

Ejercicio 1. Tenemos un TAD que modela las ventas minoristas de un comercio. Cada venta es individual (una unidad de un producto) y se quieren registrar todas las ventas. El TAD tiene un único observador:

```
Producto ES string
Monto ES  $\mathbb{Z}$ 
Fecha ES  $\mathbb{Z}$  – segundos desde 1/1/1970

TAD Comercio {
  obs ventasPorProducto : dicc(Producto, seq(tupla (Fecha, Monto)))
}
```

ventasPorProducto contiene, para cada producto, una secuencia con todas las ventas que se hicieron de ese producto. Para cada venta, se registra la fecha y el precio. Se puede considerar que todas las fechas son diferentes. Este TAD lo vamos a implementar con la siguiente estructura:

```
Módulo ComercioImpl implementa Comercio <
  var ventas: Vector<tupla<Producto, Fecha, Monto>>
  var totalPorProducto: Diccionario<Producto, Monto>
  var ultimoPrecio: Diccionario<Producto, Monto>
>
```

- **ventas** es una implementación de secuencia con todas las ventas realizadas, indicando producto, fecha y monto.
- **totalPorProducto** asocia cada producto con el dinero total obtenido por todas sus ventas.
- **ultimoPrecio** asocia cada producto con el monto de su última venta registrada.

- a) Escribir en castellano, con precisión y detalle el invariante de representación y la función de abstracción.
- b) Escribir ambos en el lenguaje de especificación.

Ejercicio 2. Dados el siguiente TAD y su implementación

```
Cientifico, Paper ES  $\mathbb{Z}$ 

TAD CitasCientíficas {
  obs publicaciones : dicc(Paper, seq(Cientifico))
  obs citas : dicc(Paper, conj(Paper))
  proc iniciarSistema() : CitasCientíficas
    // El sistema comienza con ambos diccionarios vacíos
  proc registrarPaper(inout cc : CitasCientíficas, in p : Paper, in cs : seq(Cientifico))
    // El paper no se encuentra en el sistema previamente
    // La secuencia no tiene científicos repetidos
    // Registra el paper en publicaciones con su lista de autores y en citas con vacío
  proc citarPaper(inout cc : CitasCientíficas, in p : Paper, in c : Paper)
    // Ambos papers existen en el TAD
    // Ninguno de los dos papers había sido citado previamente por el otro
    // Se agrega c al conjunto de citas de p en citas
}
```

```
Módulo CitasCientíficasImpl implementa CitasCientíficas <
  /* Todos los científicos que han publicado algún paper asociados a todos sus papers */
  var publicaciones: Diccionario<Científico, Conjunto<Paper>>

  /* Para cada paper, la lista de científicos en el orden que aparecen en la
  publicación */
  var autores: Diccionario<Paper, Vector<Científico>>
```

```
/* Tuplas donde la primer componente es el paper que cita y la segunda el
   paper citado */
var citas: Conjunto<Tupla<Paper, Paper>>

/* La primera componente de las tuplas es la cantidad de veces que el paper
   de la segunda fue citado.
   * La prioridad está determinada solamente por la primera componente */
var masCitados: ColaDePrioridadMax<Tupla<int, Paper>>

>
```

- a) Indique si cada una de las sentencias propuestas pertenecen al invariante de representación para la estructura propuesta
- i) Dados dos científicos distintos en **publicaciones**, sus significados son disjuntos
 - ii) Para todo paper de **autores** no hay repetidos en el vector de su significado
 - iii) Para todo par de tuplas del conjunto de **citas** si existe (a,b) no existe (b,a)
 - iv) No hay tuplas repetidas en **citas**
 - v) Hay la misma cantidad de claves definidas en **autores** que elementos en **citas**
 - vi) Para cada tupla de **citas**, ambos papers son claves de **autores**
- b) Proponga dos sentencias más para el invariante de representación (en castellano preciso)
- c) Escribir la función de abstracción en lógica de primer orden