Nociones Básicas

Programación Avanzada

UNRC
Pablo Castro

Conceptos Básicos

Algunos conceptos básicos:

- Expresiones, Valores, funciones,
- Igualdad,
- Razonamiento riguroso,
- Programación funcional.

Valores

Los valores son elementos de los conjuntos que usamos para programar:

Naturales: *0,1,2,3…*

Enteros: *0,-1,1,-2,2,-3…*

Strings:*"mystring", "", "hola mundo", "etc"*

Reales: *0, 0.1, 0,31415…, 0,22222*

Expresiones

Son las cadenas de texto que utilizamos para describir valores

- **Naturales**: $2 + 3, 2 * 5 + 10, 2 * 100 * 10, \ldots$
- **Booleanos**: $true, false, p \wedge q, \neg p, (p \vee q) \Rightarrow s, ...$
- **Strings**: $``Hola" + ``Mundo", ``", ...$
- **Enteros**: $-10 + 5, 10 * -1, 15 * 10 - 4, ...$

Puede haber expresiones que no denotan ningún valor

Puede haber expresiones que denotan el mismo valor “3” y “2+1”

Definición Formal

Podemos definir a las expresiones aritméticas formalmente.

- Cualquier constante es una expresión, (por ejemplo “2”)
- Si E y F son expresiones, entonces E+F, E-F, E*F, E\F son expresiones.
- Ninguna otra cadena es una expresión.

De la misma forma podemos definir expresiones *booleanas* (ejercicio…)

Expresiones vs Valores

Las expresiones **denotan** o **representan** valores pero no son valores.

- *“2”* denota el número 2, *“II”* en romano también denota el número 2 y *“10”* en binario también denota el mismo valor
- *“2+2”* denota el número 4.
- *“2+x”* denota una función (dado el valor de x retorna x mas dos)
- *“10/0”* no denota ningún valor.

Estados y Asignaciones de Valores

La noción de estado es una de las más importantes en programación imperativa:

> Dado un conjunto de variables (x, y, z, v, ..), un estado es una asignación de valores a las variables

Ejemplo:

$$x \mapsto 3, y \mapsto 2, z \mapsto 1$$

Asignaciones

```
var
        x,y : Integer;
begin
      x:=10;
      y:=15,
      y:=x+y;
end
```

X=10, y=0

X=25, y=15

Estados y Ecuaciones

Las asignaciones nos permiten evaluar expresiones:

$$x \mapsto 10, y \mapsto 5 \qquad \text{ó} \qquad \{(x, 10), (y, 5)\}$$

Por ejemplo:

$$\{(x, 10), (y, 5)\} \vDash x + y = 15$$

Estado

Se lee: “cumple”

Ecuación

Sustituciones

Dadas expresiones E y F:

$$E(x := F)$$

Los lenguaje funcionales usan sustituciones para calcular calores

Denota la expresión que resulta de reemplazar todas las ocurrencias de x por F.

Ejemplo

$$
\begin{aligned}
&(x + y)(y := 2 * z) \\
= \ &[\text{Definición de Sustitución}] \\
&x + (2 * z)
\end{aligned}
$$

La Igualdad

La igualdad es muy importante, recordemos sus propiedades:

$$X = X$$ Reflexividad

$$X = Y \Rightarrow Y = X$$ Simetria

$$X = Y \wedge Y = Z \Rightarrow X = Z$$ Transitividad

Además la regla de Leibniz:

$$X = Y \Rightarrow E[x := X] = E[x := Y]$$

Funciones

El concepto de función es muy importante:

$$f : A \to B$$

f es una función

Si:

$$f \subseteq A \times B$$

Es una relación

$$\forall x, y : x = y \Rightarrow f(x) = f(y)$$

Es determinista

Si además se cumple

$$\forall a \in A : \exists b \in B : b = f(a)$$

Se dice que es una función total

Ejemplos de Funciones

Veamos algunos ejemplos:

$$f(x) = x^3$$

Falta el perfil! $f : \mathbb{N} \to \mathbb{N}$

$$f(x, y, z) = x + y + z$$

perfil: $f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$$f(x, y) = x/y$$

perfil: $f : \mathbb{N} \to \mathbb{N}$

Función **parcial**

Notación

En la materia utilizaremos una notación cercana a los lenguajes funcionales

En vez de: $f(x)$ escribimos $f.x$

Para definir una función usamos:

Perfil de la función

$$f : A \to B$$

$$f.x \doteq E$$

Definición por medio de una expresión

Ejemplo

Veamos un ejemplo:

$$f : \mathbb{N} \to \mathbb{N}$$

$$f.x \doteq x^3$$

Para evaluar una función se usan sustituciones:

Todo el proceso es una cadena de igualdades

$$\begin{array}{l} f.2 \\ = [\text{def. de } f] \\ x^3[x := 2] \\ = [\text{sustitución}] \\ 2^3 \\ = [\text{Aritmética}] \\ 8 \end{array}$$

Esta tarea es la que hace un interprete funcional de forma rápida

Razonamiento Lógico

Sistemas de calculo:

Axiomas + Reglas

Un ejemplo para el max y min:

Axiomas

- Conmutatividad: $X \max Y = Y \max X$
- Asociatividad: $X \max (Y \max Z) = (X \max Y) \max Z$
- Idempotencia: $X \max X = X$
- Distrib.max, $+$: $X + (Y \max Z) = (X + Y \max X + Z)$
- Monotonia: $X \max Y \geq X$

Ejemplo de regla

Si P es un teorema,

entonces $P(x := E)$ también es un teorema.

Demostración

Utilizando los axiomas y las reglas podemos demostrar propiedades:

$$W \text{ max } X + Y \text{ max } Z \stackrel{?}{=} (W+Y) \text{ max } (W+Z) \text{ max } (X+Y) \text{ max } (X+Z)$$

$$(W+Y) \text{ max } (W+Z) \text{ max } (X+Y) \text{ max } (X+Z)$$
$$= [\text{Distrib. } + \text{ con respecto max}]$$
$$(W+Y \text{ max } Z) \text{ max } (X+Y) \text{ max } (X+Z)$$
$$= [\text{Distrib. } + \text{ con respecto max}]$$
$$(W+Y \text{ max } Z) \text{ max } (X + Y \text{ max } Z)$$
$$= [\text{Conmutatividad de } +]$$
$$(Y \text{ max } Z + W) \text{ max } (Y \text{ max } Z + X)$$
$$= [\text{Distrib. } + \text{ con respecto max}]$$
$$(Y \text{ max } Z) + (W \text{ max } X)$$
$$= [\text{Conmutatividad de } +]$$
$$(W \text{ max } X) + (Y \text{ max } Z)$$

Programas y Pruebas

Podemos definir programas para calcular Max y Min:

$$max.x.y \doteq \text{if } x > y \text{ then } x \text{ else } y$$

$$min.x.y \doteq \text{if } x < y \text{ then } x \text{ else } y$$

Acabamos de demostrar una propiedad sobre estos programas.

Demostrar la misma propiedad pero utilizando la definición de los programas.