y

con el flag ***-v*** se genera el fichero ***y.output*** que contiene la descripción completa del autómata y de los conflictos (si existieran).

2. Modificación de la especificación para *flex*

El estudiante deberá modificar el fichero ***omicron.l*** que haya utilizado en el desarrollo del analizador morfológico para poder ser enlazado con el fichero ***omicron.y***. Para ello, utilizará como guía las explicaciones que su profesor de prácticas realizará en su laboratorio.

3. Codificación del programa de prueba

El estudiante deberá escribir (en C) un programa de prueba del analizador sintáctico generado con ***bison***, con los siguientes requisitos:

- Nombre del programa fuente: ***pruebaSintactico.c***
- Al ejecutable correspondiente, se le invocará de la siguiente manera:

pruebaSintactico <nombre fichero entrada> <nombre fichero salida>

- La estructura de los ficheros de entrada/salida se explica a continuación.

Nota: es muy importante respetar el formato de los ficheros de salida puesto que la corrección se realizará de manera automática.

4. Descripción del fichero de entrada

El fichero de entrada contiene texto plano correspondiente a un programa escrito en el lenguaje Ómicron (no necesariamente correcto).

5. Descripción del fichero de salida

El fichero de salida se compone de un conjunto de líneas de dos tipos. En concreto:

- Una línea por cada **TOKEN** desplazado (reconocido) por el analizador morfológico.
- Una línea por cada **PRODUCCIÓN** reducida en el proceso de análisis sintáctico.

Cada línea correspondiente a un **TOKEN** desplazado debe contener la siguiente información y formato:

;D: <token>

donde

- debe escribirse literalmente para que le sea cómodo gestionar la impresión de estos textos en el compilador completo (corresponde con el símbolo de inicio de los comentarios de línea en NASM).
- **<token>** es el fragmento de entrada reconocido por el analizador morfológico.
- **D:** y **<token>** van separados por un tabulador.

Cada línea correspondiente a una regla reducida tendrá el formato siguiente:

;R: <regla>

donde

- **<regla>** es el texto de la regla reducida según aparece en la gramática.
- **R<nº regla>** y **<regla>** van separados por un tabulador.
- Los elementos que forman **<regla>** irán separados por espacios.

Recuerda que los símbolos "**;**" son para que la línea, si la dejas en el fichero ensamblador, sea un comentario.

6. Gestión de errores

Cuando se produzca un error sintáctico el analizador creado con ***bison*** deberá imprimir una línea por la salida estándar de error con el siguiente formato:

`ERROR SINTÁCTICO:<nº línea>:<nº carácter>`

donde

- **<nº linea>** es la línea en el fichero de entrada donde aparece el último **TOKEN** reconocido por el analizador léxico.
- **<nº carácter>** es el carácter en la correspondiente línea donde empieza el último **TOKEN** reconocido por el analizador morfológico.

Cuando se produzca un error morfológico, el analizador morfológico deberá imprimir por la salida estándar un mensaje del error ocurrido, que a fin de facilitar la corrección automática deberá ajustarse exactamente a lo especificado y en ningún caso contendrá acentos o eñes. Sólo se controlarán dos errores morfológicos: caracteres no permitidos e identificadores demasiado largos. Para cada error el analizador morfológico imprimirá por el stderr un mensaje de error según el siguiente formato:

- `"ERROR MORFOLÓGICO:<nº línea>:<nº carácter>:IDENTIFICADOR DEMASIADO LARGO (<token>)"`
- `“ERROR MORFOLÓGICO:<nº línea>:<nº carácter>:CARÁCTER INVÁLIDO (<token>)"`

En el caso de que se imprima un mensaje de error morfológico, no deberá imprimirse otro mensaje de error sintáctico.

7. Ejemplos

Para probar inicialmente el código desarrollado se facilitan algunos ficheros de entrada con sus correspondientes salidas que puedes encontrar en Moodle. Suponiendo que el programa ejecutable se denomina *pruebaSintactico*, la siguiente instrucción:

```
pruebaSintactico entrada_sin_1.ol misalida_sin_1.txt
```

debe generar un fichero con nombre *misalida_sin_1.txt* que sea idéntico al fichero *salida_sin_1.txt*. En particular, al hacer:

```
diff -bB salida_sin_1.txt misalida_sin_1.txt
```

no debe encontrarse ninguna diferencia entre los dos ficheros.

Entrega de la práctica:

Se entregará a través de Moodle un único fichero comprimido (*.zip*) que deberá cumplir los siguientes requisitos:

- Deberá contener todos los fuentes (ficheros *.l, .y, .h y .c*) necesarios para resolver el enunciado propuesto. No es necesario incluir los ficheros *lex.yy.c* ni *y.tab.c* puesto que puede generarse a partir del *.l* y del *.y* respectivamente.
- Deberá contener un fichero *Makefile* compatible con la herramienta make que para el objetivo ***all*** genere el ejecutable de nombre ***pruebaSintactico***.
- El nombre del fichero *.zip* seguirá el siguiente formato***:***

  sintactico_YYYY_X.zip

  donde YYYY será el número del grupo (por ejemplo 1311, 1312, 1362,...) y X será el número del grupo de trabajo que tengáis asignado.

MUY IMPORTANTE: UN ANALIZADOR SINTÁCTICO CON CONFLICTOS SE CONSIDERARÁ SUSPENSO.

Se recuerda al alumno que **las prácticas son incrementales por lo que una práctica aprobada pudiera conllevar errores importantes en las siguientes fases. Es responsabilidad del alumno subsanar totalmente los errores en cada fase, dando correcto cumplimiento a los requerimientos de los enunciados.**