Especificación del TAD List

spec List **of** T **where**

constructors

fun empty_list() **ret** l : List **of** T
{- Crea una lista vacía. -}

proc addl(**in** e : T, **in/out** l : List **of** T)
{- Agrega el elemento e al comienzo de la lista l. -}

destroy

proc destroy_list(**in/out** l : List **of** T)
{- Libera memoria en caso que sea necesario. -}

operations

fun is_empty(l : List **of** T) **ret** b : bool
{- Devuelve True si l es vacía. -}

fun head(l : List **of** T) **ret** e : T
{- Devuelve el primer elemento de la lista l -}
{- **PRE**: not is_empty(l) -}

proc tail(**in/out** l : List **of** T)
{- Elimina el primer elemento de la lista l -}
{- **PRE**: not is_empty(l) -}

proc addr(**in/out** l : List **of** T, **in** e : T)
{- Agrega el elemento e al final de la lista l. -}

fun length(l : List **of** T) **ret** n : nat
{- Devuelve la cantidad de elementos de la lista l -}

proc concat(**in/out** l : List of T, **in** l0 : List **of** T)
{- Agrega al final de l todos los elementos de l0 en el mismo orden.-}

fun index(l : List **of** T, n : nat) **ret** e : T
{- Devuelve el n-ésimo elemento de la lista l -}
{- **PRE**: length(l) > n -}

proc take(**in/out** l : List **of** T, **in** n : nat)
{- Deja en l sólo los primeros n elementos, eliminando el resto -}

proc drop(**in/out** l : List **of** T, **in** n : nat)
{- Elimina los primeros n elementos de l -}

fun copy_list(l1 : List **of** T) **ret** l2 : List **of** T
{- Copia todos los elementos de l1 en la nueva lista l2 -}

Especificación del TAD Set

spec Set **of** T **where**

constructors

fun empty_set() **ret** s : Set **of** T
{- Crea un conjunto vacío -}

proc add(**in/out** s : Set **of** T, **in** e : T)
{- Agrega el elemento e al conjunto s -}

destroy

proc destroy_set(**in/out** s : Set **of** T)
{- Libera memoria en caso que sea necesario. -}

operations

fun cardinal(s : Set **of** T) **ret** n : nat
{- Devuelve la cantidad de elementos que tiene s -}

fun is_empty_set(s : Set **of** T) **ret** b : bool
{- Devuelve True si s es vacío -}

fun member(e : T, s : Set **of** T) **ret** b : bool
{- Devuelve True si el elemento e pertenece al conjunto s -}

proc elim(**in/out** s : Set **of** T, **in** e : T)
{- Elimina el elemento e del conjunto s, en caso que esté -}

proc union(**in/out** s : Set **of** T, **in** s0 : Set **of** T)
{- Agrega a s todos los elementos de s0 -}

proc inters(**in/out** s : Set **of** T, **in** s0 : Set **of** T)
{- Elimina de s todos los elementos que NO pertenezcan a s0 -}

proc diff(**in/out** s : Set **of** T, **in** s0 : Set **of** T)
{- Elimina de s todos los elementos que pertenecen a s0 -}

fun get(s : Set **of** T) **ret** e : T
{- Obtiene algún elemento cualquiera del conjunto s -}
{- **PRE**: not is_empty_set(s) -}

```
fun copy_set(s1 : Set of T) ret s2 : Set of T
{- Copia el conjunto s1 -}
```