

Invariante de representación y función de abstracción en modelado de problemas

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: dict(Producto, seq(⟨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: SecuenciaImpl<tupla<Producto, Fecha, Monto>>
  var totalPorProducto: DiccionarioImpl<Producto, Monto>
  var ultimoPrecio: DiccionarioImpl<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.

Se pide:

- Escribir en forma coloquial y detallada el invariante de representación y la función de abstracción.
- Escribir ambos en el lenguaje de especificación.

Figura 1: Enunciado Problema 1

```
Producto = int
Fecha = int
Monto = float
Ventas = SecuenciaImpl<Tupla<Producto, Fecha, Monto>>

Modulo ComercioImpl implements Comercio {
  var ventas: Ventas
  var totalPorProducto: DiccionarioImpl<Producto, Monto>
  var ultimoPrecio: DiccionarioImpl<Producto, Monto>

  aux indiceDeFechaMasReciente (c: ComercioImpl, i1:  $\mathbb{Z}$ , i2:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = IfThenElse(
    c.ventas.s[i1]1 > c.ventas.s[i2]1, i1, i2
  );
  aux indiceUltimoPrecio (c: ComercioImpl, ventas: Ventas, inicio:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = IfThenElse(
    inicio = |ventas.s| - 1,
    inicio,
    indiceDeFechaMasReciente(c, inicio, ultimoPrecio(c, ventas, inicio + 1))
  );
  aux totalPorProducto (c: ComercioImpl, producto: Producto) :  $\mathbb{Z}$  =
     $\sum_{i=0}^{|c.ventas.s|-1}$  IfThenElse(c.ventas.s[i]0 = producto, c.ventas.s[i]2, 0);

  pred abs (ab: ArbolBinarioImpl<T>, ab': ArbolBinario<T>) {
    Abs
  }
  pred invRep (l: ListaEnlazada<T>) {
    InvRep
  }
}
```