

Algoritmos y Estructura de Datos 2

Recuperatorios

X2

Alumno: Leandro Carreira

LU: 669/18

Link a documento online (en caso que algún caracter se haya pasado mal a .pdf):

<https://docs.google.com/document/d/1YTTKsL5jPRnfuZErhZuG9wHYfTRbwfUTkjYPc99aJyU/edit?usp=sharing>

Ejercicio X2

2. Ejercicio X2 — TADs

En la playa de Mar del Plata venden empanadas y queremos llevar un registro de sus movimientos.

Inicialmente contamos con un grupo de repartidores de empanadas, que se juntan en la confitería productora de las mismas. Cuando tiene ganas, algún repartidor se lleva varias empanadas consigo (la cantidad de quiera). A medida que pasa el tiempo los grupos de personas que están frente al mar llaman a algún repartidor y le piden la cantidad de empanadas que quieran, claramente no le pueden pedir más empanadas de las que tiene el repartidor en ese momento. Cuando esté cansado el repartidor vuelve a la base inmediatamente a descansar (si le sobran empanadas se las devuelve a la confitería). Si se le acaban las empanadas, el repartidor también debe volver y descansa hasta que le toque volver a salir.

Nos interesan saber varias cosas, primero cuál fue la compra más numerosa, es decir, cuál fue la máxima cantidad de empanadas vendidas en una sola compra y por cual vendedor (en caso de empate devuelvan todas las que correspondan). Por otro lado nos interesa saber cuales fueron los vendedores que más veces vendieron todas sus empanadas.

Modelar con el TAD EMP el sistema descripto. La especificación debe estar completa, incluyendo observadores, generadores, otras operaciones, restricciones y la axiomatización correspondiente. Se sugiere escribir la igualdad observacional, pero no es de caracter obligatorio.

Detalle de colores:

rojo: observadores
azul: generadores
violeta: otras operaciones y auxiliares
naranja: comentarios

Repartidor es String

TAD EMP

géneros emp

igualdad observacional

$$\begin{aligned} (\forall e, f: \text{emp}) \quad & e =_{\text{obs}} f \Leftrightarrow \\ & \text{repartidores}(e) =_{\text{obs}} \text{repartidores}(f) \wedge \\ & \text{comprasRecord}(e) =_{\text{obs}} \text{comprasRecord}(f) \wedge \\ (\forall r: \text{repartidor}) \quad & r \in \text{repartidores}(e) \Rightarrow \\ & \text{cantEmpanadas}(e, r) =_{\text{obs}} \text{cantEmpanadas}(e, f) \\ & \text{vecesQueVendióTodas}(e, r) =_{\text{obs}} \text{vecesQueVendióTodas}(e, r) \end{aligned}$$

observadores

repartidores: emp \rightarrow conj(repartidor)
comprasRecord: emp \rightarrow tupla(nat, conj(repartidor))
cantEmpanadas: emp e x repartidor r \rightarrow nat {r \in repartidores(e)}
vecesQueVendióTodas: emp e x repartidor r \rightarrow nat {r \in repartidores(e)}

generadores

iniciar: $\text{conj}(\text{repartidor}) \text{ cr} \rightarrow \text{emp} \quad \{\neg \text{vacío?}(\text{cr})\}$
tomarEmpanadas: $\text{emp } e \text{ x repartidor } r \text{ x nat } n \rightarrow \text{emp} \quad \{r \in \text{repartidores}(e)\}$
venderEmpanadas: $\text{emp } e \text{ x repartidor } r \text{ x nat } n \rightarrow \text{emp} \quad \{r \in \text{repartidores}(e) \wedge_L n \leq \text{cantEmpanadas}(e, r)\}$

otras operaciones

vendieronTodasMásVeces: $\text{emp} \rightarrow \text{conj}(\text{repartidor})$

axiomas $\forall e: \text{emp}, \forall \text{cr}: \text{conj}(\text{repartidor}), \forall r1, r2: \text{repartidor}, \forall n: \text{nat}$

$\text{repartidores}(\text{iniciar}(\text{cr})) \equiv \text{cr}$

$\text{repartidores}(\text{tomarEmpanadas}(e, r, n)) \equiv \text{repartidores}(e)$

$\text{repartidores}(\text{venderEmpanadas}(e, r, n)) \equiv \text{repartidores}(e)$

$\text{cantEmpanadas}(\text{iniciar}(\text{cr}), r) \equiv 0$

$\text{cantEmpanadas}(\text{tomarEmpanadas}(e, r1, n), r2) \equiv$

$\text{if } r1 = r2 \text{ then}$

n

else

$\text{cantEmpanadas}(e, r2)$

fi

$\text{cantEmpanadas}(\text{venderEmpanadas}(e, r1, n), r2) \equiv$

$\text{if } r1 = r2 \text{ then}$

$\text{cantEmpanadas}(e, r2) - n$

else

$\text{cantEmpanadas}(e, r2)$

fi

$\text{comprasRecord}(\text{iniciar}(\text{cr})) \equiv \langle 0, \emptyset \rangle$

$\text{comprasRecord}(\text{tomarEmpanadas}(e, r1, n)) \equiv \text{comprasRecord}(e)$

$\text{comprasRecord}(\text{venderEmpanadas}(e, r, n)) \equiv$

$\text{if } n = \pi_1(\text{comprasRecord}(e)) \text{ then}$

// Venta igual a record presente

$\text{if } r \notin \pi_2(\text{comprasRecord}(e)) \text{ then}$

// Agrego nombre al conjunto de la tupla

$\langle \pi_1(\text{comprasRecord}(e)),$

$\text{Ag}(r, \pi_2(\text{comprasRecord}(e))) \rangle$

else

// Vendedor ya en records,

// no modifica comprasRecords.

$\text{comprasRecord}(e)$

fi

r2)

```

    else if  $n > \pi_1(\text{comprasRecord}(e))$  then
        // Nuevo record! Pisa valores anteriores
        < n, Ag(r,  $\emptyset$ ) >
    else
        // Venta sub-record, no modifica comprasRecords.
        comprasRecord(e)
    fi
fi

vecesQueVendióTodas(iniciar(cr), r)  $\equiv$  0
vecesQueVendióTodas(tomarEmpanadas(e, r1, n), r2)  $\equiv$  vecesQueVendióTodas(e,
r2)
vecesQueVendióTodas(venderEmpanadas(e, r1, n), r2)  $\equiv$ 
    if (cantEmpanadas(e, r1) - n) = 0 then
        // Vendedor r1 vendió todas sus empanadas
        if r1 = r2 then
            1 + vecesQueVendióTodas(r2)
        else
            vecesQueVendióTodas(r2)
        fi
    else
        // Vendedor r1 no vendió todas, no cambia nada
        vecesQueVendióTodas(r2)
    fi

vendieronTodasMásVeces(in e: emp)  $\rightarrow$  conj(repartidor)
    filtrarMáximos(e, repartidores(e))

filtrarMáximos(in e: emp, in cr: conj(repartidor))  $\rightarrow$  conj(repartidor)
    // Devuelve repartidores con ventas completas igual al máximo
    if #(cr) = 0 then
         $\emptyset$ 
    else
        if vecesQueVendióTodas(e, dameUno(cr))
            =
            mayorCantDeVentasCompletas(e, repartidores(e)) then
            // Es uno de los que más vendió todas, lo agrego
            Ag( dameUno(cr), filtrarMáximos(e, sinUno(cr)) )
        else
            filtrarMáximos(e, sinUno(cr))
        fi
    fi
fi
```

```

mayorCantDeVentasCompletas(in e: emp, in cr: conj(repartidor)) → nat
  // Computa el máximo de ventas completas entre repartidores
  if #(cr) = 0 then
    0
  else
    if #(cr) = 1 then
      vecesQueVendióTodas(e, dameUno(cr))
    else
      máx( vecesQueVendióTodas(e, dameUno(cr)),
           mayorCantDeVentasCompletas(e, sinUno(cr)) )
    fi
  fi

```

Fin TAD

Nota:

El tad especificado asume un **comportamiento automático** de los vendedores en donde descansan cuando tienen ganas, y de hacerlo cuando todavía tienen empanadas, **las devuelven automáticamente a la confitería.**

Esto puede verse explícitamente en la axiomatización de

cantEmpanadas(TomarEmpanadas(...))

donde al tomar nuevas empanadas, “se pisa” cualquier cantidad de empanadas que podría haber tenido el vendedor, lo que intuitivamente se corresponde con que el vendedor siempre devuelve las empanadas que le sobraron antes de pedir nuevas.

fin :)