

Instrucciones

=> Asignación (única instrucción que modifica la memoria)

$y := S[x]$ (la variable y, guarda el valor de S[x]

- 1) Buscamos el valor de x en memoria
- 2) Agregamos un sucesor a dicho valor
- 3) Guardamos el nuevo valor en y, en memoria

=> Estructuras de control (definen qué asignaciones vamos a ejecutar)

Asignación Múltiple/Simultánea => $x_1, \dots, x_N := e_1, \dots, e_N$

$[(x, S[O])] \Rightarrow x, y := S[x], x \Rightarrow [(x, S[S[O]]), (y, S[O])]$ Memoria inicial
Programa Memoria resultante

Intercambio de valores de variables:

$[(x, 1), (y, 2)] \Rightarrow x, y := y, x \Rightarrow [(x, 2), (y, 1)]$

Estructuras de control (definen el flujo del programas)

- 1) Secuencia $p_1 ; p_2$ (se ejecuta primero p_1 y luego p_2)
- 2) Selección case exp of ramas
- 3) Repetición while exp do ramas

Ejemplos

Negación Booleana ($b := !b$)

Case b of {False $\rightarrow b := \text{True}$ | True $\rightarrow b := \text{False}$ }

Conjunción Booleana ($b := b \ \&\& \ c$)

Case b of {False $\rightarrow b := \text{False}$ | True $\rightarrow b := c$ }

Disyuncion Booleana ($b := b \ || \ c$)

Case b of {True $\rightarrow b := \text{True}$ | False $\rightarrow b := c$ }

Par: ($b := \text{par}(n)$)

$b := \text{true};$

while n do {S[x] $\rightarrow n := x;$

$(b := !b)$ }

SINTAXIS

Valores $v := C \text{ vs}$

Expresiones $e := x \mid C \text{ es}$

Programas/Instrucciones:

Asignación $p := xs := es$

Secuencia $p1 ; p2$

Selección $\text{case } e \text{ of } \{rs\}$

Repetición $\text{while } e \text{ do } \{rs\}$

$\text{rama} := C \text{ xs} \rightarrow p$

p programa

$\text{type rama} = (C, xs, p)$

Memoria: lista de pares variable - valor $[(\text{variable}, \text{valor})]$

Id variable

Operaciones: Actualizar $(\leftarrow) :: \text{Mem} \rightarrow [(\text{Id}, \text{valor})] \rightarrow \text{Mem}$

Buscar $(@) :: \text{Mem} \rightarrow \text{Id} \rightarrow \text{valor}$

Vacía: $\varepsilon @ x = \text{Null } []$

Última actualización: $(m \leftarrow x \text{ vs}) @ x = \{$

$v \text{ si } (x, v) \in x \text{ vs}$

$m @ x \text{ en otro caso}$

$\}$

Evaluación de expresiones:

Variable: $y := x;$

Constructor: $S[...] \rightarrow S[...]$

Ejecución de un programa: Dado un programa y una memoria inicial, producir la memoria resultante de ejecutar el programa

$m > p > m'$

Casos:

$p := xs := es$

$\mid p1 ; p2$

$\mid \text{case } e \text{ of } rs$

$\mid \text{while } e \text{ do}$

$\text{rama} := C \text{ xs} \rightarrow p$