ANÁLISIS LÉXICO

Regex FTW

31 de agosto de 2022

Ing. Msc. Víctor Orozco

Universidad Rafael Landivar

ANÁLISIS LÉXICO

ANÁLISIS LÉXICO

```
if (i == j)
    Z = 0;
else
    Z = 1;
```

```
\tif (i == j)\n\t\tz = 0;\n\telse\n\t\tz = 1;
```

TOKENS

En inglés

Ej:

En programación

Ej:

CLASES

- Las clases de token corresponden a conjuntos de cadenas.
- Identificador: Cadenas de letras o dígitos, comenzando con una letra
- Número entero: Una cadena de dígitos no vacía
- Palabra clave: “else” o “if” o “begin” o …
- Espacio en blanco: Una secuencia no vacía de espacios en blanco, saltos de línea y tabulaciones

ANALIZADOR LÉXICO

Definición

Clasificar subconjuntos de cadenas de acuerdo a un rol y proporcionar esta clasificación al parser.

ANALIZADOR

```
\tif (i == j)\n\t\tz = 0;\n\telse\n\t\tz = 1;
```

ANALIZADOR

Para el fragmento de código descrito a continuación, seleccione el número correcto de tokens:

```
x = 0;\n\twhile (x < 10) {\n\tx++;\n}
```

- ☐ W = 9; K = 1; I = 3; N = 2; O = 9
- ☐ W = 11; K = 4; I = 0; N = 2; O = 9
- ☐ W = 9; K = 4; I = 0; N = 3; O = 9
- ☐ W = 11; K = 1; I = 3; N = 3; O = 9

RESUMEN

Un analizador léxico debe:

- ☐ Reconocer las cadenas que correspondan a los tokens (los lexemas)
- ☐ Identificar la categoría de cada lexema

ANÁLISIS LÉXICO

EJEMPLO AL

Regla de FORTRAN: Los espacios en blanco son insignificantes
`VAR1` es lo mismo que `VA R1`

EJEMPLO AL

```
DO 5  I = 1,25

DO 5  I = 1.25
```

EJEMPLO AL

- 1. El objetivo es dividir la cadena. Esto se implementa leyendo de izquierda a derecha, reconociendo un token a la vez.
- Es posible que se requiera ”mirar hacia adelante (lookahead)”para decidir dónde termina un token y comienza el siguiente token

EJEMPLO AL

```
if (i == j)
    Z = 0;
else
    Z = 1;
```

EJEMPLO AL

PL/I: Las palabras clave (keywords) no estan reservadas

```
IF ELSE THEN THEN = ELSE; ELSE ELSE = THEN
```

EJEMPLO AL

PL/I: Las palabras clave (keywords) no estan reservadas

```
DECLARE (ARG1,. . ., ARGN)
```

¿Es `DECLARE` una palabra clave o un identificador de arreglo?

EJEMPLO AL

C++ template

```
Foo<Bar>
```

C++ stream

```
cin >> var;
```

RESUMEN

- El objetivo del análisis léxico es:
  - Dividir la cadena de entrada en lexemas
  - Identificar el token de cada lexema
- Escaneo de izquierda a derecha => a veces se requiere anticipación (lookahead)

ANÁLISIS LÉXICO

LENGUAJES REGULARES

- Estructura léxica = clases (categorías) de tokens
- Podemos afirmar qué un conjunto de cadenas pertenece a una clase de tokens
- Describimo mediante expresiones regulares

LENGUAJES REGULARES

Caracter simple

LENGUAJES REGULARES

Epsilon

LENGUAJES REGULARES

Unión

LENGUAJES REGULARES

Concatenación

LENGUAJES REGULARES

Iteración

LENGUAJES REGULARES - EXPRESIONES REGULARES

Definición

Las expresiones regulares sobre $\sum$ son el conjunto más pequeño de expresiones que incluyen

LENGUAJES REGULARES

LENGUAJES REGULARES - EJERCICIO

Elija los lenguajes regulares que sea equivalentes al lenguaje: (0 + 1)*1(0 + 1)* para $\sum$ = 0, 1

1. (01 + 11)*(0 + 1)*
2. (0 + 1)*(10 + 11 + 1)(0 + 1)*
3. (1 + 0)*1(1 + 0)*
4. (0+ 1)*(0 + 1)(0 + 1)*

RESUMEN

- Las expresiones regulares especifican lenguajes regulares
- Cinco construcciones
- Dos casos base: Cadenas vacías y de 1 carácter
- Tres expresiones compuestas: Unión, concatenación, iteración

ANÁLISIS LÉXICO

LENGUAJES FORMALES

Sea $\sum$ un conjunto de caracteres (un alfabeto).

Definición

Un lenguaje sobre $\sum$ es un conjunto de cadenas de caracteres extraídos de $\sum$

LENGUAJES FORMALES

- Alfabeto = Caracteres en español
- Lenguaje = Oraciones en español

- Alfabeto = ASCII
- Lenguaje = Programas en C

LENGUAJES FORMALES

La función de significado *L* asigna la sintaxis a la semántica

LENGUAJES FORMALES

LENGUAJES FORMALES

¿Por qué usar una función de significado?

- ☐ Deja claro qué es la sintaxis, qué es la semántica.
- ☐ Nos permite considerar la notación como un tema separado
- ☐ Porque las expresiones y significados no son 1-1

LENGUAJES FORMALES

LENGUAJES FORMALES

El significado es muchos a uno:
¡nunca uno a muchos!

ANÁLISIS LÉXICO

ESPECIFICACIÓN LÉXICA

Palabra clave: “`if`” o “`else`” o “`then`” o …

ESPECIFICACIÓN LÉXICA

Entero: Una cadena no vacía de dígitos

ESPECIFICACIÓN LÉXICA

Identificador: cadenas de letras o dígitos, comenzando con una letra

ESPECIFICACIÓN LÉXICA

Espacio en blanco: una secuencia no vacía de espacios en blanco, líneas nuevas y tabulaciones

ESPECIFICACIÓN LÉXICA

juan.barney@gmail.edu.gt

ESPECIFICACIÓN LÉXICA

digit = $'0' +' 1' +' 2' +' 3' +' 4' +' 5' +' 6' +' 7' +' 8' +' 9'$
digits = $digit+$
opt_fraction = $('.'digits) + \epsilon$
opt_exponent = $('E'('+' +' -'+)digits) + \epsilon$
num = $digits\ opt\_fraction\ opt\_exponent$

ESPECIFICACIÓN LÉXICA - EJERCICIO

Elija las expresiones regulares que sean especificaciones correctas de la descripción en inglés que se proporciona a continuación:

Twelve-hour times of the form "04:13PM". Minutes should always be a two digit number, but hours may be a single digit.

1. (0 + 1)?[0-9]:[0-5][0-9](AM + PM)
2. ((0 + ε)[0-9] + 1[0-2]):[0-5][0-9](AM + PM)
3. (0*[0-9] + 1[0-2]):[0-5][0-9](AM + PM)
4. (0?[0-9] + 1(0 + 1 + 2):[0-5][0-9](A + P)M

RESUMEN

- Las expresiones regulares describen muchos lenguajes útiles
- Los lenguajes regulares son una especificación de lenguaje
  - Todavía necesitamos una implementación
- El reto, como saber si una cadena s y una rexp R, donde

$$s \in L(R)$$