

# **Especificación de un Lenguaje de Programación Mínimo y Construcción de su Procesador (5)**

---

## **Procesadores de Lenguaje**

### **Lenguaje Objeto y Máquina Virtual**

## **Lenguaje Objeto y Máquina Virtual**

---

### **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.

# El Lenguaje Objeto y la Máquina Virtual

---

- Los procesadores de lenguaje producen *programas* en un lenguaje objeto que representa el *significado* de los programas procesados.
- Los programas en lenguaje objeto adquieren su significado al ser *interpretados (ejecutados)* en *máquinas*.
- La definición del procesador del lenguaje fuente involucra, aparte de la definición de dicho lenguaje fuente:
  - la definición de los elementos del lenguaje objeto,
  - su interpretación en una máquina,
  - la *traducción* de los elementos del lenguaje fuente a los elementos del lenguaje objeto.
- Para generalizar y maximizar la portabilidad de los procesadores, esta máquina no tiene porque corresponderse con una máquina real ⇒ puede ser una *máquina virtual*.

3

## El Lenguaje Objeto

---

- En su forma más simple, una frase del lenguaje objeto será una secuencia de símbolos o instrucciones.
- No obstante, se desea generalizar esta descripción, para contemplar otros casos (ejemplo: en la construcción de un intérprete interesa traducir el programa a una representación del mismo en forma de árbol).
- Es por ello que cada frase del lenguaje objeto se representará, para su manipulación, mediante un *término básico* (p.e. una lista).
- Dicha representación de las frases de un lenguaje mediante términos básicos se basa en la sintaxis abstracta del lenguaje, frente a la caracterización gramatical de las frases que se denomina *sintaxis concreta*.

4

# El Lenguaje Objeto

---

**La definición del lenguaje objeto se centrará en la definición de su sintaxis abstracta (es decir, de la representación de sus frases como términos básicos).**

**Dicha sintaxis abstracta se puede definir de manera inductiva.**

*El concepto de sintaxis abstracta también es aplicable al lenguaje fuente*

*- Discusión (ver 2.5 , 5.2 en Aho et al.): comparar la sintaxis concreta y abstracta del lenguaje de las expresiones.*

*- Ejercicio: escribir una gramática de atributos para el lenguaje de las expresiones que construya el árbol que representa la sintaxis abstracta del lenguaje y otra que construya el árbol concreto*

5

# La Máquina Virtual

---

- **La máquina virtual se describirá como un *sistema de transición* en un conjunto de *estados de cómputo*.**
- **Dicha descripción constará de:**
  - **Una descripción de los *estados de cómputo*.**
  - **Una descripción de los *estados iniciales*.**
  - **Una descripción de los *estados finales*.**
  - **Un conjunto de *reglas de transición* que relacionan estados con sus *estados siguientes*. Dichas reglas se denotarán como  $s \rightarrow s'$**

6

# El Lenguaje Objeto y la Máquina Virtual

---

- Si para un lenguaje se proporciona una sintaxis abstracta y una máquina virtual, entonces se dice que dicho lenguaje posee una *semántica operacional*.
- En esta semántica, el *significado* de los programas viene dado por la interpretación (*ejecución*) de los mismos en la máquina virtual.
- La traducción de un lenguaje fuente a un lenguaje objeto que tenga asociado una semántica operacional proporciona, a su vez, una *semántica operacional al lenguaje fuente*.

7

# El Lenguaje Objeto y la Máquina Virtual

---

- Ejemplo 1: Máquina URM (*Unlimited Register Machine*):
  - Lenguaje objeto, formado por los *programas* URM, términos básicos, definidos como sigue:  
Instrucciones URM (su significado informal entre paréntesis)
    - $Z(n)$ , con  $n$  un natural (pone a cero el reg  $n$ ).
    - $S(n)$ , con  $n$  un natural (incrementa en 1 el reg  $n$ ).
    - $C(n, m)$ , con  $n$  y  $m$  naturales (copia el reg  $n$  en el reg  $m$ ).
    - $J(n, m, k)$ , con  $n, m$  y  $k$  naturales (salta a  $k$  si reg  $n = \text{reg } m$ ).
  - Programas URM:
    - Son *listas* de instrucciones URM. Las listas  $L(T)$  sobre un conjunto de términos, o vocabulario del lenguaje,  $T$  se definen como:
      - »  $[] \in L(T)$ .
      - » Si  $t \in T$  y  $l \in L(T)$ , entonces  $\text{cons}(t, l) \in L(T)$ .

8

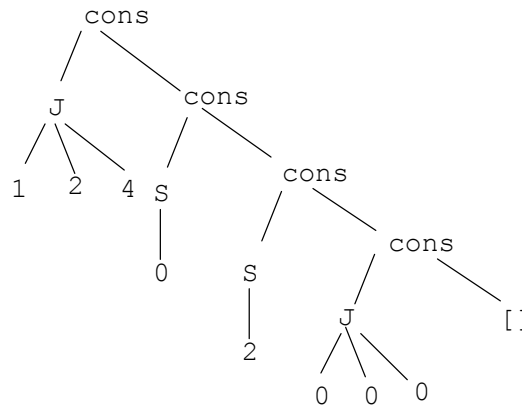
# El Lenguaje Objeto y la Máquina Virtual

---

- **Un programa URM:**

`cons(J(1,2,4),cons(S(0),cons(S(2),cons(J(0,0,0),[]))))`

- **Este programa también se puede representar como un árbol abstracto:**



*ejercicio: describir mediante una gramática una sintaxis concreta de este lenguaje*

9

# El Lenguaje Objeto y la Máquina Virtual

---

- La instrucción *i-ésima* de un programa URM  $P$  se define como  $I(P, i)$ , donde:
  - $I(\text{cons}(s, P), 0) = s$ .
  - $I(\text{cons}(s, P), i) = I(P, i - 1)$
  - *Indefinida* en cualquier otro caso.
- Los *estados* de la máquina URM son tóuplas  $\langle P, M, i \rangle$ , donde:
  - $P$  es un programa URM.
  - $M$  es una memoria de registros cuyos contenidos son números naturales.
  - $i$  es un número natural, denominado *instrucción actual*.

10

# El Lenguaje Objeto y la Máquina Virtual

---

- **Notación:**
  - $M(i)$ : Valor del registro  $i$ -ésimo en  $M$ .
  - $\{v_0, v_1, \dots, v_k\}$ : Memoria en la cuál  $M(0) = v_0, M(1) = v_1, \dots, M(k) = v_k$  y el resto de las posiciones son 0.
  - $M[d \leftarrow v]$ : Memoria  $M'$ , en la que  $M'(d) = v$  y  $M'(u) = M(u)$  para todo  $u \neq d$ .
- Los *estados iniciales* serán de la forma  $\langle P, M, 0 \rangle$ , con  $P$  un programa URM y  $M$  una memoria.
- Los *estados finales* serán de la forma  $\langle P, M, i \rangle$ , tal que  $I(P, i)$  esté indefinida.

11

# El Lenguaje Objeto y la Máquina Virtual

---

- **Reglas de transición:**
  - $\langle P, M, i \rangle \rightarrow \langle P, M', i+1 \rangle$  cuando  $I(P, i)$  no es del tipo  $J(n, m, k)$ . En este caso:
    - (R1) Si  $I(P, i)$  es  $Z(n)$ , entonces  $M' = M[n \leftarrow 0]$ .
    - (R2) Si  $I(P, i)$  es  $S(n)$ , entonces  $M' = M[n \leftarrow M(n)+1]$
    - (R3) Si  $I(P, i)$  es  $C(n, m)$  entonces  $M' = M[m \leftarrow M(n)]$
  - (R4)  $\langle P, M, i \rangle \rightarrow \langle P, M, k \rangle$  cuando  $I(P, i) = J(n, m, k)$  y  $M(n) = M(m)$
  - (R5)  $\langle P, M, i \rangle \rightarrow \langle P, M, i+1 \rangle$  cuando  $(I(P, i) = J(n, m, k) \text{ y } M(n) \neq M(m))$

12

# El Lenguaje Objeto y la Máquina Virtual

---

```
P = cons (J (1, 2, 4) , cons (S (0) , cons (S (2) , cons (J (0, 0, 0) , []) ) ) )
```

$\langle P, \{1,3\}, 0 \rangle \rightarrow^{R5} \langle P, \{1,3\}, 1 \rangle \rightarrow^{R2} \langle P, \{2,3\}, 2 \rangle \rightarrow^{R2}$   
 $\langle P, \{2,3,1\}, 3 \rangle \rightarrow^{R4} \langle P, \{2,3,1\}, 0 \rangle \rightarrow^{R5} \langle P, \{2,3,1\}, 1 \rangle \rightarrow^{R2}$   
 $\langle P, \{3,3,1\}, 2 \rangle \rightarrow^{R2} \langle P, \{3,3,2\}, 3 \rangle \rightarrow^{R4} \langle P, \{3,3,2\}, 0 \rangle \rightarrow^{R5}$   
 $\langle P, \{3,3,2\}, 1 \rangle \rightarrow^{R2} \langle P, \{4,3,2\}, 2 \rangle \rightarrow^{R2} \langle P, \{4,3,3\}, 3 \rangle \rightarrow^{R4}$   
 $\langle P, \{4,3,3\}, 0 \rangle \rightarrow^{R4} \langle P, \{4,3,3\}, 4 \rangle$

13

# El Lenguaje Objeto y la Máquina Virtual

---

- **Ejemplo 2: Máquina *aritmética* (intérprete de expresiones aritméticas).**
  - Lenguaje objeto: Está formado por *expresiones aritméticas*, términos básicos, definidos como sigue:
    - **Num(*n*)** es una expresión aritmética.
    - Si  $e_0$  y  $e_1$  son expresiones aritméticas, entonces también lo son
      - **Suma( $e_0, e_1$ )**.
      - **Resta( $e_0, e_1$ )**.
      - **Mul( $e_0, e_1$ )**.
      - **Div( $e_0, e_1$ )**.
  - **Ejemplo: Suma (Num (3) , Mul (Num (4) , Num (5) ) )**

14

# El Lenguaje Objeto y la Máquina Virtual

- Los estados de la máquina son pares del tipo  $\langle C, E \rangle$ , donde  $C$  y  $E$  son pilas.
- Los valores de  $C$  son términos de dos tipos:
  - $RED(e)$ , siendo  $e$  una expresión aritmética.
  - $OP(o)$ , con  $o \in \{+, -, *, /\}$ .
- Los valores de  $E$  son números.
- Notación:
  - $[]$ : Pila vacía.
  - $[v | R]$ : Pila con  $v$  como cabeza, y  $R$  como resto.
  - $[v] \equiv [v | []]$
  - $[v_0, v_1 | R] \equiv [v_0 | [v_1 | R]]$ .
  - $[v_0, v_1, v_2 | R] \equiv [v_0 | [v_1 | [v_2 | R]]]$ .
- Estados iniciales:  $\langle [RED(e)], [] \rangle$ , con  $e$  una expresión.
- Estados finales:  $\langle [], [v] \rangle$ .  $v$  es el *resultado* de evaluar la expresión.

15

# El Lenguaje Objeto y la Máquina Virtual

- Reglas de transición:
  - (R1)  $\langle [RED(Suma(e_0, e_1)) | R], E \rangle \rightarrow \langle [RED(e_0), RED(e_1), OP(+)|R], E \rangle$
  - (R2)  $\langle [RED(Resta(e_0, e_1)) | R], E \rangle \rightarrow \langle [RED(e_0), RED(e_1), OP(-)|R], E \rangle$
  - (R3)  $\langle [RED(Mul(e_0, e_1)) | R], E \rangle \rightarrow \langle [RED(e_0), RED(e_1), OP(*)|R], E \rangle$
  - (R4)  $\langle [RED(Div(e_0, e_1)) | R], E \rangle \rightarrow \langle [RED(e_0), RED(e_1), OP(/)|R], E \rangle$
  - (R5)  $\langle [RED(Num(n)) | R], E \rangle \rightarrow \langle R, [n|E] \rangle$
  - (R6)  $\langle [OP(+)|Rc], [v_0, v_1 | Re] \rangle \rightarrow \langle Rc, [v_1 + v_0 | Re] \rangle$
  - (R7)  $\langle [OP(-)|Rc], [v_0, v_1 | Re] \rangle \rightarrow \langle Rc, [v_1 - v_0 | Re] \rangle$
  - (R8)  $\langle [OP(*)|Rc], [v_0, v_1 | Re] \rangle \rightarrow \langle Rc, [v_1 * v_0 | Re] \rangle$
  - (R9)  $\langle [OP(/)|Rc], [v_0, v_1 | Re] \rangle \rightarrow \langle Rc, [v_1 / v_0 | Re] \rangle$

16



# El Lenguaje Objeto y la Máquina Virtual

---

$\langle [\text{RED}(\text{Suma}(\text{Num}(3), \text{Mul}(\text{Num}(4), \text{Num}(5))))], [] \rangle \rightarrow^{R1}$   
 $\langle [\text{RED}(\text{Num}(3)), \text{RED}(\text{Mul}(\text{Num}(4), \text{Num}(5))), \text{OP}(+)], [] \rangle \rightarrow^{R5}$   
 $\langle [\text{RED}(\text{Mul}(\text{Num}(4), \text{Num}(5))), \text{OP}(+)], [3] \rangle \rightarrow^{R3}$   
 $\langle [\text{RED}(\text{Num}(4)), \text{RED}(\text{Num}(5)), \text{OP}(*), \text{OP}(+)], [3] \rangle \rightarrow^{R5}$   
 $\langle [\text{RED}(\text{Num}(5)), \text{OP}(*), \text{OP}(+)], [4, 3] \rangle \rightarrow^{R5}$   
 $\langle [\text{OP}(*), \text{OP}(+)], [5, 4, 3] \rangle \rightarrow^{R8}$   
 $\langle [\text{OP}(+)], [20, 3] \rangle \rightarrow^{R6}$   
 $\langle [], [23] \rangle$

17

## El Lenguaje Objeto y la Máquina Virtual Máquina P

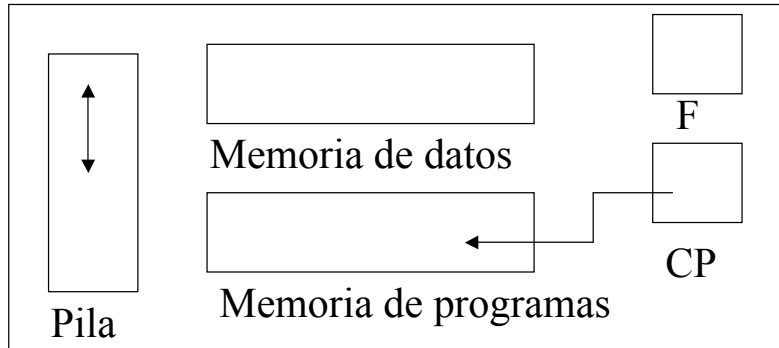
---

- Para el procesador del lenguaje ejemplo se utilizará un tipo de máquina virtual denominada *máquina P* (versiones de la máquina P han sido utilizadas en los procesadores de Pascal, Modula, Smalltalk y Java)
- Se definirá una máquina P mínima que dé soporte al lenguaje ejemplo. Dicha máquina se ampliará posteriormente (ver sección 2.8 Aho et al.).
- Descripción informal:
  - El componente central de esta máquina es la *pila* de operandos.
  - Todos los cálculos se realizan en dicha pila.
  - Dicha máquina posee también una *memoria de datos*, cuyas celdas pueden almacenar valores numéricos.
  - Por último, la máquina posee un *contador de programa*, una *memoria de programa* donde se almacena el programa que se está ejecutando, y un *indicador de estado*.

18

# El Lenguaje Objeto y la Máquina Virtual Máquina P

---



19

# El Lenguaje Objeto y la Máquina Virtual Máquina P

---

- **Lenguaje objeto:**
  - **Instrucciones:**
    - suma, resta, multiplica, divide.
    - $\text{apila}(n)$ , con  $n$  un número.
    - $\text{apila-dir}(d)$ , con  $d$  un número.
    - $\text{desapila-dir}(d)$ , con  $d$  un número.
  - **Programas (secuencias de instrucciones):** Son listas de instrucciones.
  - **Azúcar sintáctico:**
    - $\text{cons}(s, p) \equiv s ; p$  (si  $p \neq []$ )
    - $\text{cons}(s, []) \equiv s$ .

20

# El Lenguaje Objeto y la Máquina Virtual

## Máquina P

---

- Significado informal de las instrucciones
  - *suma, resta, multiplica y divide*: desapilan la cima (2º operando) y la subcima (1º operando) de la pila, operan con ellas, y apilan el resultado.
  - *apila(n)*: apila el valor *n*.
  - *apila-dir(d)*: apila el contenido de la celda *d* de la memoria de datos.
  - *desapila-dir(d)*: desapila y almacena el valor en la celda *d* de la memoria de datos.

21

# El Lenguaje Objeto y la Máquina Virtual

## Máquina P

---

- Traducción *manual* de un programa a código P

```
tiempo; espacio; velocidad  
&  
espacio := 25;  
tiempo := 30;  
velocidad := espacio / tiempo
```



```
apila(25);  
despila-dir(1);  
apila(30);  
despila-dir(0);  
apila-dir(1);  
apila-dir(0);  
divide;  
desapila-dir(2).
```

22

# El Lenguaje Objeto y la Máquina Virtual

## Máquina P

- La instrucción  $k$ -ésima de un programa objeto es  $I(P, k)$ , definida como en el caso de los programas URM.
- Los estados de la máquina P vienen dados por las tuplas  $\langle P, S, M, CP, F \rangle$  donde:
  - P es un programa.
  - S es la pila.
  - M es una memoria RAM cuyos contenidos son números.
  - CP es el valor del contador de programa (un número).
  - F es un *indicador* que vale  $r$  (máquina en ejecución),  $s$  (máquina parada),  $e$  (indicación de error).
- Estados iniciales:  $\langle P, [], \{ \}, 0, r \rangle$
- Estados finales:  $\langle P, S, M, CP, s \rangle$  y  $\langle P, S, M, CP, e \rangle$

23

# El Lenguaje Objeto y la Máquina Virtual

## Máquina P

### Reglas de transición:

- (R1)  $\langle P, [v_0, v_1 | R], M, i, r \rangle \rightarrow \langle P, [v_1 + v_0 | R], M, i+1, r \rangle$  cuando  $I(P, i) = \text{suma}$ .
- (R2)  $\langle P, [v_0, v_1 | R], M, i, r \rangle \rightarrow \langle P, [v_1 - v_0 | R], M, i+1, r \rangle$  cuando  $I(P, i) = \text{resta}$ .
- (R3)  $\langle P, [v_0, v_1 | R], M, i, r \rangle \rightarrow \langle P, [v_1 * v_0 | R], M, i+1, r \rangle$  cuando  $I(P, i) = \text{multiplica}$ .
- (R4)  $\langle P, [v_0, v_1 | R], M, i, r \rangle \rightarrow \langle P, [v_1 / v_0 | R], M, i+1, r \rangle$  cuando  $I(P, i) = \text{divide}$ .
- (R5)  $\langle P, S, M, i, r \rangle \rightarrow \langle P, [n | S], M, i+1, r \rangle$  cuando  $I(P, i) = \text{apila}(n)$ .
- (R6)  $\langle P, S, M, i, r \rangle \rightarrow \langle P, [M(d) | S], M, i+1, r \rangle$  cuando  $I(P, i) = \text{apila-dir}(d)$ .
- (R7)  $\langle P, [v | S], M, i, r \rangle \rightarrow \langle P, S, M[d \leftarrow v], i+1, r \rangle$  cuando  $I(P, i) = \text{desapila-dir}(d)$ .
- (R8)  $\langle P, S, M, i, r \rangle \rightarrow \langle P, S, M, i, s \rangle$  cuando  $I(P, i)$  está indefinida.
- (R9)  $\langle P, S, M, i, r \rangle \rightarrow \langle P, S, M, i, e \rangle$  en otro caso (por defecto, la máquina entra en estado de error).

*discusión: ejemplos de implementación*

24