

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

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

Lollapatuza

Especificación

Integrante	LU	Correo electrónico
Mamani, Carlos	496/16	mamanimezacarlos@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega	<i>Fermín</i>	<i>A</i>

1. Aclaraciones

En la axiomatización se utiliza puestosDCSD para hacer referencia a puestosDondeConsumioSinDescuento.
Además se utiliza la función aplicarDescuento provista en el enunciado.

2. TADs

TADs PERSONA, ITEM son STRING

TAD PRECIO es NAT

TAD PEDIDO es $\text{dicc}(\text{item}, \text{cantidad} : \text{NAT})$ donde $\text{cantidad} > 0$ *esto lo tenés que definir como restricción donde lo usas*

TAD LOLLAPATUZA

usa Persona, Puesto, Nat

exporta observadores

géneros lola

igualdad observacional

$$(\forall l, l' : \text{lola}) \left(l =_{\text{obs}} l' \iff \left(\begin{array}{l} \text{personas}(l) =_{\text{obs}} \text{personas}(l') \wedge \text{puestos}(l) =_{\text{obs}} \text{puestos}(l') \wedge \\ (\forall s : \text{persona})(s \in \text{personas}(l) \Rightarrow_{\text{L}} \\ \# \text{plataGastada}(l, s) =_{\text{obs}} \# \text{plataGastada}(l', s) \wedge \\ (\forall p : \text{puesto})(\forall e : \text{item})(p \in \text{puestos}(l) \Rightarrow_{\text{L}} \\ \# \text{itemsConsumidosSinDescuento}(l, p, s, e) \\ =_{\text{obs}} \# \text{itemsConsumidosSinDescuento}(l', p, s, e))) \end{array} \right) \right)$$

observadores básicos

$\text{personas} : \text{lola } l \rightarrow \text{conj}(\text{persona})$

$\text{puestos} : \text{lola } l \rightarrow \text{conj}(\text{puesto})$

$\# \text{plataGastada} : \text{lola } l \times \text{persona } p \rightarrow \text{nat} \quad \{p \in \text{personas}(l)\}$

$\# \text{itemsConsumidosSinDescuento} : \text{lola } l \times \text{puesto } p \times \text{persona } s \times \text{item } e \rightarrow \text{nat} \quad \{p \in \text{puestos}(l) \wedge s \in \text{personas}(l)\}$

generadores

$\text{crearLola} : \text{conj}(\text{puesto}) c \times \text{conj}(\text{persona}) p \rightarrow \text{lola}$
 $\left\{ \begin{array}{l} \neg \emptyset?(c) \wedge \neg \emptyset?(p) \wedge \\ (\forall a, b : \text{puesto})(\forall e : \text{item})((a, b \in c \wedge \text{estaEnMenu?}(a, e) \wedge \text{estaEnMenu?}(b, e)) \Rightarrow_{\text{L}} \text{precio}(a, e) = \text{precio}(b, e)) \end{array} \right\}$

$\text{comprar} : \text{lola } l \times \text{puesto } p \times \text{persona } s \times \text{pedido } c \rightarrow \text{lola}$
 $\left\{ \begin{array}{l} \neg \emptyset?(\text{claves}(c)) \wedge p \in \text{puestos}(l) \wedge s \in \text{personas}(l) \wedge \\ \text{claves}(c) \subseteq \text{claves}(\text{menu}(p)) \wedge \\ (\forall e : \text{item})(e \in \text{claves}(c) \Rightarrow_{\text{L}} 0 < \text{obtener}(e, c) \leq \text{stock}(p, e)) \end{array} \right\}$

$\text{hackear} : \text{lola } l \times \text{persona } p \times \text{item } e \rightarrow \text{lola}$
 $\{p \in \text{personas}(l) \wedge_{\text{L}} (\exists t : \text{puesto})(t \in \text{puestos}(l) \wedge_{\text{L}} \# \text{itemsConsumidosSinDescuento}(l, t, p, e) > 0)\}$

otras operaciones

$\text{puestosDondeConsumioSinDescuento} : \text{lola } l \times \text{conj}(\text{puesto}) c \times \text{persona } p \times \text{item } e \rightarrow \text{conj}(\text{puesto})$
 $\left\{ \begin{array}{l} p \in \text{personas}(l) \wedge (\exists t : \text{puesto})(t \in \text{puestos}(l) \wedge_{\text{L}} \text{estaEnMenu?}(t, e)) \wedge \\ c \subseteq \text{puestos}(l) \end{array} \right\}$

$\text{masGastador} : \text{lola } l \rightarrow \text{persona} \quad \{\}$

$\text{masGastadores} : \text{lola } l \times \text{conj}(\text{persona}) c \rightarrow \text{conj}(\text{persona}) \quad \{c \subseteq \text{personas}(l)\}$

$\text{maximaPlataGastada} : \text{lola } l \rightarrow \text{nat} \quad \{\}$

$\text{maximaPlataGastadaAux} : \text{lola } l \times \text{conj}(\text{persona}) \ c \longrightarrow \text{nat} \quad \{c \subseteq \text{personas}(l)\}$

axiomas

$\text{personas}(\text{crearLola}(c, p)) \equiv p$

$\text{personas}(\text{comprar}(l, p, s, c)) \equiv \text{personas}(l)$

$\text{personas}(\text{hackear}(l, p, e)) \equiv \text{personas}(l)$

$\text{puestos}(\text{crearLola}(c, p)) \equiv c$

$\text{puestos}(\text{comprar}(l, p, s, c)) \equiv (\text{puestos}(l) - \{p\}) \cup \{\text{venderPedido}(p, c)\}$ ✓

$\text{puestos}(\text{hackear}(l, p, e)) \equiv (\text{puestos}(l) - \{\text{dameUno}(\text{puestosDCSD}(l, \text{puestos}(l), p, e))\}) \cup \{\text{agregarStock}(\text{dameUno}(\text{puestosDCSD}(l, \text{puestos}(l), p, e)), e, 1)\}$ ✓

$\# \text{plataGastada}(\text{crearLola}(a, b), p) \equiv 0$

$\# \text{plataGastada}(\text{comprar}(l, p, s, c), a) \equiv \text{if } a \neq s \text{ then } \# \text{plataGastada}(l, a) \text{ else } \# \text{plataGastada}(l, a) + \text{costo}(p, c) \text{ fi}$ ✓

$\# \text{plataGastada}(\text{hackear}(l, p, e), p') \equiv \text{if } p' == p \text{ then } \# \text{plataGastada}(l, p') - \text{precio}(\text{dameUno}(\text{puestosDCSD}(l, \text{puestos}(l), p', e))), e) \text{ else } \# \text{plataGastada}(l, p') \text{ fi}$ ✓

$\# \text{itemsConsumidosSinDescuentos}(\text{crearLola}(a, b), p, s, e) \equiv 0$

$\# \text{itemsConsumidosSinDescuentos}(\text{comprar}(l, p, s, c), p', s', e) \equiv \# \text{itemsConsumidosSinDescuentos}(l, p', s', e) + \text{if } (p' == p \wedge s' == s \wedge e \in \text{claves}(c) \wedge \text{descuentoAAplicar}(p', e, \text{obtener}(e, c)) == 0) \text{ then } \text{obtener}(e, c) \text{ else } 0 \text{ fi}$ ✓

$\# \text{itemsConsumidosSinDescuentos}(\text{hackear}(l, p, e), t, a, e') \equiv \text{if } a == p \wedge e' == e \text{ then } \text{if } t == \text{dameUno}(\text{puestosDCSD}(l, \text{puestos}(l), a, e')) \text{ then } \# \text{itemsConsumidosSinDescuentos}(l, t, a, e') - 1 \text{ else } \# \text{itemsConsumidosSinDescuentos}(l, t, a, e') \text{ fi}$ ✓

$\text{puestosDCSD}(l, c, p, e) \equiv \text{if } \emptyset?(c) \text{ then } \emptyset \text{ else } \text{if } \# \text{itemsConsumidosSinDescuentos}(l, \text{dameUno}(c), p, e) > 0 \text{ then } \text{Ag}(\text{dameUno}(c), \text{puestosDCSD}(l, \text{sinUno}(c), p, e)) \text{ else } \text{puestosDCSD}(l, \text{sinUno}(c), p, e) \text{ fi}$ ✓

$\text{masGastador}(l) \equiv \text{dameUno}(\text{masGastadores}(l, \text{personas}(l)))$

$\text{masGastadores}(l, c) \equiv \text{if } \emptyset?(c) \text{ then } \emptyset \text{ else } \text{if } \# \text{plataGastada}(l, \text{dameUno}(c)) == \text{maximaPlataGastada}(l) \text{ then } \text{Ag}(\text{dameUno}(c), \text{masGastadores}(l, \text{sinUno}(c))) \text{ else } \text{masGastadores}(l, \text{sinUno}(c)) \text{ fi}$ ✓

$\text{maximaPlataGastada}(l) \equiv \text{maximaPlataGastadaAux}(l, \text{personas}(l))$

```

maximaPlataGastadaAux(l, c) ≡ if ∅?(c) then 0
                             else
                               max(#plataGastada(l, dameUno(c)),
                                   maximaPlataGastadaAux(l, sinUno(c)) )
                             fi

```

Fin TAD**TAD PUESTO****usa** Item, Combo, Precio, Nat**exporta** observadores, generadores, estaEnMenu?, precio, costo, descuentoAAplicar, venderPedido**géneros** puesto**igualdad observacional**

$$(\forall p, p' : \text{puesto}) \left(p =_{\text{obs}} p' \iff \left(\begin{array}{l} \text{menu}(p) =_{\text{obs}} \text{menu}(p') \wedge \\ \text{descuentos}(p) =_{\text{obs}} \text{descuentos}(p') \wedge \\ (\forall e : \text{item})(\text{estaEnMenu?}(p, e) \Rightarrow_{\text{L}} \text{stock}(p, e) = \text{stock}(p', e)) \end{array} \right) \right)$$

observadores básicosmenu : puesto $p \rightarrow \text{dicc}(\text{item}, \text{precio})$ { }descuentos : puesto $p \rightarrow \text{dicc}(\text{item}, \text{conj}(\text{combo}))$ { }stock : puesto $p \times \text{item } e \rightarrow \text{nat}$ {estaEnMenu?(p, e)} ✓**generadores**

nuevoPuesto : $\text{dicc}(\text{item}, \text{precio}) \text{ menu} \times \text{dicc}(\text{item}, \text{conj}(\text{combo})) \text{ descuentos} \rightarrow \text{puesto}$

$$\left\{ \begin{array}{l} \neg \emptyset?(\text{claves}(\text{menu})) \wedge \text{claves}(\text{descuentos}) \subseteq \text{claves}(\text{menu}) \wedge_{\text{L}} \\ (\forall e : \text{item})(\text{estaEnMenu?}(p, e) \Rightarrow_{\text{L}} \text{obtener}(e, \text{menu}) > 0) \wedge_{\text{L}} \\ (\forall e : \text{item})(e \in \text{claves}(\text{descuentos}) \Rightarrow_{\text{L}} \\ \neg \emptyset?(\text{obtener}(e, \text{descuentos})) \wedge_{\text{L}} \\ (\forall a, b : \text{combo})(a, b \in \text{obtener}(e, \text{descuentos}) \Rightarrow_{\text{L}} \text{cantidad}(a) \neq \text{cantidad}(b)) \end{array} \right\}$$
agregarStock : puesto $p \times \text{item } e \times \text{nat } cant \rightarrow \text{puesto}$ {estaEnMenu?(p, e) ∧ cant > 0 }vender : puesto $p \times \text{item } e \times \text{nat } c \rightarrow \text{puesto}$ {estaEnMenu?(p, e) ∧ 0 < c ≤ stock(p, e) } ✓**otras operaciones**estaEnMenu? : puesto $\times \text{item} \rightarrow \text{bool}$ precio : puesto $p \times \text{item } e \rightarrow \text{nat}$ {estaEnMenu?(p, e)}precioCantidad : puesto $p \times \text{item } e \times \text{nat } cant \rightarrow \text{nat}$ {estaEnMenu?(p, e) ∧ cant > 0 }costo : puesto $p \times \text{pedido } c \rightarrow \text{nat}$ {claves(c) ⊆ claves(menu(p))} ✓descuentoAAplicar : puesto $p \times \text{item } e \times \text{nat } cant \rightarrow \text{nat}$ {estaEnMenu?(p, e) ∧ cant > 0 }descuentoVigente : $\text{conj}(\text{combo}) c \times \text{nat } desc \times \text{nat } low \times \text{nat } high \rightarrow \text{nat}$ {low ≤ high ∧ desc < 100}

venderPedido : puesto $p \times \text{pedido } c \rightarrow \text{puesto}$

$$\left\{ \begin{array}{l} \text{claves}(c) \subseteq \text{claves}(\text{menu}(p)) \wedge \\ (\forall e : \text{item})(e \in \text{claves}(c) \Rightarrow_{\text{L}} 0 < \text{obtener}(e, c) \leq \text{stock}(p, e)) \end{array} \right\}$$
axiomas

menu(nuevoPuesto(m, d)) ≡ m

menu(agregarStock(p, e, c)) ≡ menu(p)

menu(vender(p, e, c)) ≡ menu(p)

descuentos(nuevoPuesto(m, d)) ≡ d

descuentos(agregarStock(p, e, c)) ≡ descuentos(p)

```

descuentos(vender(p, e, c)) ≡ descuentos(p)
stock(nuevoPuesto(m, d), e) ≡ 0
stock(agregarStock(p, e, c), e') ≡ stock(p, e') + if e' == e then c else 0 fi ✓
stock(vender(p, e, c), e') ≡ if e' == e then stock(p, e') - c else stock(p, e') fi ✓
estaEnMenu?(p, e) ≡ e ∈ claves(menu(p))
precio(p, e) ≡ obtener(e, menu(p))
precioCantidad(p, e, cant) ≡ aplicarDescuento(precio(p, e) × cant,
                                             descuentoAAplicar(p, e, cant))
costo(p, pedido) ≡ if ∅?(claves(pedido)) then
    0
else
    precioCantidad(p, dameUno(claves(pedido)),
                  obtener(dameUno(claves(pedido)), pedido) )
    + costo(p, borrar(dameUno(claves(pedido)), pedido) )
fi
descuentoAAplicar(puesto, item, cant) ≡ if item ∉ claves(descuentos(puesto)) then
    0
else ✓
    descuentoVigente(obtener(item, descuentos(puesto)), 0, 0, cant)
fi
descuentoVigente(c, desc, low, high) ≡ if ∅?(c)
then
    desc
else
    if low < cantidad(dameUno(c)) ≤ high ✓
    then descuentoVigente(sinUno(c), descuento(dameUno(c)),
                          cantidad(dameUno(c)), high)
    else descuentoVigente(sinUno(c), desc, low, high) fi
fi
venderPedido(p, pedido) ≡ if ∅?(claves(pedido)) then p ✓
else
    vender( venderPedido(p, borrar(dameUno(claves(pedido)), pedido) ),
           dameUno(claves(pedido)),
           obtener(dameUno(claves(pedido)), pedido) )
fi

```

Fin TAD

TAD COMBO

usa Nat

exporta observadores

géneros combo

observadores básicos

cantidad : combo → nat

descuento : combo → nat

generadores

nuevoCombo : nat *cant* × nat *d* → combo

{0 < *cant* ∧ 0 < *d* < 100 }

axiomas

cantidad(nuevoCombo(*c*, *d*)) ≡ *c*

$\text{descuento}(\text{nuevoCombo}(c, d)) \equiv d$

Fin TAD