

# **Especificación de un Lenguaje de Programación Mínimo y Construcción de su Procesador (6)**

---

## **Procesadores de Lenguaje**

### **Especificación formal de la traducción**

## Especificación formal de la traducción

---

### **1. (cont.) Especificación de un lenguaje de programación mínimo y construcción de su procesador**

- Lenguaje objeto y máquina virtual.
- **Especificación formal de la traducción.**
  - **Gramáticas de atributos y semántica operacional**
- Construcción del procesador.
  - Analizadores recursivos descendentes y procesadores de un solo paso.
  - Transformaciones en las gramáticas de atributos.
  - Esquemas de traducción.
- Fases de un compilador.
- Reingeniería de dos ejemplos de procesadores de lenguajes sencillos.

# Especificación de la Traducción

---

- El último aspecto que debe abordarse en la especificación del procesador de lenguaje es la traducción del lenguaje fuente al lenguaje objeto.
- Dicha traducción puede especificarse mediante una gramática de atributos, que indique cómo construir un atributo "código" cuyo valor sea el *programa* objeto.
- Será necesario utilizar la tabla de símbolos para obtener información adicional necesaria para la traducción: p.e. la dirección de las variables.

3

# Especificación de la Traducción

---

- En el lenguaje ejemplo será necesario:
  - Asociar direcciones a los identificadores durante la construcción de la tabla de símbolos.
  - Especificar la traducción de las expresiones e instrucciones de asignación del programa a código de la máquina P.
- Discusión:
  - ¿que otra información podría ser necesario añadir a la tabla de símbolos?
  - ¿para qué serviría esta información?

4



# Especificación de la Traducción

## Generación de código

---

Atributos:

```
Prog, Is, I, IAsig, Exp, Term, Fact: sintetizados: cod  
OpAd, OpMul: sintetizados: cod  
iden: sintetizados: lex, dir  
número: sintetizados: lex, val  
Is, I, Iasig, Exp, Term, Fact:  
heredados:  $ts_h$  (especificación incompleta)
```

Gramática (fragmento):

```
Prog --> Decs & Is      Prog.cod = Is.cod  
Decs ...  
  
Is --> I                Is.cod = I.cod  
...
```

7

# Especificación de la Traducción

## Generación de código

---

```
...  
Is --> Is ; I      Is0.cod = Is1.cod || I.cod  
  
I --> IAsig       I.cod = IAsig.co  
  
IAsig --> iden := Exp  
          IAsig.cod = Exp.cod || despila-dir(iden.dir)  
          Iden.dir = dirID(IAsig.tsh, iden.lex)
```

8

# Especificación de la Traducción

## Generación de código

---

```
Gramática:
Exp --> Exp OpAd Term
           Exp0.cod = Exp1.cod || Term.cod || OpAd.cod
Exp --> Term
           Exp.cod = Term.cod

Term --> Term OpMul Fact
           Term0.cod = Term1.cod || Fact.cod || OpMul.cod
Term --> Fact
           Term.cod = Fact.cod

...
```

9

# Especificación de la Traducción

## Traducción

```
...
Fact --> iden
           Fact.cod = apila-dir(iden.dir)
           iden.dir = dirID(Fact.tsh, iden.lex)

Fact --> numero
           Fact.cod = apila(numero.val)
           numero.val = valorDe(numero.lex)

Fact --> ( Exp )   Fact.cod = Exp.cod
OpAd --> +         OpAd.cod = '+'
OpAd --> -         OpAd.cod = '-'
OpMul --> *        OpMul.cod = '*'
OpMul --> \        OpMul.cod = '\\'
```

10

# Especificación de la Traducción

## Generación de código

---

### *Ejercicios*

1. *Representa gráficamente las producciones y el flujo de información en las mismas en la gramática de atributos anterior.*
2. *Completa la gramática anterior escribiendo las ecuaciones y los atributos necesarios para propagar la tabla de símbolos*

# Especificación de la Traducción

## Generación de código

---

### *Discutir:*

- *utilización de atributos sintetizados vs. atributos heredados en la generación de código*
- *utilización de atributos remotos en el sentido de Knuth*