

Implementación Correcta

Yolanda Ortega Mallén

Dpto. de Sistemas Informáticos y Computación

Universidad Complutense de Madrid

Cartagena99

CLASES PARTICULARES TUTORÍAS
LLAMA O ENVIA WHATSAPP. 689 45 44 70
ONLINE PRIVATE LESSONS FOR SC
CALL OR WHATSAPP. 689 45 44 70

Sumario

- La máquina abstracta.
- Traducción.
- Corrección.
- Bisimulación

Cartagena99

CLASES PARTICULARES TUTORÍAS
LLAMA O ENVIA WHATSAPP. 689 45 44 70
ONLINE PRIVATE LESSONS FOR SC
CALL OR WHATSAPP. 689 45 44 70

Configuraciones + Instrucciones

Configuraciones $\langle c, e, s \rangle \in \mathbf{Code} \times \mathbf{Stack} \times \mathbf{State}$

- c **código**, secuencia de instrucciones
- e **pila de evaluación**, $\mathbf{Stack} = (\mathbb{Z} \cup \mathbf{T})^*$
- s **almacén**

```

inst ::= PUSH-n | ADD | MULT | SUB
      | TRUE | FALSE | EQ | LE | NEG | AND
      | FETCH-x | STORE-x
      | NOOP | BRANCH(c,c) | LOOP(c,c)
c ::=  $\epsilon$  | inst : c
  
```

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Semántica operacional

$\langle \text{PUSH-}n : c, e, s \rangle$	\triangleright	$\langle c, \mathcal{N}[[n]] : e, s \rangle$	
$\langle \text{ADD} : c, z_1 : z_2 : e, s \rangle$	\triangleright	$\langle c, (z_1 \oplus z_2) : e, s \rangle$	si $z_1, z_2 \in \mathbb{Z}$
$\langle \text{MULT} : c, z_1 : z_2 : e, s \rangle$	\triangleright	$\langle c, (z_1 \otimes z_2) : e, s \rangle$	si $z_1, z_2 \in \mathbb{Z}$
$\langle \text{SUB} : c, z_1 : z_2 : e, s \rangle$	\triangleright	$\langle c, (z_1 \ominus z_2) : e, s \rangle$	si $z_1, z_2 \in \mathbb{Z}$
$\langle \text{TRUE} : c, e, s \rangle$	\triangleright	$\langle c, \mathbf{tt} : e, s \rangle$	
$\langle \text{FALSE} : c, e, s \rangle$	\triangleright	$\langle c, \mathbf{ff} : e, s \rangle$	
$\langle \text{EQ} : c, z_1 : z_2 : e, s \rangle$	\triangleright	$\langle c, (z_1 = z_2) : e, s \rangle$	si $z_1, z_2 \in \mathbb{Z}$
$\langle \text{LE} : c, z_1 : z_2 : e, s \rangle$	\triangleright	$\langle c, (z_1 \leq z_2) : e, s \rangle$	si $z_1, z_2 \in \mathbb{Z}$

$$\langle \text{AND} : c, t_1 : t_2 : e, s \rangle \triangleright \begin{cases} \langle c, \mathbf{tt} : e, s \rangle & \text{si } t_1 = \mathbf{tt} \text{ y } t_2 = \mathbf{tt} \\ \langle c, \mathbf{ff} : e, s \rangle & \text{si } t_1 = \mathbf{ff} \text{ o } t_2 = \mathbf{ff}, \text{ y } t_1, t_2 \in \mathbf{T} \end{cases}$$

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Semántica operacional

$\langle \text{FETCH-}x : c, e, s \rangle$	\triangleright	$\langle c, (s \ x) : e, s \rangle$
$\langle \text{STORE-}x : c, z : e, s \rangle$	\triangleright	$\langle c, e, s[x \mapsto z] \rangle$ si $z \in \mathbb{Z}$
$\langle \text{NOOP} : c, e, s \rangle$	\triangleright	$\langle c, e, s \rangle$
$\langle \text{BRANCH}(c_1, c_2) : c, t : e, s \rangle$	\triangleright	$\begin{cases} \langle c_1 : c, e, s \rangle & \text{si } t = \mathbf{tt} \\ \langle c_2 : c, e, s \rangle & \text{si } t = \mathbf{ff} \end{cases}$
$\langle \text{LOOP}(c_1, c_2) : c, e, s \rangle$	\triangleright	$\langle c_1 : \text{BRANCH}(c_2 : \text{LOOP}(c_1, c_2), \text{NOOP}) : c, e, s \rangle$

Secuencia de cómputo $\gamma_0 \triangleright \gamma_1 \triangleright \gamma_2 \triangleright \dots \triangleright \gamma_k$

- finita / termina

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Propiedades

Inducción sobre la **longitud de la secuencia de cómputo**

Ejercicio 4.4

Demostrar que se puede extender el código y la pila de evaluación de MA sin que cambie su comportamiento:

$$\langle c_1, e_1, s \rangle \triangleright^k \langle c', e', s' \rangle \implies \langle c_1 : c_2, e_1 : e_2, s \rangle \triangleright^k \langle c' : c_2, e' : e_2, s' \rangle$$

Ejercicio 4.5

Demostrar que la ejecución de una secuencia de instrucciones puede fraccionarse:

$$\langle c_1 : c_2, e, s \rangle \triangleright^k \langle \varepsilon, e'', s'' \rangle \implies \\ \exists \langle \varepsilon, e', s' \rangle \exists k_1, k_2. \langle c_1, e, s \rangle \triangleright^{k_1} \langle \varepsilon, e', s' \rangle \wedge \langle c_2, e', s' \rangle \triangleright^{k_2} \langle \varepsilon, e'', s'' \rangle \\ \text{con } k = k_1 + k_2.$$

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Función semántica

Significado de una secuencia de instrucciones:

$$\mathcal{M} : \mathbf{Code} \longrightarrow (\mathbf{State} \leftrightarrow \mathbf{State})$$

$$\mathcal{M}[[c]]s = \begin{cases} s' & \text{si } \langle c, \varepsilon, s \rangle \triangleright^* \langle \varepsilon, e, s' \rangle \\ \text{INDEFINIDO} & \text{e.c.c.} \end{cases}$$

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVIA WHATSAPP. 689 45
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Modificaciones

Ejercicio 4.7

Modificar MA para referirse a las variables por su **dirección**:

- las configuraciones son $\langle c, e, m \rangle$, donde $m \in \mathbb{Z}^*$ representa la **memoria** ($m[n]$ selecciona el n -ésimo valor de la lista m);
- sustituir **FETCH- x** y **STORE- x** por **GET- n** y **PUT- n** , siendo $n \in \mathbb{N}$ una dirección.

Dar la semántica operacional de la máquina modificada MA_1 .

Ejercicio 4.8

Modificar MA_1 para introducir **instrucciones de salto**:

- las configuraciones son $\langle pc, c, e, m \rangle$, donde $pc \in \mathbb{N}$ es el **contador de programa** que apunta a alguna instrucción en c ($c[pc]$ indica la instrucción en c apuntada por pc);

Cartagena99

CLASES PARTICULARES TUTORÍAS
LLAMA O ENVÍA WHATSAPP. 689 45
ONLINE PRIVATE LESSONS FOR SC
CALL OR WHATSAPP. 689 45 44 70

Traducción de expresiones

Expresiones aritméticas

$$CA : Aexp \rightarrow Code$$

$$CA[[n]] = PUSH-n$$

$$CA[[x]] = FETCH-x$$

$$CA[[a_1 + a_2]] = CA[[a_2]] : CA[[a_1]] : ADD$$

$$CA[[a_1 \times a_2]] = CA[[a_2]] : CA[[a_1]] : MULT$$

$$CA[[a_1 - a_2]] = CA[[a_2]] : CA[[a_1]] : SUB$$

Expresiones booleanas

$$CB : Bexp \rightarrow Code$$

$$CB[[true]] = TRUE$$

$$CB[[false]] = FALSE$$

$$CB[[a_1 \wedge a_2]] = CB[[a_1]] : CB[[a_2]] : AND$$

$$CB[[a_1 \vee a_2]] = CB[[a_1]] : CB[[a_2]] : OR$$

$$CB[[\neg a_1]] = CB[[a_1]] : NOT$$

$$CB[[a_1 \Rightarrow a_2]] = CB[[a_1]] : CB[[a_2]] : IMPL$$

$$CB[[a_1 \Leftrightarrow a_2]] = CB[[a_1]] : CB[[a_2]] : XNOR$$

$$CB[[a_1 \Rightarrow a_2]] = CB[[a_1]] : CB[[a_2]] : IMPL$$

$$CB[[a_1 \Leftrightarrow a_2]] = CB[[a_1]] : CB[[a_2]] : XNOR$$

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Traducción de sentencias

$$CS : \text{Stm} \rightarrow \text{Code}$$

$$CS[x := a] = CA[a] : \text{STORE-}x$$

$$CS[\text{skip}] = \text{NOOP}$$

$$CS[S_1; S_2] = CS[S_1] : CS[S_2]$$

$$CS[\text{if } b \text{ then } S_1 \text{ else } S_2] = CB[b] : \text{BRANCH}(CS[S_1], CS[S_2])$$

$$CS[\text{while } b \text{ do } S] = \text{LOOP}(CB[b], CS[S])$$

Ejercicio 4.11

Aunque $\mathcal{A}[(a_1 + a_2) + a_3] = \mathcal{A}[a_1 + (a_2 + a_3)]$, demostrar que, en general, $CA[(a_1 + a_2) + a_3] \neq CA[a_1 + (a_2 + a_3)]$; sin embargo se *comportan* de forma similar.

Ejercicio 4.11



CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Función semántica

Significado de una sentencia: Traducir a código de MA y ejecutarlo.

$$\mathcal{S}_{\text{ma}} : \text{Stm} \longrightarrow (\text{State} \leftrightarrow \text{State})$$

$$\mathcal{S}_{\text{ma}}[[S]] = (\mathcal{M} \circ \mathcal{CS})[[S]]$$

Ejercicio 4.16

Modificar la generación de código para traducir **While** a MA_1 .

Utilizar entornos $\text{env} : \text{Var} \longrightarrow \mathbb{N}$ que asocian cada variable a su dirección.

Ejercicio 4.17

Modificar la generación de código para traducir **While** a MA_2 .

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Corrección de la implementación de expresiones

Lema 8:

$$\forall a \in \mathbf{Aexp}, \forall s \in \mathbf{State}. \langle \mathcal{CA}[[a]], \varepsilon, s \rangle \triangleright^* \langle \varepsilon, \mathcal{A}[[a]]s, s \rangle$$

Además, en todas las configuraciones intermedias de esta secuencia de cómputo, la pila de evaluación es no vacía.

Ejercicio 4.19

Demostrar un resultado similar para las expresiones booleanas:

$$\forall b \in \mathbf{Bexp}, \forall s \in \mathbf{State}. \langle \mathcal{CB}[[b]], \varepsilon, s \rangle \triangleright^* \langle \varepsilon, \mathcal{B}[[b]]s, s \rangle$$

Demostrar además que en todas las configuraciones intermedias de esta secuencia de cómputo, la pila de evaluación es no vacía.

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

Corrección de la implementación de sentencias

Teorema 9:

$$\forall S \in \mathbf{Stm}. \mathcal{S}_{bs} \llbracket S \rrbracket = \mathcal{S}_{ma} \llbracket S \rrbracket$$

Lema 10:

$$\forall S \in \mathbf{Stm}, \forall s, s' \in \mathbf{State}. \langle S, s \rangle \rightarrow s' \implies \langle \mathcal{CS} \llbracket S \rrbracket, \varepsilon, s \rangle \triangleright^* \langle \varepsilon, \varepsilon, s' \rangle$$

Lema 11:

$$\forall S \in \mathbf{Stm}, \forall s, s' \in \mathbf{State}. \langle \mathcal{CS} \llbracket S \rrbracket, \varepsilon, s \rangle \triangleright^k \langle \varepsilon, e, s' \rangle \implies \langle S, s \rangle \rightarrow s' \wedge e = \varepsilon$$

Resumen demostración corrección

① **Inducción sobre el árbol de derivación**

Para cada árbol de derivación en la semántica de paso largo existe la correspondiente secuencia de cómputo finita en la máquina abstracta.

Cartagena99

CLASES PARTICULARES TUTORÍAS
 LLAMA O ENVÍA WHATSAPP. 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SC
 CALL OR WHATSAPP. 689 45 44 70

semántica de paso largo.

Corrección de la implementación de sentencias

Ejercicio 4.23

Modificar la función de traducción de forma que $\mathcal{CS}[\text{skip}] = \varepsilon$.
¿Cómo afecta este cambio a la demostración del Teorema 9?

Ejercicio 4.24

Extender la demostración del Teorema 9 para incluir la sentencia
`repeat S until b`.

Ejercicio 4.25

Demostrar la corrección del código generado para MA_1 .
¿Qué hay que asumir para *env*?

Ejercicio 4.26

Cartagena99

CLASES PARTICULARES TUTORÍAS
LLAMA O ENVIA WHATSAPP. 689 45 44 70
ONLINE PRIVATE LESSONS FOR SC
CALL OR WHATSAPP. 689 45 44 70

Demostración alternativa de corrección

- Utilizar la semántica de **paso corto**.
- Definir una relación de **bisimulación**:

$$\begin{aligned}\langle S, s \rangle &\approx \langle \mathcal{CS}[[S]], \varepsilon, s \rangle \\ s &\approx \langle \varepsilon, \varepsilon, s \rangle\end{aligned}$$

Ejercicio 4.27 + 4.28

- 1 Demostrar que a cada paso de la semántica de paso corto le corresponde una secuencia de pasos en la máquina abstracta:

$$\gamma_{ss} \approx \gamma_{ma} \wedge \gamma_{ss} \Rightarrow \gamma'_{ss} \Longrightarrow \exists \gamma'_{ma} \cdot \gamma'_{ss} \approx \gamma'_{ma} \wedge \gamma_{ma} \triangleright^+ \gamma'_{ma}$$

Inferir que $\langle S, s \rangle \Rightarrow^* s' \Longrightarrow \langle \mathcal{CS}[[S]], \varepsilon, s \rangle \triangleright^* \langle \varepsilon, \varepsilon, s' \rangle$

- 2 Consideramos secuencias de cómputo que parten de una pila de evaluación vacía y terminan con la pila vacía.

Asumiendo que $\gamma_{ss} \approx \gamma_{ma}^1$ y que $\gamma_{ma}^1 \triangleright \gamma_{ma}^2 \triangleright \dots \triangleright \gamma_{ma}^k$, con $k > 1$ y la

**CLASES PARTICULARES TUTORÍAS
LLAMA O ENVÍA WHATSAPP. 689 45
ONLINE PRIVATE LESSONS FOR SC
CALL OR WHATSAPP. 689 45 44 70**

Cartagena99

Demostración alternativa de corrección

$$\forall S \in \mathbf{Stm}. \mathcal{S}_{ss}[[S]] = \mathcal{S}_{ma}[[S]]$$

Resumen demostración corrección con bisimulación

- 1 Un paso en la semántica de paso corto puede simularse mediante una secuencia no vacía de pasos en la máquina abstracta.
Extender a secuencias de pasos en la semántica de paso corto.
- 2 Una secuencia no vacía de pasos en la máquina abstracta que cumple algunas restricciones puede simularse mediante un paso en la semántica de paso corto. Extender a secuencias más generales en la máquina abstracta.

Ejercicio 4.30

Sustituir la regla para los bucles en la semántica de paso corto por dos axiomas:

$$\begin{aligned} \langle \text{while } b \text{ do } S, s \rangle &\Rightarrow \langle S; \text{while } b \text{ do } S, s \rangle && \text{si } \mathcal{B}[[b]]s = \mathbf{tt} \\ \langle \text{while } b \text{ do } S, s \rangle &\Rightarrow s && \text{si } \mathcal{B}[[b]]s = \mathbf{ff} \end{aligned}$$

Cartagena99

CLASES PARTICULARES TUTORÍAS
LLAMA O ENVÍA WHATSAPP. 689 45
ONLINE PRIVATE LESSONS FOR SC
CALL OR WHATSAPP. 689 45 44 70