

Algoritmos y Estructuras de Datos 2.

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

TP 1: Lollapatuza.

Integrante	LU	Correo electrónico
Nicolás Pachón Pintos	301/20	pintosn1511@gmail.com
Kevin Alexander Perez Marzo	770/16	stixerks@gmail.com
Javier Mareque	112/22	javiermaxmareque@gmail.com
Pablo Villavicencio	799/07	pablovillavicencio87@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega	Santiago	Aprobado
Segunda entrega		

1. Resolución

TAD COMIDA es STRING ✓

TAD PUESTO DE COMIDA

usa dicc, nat, bool

exporta Observadores, Igualdad Observacional, Generadores, Otras operaciones

géneros PDC

observadores básicos

menu : PDC \rightarrow dicc(Comida, nat) ✓

stockProducto : PDC $p \times$ Comida $c \rightarrow$ nat ✓ {def?(c,menu(p))}

promocionesProducto : PDC $p \times$ Comida $c \rightarrow$ dicc(nat, nat) ✓ {def?(c,menu(p))}

igualdad observacional

$$(\forall a, b : \text{PDC}) \left(a =_{\text{obs}} b \iff \begin{pmatrix} \text{menu}(a) =_{\text{obs}} \text{menu}(b) \wedge_L \\ (\forall c : \text{Comida})(c \in \text{claves}(\text{menu}(a)) \Rightarrow_L (\text{stock-} \\ \text{Producto}(a,c) =_{\text{obs}} \\ \text{stockProducto}(b,c) \wedge \\ \text{promocionesProducto}(a,c) =_{\text{obs}} \\ \text{promocionesProducto}(b,c) \end{pmatrix} \right)$$

generadores

abrirPuesto : dicc(Comida, nat) \rightarrow PDC

agregarStock : PDC $p \times$ Comida $c \times$ nat \rightarrow PDC ✓

NuevaPromocion : PDC $p \times$ Comida $c \times$ nat $k \times$ nat $d \rightarrow$ PDC podría estar desde la apertura.
{ $c \in \text{claves}(\text{menu}(p)) \wedge_L \neg \text{def?}(k, \text{promocionProducto}(p,c)) \wedge d < 100$ }

venta : PDC $p \times$ Comida $c \times$ int $k \rightarrow$ PDC { $c \in \text{claves}(\text{menu}(p)) \wedge_L \text{hayStock}(p,c,k)$ }

otras operaciones

precioDeCompra : PDC $p \times$ Comida $c \times$ int $k \rightarrow$ nat ✓

{ $c \in \text{claves}(\text{menu}(p)) \wedge_L \text{hayStock}(p,c,k)$ }

hayStock : PDC \times Comida \times nat \rightarrow bool ✓

hayPromocion : PDC \times Comida \times nat \rightarrow bool ✓

axiomas

$\forall p: \text{PDC}, \forall c, c_g, c_o : \text{comida}, \forall k, d : \text{nat}, \forall m : \text{dicc}(\text{comida}, \text{nat})$

$\text{hayStock}(p, c, k) \equiv k \leq \text{stockProducto}(p, c)$ ✓

$\text{hayPromocion}(p, c, k) \equiv \text{if } k = 0 \text{ then false else}$
 $\quad \text{if } \text{def?}(k, \text{promocionesProducto}(p, c)) \text{ then true}$
 $\quad \text{else } \text{hayPromocion}(p, c, k - 1) \text{ fi}$ ✓

$\text{menu}(\text{abrirPuesto}(m)) \equiv m$ ✓

$\text{menu}(\text{agregarStock}(p, c, k)) \equiv \text{menu}(p)$ ✓

$\text{menu}(\text{NuevaPromocion}(p, c, k, d)) \equiv \text{menu}(p)$ ✓

$\text{menu}(\text{venta}(p, c, k)) \equiv \text{menu}(p)$ ✓

$\text{stockProducto}(\text{abrirPuesto}(m), c) \equiv 0$ conceptualizaron stock inicial cero... me parece bien.

$\text{stockProducto}(\text{agregarStock}(p, c_g, k), c_o) \equiv \text{if } c_g = c_o \text{ then}$
 $\quad k + \text{stockProducto}(p, c_g)$
 $\quad \text{else}$
 $\quad \text{stockProducto}(p, c_o)$ ✓

$\text{stockProducto}(\text{nuevaPromocion}(p, c_g, k, d), c_0) \equiv \text{stockProducto}(p, c_0)$

$\text{stockProducto}(\text{venta}(p, c_g, k), c_0) \equiv \text{if } c_g = c_0 \text{ then}$
 $\quad \text{stockProducto}(p, c_g) - k$
 $\quad \text{else}$
 $\quad \text{stockProducto}(p, c_0)$ ✓

$\text{promocionesProducto}(\text{abrirPuesto}(m), c) \equiv \text{vacio}$

$\text{promocionesProducto}(\text{agregarStock}(p, c_g, k), c_o) \equiv \text{promocionesProducto}(p, c_o)$

$\text{promocionesProducto}(\text{nuevaPromocion}(p, c_g, k, d), c_0) \equiv \text{if } c_g = c_0 \text{ then}$
 $\quad \text{definir}(k, d, \text{promocionesProducto}(p, c_g))$
 $\quad \text{else}$
 $\quad \text{promocionesProducto}(p, c_0)$ ✓

$\text{promocionesProducto}(\text{venta}(p, c_g, k), c_0) \equiv \text{promocionesProducto}(p, c_0)$

```

precioDeCompra(p,c,k)  $\equiv$  if hayPromocion(p,c,k) then
    aplicarDescuento(precioTotal(p,c,k),mayorDescuentoPosible(p,c,k))
else
    precioTotal(p,c,k)
fi

```

precioTotal : PDC \times Comida \times nat \longrightarrow nat

precioTotal(p,c,k) \equiv obtener(c,menu(p))*k

mayorDescuentoPosible : PDC \times Comida \times nat \longrightarrow nat

```

mayorDescuentoPosible(p,c,k)  $\equiv$  if def?(k,promocionesProducto(p,c)) then
    obtener(k,promocionesProducto(p,c))
else
    mayorDescuentoPosible(p,c,k - 1)
fi

```

div : nat \times nat $k \longrightarrow$ nat

{k>0}

div(n,k) \equiv **if** n < k **then** 0 **else** 1 + div(n - k,k) **fi**

aplicarDescuento : nat \times nat $d \longrightarrow$ nat

{d<100}

aplicarDescuento(p,d) \equiv div(p*(100-d),100)

Fin TAD

TAD PERSONA es STRING

TAD COMPRA es TUPLA(COMIDA,NAT,PDC)

TAD LOLLAPATUZA

usa multiconj, conj, nat, bool, PDC

exporta Observadores, Igualdad Observacional, Generadores, Otras operaciones

géneros LPZ

observadores básicos

puestos : LPZ \rightarrow conj(PDC) **multiconj. Podrías tener puestos con mismo menú y stock.**

personasHabilitadas : LPZ \rightarrow conj(Personas)

compras : LPZ $l \times$ Persona $n \rightarrow$ multiconj(Compra) $\{n \in \text{personasHabilitadas}(l)\}$

igualdad observacional

$$(\forall a, b : \text{LPZ}) \left(a =_{\text{obs}} b \iff \begin{pmatrix} (\text{puestos}(a) =_{\text{obs}} \text{puestos}(b) \wedge \\ \text{personasHabilitadas}(a) =_{\text{obs}} \\ \text{personasHabilitadas}(b)) \wedge_L \\ (\forall n : \text{Persona})(n \in \text{personasHabilitadas}(a) \wedge \Rightarrow_L \\ \text{compras}(a,n) =_{\text{obs}} \text{compras}(b,n)) \end{pmatrix} \right)$$

generadores

lollapatuza : conj(PDC) $ps \times$ conj(Personas) \rightarrow LPZ $\{\text{cumplePoliticassLollapatuza}(ps)\}$

realizaCompra : LPZ $l \times$ PDC $p \times$ Persona $n \times$ Comida $c \times$ nat $k \rightarrow$ LPZ $\left\{ \begin{array}{l} p \in \text{puestos}(l) \wedge \text{def?}(c, \text{menu}(p)) \wedge n \in \text{personasHabilitadas}(l) \wedge k > 0 \\ \wedge \text{hayStock}(p, c, k) \end{array} \right\}$

realizarHackeo : LPZ $l \times$ Comida $c \times$ Persona $n \rightarrow$ LPZ \downarrow

realizar hackeo podría ser otras operaciones.

$\{\text{existeCompraSinPromo}(l, n, c)\}$

otras operaciones

consumoTotal : LPZ $l \times$ Persona $n \rightarrow$ nat

$\{n \in \text{personasHabilitadas}(l)\}$

personaQueMasGasto : LPZ \rightarrow Persona

axiomas

$\forall ps : \text{conj(PDC)}, \forall ns : \text{conj(personas)}, \forall l : \text{LPZ}, \forall p : \text{PDC}, \forall n, n_g, n_o : \text{persona}, \forall c : \text{comida},$

$\forall k : \text{nat}$

$\text{puestos}(\text{lollapatuza}(ps, ns)) \equiv ps$

$\text{personasHabilitadas}(\text{lollapatuza}(ps, ns)) \equiv ns$

$\text{compras}(\text{lollapatuza}(ps, ns), n) \equiv \emptyset$

puestos(realizaCompra(l,p,n,c,k)) \equiv Ag(venta(p,c,k),puestos(l) - {p}) ✓

personasHabilitadas(realizaCompra(l,p,n,c,k)) \equiv personasHabilitadas(l) ✓

compras(realizaCompra(l,p,n_g,c,k),n_o) \equiv **if** n_g = n_o **then**

Ag(\langle c, k, p \rangle , compras(l,n_g)) ✓

else

Compras(l,n_o)

fi

puestos(realizarHackeo(l,c,n)) \equiv Ag(

agregarStock(dameUno(puestosComproSD(compras(l,n),c),c,1),

puestos(l) - {dameUno(puestosComproSD(compras(l,n),c))}) ✓

puestosComproSD : multiconj(Compra) \times Comida \longrightarrow Conj(PDC)

\forall cs : multiconj(Compra), \forall c: comida

puestosComproSD(cs,c) \equiv **if** vacio?(cs) **then**

\emptyset

else

if $\pi_1(\text{dameUno}(cs)) = c \wedge \neg \text{hayPromocion}(\pi_3(\text{dameUno}(cs)),$

$\pi_1(\text{dameUno}(cs)), \pi_2(\text{dameUno}(cs)))$ **then**

Ag($\pi_3(\text{dameUno}(cs)), \text{puestosComproSD}(\text{sinUno}(cs))$)

else puestosComproSD(sinUno(cs)) **fi** ✓

fi

personasHabilitadas(realizarHackeo(l,c,n)) \equiv personasHabilitadas(l) ✓

compras(realizarHackeo(l,c,n_h),n_o) \equiv **if** n_h = n_o **then**

reduceCantDeCompra(c,compras(l,n_h),

dameUno(puestosComproSD(compras(l,n_h),c)))

else

compras(l,n_o) ✓

fi

$\text{reduceCantDeCompra} : \text{Comida} \times \text{multiconj}(\text{Compra}) \times \text{PDC} \longrightarrow \text{multiconj}(\text{Compra})$
 $\forall c : \text{comida} , \forall cs : \text{multiconj}(\text{compra}) , \forall p : \text{PDC}$
 $\text{reduceCantDeCompra}(c,cs,p) \equiv \text{if } \pi_1(\text{dameUno}(cs)) = c \wedge \pi_3(\text{dameUno}(cs)) = p$
 $\quad \wedge \neg \text{hayPromocion}(p,c,\pi_2(\text{dameUno}(cs))) \text{ then}$
 $\quad \text{Ag}(\langle \pi_1(\text{dameUno}(cs)), \pi_2(\text{dameUno}(cs)) - 1,$
 $\quad \pi_3(\text{dameUno}(cs)) \rangle, \text{sinUno}(cs))$
 else
 $\quad \text{Ag}(\text{dameUno}(cs), \text{reduceCantDeCompra}(c,\text{sinUno}(cs)))$
 fi

Me parece que les falta el caso de las compras de un solo item. En ese caso deberían borrar la compra entera

$\text{consumoTotal}(l,n) \equiv \text{todosLosConsumos}(\text{compras}(l,n))$
 $\text{todosLosConsumos} : \text{multiconj}(\text{Compra}) \longrightarrow \text{nat}$
 $\forall cs : \text{multiconj}(\text{compra})$
 $\text{todosLosConsumos}(cs) \equiv \text{if vacio?}(cs) \text{ then}$
 $\quad \text{totalDePlataGastada} \quad 0$
 else
 $\quad \text{precioDeCompra}(\pi_3(\text{dameUno}(cs)), \pi_1(\text{dameUno}(cs)), \pi_2(\text{dameUno}(cs)))$
 $\quad +$
 $\quad \text{todosLosConsumos}(\text{sinUno}(cs))$
 fi

$\text{personaQueMasGasto}(l) \equiv \text{mayorGasto}(l, \text{sinUno}(\text{personasHabilitadas}(l)),$
 $\quad \text{dameUno}(\text{personasHabilitadas}(l)))$

$\text{mayorGasto} : \text{LPZ} \times \text{conj}(\text{Personas}) \times \text{Persona} \longrightarrow \text{Persona}$
 $\forall l : \text{LPZ} , \forall ns : \text{conj}(\text{personas}), \forall n : \text{persona}$
 $\text{mayorGasto}(l,ns,n) \equiv \text{if vacio?}(ns) \text{ then}$
 $\quad n$
 else
 $\quad \text{if consumoTotal}(l,n) \geq \text{consumoTotal}(l,\text{dameUno}(ns)) \text{ then}$
 $\quad \text{mayorGasto}(l,\text{sinUno}(ns),n) \text{ else}$
 $\quad \text{mayorGasto}(l,\text{sinUno}(ns),\text{dameUno}(ns)) \text{ fi}$
 fi

Fin TAD

2. Anexo: Predicados



$$\begin{aligned} \text{cumplePoliticasyLollapatuza}(\text{ps}) &\equiv (\forall a, b: \text{PDC})(\forall c: \text{Comida})(a \in \text{ps} \wedge b \in \text{ps} \wedge \text{def?}(c, \text{menu}(a)) \\ &\quad \wedge \text{def?}(c, \text{menu}(b)) \Rightarrow_{\text{L}} \text{obtener}(c, \text{menu}(a)) = \text{obtener}(c, \text{menu}(b))) \\ \text{existeCompraSinPromo}(l, n, c) &\equiv (\exists a : \text{Compra})((a \in \text{compras}(l, n) \wedge \pi_2(a) = c) \wedge_{\text{L}} \\ &\quad \neg \text{hayPromocion}(\pi_3(a), \pi_1(a), \pi_2(a))) \end{aligned}$$