

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

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

Lollapatuza

TADgrupo

Integrante	LU	Correo electrónico
Aquilante, Juan Pablo	755/18	aquilantejp@outlook.es
Bocanegra, Oscar Iván	537/22	oscarbocanegraush@gmail.com
Flores Galvan, Silvina	689/22	silu204@gmail.com
Sanchez, Gastón	361/22	gasanchez@dc.uba.ar

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

TAD PUESTO DE COMIDA**igualdad observacional**

$$(\forall p1, p2 : \text{puesto}) \left(p1 =_{\text{obs}} p2 \iff \left(\begin{array}{l} \text{menuPuesto}(p1) =_{\text{obs}} \text{menuPuesto}(p2) \wedge \\ \text{stockPuesto}(p1) =_{\text{obs}} \text{stockPuesto}(p2) \wedge \\ \text{promos}(p1) =_{\text{obs}} \text{promos}(p2) \wedge \\ \text{ID}(p1) =_{\text{obs}} \text{ID}(p2) \end{array} \right) \right)$$

géneros puesto**exporta** puesto, generadores, observadores, precioItem, promosItemPuesto, promosItem, mejorDescuento**usa** NAT, PRECIO, ITEM, CANTIDAD, DESCUENTO, MENU, STOCK, CONJUNTO(PROMO), PROMO**observadores básicos**

ID	: puesto	→ nat
menuPuesto	: puesto	→ menú
stockPuesto	: puesto	→ stock
promos	: puesto	→ conj(promo)

generadores

crearPuesto	: nat × menu m × stock s × conj(promo) prm	→ puesto	$\left\{ \begin{array}{l} \neg \text{vacío?}(\text{claves}(m)) \wedge_{\text{L}} \text{claves}(m) = \text{claves}(s) \\ \wedge_{\text{L}} (\forall pr: \text{promo}) (\text{pr} \in \text{prm} \Rightarrow_{\text{L}} \pi_1(pr) \in \text{claves}(m) \wedge 0 < \pi_3(pr) \leq 100) \end{array} \right\}$
reponerItem	: puesto p × item i	→ puesto	{def?($i, \text{stockPuesto}(p)$)}
venderItems	: puesto p × item i × cantidad c	→ puesto	{def?($i, \text{menuPuesto}(p)$) $\wedge_{\text{L}} c \leq \text{obtener}(i, \text{stockPuesto}(p))$ }

otras operaciones

precioItem	: puesto p × item i	→ nat	{def?($i, \text{menuPuesto}(p)$)}
promosItemPuesto	: puesto × item	→ conj(promo)	
promosItem	: conj(promo) × item	→ conj(promo)	
mejorDescuento	: conj(promo) × cantidad	→ nat	

axiomas $\forall p: \text{puesto}, \forall i: \text{item}, \forall n: \text{nat}, \forall prms: \text{conj}(\text{promo}), \forall c: \text{cantidad}, \forall m: \text{menu}, \forall s: \text{stock}$

ID(crearPuesto(n, m, s, prm))	$\equiv n$
ID(reponerItem(p, i))	$\equiv \text{ID}(p)$
ID(venderItems(p, i, c))	$\equiv \text{ID}(p)$
menuPuesto(crearPuesto($m, s, prms$))	$\equiv m$
menuPuesto(reponerItem(p, i))	$\equiv \text{menuPuesto}(p)$
menuPuesto(venderItems(p, i, c))	$\equiv \text{menuPuesto}(p)$
stockPuesto(crearPuesto($m, s, prms$))	$\equiv s$
stockPuesto(reponerItem(p, i))	$\equiv \text{definir}(i, \text{obtener}(i, \text{stock}(p)) + 1, \text{stock}(p))$
stockPuesto(venderItems(p, i, c))	$\equiv \text{definir}(i, \text{obtener}(i, \text{stock}(p)) - c, \text{stock}(p))$
promos(crearPuesto($m, s, prms$))	$\equiv prms$
promos(reponerItem(p, i))	$\equiv \text{promos}(p)$
promos(venderItems(p, i, c))	$\equiv \text{promos}(p)$
precioItem(p, i)	$\equiv \text{obtener}(i, \text{menuPuesto}(p))$
promosItemPuesto(p, i)	$\equiv \text{promosItem}(\text{promos}(p), i)$

```

promosItem(prms,i)                                     ≡ if vacía?(prms) then
  ∅
else
  if ( $\pi_1(\text{dameUno}(\text{prms}))=i$ ) then
    Ag(dameUno(prms),promosItem(sinUno(prms),i))
  else
    promosItem(sinUno(prms),i)
  fi
fi

mejorDescuento(prms,c)                                ≡ if vacía?(prms) then
  0
else
  if  $\pi_2(\text{dameUno}(\text{prms})) > c$  then
    mejorDescuento(sinUno(prms),c)
  else
    max( $\pi_3(\text{dameUno}(\text{prms}))$ , mejorDescuento(sinUno(prms),c))
  fi
fi

```

Fin TAD**TAD** PRECIO es NAT**TAD** ITEM es STRING**TAD** CANTIDAD es NAT**TAD** DESCUENTO es NAT**TAD** MENU es DICC(ITEM,PRECIO)**TAD** STOCK es DICC(ITEM,CANTIDAD)**TAD** PROMO es TUPLA(ÍTEM,CANTIDAD,DESCUENTO)

TAD LOLLAPATUZA**igualdad observacional**

$$(\forall l_1, l_2 : \text{lollapatuza}) \left(l_1 =_{\text{obs}} l_2 \iff \left(\text{personas}(l_1) =_{\text{obs}} \text{personas}(l_2) \wedge \text{puestos}(l_1) =_{\text{obs}} \text{puestos}(l_2) \wedge_{\text{L}} \right) \right)$$

$$\left((\forall p: \text{puesto}) (p \in \text{puestos}(l_1) \Rightarrow_{\text{L}} \text{ventas}(l_1, p) =_{\text{obs}} \text{ventas}(l_2, p)) \right)$$

géneros lollapatuza**exporta** lollapatuza, generadores, observadores, quienGastoMas?**usa** PUESTO, PERSONA, VENTA, CONJUNTO(PUESTO), CONJUNTO(PERSONA), MULTICONJUNTO(VENTA), ITEM, CANTIDAD, BOOL, NAT**observadores básicos**

personas	: lollapatuza	→ conj(persona)
puestos	: lollapatuza	→ conj(puesto)
ventas	: lollapatuza $l \times$ puesto p	→ multiconj(venta) $\{p \in \text{puestos}(l)\}$

generadores

nuevoLolla	: conj(puesto) \times conj(persona)	→ lollapatuza
vender	: lollapatuza $l \times$ puesto $p \times$ venta v $\{p \in \text{puestos}(l) \wedge \pi_1(v) \in \text{personas}(l) \wedge \text{def}?(\pi_2(v), \text{menuPuesto}(p)) \wedge \pi_3(v) \leq \text{obtener}(\pi_2(v), \text{stockPuesto}(p)) \}$	→ lollapatuza
hackearLolla	: lollapatuza $l \times$ persona $per \times$ item i $\{per \in \text{personas}(l) \wedge_{\text{L}} (\exists p: \text{puesto}) (p \in \text{puestos}(l) \wedge_{\text{L}} \text{hayVentaSinDescuento}(l, per, i, p, \text{ventas}(l, p))) \}$	→ lollapatuza

otras operaciones

ventasPostHackeo	: lollapatuza $l \times$ persona $per \times$ item $i \times$ multiconj(venta) mcv \times puesto p	→ multiconj(venta) $\{per \in \text{personas}(l) \wedge p \in \text{puestos}(l) \wedge_{\text{L}} mcv \subseteq \text{ventas}(l, p) \}$
damePuestoHackeable	: lollapatuza $l \times$ persona $per \times$ item $i \times$ conj(puesto) cp $\{cp \subseteq \text{puestos}(l) \wedge_{\text{L}} (\exists p: \text{puesto}) (p \in cp \wedge_{\text{L}} \text{hayVentaSinDescuento}?(l, per, i, p, \text{ventas}(l, p))) \}$	→ puesto
hayVentaSinDescuento?	: lollapatuza $l \times$ persona $per \times$ item $i \times$ puesto $p \times$ multi- conj(venta) mcv	→ bool $\{mcv \subseteq \text{ventas}(l, p) \}$
hayDescuento?	: venta $v \times$ conj(promo) $prms$	→ bool $\{(\forall pr: \text{promo}) (pr \in prms \Rightarrow_{\text{L}} \pi_1(pr) = \pi_2(v)) \}$
minCantPromos	: conj(promo) $prms$	→ nat $\{\neg \text{vacio}?(prms) \}$
quienGastoMas?	: lollapatuza $l \times$ conj(persona) $cper$	→ persona $\{cper \subseteq \text{personas}(l) \wedge \neg \emptyset?(cper) \}$
gastoMasGrande	: lollapatuza $l \times$ conj(persona) $cper$	→ nat $\{cper \subseteq \text{personas}(l) \wedge \neg \emptyset?(cper) \}$
gastoPersona	: lollapatuza $l \times$ persona per	→ nat
gastoPersonaPuestos	: lollapatuza $l \times$ conj(puesto) $cp \times$ persona per	→ nat $\{cp \subseteq \text{puestos}(l) \}$
gastoPersonaEnPuesto	: puesto $p \times$ multiconj(venta) $mcv \times$ persona	→ nat $\{(\forall v: \text{venta}) (v \in mcv \Rightarrow_{\text{L}} \pi_2(v) \in \text{claves}(\text{menuPuesto}(p))) \}$
gastoVenta	: venta $v \times$ puesto p	→ nat $\{\pi_2(v) \in \text{claves}(\text{menuPuesto}(p)) \}$
aplicarDescuento	: precio \times descuento d	→ nat $\{d < 100 \}$
div	: nat $n \times$ nat k	→ nat $\{0 < k \}$

axiomas	$\forall per: persona, \forall p, p': puesto, \forall i: item, \forall c: cantidad, \forall cper: conj(persona), \forall cp: conj(puesto), \forall mcv: multi-conj(venta), \forall v: venta, \forall pre: precio$
personas(nuevoLolla($cp, cper$))	$\equiv cper$
personas(vender(l, p, v))	$\equiv personas(l)$
personas(hackearLolla(l, per, i))	$\equiv personas(l)$
puestos(nuevoLolla($cp, cper$))	$\equiv cp$
puestos(vender(l, p, v))	$\equiv Ag(venderItems(p, \pi_2(v), \pi_3(v)), puestos(l) - \{p\})$
puestos(hackearLolla(l, per, i))	$\equiv Ag(reponerItem(damePuestoHackeable(l, puestos(l), per, i), i), puestos(l) - \{damePuestoHackeable(l, per, i)\})$
ventas(nuevoLolla($cp, cper$), p)	$\equiv \emptyset$
ventas(vender(l, p', v), p)	$\equiv \text{if } p = p' \text{ then } Ag(v, ventas(l, p)) \text{ else } ventas(l, p) \text{ fi}$
ventas(hackearLolla(l, per, i), p)	$\equiv \text{if } damePuestoHackeable(l, per, i) = p \text{ then } ventasPostHackeo(l, per, i, ventas(l, p), p) \text{ else } ventas(l, p) \text{ fi}$
ventasPostHackeo(l, per, i, mcv, p)	$\equiv \text{if } \pi_1(dameUno(mcv)) = per \wedge \pi_2(dameUno(mcv)) = i \wedge (promosItem(p, i) = \emptyset \vee_L \pi_3(dameUno(mcv)) < \minCantPromos(promosItem(p, i))) \text{ then } Ag(\langle per, i, \pi_3(dameUno(mcv)) - 1 \rangle, ventas(l, p) - dameUno(mcv)) \text{ else } ventasPostHackeo(l, per, i, sinUno(mcv), p) \text{ fi}$
damePuestoHackeable(l, per, cp, i)	$\equiv \text{if } hayVentaSinDescuento?(l, per, i, dameUno(cp), ventas(l, dameUno(cp))) \text{ then } dameUno(cp) \text{ else } damePuestoHackeable(l, per, sinUno(cp), i) \text{ fi}$
hayVentaSinDescuento?(l, per, i, p, mcv)	$\equiv \text{if } \emptyset?(mcv) \text{ then } false \text{ else } \text{if } hayDescuento?(dameUno(mcv), promos(p)) \wedge \pi_1(dameUno(mcv)) = per \wedge \pi_2(dameUno(mcv)) = i \text{ then } true \text{ else } hayVentaSinDescuento?(l, per, i, p, sinUno(mcv)) \text{ fi}$
hayDescuento?($v, prms$)	$\equiv \text{if } vacio?(prms) \text{ then } false \text{ else } \text{if } \pi_3(v) \geq \pi_2(dameUno(prms)) \text{ then } true \text{ else } hayDescuento?(v, sinUno(prms)) \text{ fi}$
minCantPromos($prms$)	$\equiv \text{if } \#(prms) = 1 \text{ then } \pi_3(dameUno(prms)) \text{ else } \min\{\pi_3(dameUno(prms)), \minCantPromos(sinUno(prms))\} \text{ fi}$

```

quienGastoMas?(l,cper)      ≡ if    gastoPersona(l,dameUno(cper))=gastoMasGrande(l,personas(l))
                             then
                             dameUno(cper)
                             else
                             quienGastoMas?(l,sinUno(cper))
                             fi

gastoMasGrande(l,cper)      ≡ if #(cper)=1 then
                             gastoPersona(l,dameUno(mcp))
                             else
                             max{gastoPersona(l,dameUno(mcp)),
                                gastoMasGrande(l,sinUno(mcp))}
                             fi

gastoPersona(l,per)         ≡ gastoPersonaPuestos(l,puestos(l),per)

gastoPersonaPuestos(l,cp,per) ≡ if ∅?(cp) then
                             0
                             else
                             gastoPersonaEnPuesto(dameUno(cp),ventas(l,dameUno(cp)),per)
                             + gastoPersonaPuestos(l,sinUno(cp),per)
                             fi

gastoPersonaEnPuesto(p,mcv,per) ≡ if vacio?(mcv) then
                             0
                             else
                             if  $\pi_1(mcv)=per$  then
                             gastoVenta(dameUno(mcv),p)
                             +gastoPersonaEnPuesto(p,sinUno(mcv),per)
                             else
                             gastoPersonaEnPuesto(p,sinUno(mcv),per)
                             fi
                             fi

gastoVenta(v,p)            ≡ if hayDescuento?(v,promosItem(p, $\pi_2(v)$ )) then
                             aplicarDescuento(precioItem(p, $\pi_2(v)$ ),
                             mejorDescuento(promosItemPuesto(p, $\pi_2(v)$ ), $\pi_3(v)$ ))  $\times \pi_3(v)$ 
                             else
                             precioItem(p,  $\pi_2(v)$ )  $\times \pi_3(v)$ 
                             fi

aplicarDescuento(pre,d)    ≡ div(pre  $\times$  (100 - d), 100)

div(n,k)                   ≡ if n < k then 0 else 1+div(n-k,k) fi

```

Fin TAD**TAD PERSONA** es NAT**TAD VENTA** es TUPLA(PERSONA,ITEM,CANTIDAD)