Lenguajes de programación

Elementos de un lenguaje

Especificación de un lenguaje de programación

- LÉXICO
  - Vocabulario del lenguaje.
- SINTAXIS
  - Reglas para combinar los vocablos del lenguaje conformando frases válidas.
- SEMÁNTICA
  - Reglas para usar correctamente a los elementos del lenguaje en cuanto al significado que deben tener las frases.

LÉXICO

- De un lenguaje natural:
  - Palabras que se encuentran en el diccionario: sustantivos, adjetivos, verbos, artículos, etc.
  - Signos de puntuación.
- De un lenguaje de programación:
  - Palabras reservadas
  - Identificadores
  - Valores constantes
  - Símbolos especiales: Operadores, Delimitadores.

SINTAXIS

- Orden en que se combinan las palabras.
- De un lenguaje natural:
  - *"Primero el sujeto y luego el predicado…"*
- De un lenguaje de programación:
  - *if (expresión) estatuto; else estatuto;*

SEMÁNTICA

- Significado congruente de una frase.
- De un lenguaje natural:
  - Congruencia de género, número y conjugación.
- De un lenguaje de programación:
  - Congruencia de tipos de datos, de unicidad de nombres, etc.

Ejemplo

- Frase con un léxico, sintaxis y semántica correctos:

“ Los alumnos de la clase de Lenguajes de programación son los mejores de la escuela.”

Ejemplo

- Frase con un léxico y sintaxis correctos pero con una semántica incorrecta:

“ El alumnos de los clase de Lenguajes con programación es lo mejores de la escuelas.”

Ejemplo

- Frase con un léxico correcto pero con una sintaxis y semántica incorrecta:

“ De la escuelas el alumnos de Lenguajes con programación de los clase es lo mejores”

Ejemplo

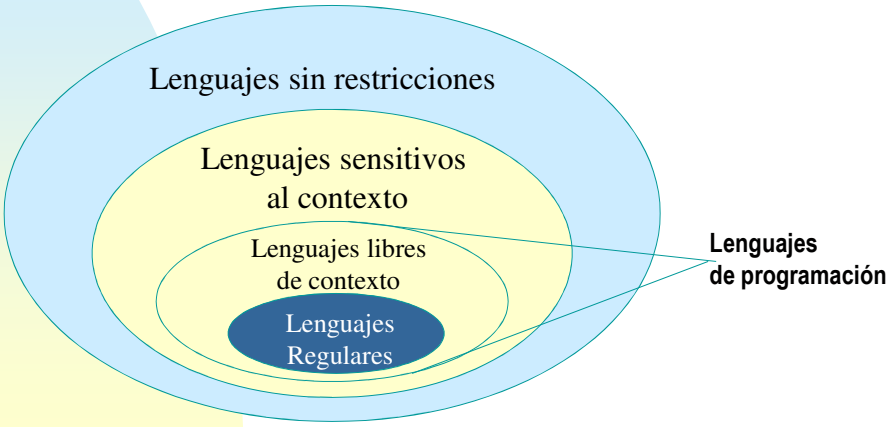
- Frase con una sintaxis y semántica correctas, pero un léxico incorrecto:

" Los students de la clase de Lenguajes de programación son los best de la school."

Formas de describir a un lenguaje de programación

- Informalmente, usando el lenguaje natural. *(problema: ambigüedad)*
- Gráficamente, usando diagramas.
  - Diagramas de estados o transiciones (autómatas).
  - Diagramas de sintaxis.
- Formalmente, usando fundamentos y notación matemática.

Jerarquía de Chomsky para los lenguajes



Herramientas formales

- Para describir el **léxico**:
  - Expresiones regulares
  - Autómatas finitos (máquina de estados finitos)
  - Gramáticas regulares
  - Diagrama de sintaxis
- Para describir la **sintaxis**:
  - Gramáticas libres de contexto
  - Diagrama de sintaxis
  - Autómatas de pila
- Para describir la **semántica**:
  - Lenguaje natural
  - Lenguaje operacional o denotacional o axiomático

Por lo tanto...

- Las reglas del léxico de un lenguaje de programación se pueden definir como un lenguaje regular.
- Las reglas de sintaxis de un lenguaje de programación se pueden definir como un lenguaje libre de contexto.

Diagramas de Sintaxis

Forma gráfica de representar
Gramáticas Libres de Contexto

Nomenclatura

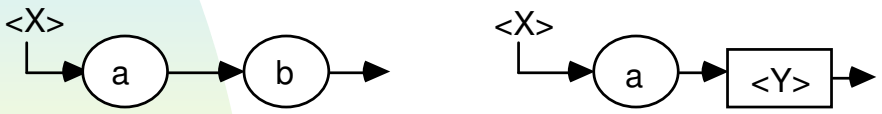
<Nombre> Representa el nombre de una Estructura Sintáctica

No-T
Representa un símbolo No-terminal,
toda una Estructura Sintáctica que debe ser expandida.

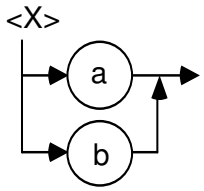
T T Representan a un símbolo terminal (token)

Representa el orden ó flujo de control en un diagrama.

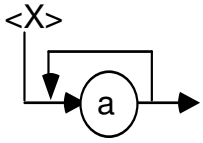
Estructuras básicas: SECUENCIA

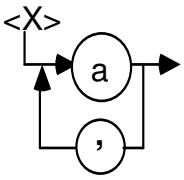


Estructuras básicas: ALTERNATIVA



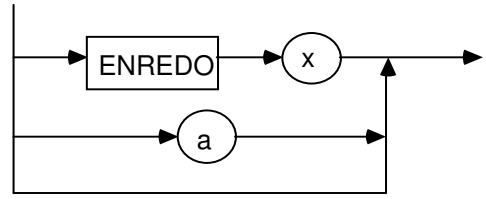
Estructuras básicas: REPETICIÓN





Uso de Recursividad

< ENREDO >



¿Cuál es el lenguaje que reconoce el diagrama?

Cadenas que pueden o no comenzar con ***a***
seguidas de cero o más ***x***'s

Ejemplo de Léxico

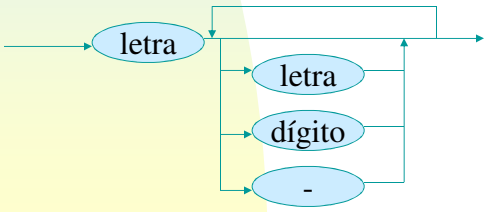
- Regla para describir un identificador:

- INFORMALMENTE: *"Un identificador comienza con una letra, seguida de cero o más letras o dígitos o el caracter de subrayado"*

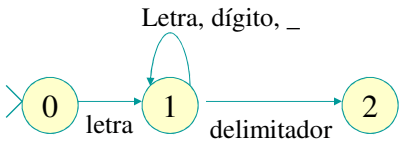
Ejemplo de Léxico

- Regla para describir un identificador GRÁFICAMENTE:

Diagrama de Sintaxis



Autómata finito



Ejemplo de Léxico

- Regla para describir un identificador FORMALMENTE:
  - EXPRESIÓN REGULAR: *letra (letra | dígito | _)**
  - GRAMÁTICA REGULAR:
    - <ID> ::= letra <ID1>
    - <ID1> ::= letra <ID1>
    - <ID1> ::= dígito <ID1>
    - <ID1> ::= _ <ID1>
    - <ID1> ::= vacío

Ejemplo de Sintaxis

- Regla para escribir expresiones aritméticas:

- INFORMALMENTE: *"Una expresión aritmética es un número, una expresión aritmética entre paréntesis o una operación binaria de expresiones aritméticas en notación infija"*

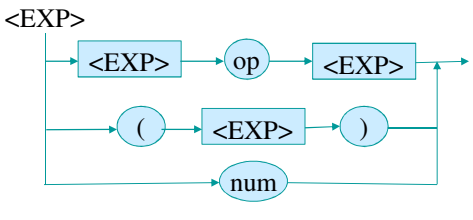
Ejemplo de Léxico

- Regla para describir un identificador GRÁFICAMENTE:

Gramática Libre de Contexto

<EXP> ::= <EXP> op <EXP>
<EXP> ::= (<EXP>)
<EXP> ::= num

Diagrama de Sintaxis



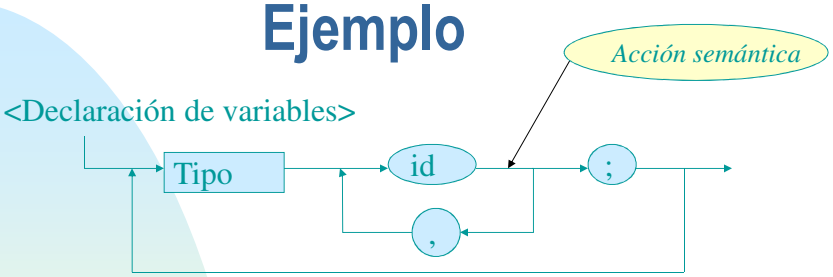
Análisis de SEMÁNTICA

- Verifica la congruencia de las frases gramaticales (estatutos).
- Tipos de verificación:
  - COMPROBACIÓN DE TIPOS: Compatibilidad de tipos de datos en base a sus rangos.
  - COMPROBACIÓN DE NOMBRES: Uso del mismo identificador en 2 o más lugares.
  - COMPROBACIÓN DE UNICIDAD: Definición única de elementos.

¿Cómo se construye un Analizador Semántico?

- A nivel diseño, se incorporan acciones de verificación semántica dentro de las producciones en la gramática del lenguaje.
- Estas acciones se implementan según el método de análisis sintáctico que se esté utilizando.

Ejemplo



Acción semántica:
Da de alta el lexema del identificador en una tabla, y verifica que no se repita. Si se repite, marca error.

Generación de código

- Se colocan acciones de generación de código en la gramática del lenguaje.
- Dependiendo del código que se quiera manejar, se diseñan las acciones.
- Tipos de código intermedio típicos: three-address code (TAC o 3AC), Static Single Assignment Form (SSA), P-code, etc.

Esta es una historia que continuará en la clase de Traductores…

Ejemplo

Código en C

```
int main(void) {
    int i;
    int b[10];
    for (i = 0; i < 10; ++i) {
        b[i] = i*i;
    }
}
```

Triplo: three-address-code

```
    i := 0                      ; asignación
L1: if i < 10 goto L2           ; salto condicional
    goto L3                     ; salto incondicional
L2: t0 := i*i
    t1 := &b                    ; operación de dirección
    t2 := t1 + i        ; t2 tiene la dirección de b[i]
    *t2 := t0           ; almacena mediante apuntador
    i := i + 1
    goto L1
L3:
```