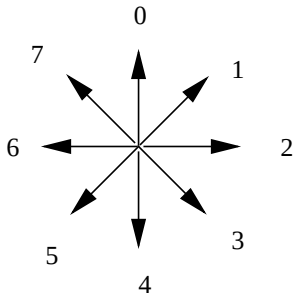
13048 : Procesadores de Lenguaje
Práctica 1: Diseño de un Analizador Léxico y Sintáctico
Curso: 2005-2006

Objetivos: Diseñar e implementar un Analizador Léxico y un Analizador Sintáctico predictivo recursivo para el reconocimiento de un lenguaje de diseño gráfico por ordenador que permite la generación de figuras gráficas.

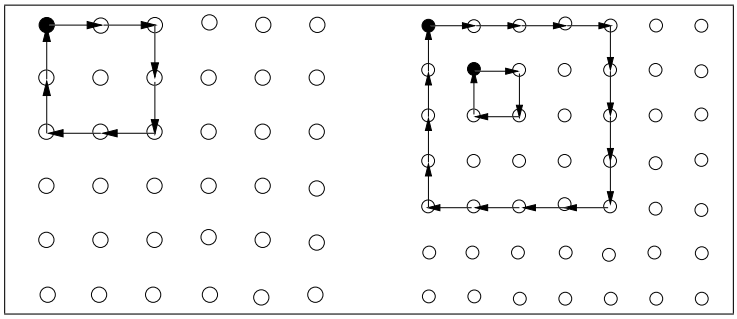
Descripción de la Práctica

Uno de los métodos para describir el contorno de los objetos que aparecen en una imagen es mediante el código cadena. Los códigos de cadenas se usan para representar un contorno mediante una secuencia de segmentos de líneas de una longitud y dirección específica (las primitivas). Cada segmento se codifica usando una numeración como se muestra a continuación:



Podemos generar un código de cadena siguiendo el contorno del objeto en la dirección de las agujas del reloj y asignando una dirección a cada segmento que conecta cada par de puntos. Por ejemplo, para el cuadrado de la figura el código adena correspondiente es 2 2 4 4 6 6 0 0.

Supongamos que el formato de ficheros de un sistema de diseño gráfico asistido por ordenador es el siguiente. Cada figura viene determinada por las palabras `begin_figure` y `end_figure`. Para cada figura es necesario dar el punto del inicio y el código cadena correspondiente. Una figura puede además contener subfiguras que a su vez pueden contener nuevas subfiguras. En cada fichero de datos pueden aparecer varias figuras. Cada figura puede tener además una etiqueta en donde se puede colocar un nombre (un identificador).



Por ejemplo para la figura de la izquierda el fichero de datos es:

```
beginfigure
      startpoint 1 , 1
      chaincode 2 2 4 4 6 6 0 0
      label cuadrado
endfigure
```

Y para la figura de la derecha el fichero de datos es:

```
beginfigure
      startpoint 1 , 1
      chaincode 2 2 2 2 4 4 4 4 6 6 6 6 0 0 0 0
      label cuadrado
      beginfigure
           startpoint 2 , 2
           chaincode 2 4 6 0
      endfigure
%esto es un cuadrado de lado 4 con otro dentro de lado 1
endfigure
```

Análisis Léxico (1ª sesión)

Se trata de implementar un analizador léxico, en el que exista una función que devuelva el siguiente token dentro del fichero de datos, es decir, que devuelva el tipo de token en forma de constante entera, y su lexema, en forma de cadena de caracteres. En esta primera fase de la práctica, y con la finalidad de comprobar que funciona correctamente, esta función será llamada por la función main, que solicitará nuevos tokens hasta que se agote el texto del fichero de entrada. Vuestro programa deberá imprimir una lista de tokens de la forma (tipo_de_token, lexema) como resultado, deteniéndose en el caso de un error léxico e indicando la línea y columna del texto fuente donde éste se produjo.

Los componentes léxicos a reconocer son los siguientes:

Constantes

Sólo existe un tipo de constante numérica definida por:

TKN_NUM = digito+

y un tipo de identificadores:

TKN_ID = letra (letra — digito) *

siendo digito = 0 | 1 | ... | 9

letra = a | ... | z | A | B | ... | Z

Tokens de Puntuación

, TKN_COMA

Tokens asociados a palabras reservadas:

beginfigure TKN_BEGIN
endfigure TKN_END
startpoint TKN_START
chaincode TKN_CODE
label TKN_LABEL

También reconoceremos comentarios dentro del fichero de datos. Un comentario empieza con el carácter % y termina con el final de línea (como en MATLAB).

NOTA: La distinción entre identificadores y palabras reservadas debe hacerse mediante el método de añadir las palabras reservadas a una Tabla de tokens predefinidos y comprobar cada candidato si es identificador o palabra reservada.

Análisis Sintáctico (2ª y 3ª sesión)

Partiendo del analizador léxico ya programado, se trata de implementar un analizador sintáctico direccional determinista descendente basado en el uso de una gramática LL(1). El programa deberá imprimir como resultado la lista de producciones que generan la derivación más a la izquierda del fichero de entrada. En el caso de que haya errores sintácticos, el programa debe proporcionar la información sobre la línea y columna del texto original donde se produjo el error, e intentar efectuar una recuperación del error en modo de pánico, informando de lo que ha encontrado en la entrada y lo que se esperaba en ese momento.

PRIMER APARTADO

El primer apartado de esta parte consiste en diseñar una Gramática LL(1) que genere el lenguaje a reconocer que viene explicado a continuación. Se trata de un lenguaje para la generación de un fichero de datos con figuras. Las consideraciones a tener en cuenta para la definición de la gramática que representa la estructura del fichero son las siguientes:

- Un fichero se compone de al menos una figura.
- Una figura puede tener opcionalmente una o varias subfiguras. Y una subfigura a su vez puede o no tener una o más subfiguras.
- El campo del punto de inicio no puede estar vacío.
- Un código cadena debe estar al menos formado por un segmento.
- La etiqueta de la figura no es obligatoria. La etiqueta es un identificador.
- El orden de los campos de la figura siempre es el mismo: el punto de inicio, el código cadena, la etiqueta (si existe) y las subfiguras (si existen).

SEGUNDO APARTADO

Como segunda parte se trata de construir el analizador sintáctico con las características especificadas al anteriormente.

Material de entrega de la práctica

1. La implementación del analizador léxico y sintáctico que reconozca el formato de los ficheros de dibujo.
2. Una gramática LL(1) que genere el lenguaje del fichero de datos (entregar un documento con las producciones y la comprobación de que se trata de una gramática LL(1)).
3. Implementar un método de recuperación de errores sintácticos en modo de pánico, informando de lo que ha encontrado en la entrada y lo que se esperaba en ese momento (conjunto de símbolos de PREDICCION).
4. Para aquellos ficheros correctos, dibujar en la pantalla las figuras. Si quieres puedes hacerlo en una página Web.
5. Los errores a contemplar son tanto de tipo léxico, sintáctico como semánticos: i) Error léxico si se encuentra un carácter desconocido no contemplado en el vocabulario del lenguaje. ii) Error de sintaxis en el formato de la figura, la entrada no puede ser generada por la gramática. iii) Error semántico si el código cadena es incorrecto (se han usado dígitos diferentes del 0 al 7).

Duración: 3 sesiones. Entrega: 30 de Diciembre, todos los grupos.

Enviar al profesor correspondiente por correo electrónico. e-mail: elena.diaz@uv.es, ariadna.fuertes@uv.es, j.francisco.garcia@uv.es