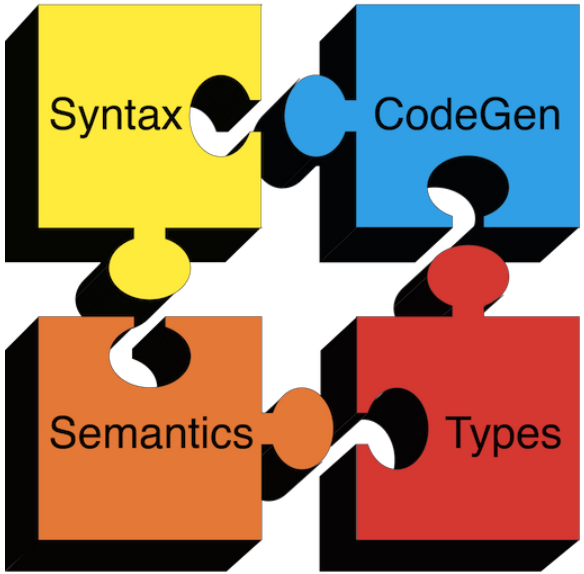


Compiladores

Análisis léxico

Análisis léxico

1. Análisis léxico
2. Análisis sintáctico (Parsing)
3. Análisis semántico
4. Optimización
5. Generación de código

Análisis léxico

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

Un analizador léxico descompone el código en sus componentes fundamentales, tales como palabras clave, nombres de variables y operadores relacionales. A diferencia de los humanos, que utilizan pistas visuales, el analizador ve una secuencia de caracteres lineales y su tarea es colocar marcadores para identificar dónde comienzan y terminan los distintos elementos del código.

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

Análisis léxico

En el análisis léxico, el trabajo no se limita a simplemente colocar divisores en la secuencia de caracteres. Más allá de identificar estas subcadenas, es esencial ***categorizar*** cada elemento de la cadena según su función específica dentro del lenguaje de programación. A estas categorías las denominamos 'clases de token'.

- Clase de token (or Class)
  - En inglés:

    Noun, verb, adjective, ...

  - En un lenguaje de programación:

    Identifier, Keywords, '(', ')', Numbers, ...

Lexical Analysis

- Las clases de token corresponden a conjuntos de cadenas.

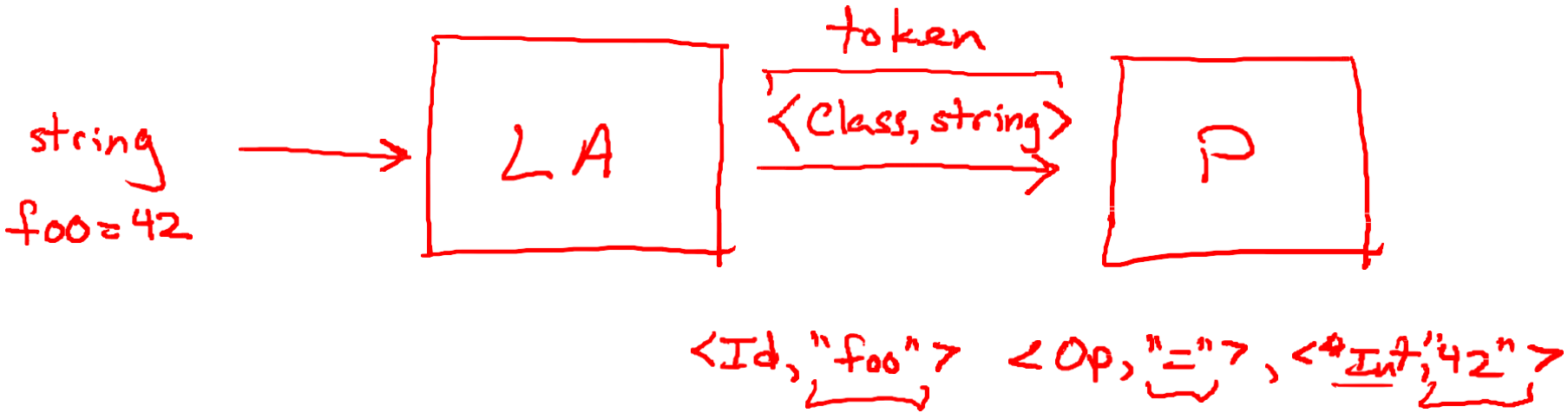
- Clase de token (Identificador): A1 Foo B17
  - *cadenas de letras o dígitos, que comienzan con una letra*
- Clase de token (Entero):
  - *una cadena de dígitos no vacía* 0 12 001 00
- Clase de token (Palabra clave):
  - *"else" or "if" or "begin" or …*
- Clase de token (Espacios en blanco):
  - *Una secuencia no vacía de espacios en blanco, saltos de línea y tabulaciones* if (whitespace

Análisis léxico

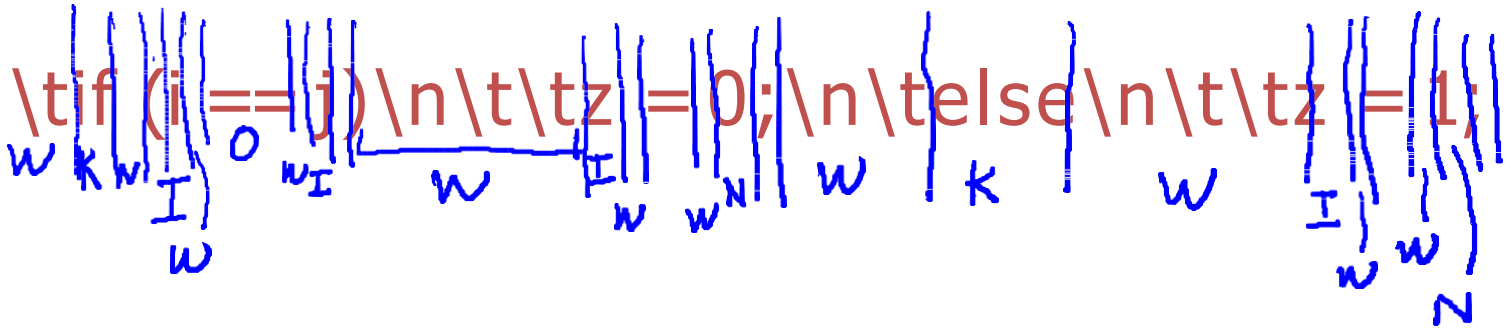
El propósito fundamental del análisis léxico es categorizar las subcadenas de un programa de acuerdo con su función específica en el lenguaje de programación, asignándoles una 'clase de token'. Estas clases pueden identificar si una subcadena es una palabra clave, un identificador de variable, entre otros roles.

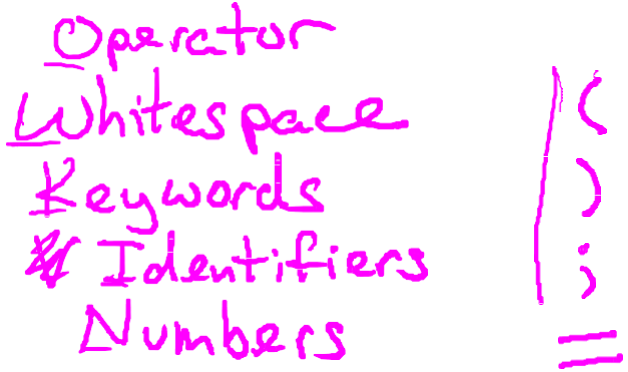
- Clasificar las subcadenas del programa según el rol
- Comunicar tokens al analizador

class token



Análisis léxico





Durante este proceso, cada subcadena del código será identificada y etiquetada con su clase de token correspondiente. Esto incluye agrupar secuencias de espacios en blanco en un único token, identificar palabras clave, operadores, números, identificadores y signos de puntuación, cada uno en su respectiva clase.

Análisis léxico

Para el siguiente fragmento de código,
Elija el número correcto de tokens en
cada clase que aparece en el fragmento de código

```
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

W: Whitespace
K: Keyword
I: Identifier
N: Number
O: Other Tokens:
{ } ( ) < ++ ; =

Análisis léxico

Para resumir, la implementación del análisis léxico implica dos tareas fundamentales. La primera tarea es identificar las subcadenas en la entrada que corresponden a tokens. En la terminología de compiladores, estas subcadenas se denominan 'lexemas'. En esencia, los lexemas son las palabras o elementos del programa que se analizan.

- Una implementación debe hacer dos cosas:
  1. Reconocer subcadenas correspondientes a tokens
     - Los lexemas
  2. Identificar la clase de token de cada lexema