LEX. Generador de reconocedores léxicos

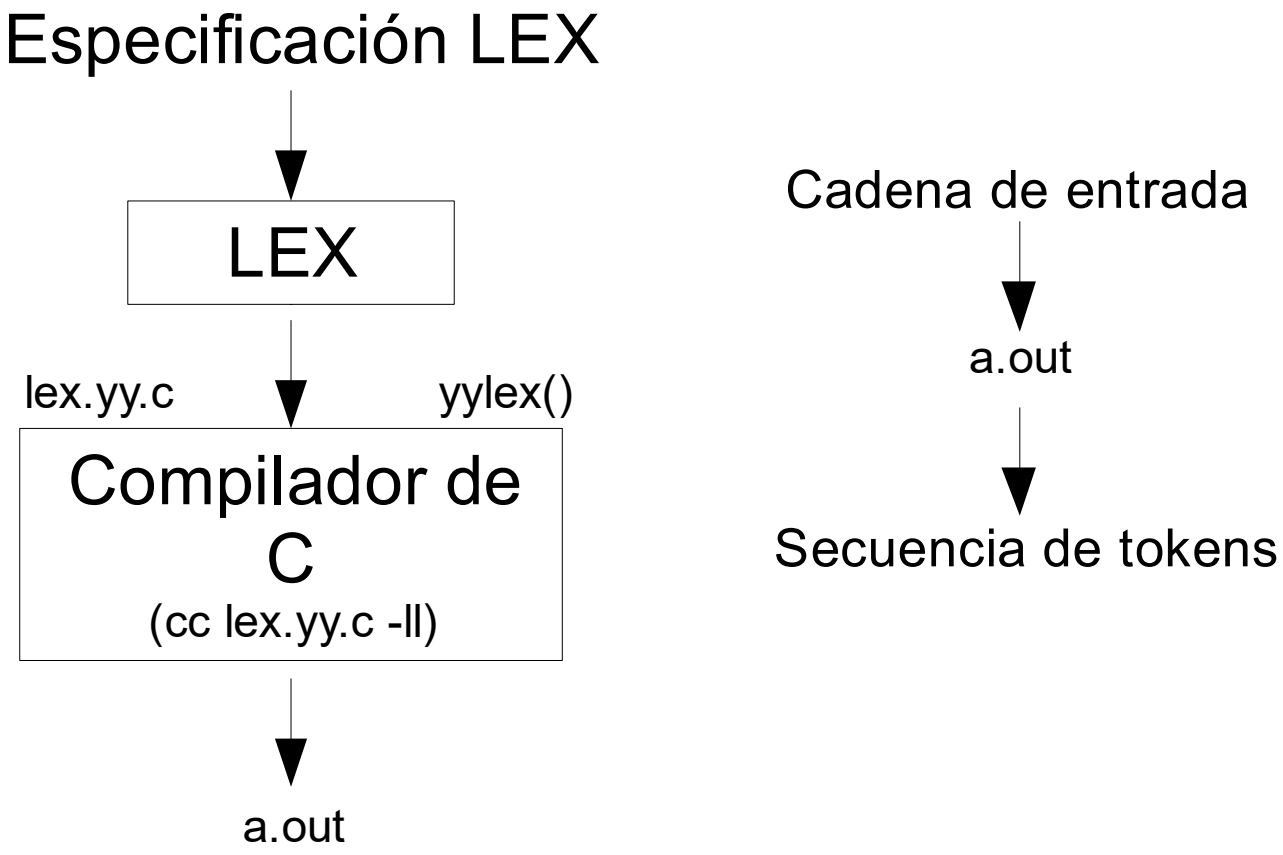
Compilación I. Ingeniería en Informática.

Facultad de Informática de San Sebastián.

¿Qué es LEX?

- Generación automática de un programa (C) a partir de una especificación de su comportamiento.
  - Asignación de un tratamiento a la aparición de instancias de expresiones regulares en su entrada.
  - Inclusión de fragmentos de código C en la especificación LEX, que se copian en el programa de salida.
- Comportamiento de un programa generado por LEX:
  - Emparejar secuencias de caracteres con las expresiones regulares definidas en la especificación LEX.
  - Ejecutar la acción asociada a la especificación LEX emparejada.

¿Qué es LEX?


Especificación LEX
LEX
lex.yy.c
yylex()
Compilador de C
(cc lex.yy.c -ll)
a.out
Cadena de entrada
a.out
Secuencia de tokens

Especificaciones LEX.

- Instruir a LEX para que nos genere un programa que analice un texto de entrada y realice una serie de acciones en función de las cadenas de caracteres que encuentre.
- Especificación LEX:
  - Sección de declaraciones.
    - Definición de tipos, macros, inclusión de ficheros.
  - Sección de reglas.
    - Definición de ER-LEX junto con las acciones a realizar.
  - Sección de funciones del usuario.
    - Definición de funciones auxiliares.

Especificaciones LEX.

Declaraciones

%%

Reglas

%%

Funciones de usuario

- Funciones de usuario → opcional.
- Mínimo programa:

  %%

Sección de declaraciones.

- Tiene dos funciones:
  1. Introducir código C global a la función *yylex()*.
     - Definición de variables, constantes, tipos.
     - LEX reconoce como código C:
       - Líneas cuya primera columna es un espacio en blanco o un tabulador.
       - Todo el texto que vaya entre %{ y %}.
  2. Definir definiciones regulares.
     - Pares de la forma: *nombre expresión_regular*
     - Nombre:
       - Debe comenzar en la primera columna de la línea.
       - Puede utilizarse posteriormente como sinónimo de expresión regular, siempre que vaya entre '{' y '}'.

Sección de funciones de usuario.

- Definición de funciones propias del usuario.
- Definición propia de ‘main()’ o ‘yywrap()’.

Sección de reglas.

- Pares de la forma:
  expresion_regular_LEX acción

  - *expresion_regular_LEX*: expresiones regulares en notación LEX.
  - *acciones*: bloques de código C.

Expresiones regulares LEX

- Σ = { caracteres ASCII }
- operadores = { \ [ ] ^ - ? . * + | ( ) $ / { } % < > }

- Notación: Sean…
  - x,z – 2 caracteres ASCII que no pertenecen al conjunto *operadores*.
  - y – carácter ASCII tal que \y no es un carácter de control en C.
  - W – cualquier secuencia de caracteres ASCII.

Expresiones regulares LEX.

- x L(x) = {x}.
- “W” L(“W”) = {W}. Las comillas no son necesarias a no ser que incluyan símbolos especiales.
- \y L(\y) = {y} aunque *y* sea un operador.
- . L(.) = Σ – {\n}
- [x-z] L([x-z]) = {caracteres ASCII comprendidos entre ‘x’ y ‘z’}.
  - **[abc] es como a|b|c**
- [^$y_1 \ldots y_n$] L([^$y_1 \ldots y_n$]) = Σ – {x : x ϵ L([$y_1 \ldots y_n$])}
  - **[^abc] incluye cualquier carácter menos ‘a’ ‘b’ o ‘c’**

Expresiones regulares LEX.

Sean *e,f* dos exp. regulares; *m* y *n* números naturales.

- e? L(e?) = L(e) U L(ξ). Opcional.
- e|f L(e|f) = L(e) U L(f). Indica opción.
- ef L(ef) = L(e) · L(f)
- (e) L((e)) = L(e)
- e{m} L(e{m}) = $L(e)^m$
- e{m,n} L(e{m,n}) = $L(e)^m$ U $L(e)^{m+1}$ U… U $L(e)^{m+n}$
  = conjunto de palabras formadas por la concatenación de ‘i’ instancias de e tal que m<=i<=n.

Expresiones regulares LEX.

Sean e,f dos exp. regulares; m y n números naturales.

- e+ L(e+) = conjunto de palabras formadas por la concatenación de un número de instancias de ‘e’ mayor o igual a una.
- e* L(e*) = L(e+) U {ξ}.

Expresiones regulares LEX.

Sensibilidad al contexto.

- ^e — L(^e) = conjunto de palabras formadas por instancias de ‘e’ siempre que se hayen al principio de una línea.
- e$ — L(e$) = conjunto de palabras formadas por instancias de ‘e’ siempre que se encuentren al final de la línea.
- e/f — L(e/f) = conjunto de instancias de ‘e’ si éstas van precedidas de la instancia ‘f’.

Acciones.

- Código C que se ejecuta cuando se reconoce una instancia de la ER-LEX asociada.
- Acción léxica por defecto: copiar los caracteres de entrada en la salida estándar.
- Acción nula: **;**

- Uso de variables y macros de utilidad:
  - **yytext[]**: última secuencia de caracteres emparejada.
  - **yyleng**: longitud de yytext[].
  - **ECHO**: copia en la salida estándar la secuencia de caracteres reconocida.

Acciones.

- Tratamiento de la ambigüedad:
  - Una secuencia de caracteres de entrada puede derivarse de más de una expresión regular de las reglas.
- Comportamiento de LEX:
  a. se toman las coincidencias más largas.
  b. primera regla en el orden de entrada LEX.
- Ejemplo:

```
if          {acción 1}
[a-z]+      {acción 2}
```

El uso de LEX.

- Comando UNIX *lex*:

  lex nombre-fichero (genera *lex.yy.c*)

- lex.yy.c:
  - función *int yylex()*:
  - Componente fundamental del programa resultante.
  - Cuando *'yylex()'* detecta el final del texto de entrada llama a *'yywrap()'*.
  - Finaliza cuando *'yywrap()'* devuelve el valor 1. En ese caso *'yylex()'* devuelve el valor 0.