

1 Macros TAD, proc, pred y aux

// Macros para TAD y spec.

Bloque **ES** Struct $\langle id : \mathbb{Z}, lista : Seq\langle \dots \rangle \rangle$

TAD nombreTad {

}

TAD Diccio $\langle K, V \rangle$ {

obs $d : Dict\langle K, V \rangle$

proc definirYValidar(inout D : Diccio, in k : K, in v : V) : Bool {

requiere : $\{D = D_0\}$

asegura : $\{D.d = \text{setKey}(D_0, k, v)\}$

asegura : $\{res = \text{true} \longleftrightarrow k \in D.d\}$

 }

pred esLlave(D : Diccio, k : K) {

$res = \text{true} \longleftrightarrow k \in D.d$

 }

pred soloValoresRepetidos(D : Diccio) {

 Devuelve true si los valores son todos iguales...

 }

aux sumaElementosSiSonPrimos(s : Seq $\langle \mathbb{Z} \rangle$) : $\mathbb{Z} = \dots$

$|s|-1$

$\sum_{i=0} \text{IfThenElse}(\text{esPrimo}(s[i]), s[i], 0)$

}

2 Ejemplo de comandos

nombre **ES** definicion

obs *var* : Tipo

requiere : $\{\dots\}$

asegura : $\{\dots\}$

in *a* : TipoA

out *b* : TipoB

inout *c* : TipoC

// Tipos

\mathbb{N} , \mathbb{Z} , \mathbb{R}

Bool, Float, Char, String

T , T

Tuple $\langle \mathbb{Z}, T \rangle$

Struct $\langle id : \mathbb{Z}, sueldo : \mathbb{R} \rangle$

Seq $\langle \text{Char} \rangle$

Conj $\langle \dots \rangle$

Dict $\langle K, V \rangle$

// Parentesis grandes (para sumas y uniones sirve)

$\left(\left(\dots \right) \right)$

// Suma, union y otros

$\sum_{i=0}^{\infty} \dots$

$\sum_{i=0}^{|A|-1} \left(\sum_{j=0}^{|A[i]|-1} A[i][j] \right)$

$\sum_{a \in A} \text{func}(a)$

superindice

$\bigcup_{\text{subindice}} \text{aplicacion}$

// Llamado a funciones, sirve para el modo math

funcion(*a*, *b*)

versus:

funcion(*a*, *b*)

```

// Logica
true, false,  $\wedge$ ,  $\vee$ ,  $\longrightarrow$ ,  $\longleftrightarrow$ 
 $\wedge_L$ ,  $\vee_L$ ,  $\longrightarrow_L$ 
// Version corta de logica (por si esta muy grande)

 $\rightarrow y \rightarrow_L$ 
// Dos versiones de IfThenElse

IfThenElse( $B, S_1, S_2$ )
if  $B$  then  $S_1$  else  $S_2$  fi

```