

Algoritmos y Estructuras de Datos II - 1º cuatrimestre 2018

Práctico 2 - Parte 2

1. El tipo de los valores de verdad es muy elemental pero no siempre es uno de los tipos concretos en los lenguajes de programación. Especificá el TAD *Booleano* incluyendo constructores (*Verdadero* y *Falso*), las operaciones usuales ($\wedge, \vee, \Rightarrow, \equiv, \neg$), y las ecuaciones que las describen.
2. Especificá el TAD *Natural* de los números naturales utilizando dos constructores (*Cero* y *Sucesor*), y las operaciones aritméticas (más, por, pot) y relacionales ($<, \leq$, etc.) habituales. Para estas últimas utilizá el TAD Booleano del ejercicio anterior.
3. Algunos lenguajes de programación no incluyen el tipo *lista* como tipo concreto. Especificá el TAD *Lista* de valores de tipo T a partir de los constructores usuales: $[\]$ para crear la lista vacía y \triangleright para agregar un elemento al principio de una lista.

Otras operaciones a incluir son: \triangleleft para insertar un elemento por detrás, *tamaño* para observar la longitud de la lista, *cabeza* para observar el primer elemento de la lista, *cola* para eliminar el primer elemento de la lista, *es_vacia* para observar si la lista es vacía, *concat* para concatenar dos listas, \cdot y \downarrow para observar el elemento que se encuentra en una posición dada, y la sublista en esa posición. Notar que algunas operaciones tienen restricciones.

4. El TAD *cola* es ampliamente usado para resolver una gran variedad de problemas. Este uso se puede extender aún más si se incluyen operaciones para agregar, observar y eliminar elementos por ambos extremos. Una variante con estas características se denomina *cola doble* o *bicola*. Especificá este TAD que además de las 5 operaciones habituales del TAD *cola*, incluye operaciones para: devolver el último elemento, eliminar el último elemento, y agregar un elemento al comienzo de la bicola. Detallá los constructores, operaciones y ecuaciones necesarias.
5. Especificá el TAD *pila reversible*, que tiene los mismos constructores y operaciones que el TAD *pila*, y además la operación *invertir*.

Ayuda: Puede ser conveniente introducir operaciones auxiliares para caracterizar la operación *invertir*.

6. Especificá el tipo abstracto *conjunto finito* (de elementos de tipo T). Debe tener (al menos) operaciones para obtener la unión, intersección y diferencia de dos conjuntos, determinar si un conjunto es vacío y si un elemento pertenece a un conjunto. Se puede utilizar como constructores *vacío* o \emptyset (crea el conjunto vacío), *agregar* (agrega un elemento dado a un conjunto finito). Escribir las ecuaciones entre constructores que expresen las propiedades de conjuntos.
7. Especificá una expendedora de café y chocolate como tipo abstracto. El TAD tendrá, además del constructor vacío, tres constructores más: uno para reponer (una dosis de) café, otro para reponer (una dosis de) chocolate y otro para reponer un vaso descartable. Expresá que el orden en que se agreguen estos elementos es irrelevante.

La máquina expende café (respectivamente chocolate) con sólo oprimir el botón correspondiente.

Otras operaciones a incluir son: una que averigua si el dispositivo tiene café, otra que averigua si tiene chocolate, otra que averigua si tiene vasos. Finalmente, habrá dos operaciones de eliminación (una de una dosis de café y otra de una dosis de chocolate), que corresponden a la función de la máquina de expender café (o chocolate) al usuario. Ambas operaciones eliminarán también un vaso descartable. Estas operaciones sólo se aplican en el caso de haber café (respectivamente chocolate) y vasos.

Se pide también especificar tres operaciones que devuelvan, respectivamente, el número de dosis de café, de dosis de chocolate y de vasos que dispone la máquina, y tres operaciones correspondientes a la reposición, que permitan agregar n dosis de café, n dosis de chocolate, y n vasos.

ejercicios adicionales

8. Especificá un TAD *tablero* para mantener el tanteador en contiendas deportivas entre dos equipos (equipo A y equipo B). Deberá tener un constructor para el comienzo del partido (tanteador inicial), un constructor para registrar un nuevo tanto del equipo A y uno para registrar un nuevo tanto del equipo B. El tablero sólo registra el estado actual del tanteador, por lo tanto el orden en que se fueron anotando los tantos es irrelevante.

Además se requiere operaciones para comprobar si el tanteador está en cero, si el equipo A ha anotado algún tanto, si el equipo B ha anotado algún tanto, una que devuelva verdadero si y sólo si el equipo A va ganando, otra que devuelva verdadero si y sólo si el equipo B va ganando, y una que devuelva verdadero si y sólo si se registra un empate.

Finalmente habrá una operación que permita anotarle un número n de tantos a un equipo y otra que permita “castigarlo” restándole un número n de tantos. En este último caso, si se le restan más tantos de los acumulados equivaldrá a no haber anotado ninguno desde el comienzo del partido.

9. Un programa de computación simbólica permite a sus usuarios manipular *polinomios con coeficientes enteros*. Para ello se utilizará un TAD que entre sus operaciones incluye evaluar en la indeterminada x y devolver el coeficiente de grado k .

Especificá el TAD. Se puede definir como constructores uno que devuelve el polinomio nulo y un segundo que suma un monomio (de la forma ax^k) a un polinomio.