Conceptos Básicos

Diseño de Lenguajes de Programación

Ingeniería Informática

Universidad de Oviedo

(v1.12)

Raúl Izquierdo Castanedo

Procesador de Lenguaje

Procesadores de lenguaje es el nombre genérico que reciben todas las aplicaciones informáticas en las cuales uno de los datos fundamentales *de entrada* es un lenguaje.

Entradas triviales

- Texto
  - *readline*
- Formato estructurado

```
José Pérez; 12; 13483748W
Mariano Antón; 132; 13243748M
Pepe Suárez; 22; 5583748Z
Raúl Alonso; 83; 81388748K
```

Entrada no-trivial

- Lenguaje de programación

```
3 + 4 * 5
```

```
(3 + 4) * 5
```

```
a:int;
a = 5;
f(a++);
```

Problemas

- Complejidad
- Validación

Objetivo

```
struct Persona {
  int edad;
  float altura;
};
Persona[11] equipo;

void inicia(int i) {
  int j;
  j = 0;
  while (j < i) {
      equipo[j].edad =  j * 2;
      equipo[j].altura = equipo[j].edad * (j / 2);
      j = j + 1;
  }
}
real sumaSueldos(int elementos) {
  int i;
  real acumulado;

  acumulado = 0;
  i = 0;

  while (i < elementos) {
      acumulado = acumulado + equipo[i].sueldo;
      i = i + 1;
  }
  return acumulado;
}
```

```
main {
  int num;

  read num;
  if (num < 11)
      inicia(num);
  else
      inicia(num % 11);
  print sumaSueldos(11) / 11;
}
```

Procesadores de Lenguajes

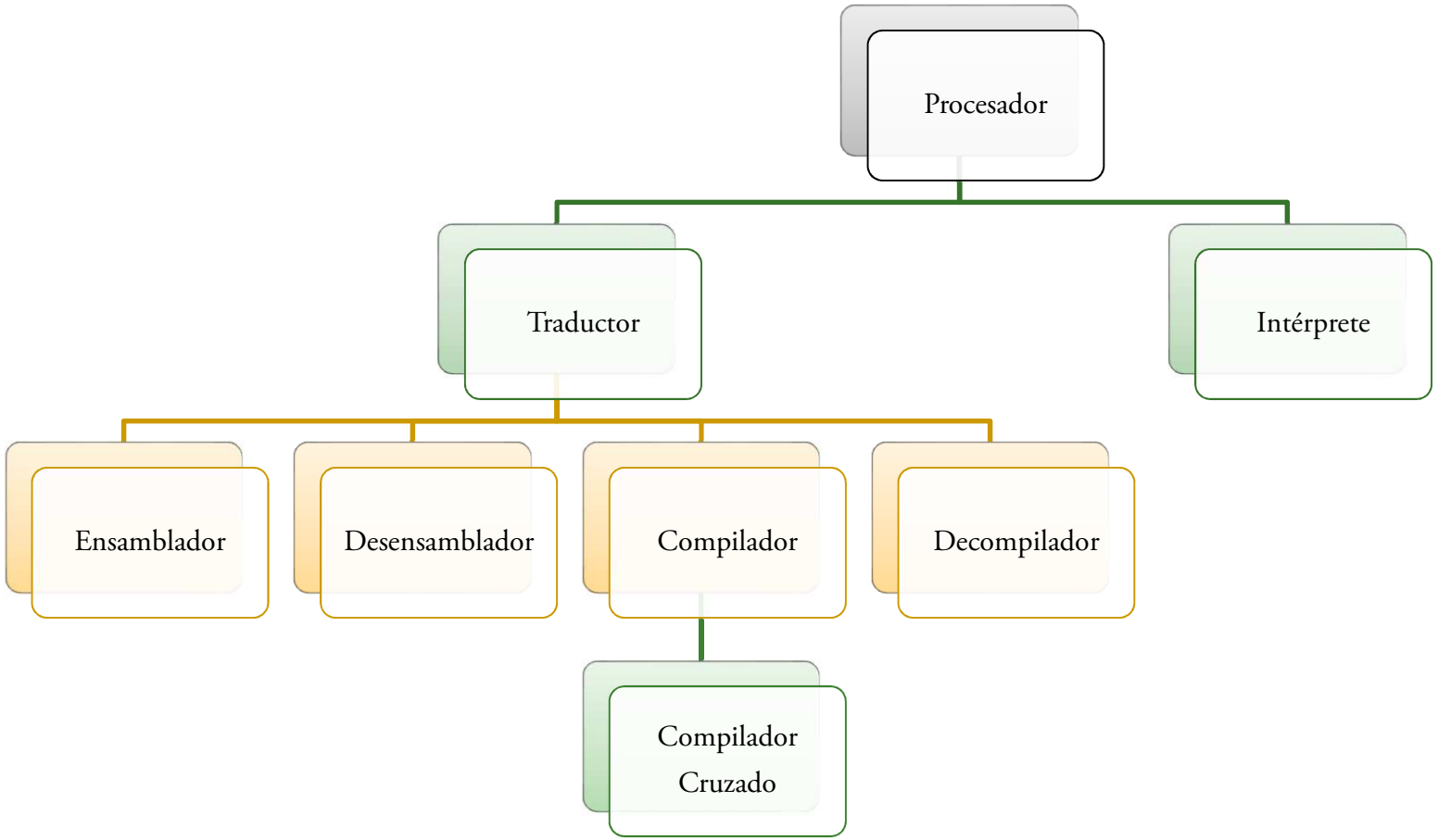
¿Por qué ver construcción de Procesadores de Lenguajes?

- Porque somos sus principales usuarios
- Porque somos los que deberían ser capaces de diseñarlos e implementarlos

Pero…

- ¿Solo los programadores usan procesadores de lenguajes?
  - ¡El programa que más utilizan los usuarios no-informáticos es un procesador!
- ¿Solo utiliza estas técnicas el que se dedica a hacer compiladores?

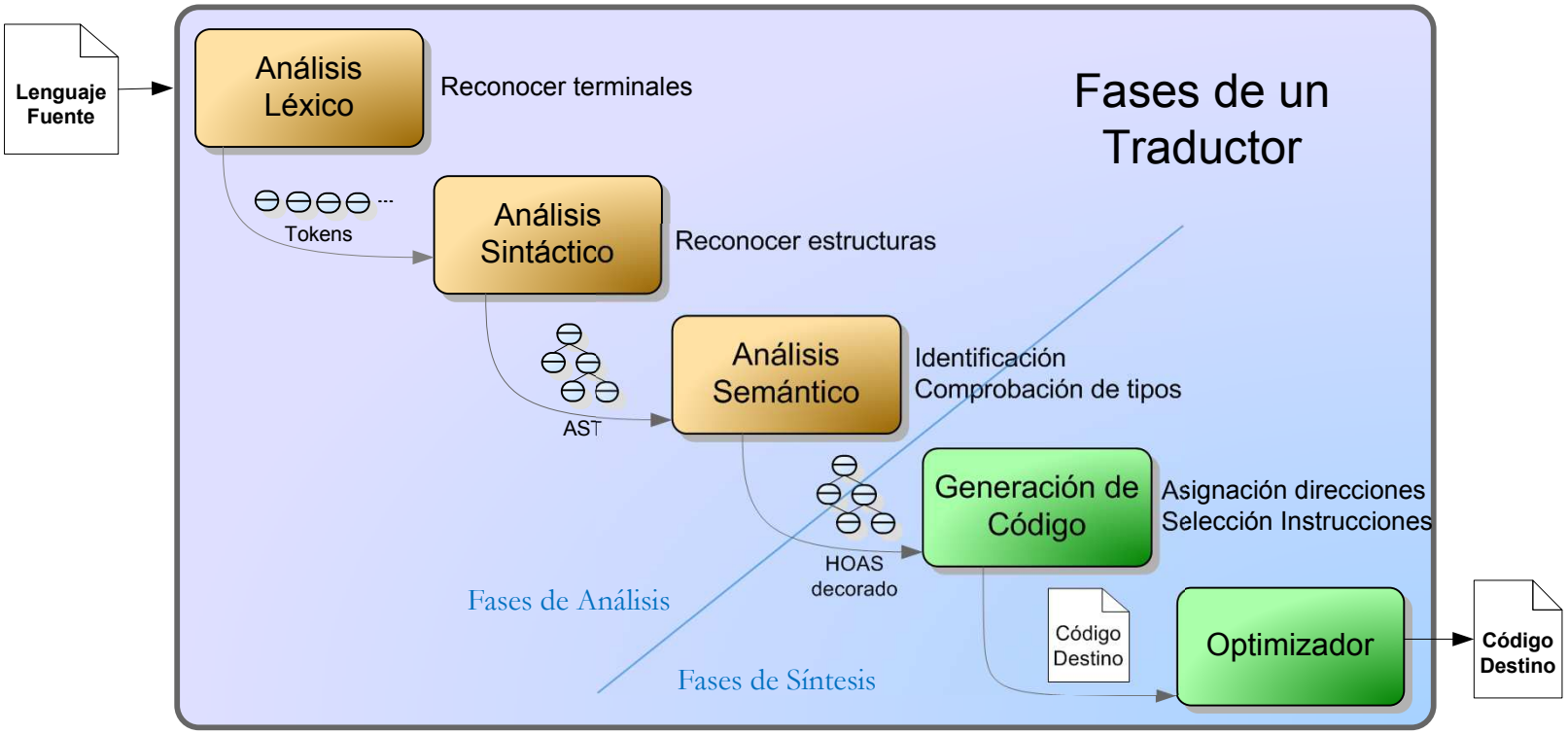
Tipos de Procesadores de Lenguaje



Fases de un Traductor

Fases de un Traductor

Se divide el proceso en fases más manejables



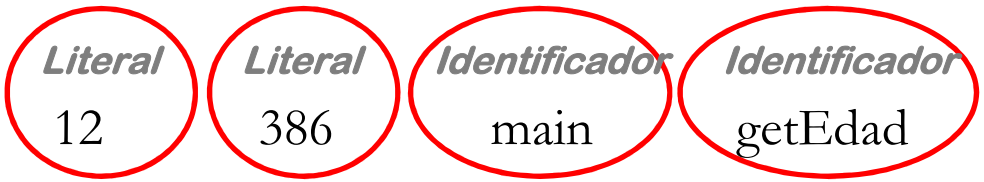
Análisis Léxico

Definiciones

- Lexema
  - Agrupación de caracteres que constituyen los símbolos con los que se forman las sentencias del lenguaje

12 386 main getEdad

- Token
  - Conjunto de lexemas que puede ser tratado como una unidad sintáctica



Análisis Léxico

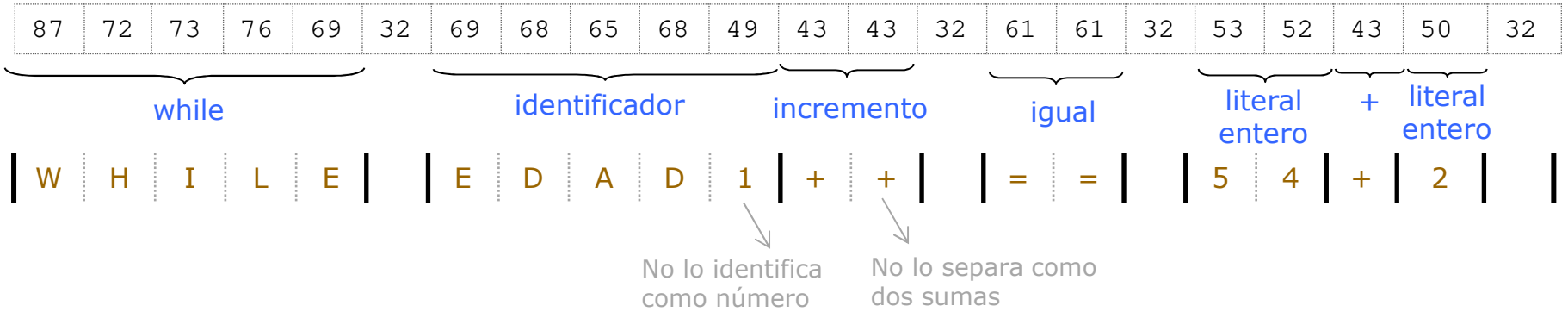
Funciones

- Separar los lexemas
- Clasificarlos en tokens
  - Rechazar secuencias de caracteres que no formen lexemas válidos

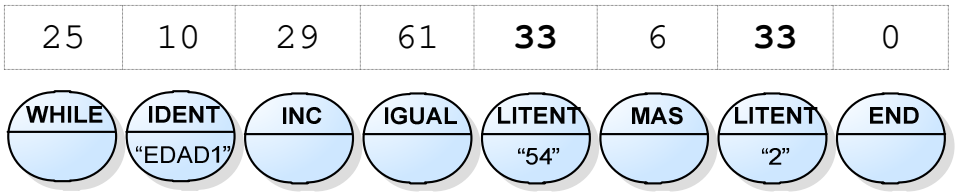
Ejemplo:

- Entrada: 22 caracteres

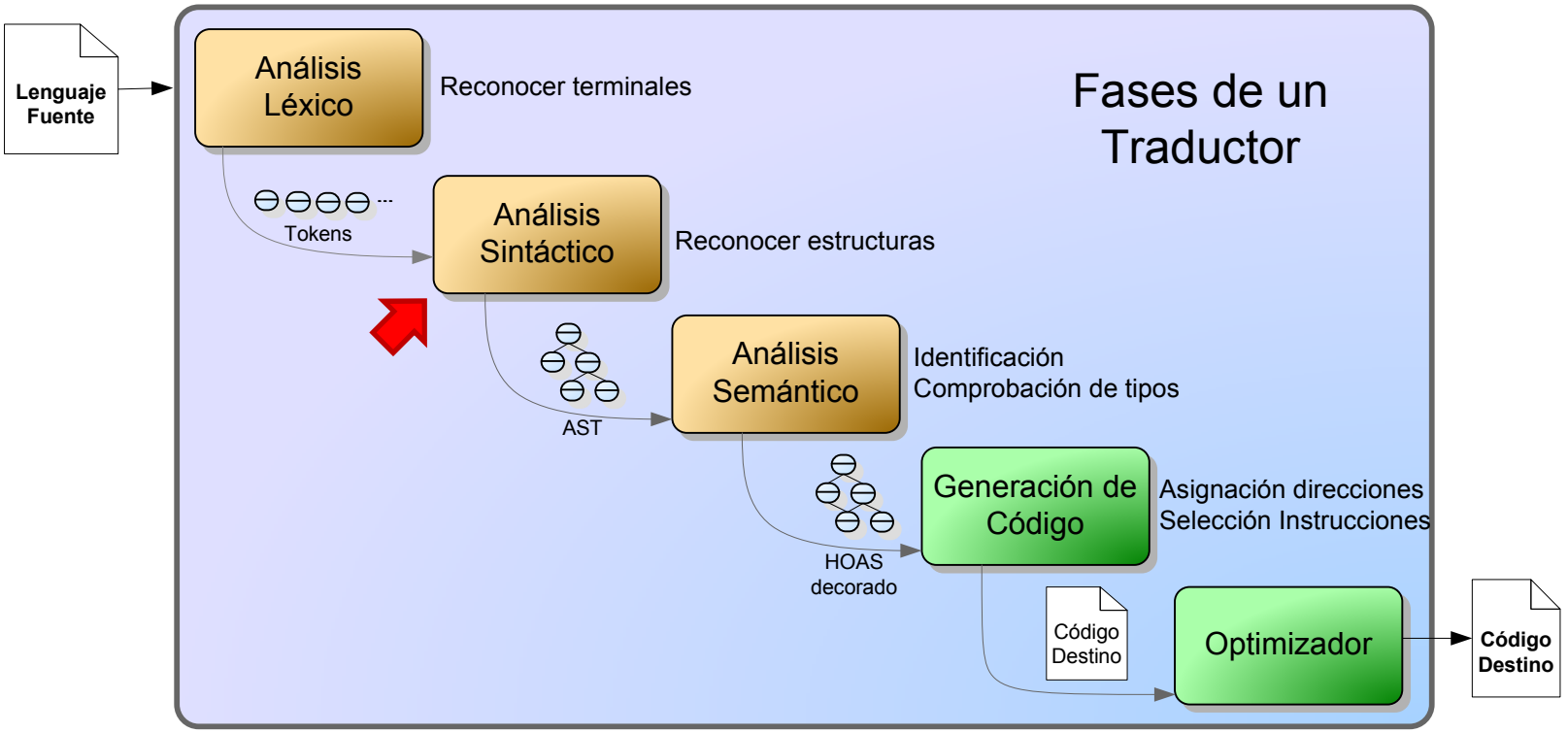
WHILE EDAD1++ == 54+2



- Salida: 8 Tokens (y sus lexemas)



Fases de un Traductor



Análisis Sintáctico (I)

Funciones

- Determina si la secuencia de tokens es válida

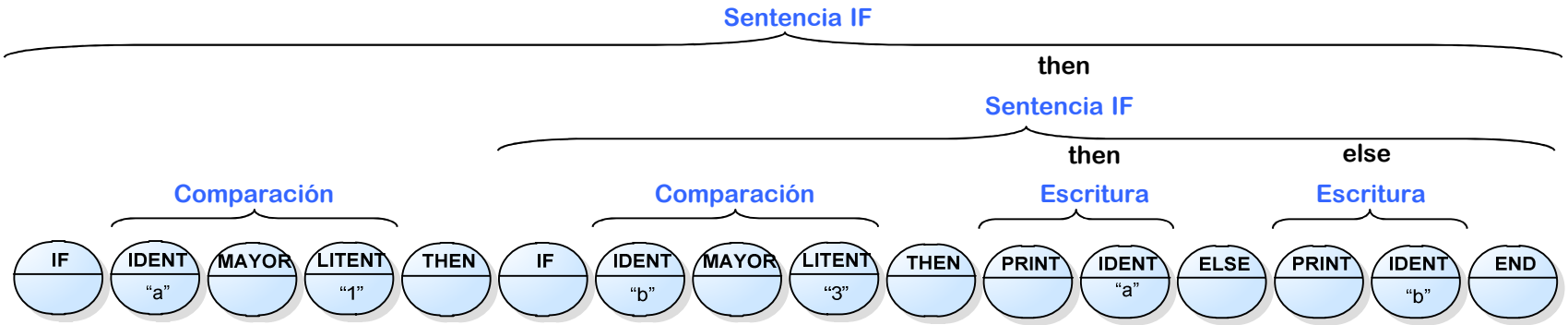
```
a > b (if)
```

- Si lo es, identifica las estructuras sintácticas que forman

```
if a>1 then if b>3 then print a else print b
```

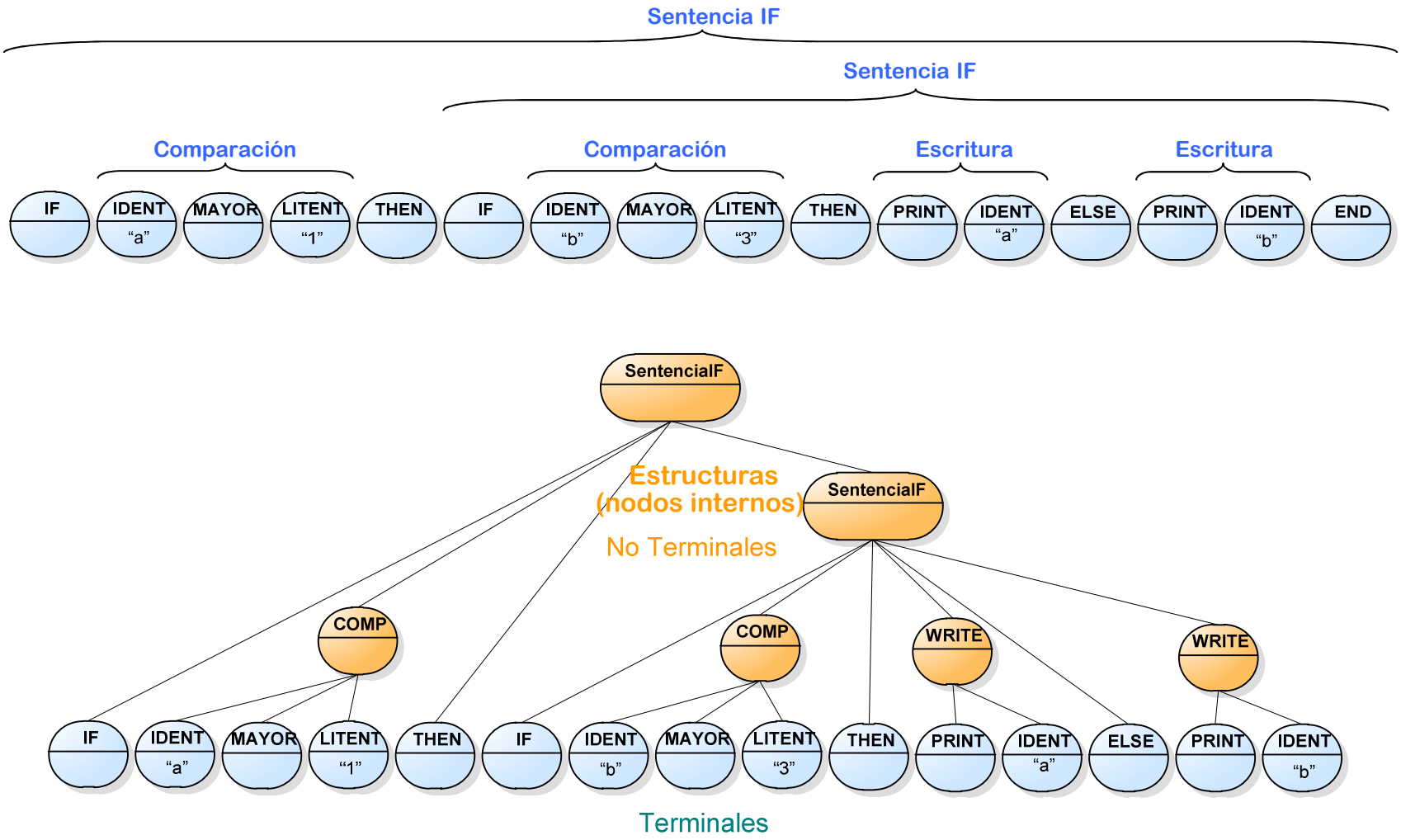
```
if a>1 then
    if b>3 then
        print a
    else
        print b
```

```
if a>1 then
    if b>3 then
        print a
else
    print b
```



Análisis Sintáctico (II)

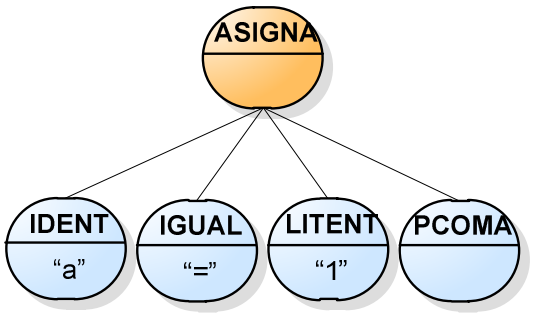
¿Cómo representar la estructura?

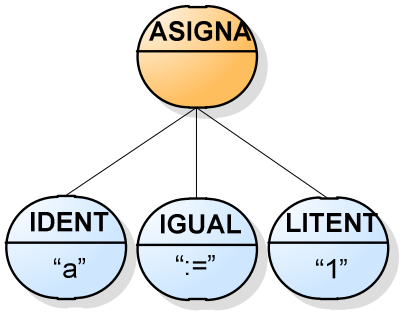


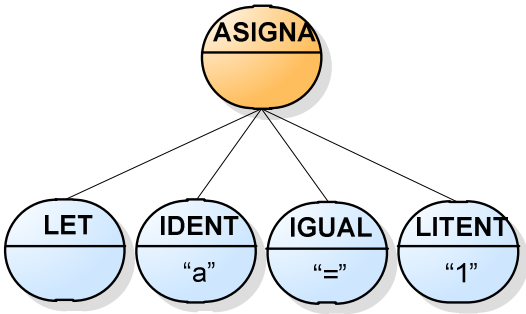
Análisis Sintáctico. AST (I)

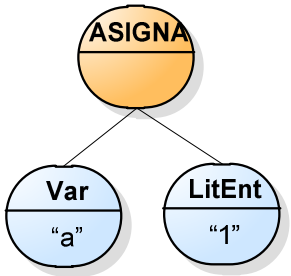
Asignación de variables

- a = 1;
- a := 1
- Let a = 1



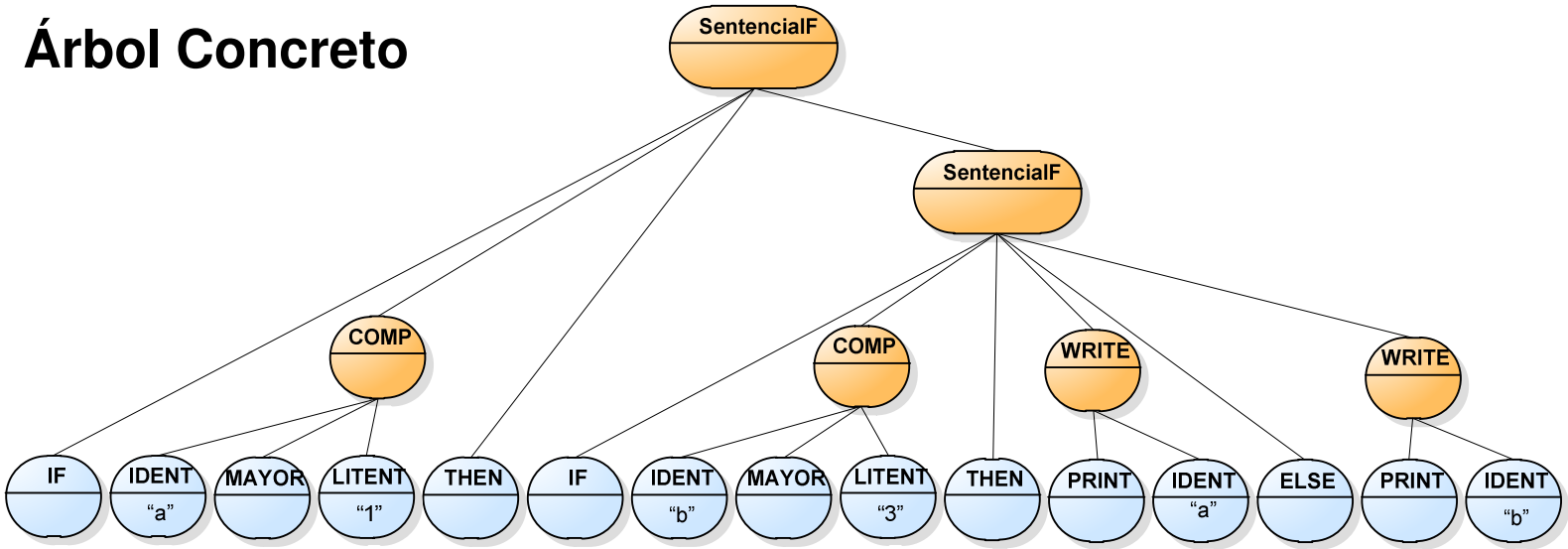




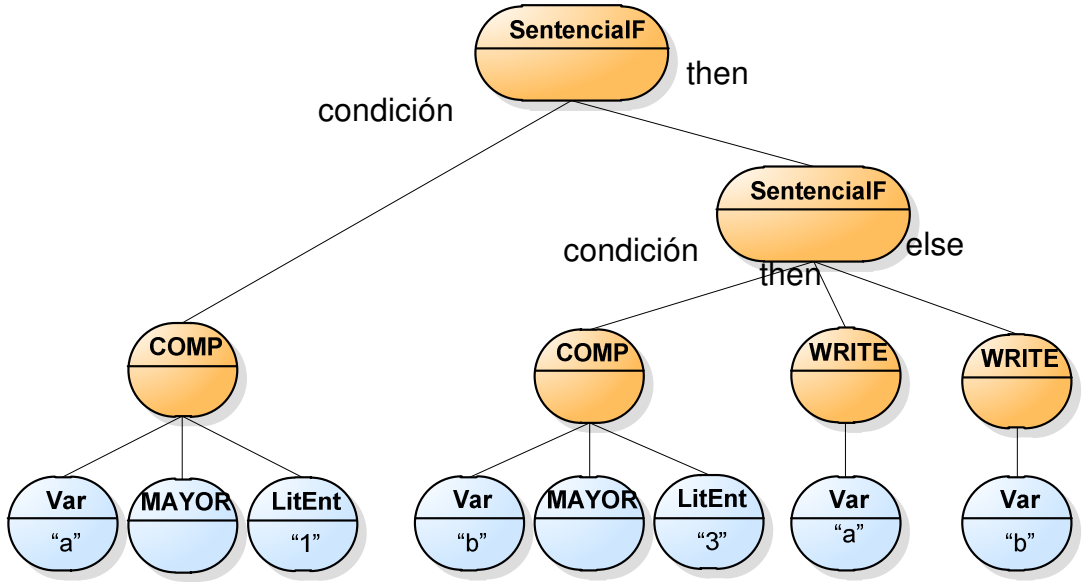


Análisis Sintáctico. AST (II)

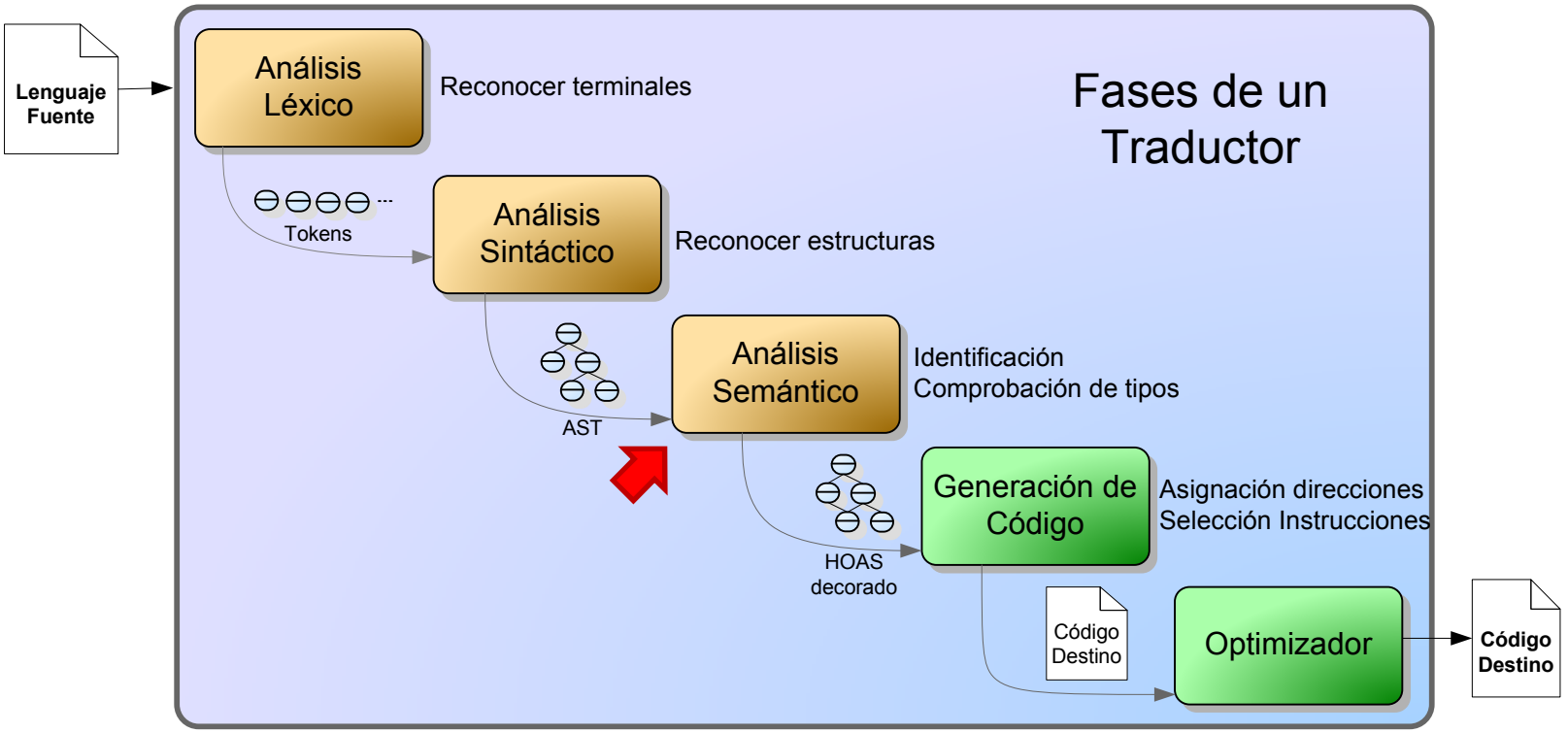
Árbol Concreto



Árbol Abstracto (AST)



Fases de un Traductor



Análisis Semántico

¿Sentencia correcta en Java?

```
a = b + 5;
```

Análisis Semántico

- Dado un árbol comprueba que el programa cumple las restricciones contextuales del lenguaje
  - Análisis Contextual

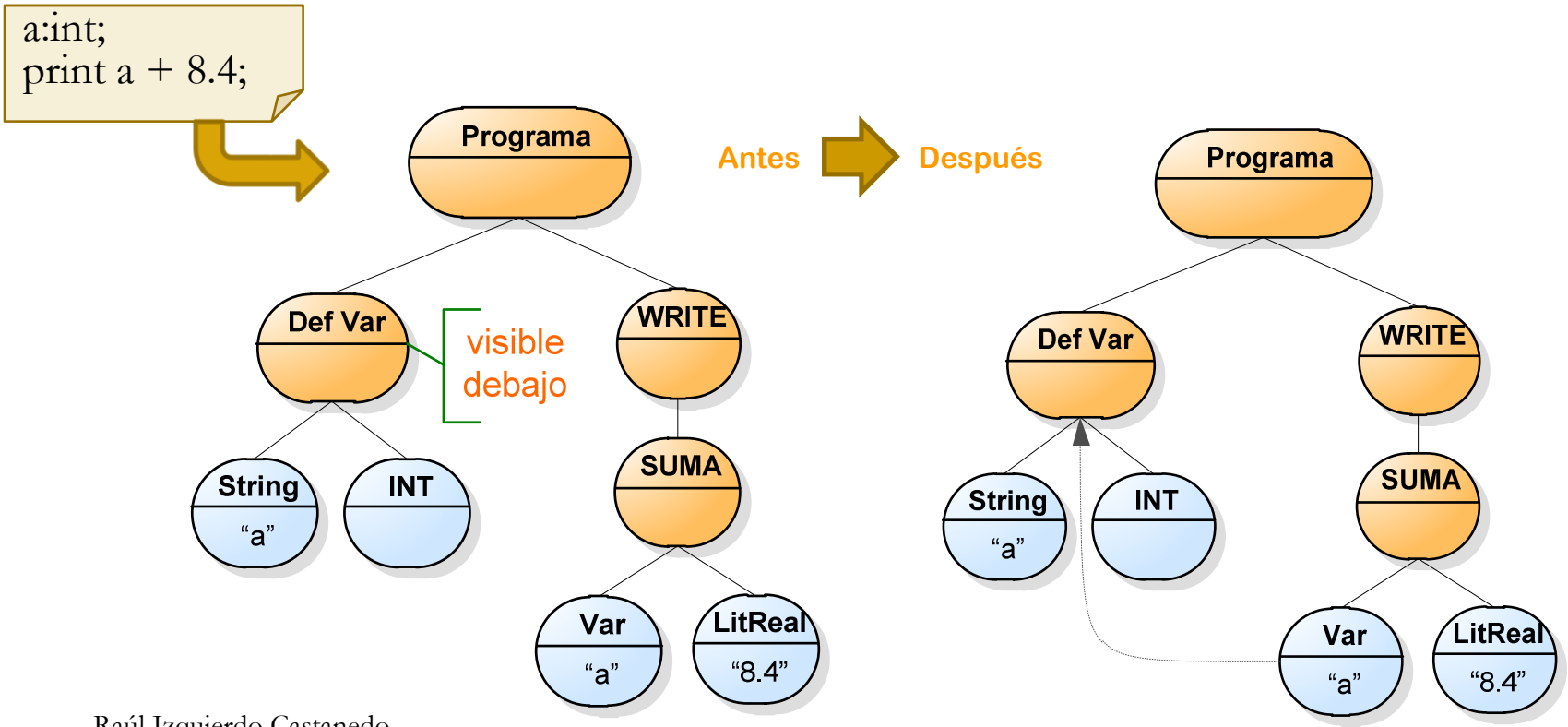
Validaciones más comunes

- Reglas de Ámbito
  - Fase de Identificación
- Reglas de Tipo
  - Fase de Inferencia de Tipos

Análisis Semántico. Fase de Identificación

Tareas de la Fase de Identificación

- Comprueba las Reglas de Ámbito
  - Determinan
    - Alcance de un símbolo
      - ¿Ejemplos?
    - Resolución de colisiones
      - ¿Ejemplos?
- Enlaza los símbolos con sus definiciones



Análisis Semántico. Reglas de Tipo

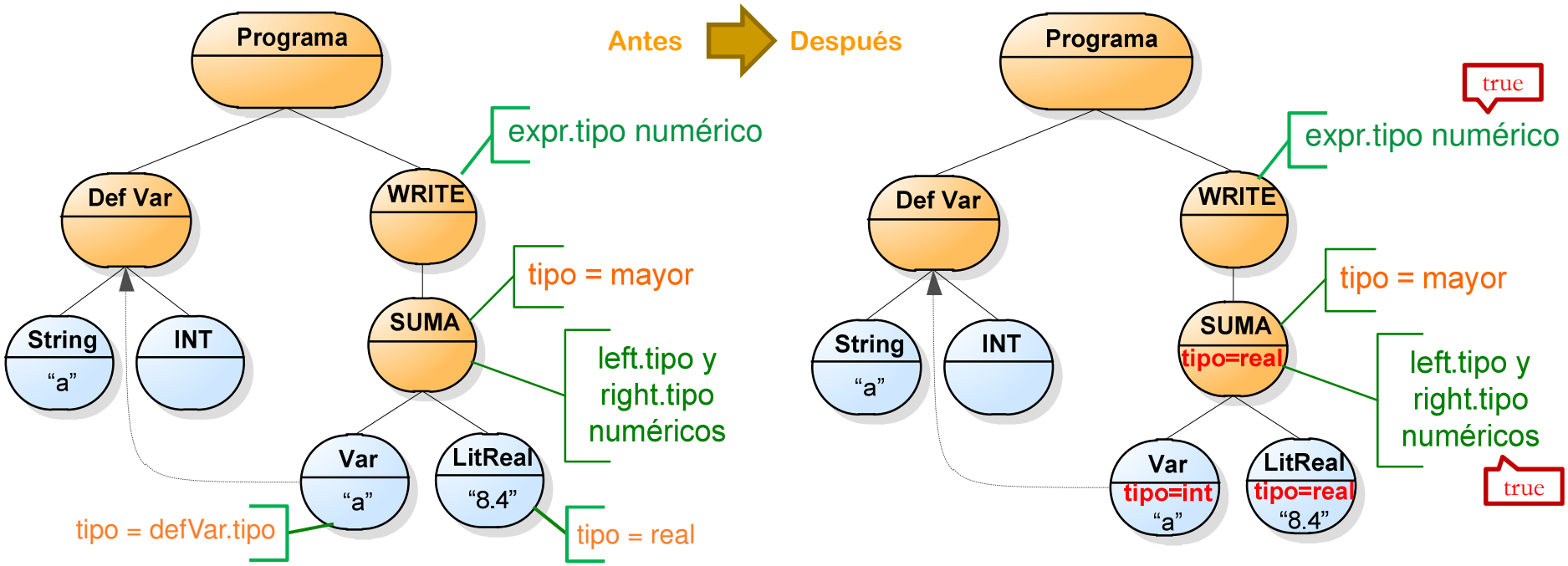
Reglas de Tipo

- Permiten comprobar de manera estática el uso adecuado de las expresiones
- Indican qué tipos deben tener las expresiones que formen parte de una estructura (predicados)…
  - La suma solo puede aplicarse a tipos numéricos
  - Los corchetes deben aplicarse solo a una *expresión* de tipo…
  - El tipo de los argumentos debe coincidir con el de los parámetros… o no
  - La condición del *if* debe ser de tipo…
- … y cómo inferir el tipo de una expresión (funciones de tipo)
  - El tipo de una suma es el tipo mayor de sus operandos
  - El tipo de un operador relacional es…
  - El tipo de un acceso a un array es…
  - El tipo de una llamada a una función…

Análisis Semántico. Fase de Inferencia de Tipos

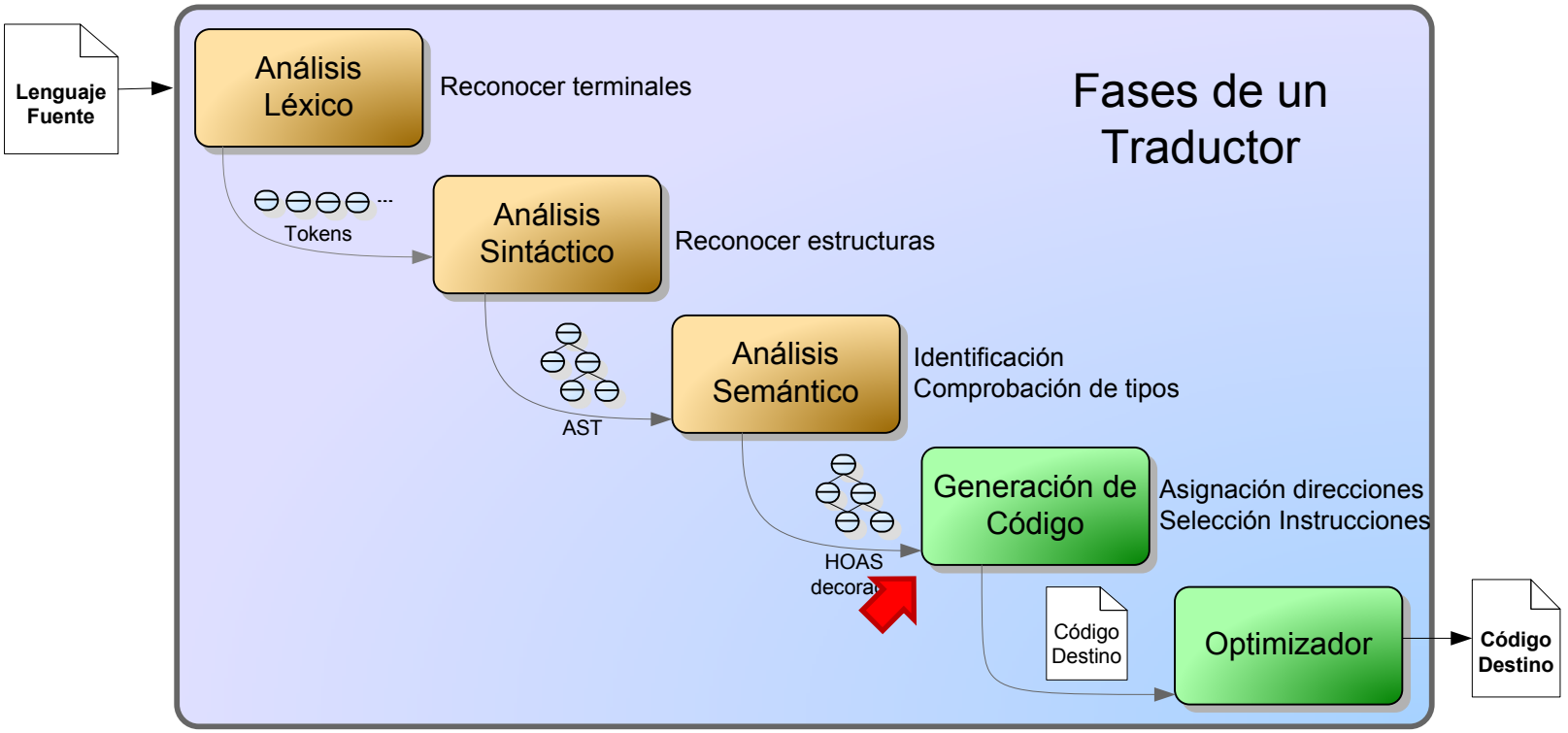
Tareas de la Fase de Inferencia de Tipos

- Asigna un atributo *tipo* a cada expresión (usando **funciones de tipo**)
- Comprueba todos los **predicados**



- ¿Cómo se expresaría que el lenguaje no tiene conversiones implícitas?

Fases de un Traductor



Generación de Código

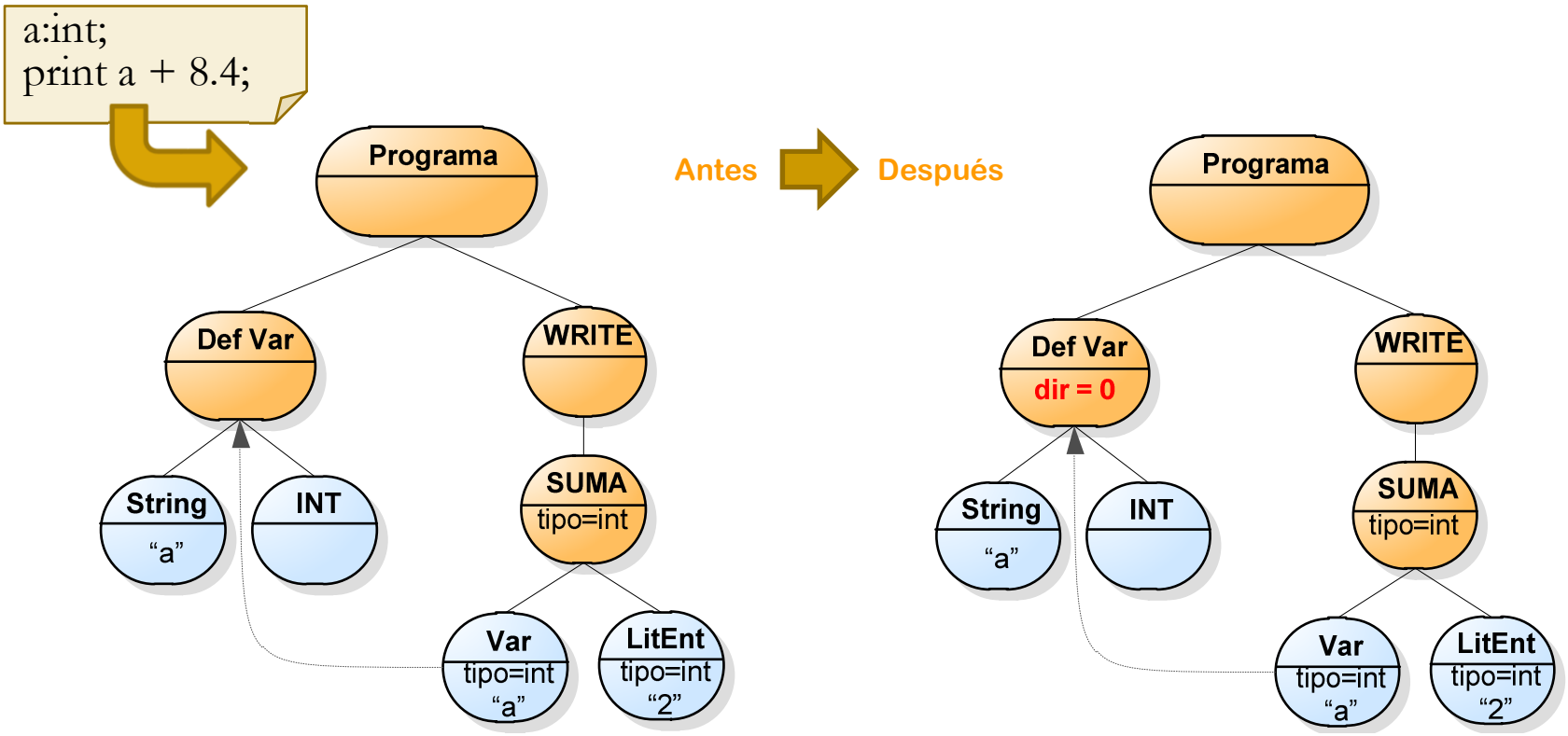
Tareas Fundamentales:

- Gestión de Memoria (asignación de direcciones)
- Selección de Instrucciones
- Gestión de Registros

Generación de Código (I)

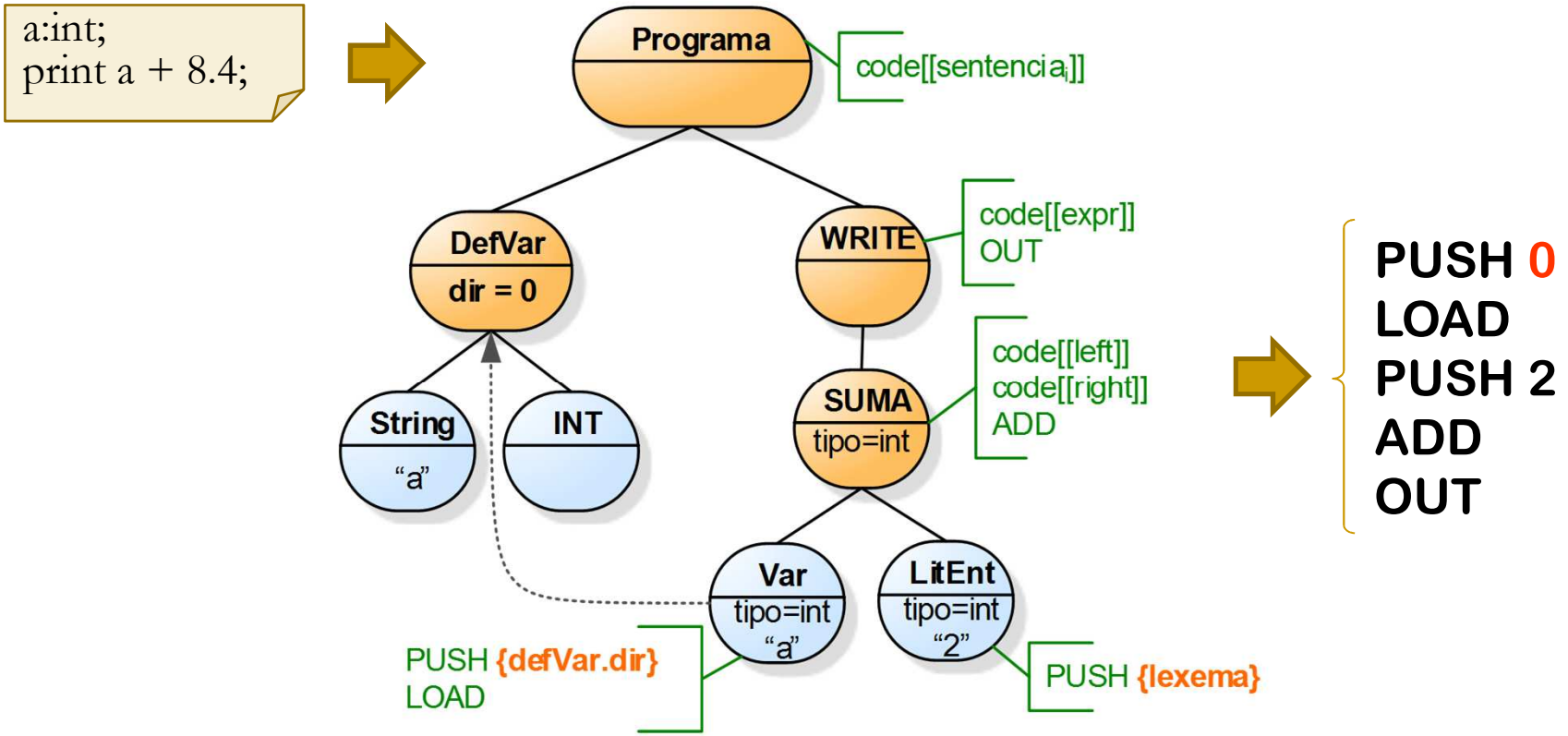
Gestión de Memoria

- Determinar la estructura de cada símbolo (tamaño y formato)
- Determinar las direcciones de cada símbolo
  - Asignación de direcciones

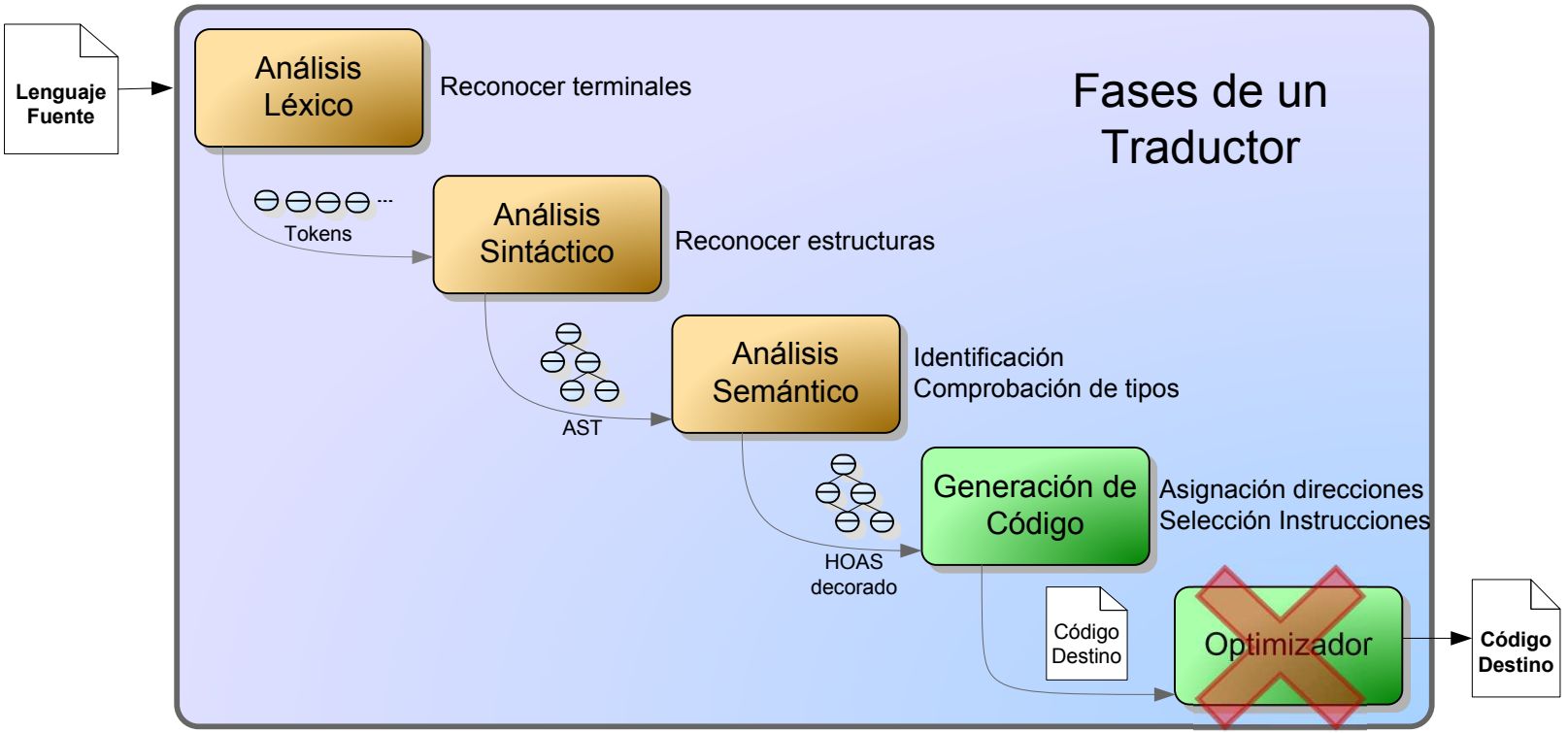


Generación de Código (II)

Selección de Instrucciones



Fases de un Traductor. Resumen

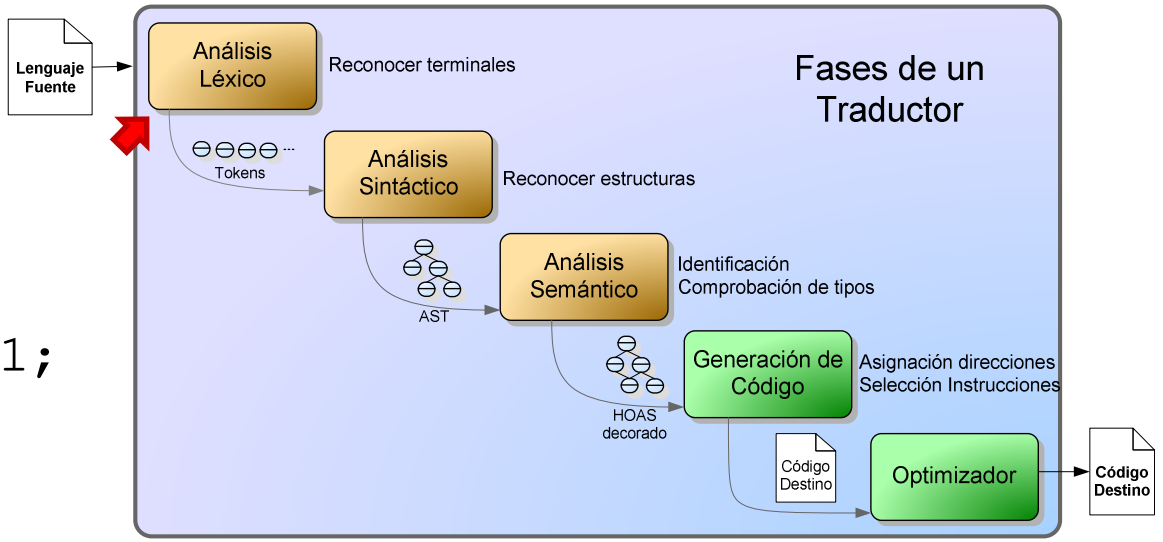


Ejercicio
Conceptos Básicos

Ejercicio

Entrada

```
ready, x2:real;
Read ready;
x2 = ready + 55 * 11;
```

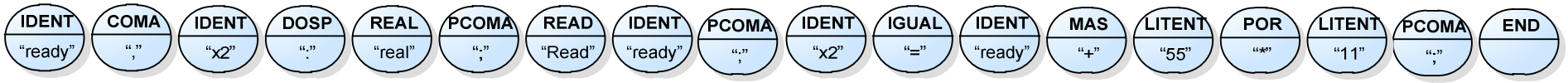


Fase 1. Léxico

Ejercicio

Fase 1. Resultado del Análisis Léxico

```
ready, x2:real;
Read ready;
x2 = ready + 55 * 11;
```

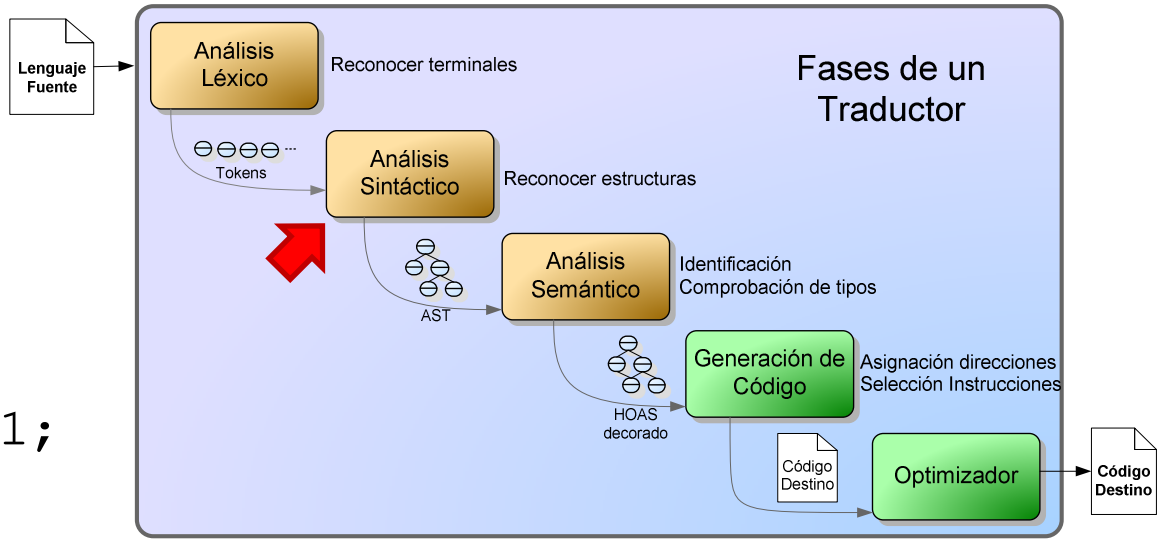


- Entran 50 números y salen 18

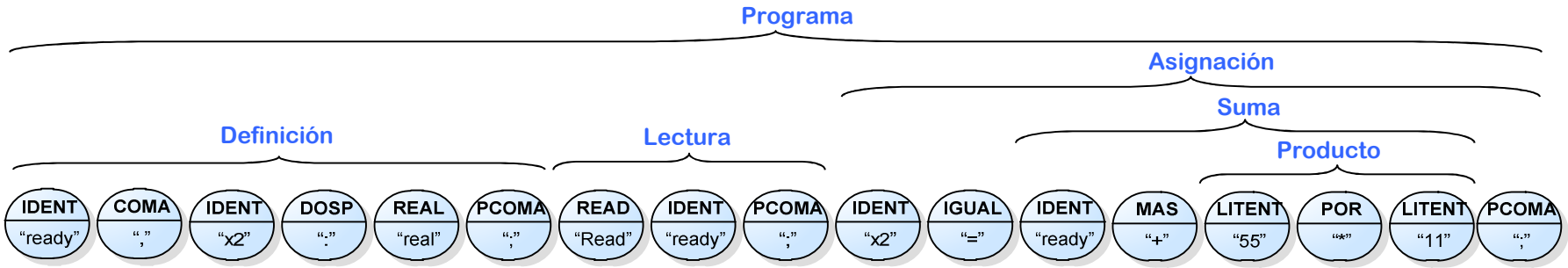
Ejercicio

Fase 2. Sintáctico

```
ready, x2:real;
Read ready;
x2 = ready + 55 * 11;
```

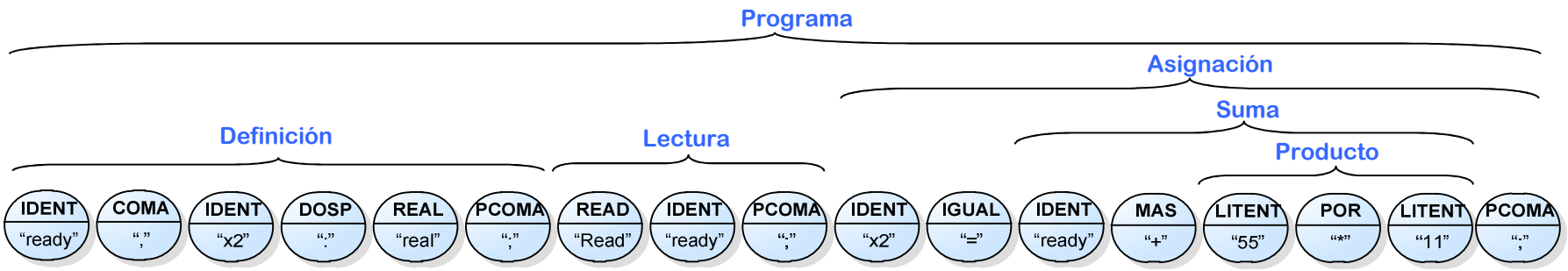


- Árbol Concreto

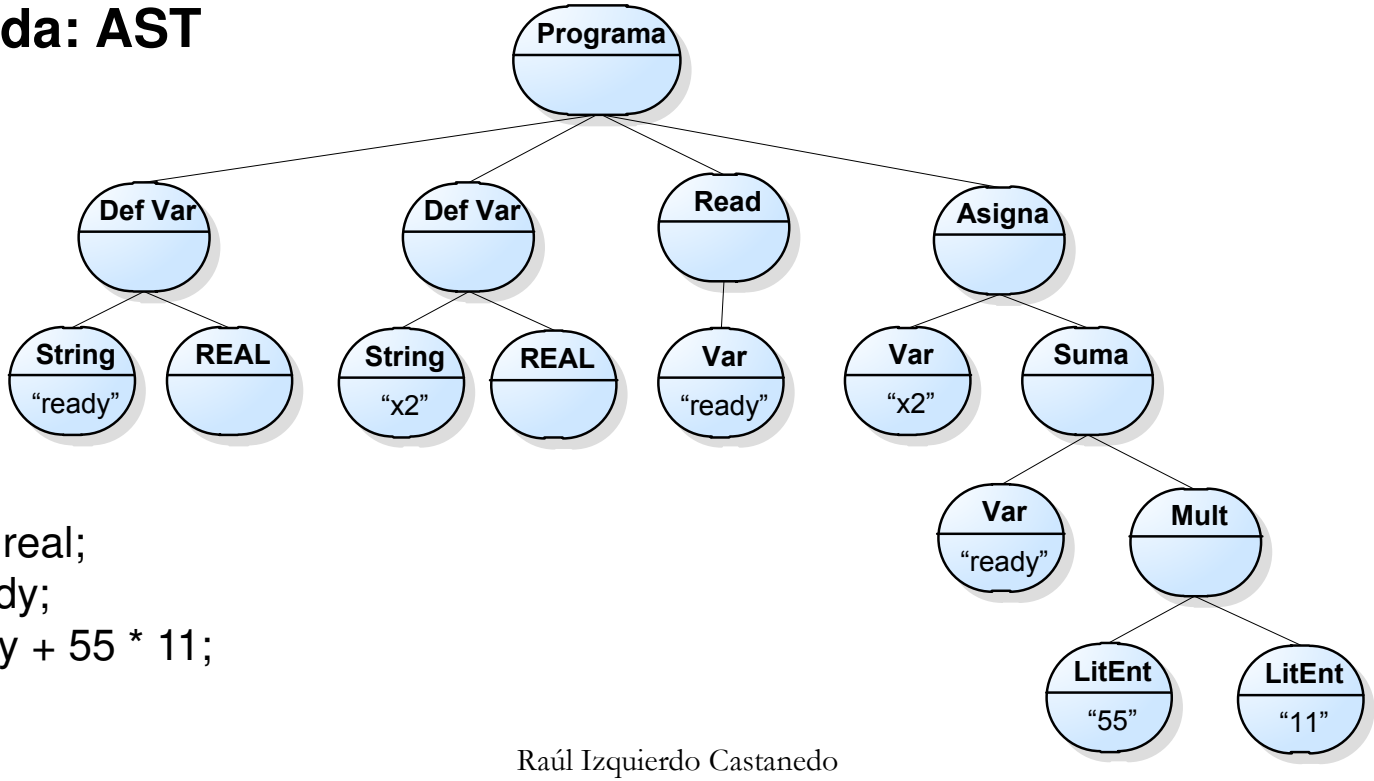


- ¿Árbol Abstracto (AST)?

Ejercicio

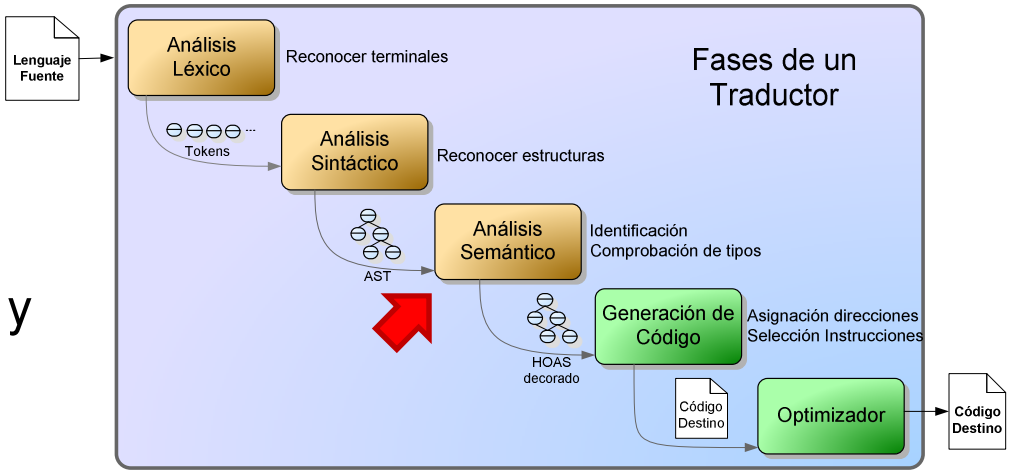


Salida: AST



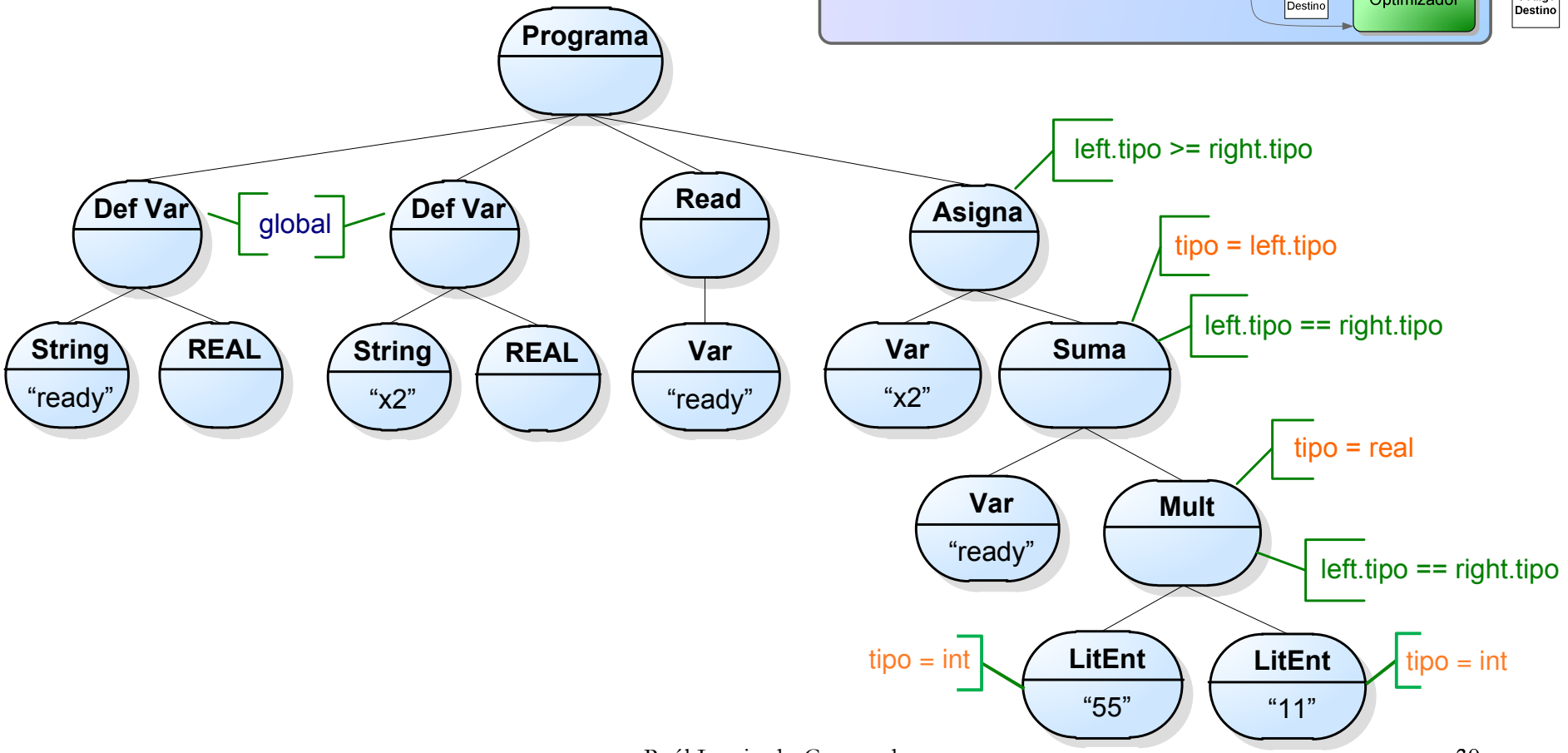
ready, x2:real;
Read ready;
x2 = ready + 55 * 11;

Ejercicio



Fase 3. Semántico

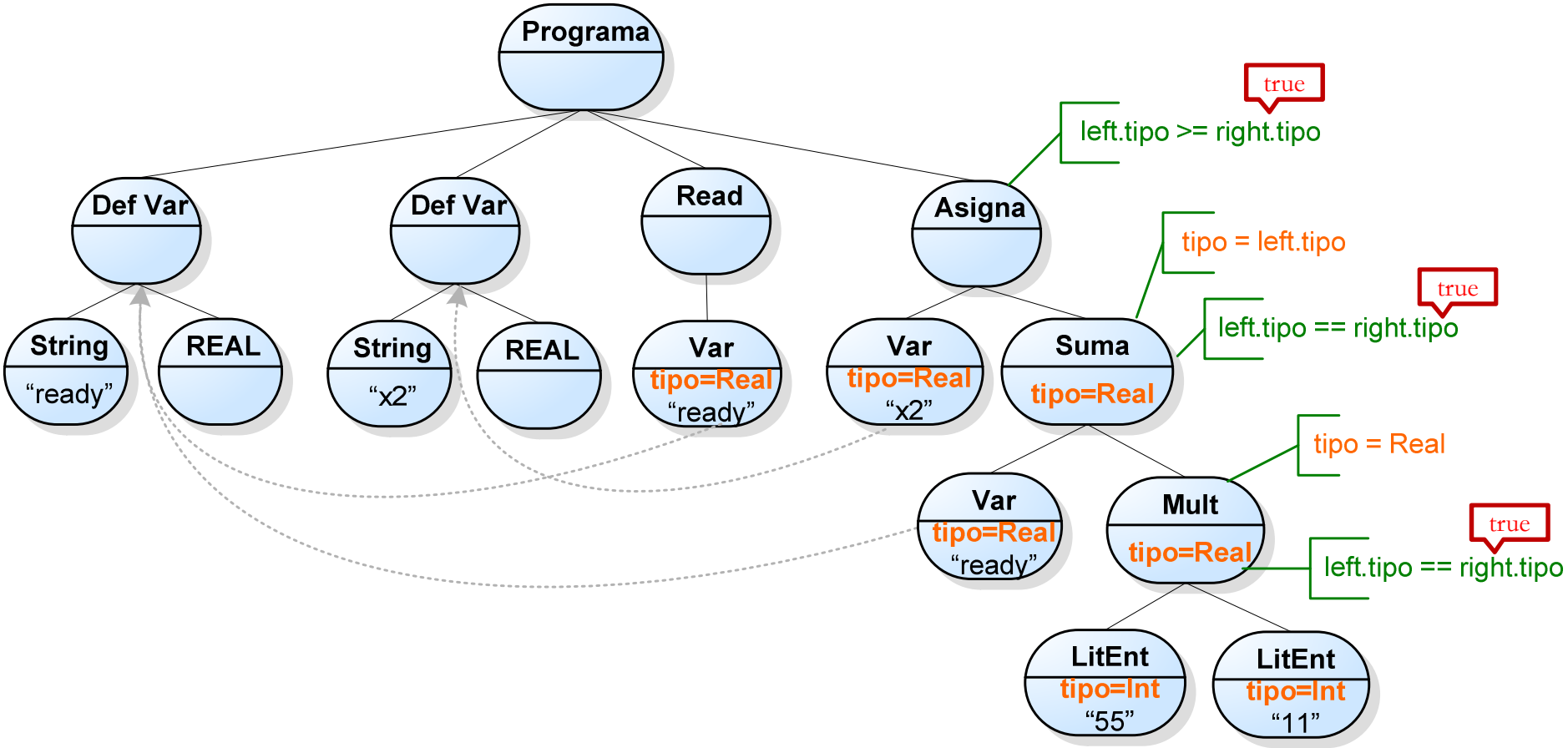
Reglas de Ámbito (azul) y
Reglas de Tipo (predicados y funciones de tipo)



Ejercicio

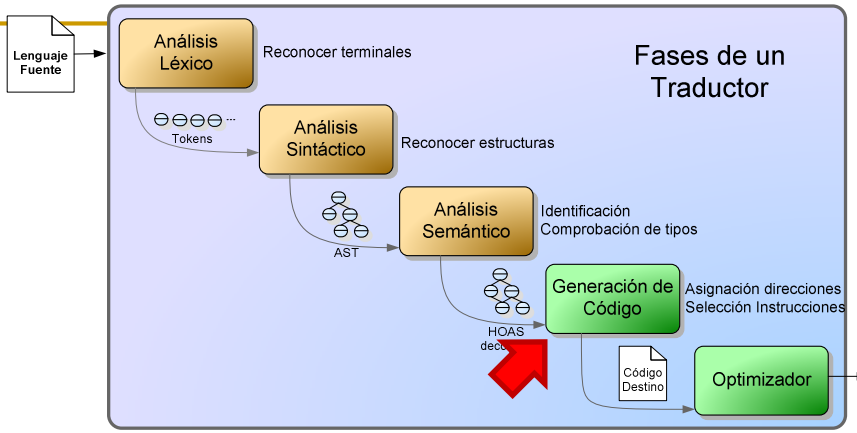
Fase 3. Semántico

Resultado de las Fases de Identificación y de Inferencia de Tipos

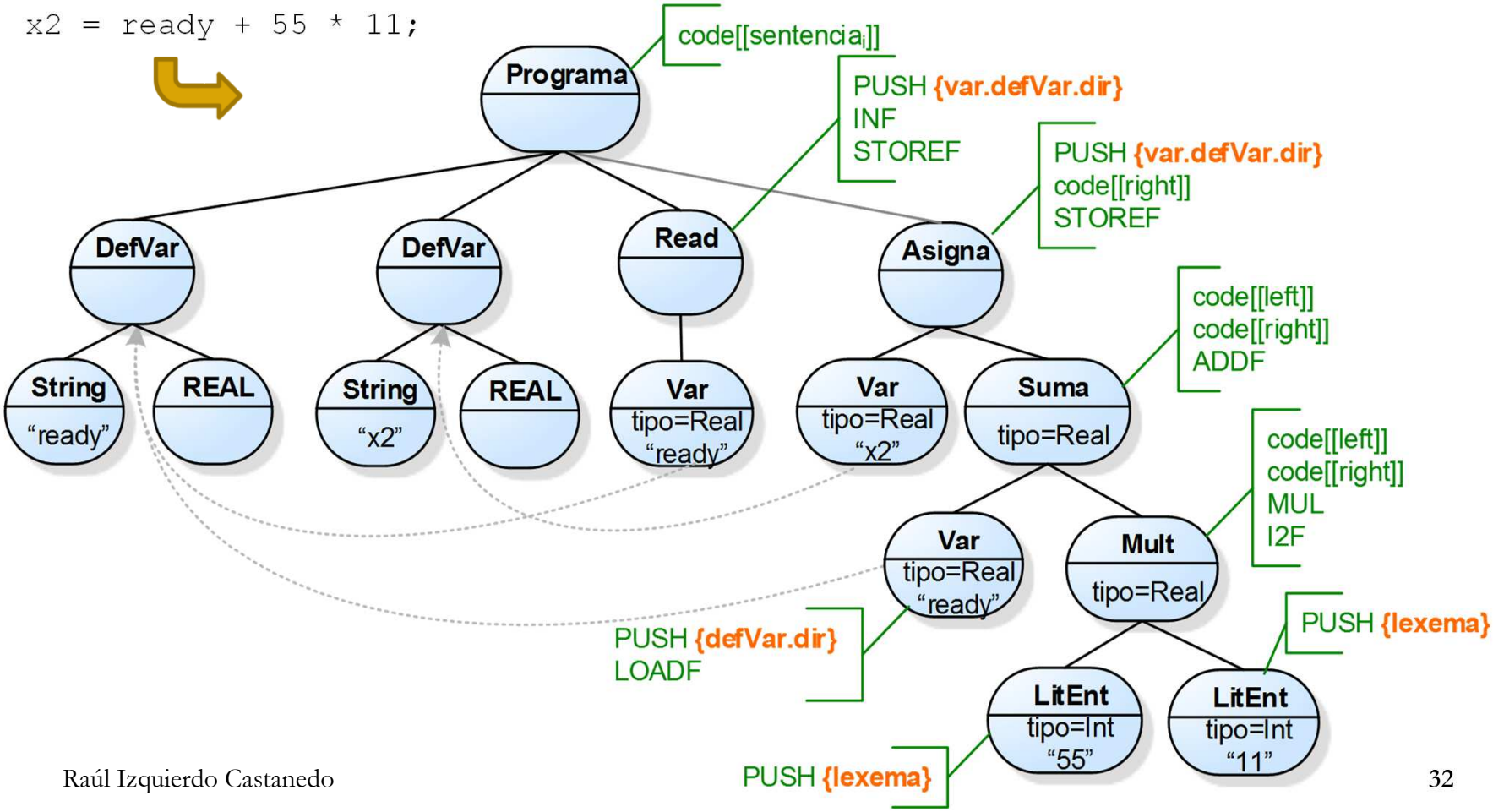


Ejercicio

Fase 4. Generación de Código



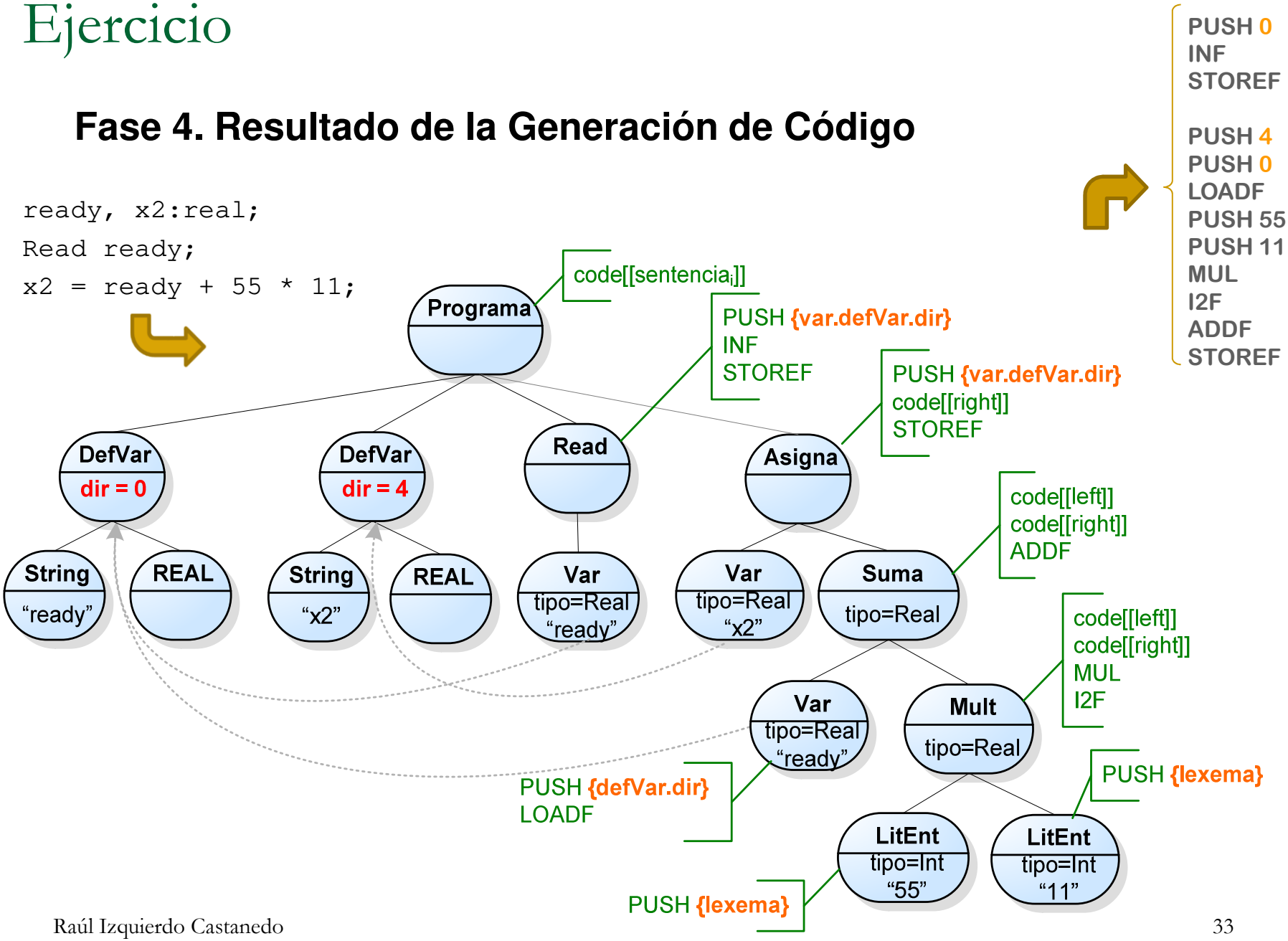

```
ready, x2:real;
Read ready;
x2 = ready + 55 * 11;
```



Ejercicio

Fase 4. Resultado de la Generación de Código

```
ready, x2:real;
Read ready;
x2 = ready + 55 * 11;
```



Fases de un Traductor. Resumen