

El generador de analizadores léxicos *lex*.

III

Teoría de Autómatas y lenguajes formales
Federico Simmross Wattenberg (fedesim@infor.uva.es)
Universidad de Valladolid

1. Más sobre contextos: reglas condicionales

Además de las mínimas posibilidades ya vistas (los metacaracteres ‘^’, ‘\$’ y ‘/’), *lex* incorpora un mecanismo más potente que permite reconocer patrones “sólo bajo ciertas circunstancias”: las reglas condicionales.

Lex permite definir contextos (llamados “estados”), y asociar ciertas reglas a uno o más de estos contextos, de forma que sólo se las tenga en cuenta cuando esté activo uno de sus estados. Consideremos el problema:

“En un programa en C, elimínense los comentarios de tipo // que se encuentren dentro de un comentario de tipo / */”*

Este problema puede resolverse sin reglas condicionales, buscando comentarios */* */* con un patrón *lex*, y dentro de su acción en *C*, buscando a mano la secuencia *//* y eliminando los caracteres hasta el fin de la línea, pero *lex* puede hacerlo automáticamente, si especificamos una regla que encaje con un comentario *//* y que sólo se active dentro de un comentario */* */*:

```
%s COMENTARIO

%%
<INITIAL>"/*"                               {ECHO; BEGIN (COMENTARIO); }
<COMENTARIO>"/" (.*) $                       ;
<COMENTARIO>"/"                               {ECHO; BEGIN (INITIAL); }
```

- En la sección de definiciones, declaramos el estado “COMENTARIO” mediante la directriz `%s <nombre_de_estado>`.
- En la sección de reglas, para indicar a *lex* que una regla sólo debe tenerse en cuenta en un determinado estado, se la precede de la cadena “<nombre_de_estado>”. Si una regla ha de tenerse en cuenta en más de un estado, se puede especificar una lista de estados separados por comas: “<estado1, estado2, estado3, ...>”.
- En el campo de acción, podemos activar un estado mediante la macro `BEGIN(nombre_de_estado)`.

- El estado por defecto, aquél en el que sólo se ejecutan las reglas incondicionales, se llama `INITIAL`, así que para desactivar un estado, hacemos un `BEGIN (INITIAL)`.
- Obsérvese que las reglas incondicionales (en este ejemplo, la única es la regla por defecto) se ejecutan tanto en el estado `INITIAL` como en el estado `COMENTARIO`. Esto se debe a que hemos declarado el estado `COMENTARIO` mediante la directriz `%s`, que indica un estado *inclusivo*, en lugar de utilizar la otra posibilidad, `%x`.
- Los estados declarados con `%x` son estados *exclusivos*. Dentro de éstos, sólo se tienen en cuenta las reglas asociadas explícitamente a este estado, y se ignoran todas las demás, incluidas las reglas incondicionales.

2. El fichero de entrada `yyin`

Por defecto, todos los analizadores generados por `lex` consideran que el fichero de entrada que deben procesar es la entrada estándar (el stream de C llamado `stdin`), de manera que típicamente es necesario redirigir la entrada estándar para poder aplicar un analizador a un fichero cualquiera. Sin embargo, no es del todo cierto que `lex` utilice `stdin` como fichero de entrada. En realidad, la función `yylex()` siempre lee el texto a procesar de un stream llamado `yyin`; lo que ocurre es que por defecto, `yyin` apunta a `stdin`.

Si alteramos `yyin` y lo hacemos apuntar a otro fichero, conseguiremos que el analizador generado por `lex` procese el texto de dicho fichero en lugar de la entrada estándar. Es típico en todo programa `lex` incluir una función `main()` que compruebe si el usuario ha suministrado algún argumento en la línea de comando para utilizarlo como fichero de entrada:

```

.
.
.
%%
main(int argc, char **argv) {
    if (argc > 1)
        yyin = fopen(argv[1], "r");

    yylex();
    return 0;
}

```

Así, podemos proporcionar un fichero como argumento, y en caso de que no lo hagamos, `yyin` seguirá apuntando a la entrada estándar. El analizador generado funcionará correctamente tanto si proporcionamos un nombre de fichero como si no:

\$./a.out quijote.txt Proporcionamos un fichero como argumento

\$./a.out <quijote.txt No pasamos argumentos. Se usará la entrada estándar.

Nótese que este ejemplo no controla los errores que pueden producirse si hay problemas al abrir el fichero.

3. Ejercicios

- 1- Pasar a mayúsculas las letras pares de las palabras impares de las líneas con número de orden primo de un fichero que se especificará como primer argumento al analizador.