

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

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

Especificación

Lollapatuza

Integrante	LU	Correo electrónico
Pacio, Francisco	37/22	f@pacio.ar
Tabarez, Diego Rafael	131/22	diegortabarez@gmail.com
Peré Schröder, Máximo Adriel	446/13	maximo.pere@bue.edu.ar
Hajek Pulido, David Alejandro	52/21	hajekuba96@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. TAD LOLLAPATUZA

TAD Lollapatuza

géneros Lolla

usa PuestoDeComida, Persona, Item, Precio, Cantidad, Compra, Conjunto

igualdad observacional

$$(\forall n, m : \text{Lolla}) \left(n =_{\text{obs}} m \iff \left(\begin{array}{l} \text{verPuestos}(n) =_{\text{obs}} \text{verPuestos}(m) \wedge \text{verPersonas}(n) =_{\text{obs}} \\ \text{verPersonas}(m) \wedge \text{verCompras}(n) =_{\text{obs}} \text{verCompras}(m) \end{array} \right) \right)$$

generadores

inaugurar : \longrightarrow Lolla

habilitarPersona : $\text{Lolla } l \times \text{Persona } p \longrightarrow \text{Lolla}$

habilitarPuestoDeComida : $\text{Lolla } l \times \text{Puesto } p \longrightarrow \text{Lolla}$ {estaEnRegla?(l,p)}

realizarCompra : $\text{Lolla } l \times \text{Persona } pe \times \text{Item } i \times \text{Puesto } pu \times \text{Cantidad } c \longrightarrow \text{Lolla}$
{pe \in verPersonas(l) \wedge pu \in verPuestos(l) \wedge i \in verMenu(pu) \wedge_L c \leq verStockItem(pu, i)}

hackear : $\text{Lolla } l \times \text{Persona } p \times \text{Item } i \longrightarrow \text{Lolla}$
{p \in verPersonas(l) \wedge ($\exists C \in \text{verCompras}(l)$ tq p = C_{persona} \wedge i = C_{item} \wedge $C_{\text{descuento}} = 0$)}

observadores básicos

verPuestos : $\text{Lolla} \longrightarrow \text{MULTI}(\text{CONJ}(\text{PUESTO}))$

verPersonas : $\text{Lolla} \longrightarrow \text{CONJ}(\text{PERSONA})$

verCompras : $\text{Lolla} \longrightarrow \text{MULTI}(\text{CONJ}(\text{COMPRA}))$

otras operaciones

comprasPersona : $\text{Lolla } l \times \text{Persona } p \longrightarrow \text{MULTI}(\text{CONJ}(\text{COMPRA}))$ {p \in verPersonas(l)}

verMaximoGastador : $\text{Lolla} \longrightarrow \text{Persona}$

obtenerMaximo : $\text{Lolla } l \times \text{Persona } p \times \text{Conj}(\text{Persona}) ps \longrightarrow \text{Persona}$
{p \in verPersonas(l) \wedge ps \subset verPersonas(l)}

gastoTotalPersona : $\text{Lolla } l \times \text{Persona } p \longrightarrow \text{Precio}$ {p \in verPersonas(l)}

sumarGastos : $\text{Multi}(\text{Conj}(\text{Compra})) \longrightarrow \text{Precio}$

estaEnRegla? : $\text{Lolla} \times \text{Puesto} \longrightarrow \text{Bool}$

menuValido? : $\text{Menu} \times \text{Menu} \longrightarrow \text{Bool}$

agruparGastos : $\text{Persona} \times \text{Multi}(\text{Conj}(\text{Compra})) \longrightarrow \text{Multi}(\text{Conj}(\text{Compra}))$

verItemsHabilitados : $\text{Lolla} \longrightarrow \text{Menu}$

unirMenus : $\text{Conj}(\text{Puesto}) \longrightarrow \text{Menu}$

compraHackeada : $\text{Conj}(\text{Compras}) cs \times \text{Persona } pe \times \text{Item } i \longrightarrow \text{Compra}$
{ $\exists C \in cs$ tq p = C_{persona} \wedge i = C_{item} \wedge $C_{\text{descuento}} = 0$ }

axiomas

verPuestos(inaugurar) $\equiv \emptyset$

verPuestos(habilitarPersona(l, p)) $\equiv \text{verPuestos}(l)$

verPuestos(realizarCompra(l, pe, i, pu, c)) $\equiv \text{Ag}(\text{descontarStock}(pu, i, c), \text{verPuestos}(l) - \{pu\})$

verPuestos(hackear(l, pe, i)) $\equiv \text{Ag}(\text{reponerStock}(C_{\text{puesto}}, i, c), \text{verPuestos}(l) - \{C_{\text{puesto}}\})$ donde C = compraHackeada(verCompras(l), pe, i)

verPuestos(habilitarPuestoDeComida(l, p)) $\equiv \text{Ag}(p, \text{verPuestos}(l))$

verPersonas(inaugurar) $\equiv \emptyset$

```

verPersonas(habilitarPersona(l, p)) ≡ Ag(p, verPersonas(l))
verPersonas(habilitarPuestoDeComida(l,p)) ≡ verPersonas(l)
verPersonas(realizarCompra(l,pe,i,pu,c)) ≡ verPersonas(l)
verPersonas(hackear(l, pe, i)) ≡ verPersonas(l)
verCompras(inaugurar) ≡ ∅
verCompras(habilitarPersona(l, p)) ≡ verCompras(l)
verCompras(habilitarPuestoDeComida(l,p)) ≡ verCompras(l)
verCompras(realizarCompra(l,pe,i,pu,c)) ≡ Ag(⟨pe, pu, i, c, cuantoDescuento?(pu,i,c)⟩, verCompras(l))
verCompras(hackear(l,pe,i)) ≡ verCompras(l) - {compraHackeada(l,pe,i)}
estaEnRegla(l,p) ≡ menuValido?(verItemsHabilitados(l), verMenu(p))
menuValido?(h,m) ≡ if esVacio?(claves(m)) then
    True
else
    (dameUno(claves(m)) ∉ claves(h) ∨L obtener(dameUno(claves(m)), m) = obtener(dameUno(claves(m)), h) ) ∧ menuValido?(h, borrar(dameUno(claves(m)), m)
fi
verMaximoGastador(l) ≡ obtenerMaximo(l, dameUno(verPersonas(l)), sinUno(verPersonas(l))
obtenerMaximo(l, p, ps) ≡ if #ps = 0 then p else if gastoTotalPersona(l,p) >= gastoTotalPersona(l,
    dameUno(ps)) then
    obtenerMaximo(l, p, sinUno(ps))
else
    obtenerMaximo(l, dameUno(ps), sinUno(ps)) fi
fi
comprasPersona(l,p) ≡ agruparGastos(p, verCompras(l))
agruparGastos(p,c) ≡ if ∅?(c) then c else if dameUno(c)persona = p then
    Ag(dameUno(c), agruparGastos(sinUno(c))
else
    agruparGastos(sinUno(c)) fi
fi
gastoTotalPersona(l,p) ≡ sumarGastos(comprasPersona(l,p))
sumarGastos(m) ≡ if ∅?(m) then
    0
else
    aplicarDescuento(dameUno(m)precio, dameUno(m)descuento) + sumarGastos(sinUno(m))
fi
verItemsHabilitados(l) ≡ unirMenus(verPuestos(l))
unirMenus(ps) ≡ if vacio?(ps) then vacio else verMenu(dameUno(ps)) ∪ unirMenus(sinUno(ps)) fi
compraHackeada(cs,pe,i) ≡ if Cpersona = pe ∧ Citem = i ∧ Cdescuento = 0 then
    C
else
    compraHackeada(sinUno(cs), pe, i)
fi donde C = dameUno(cs)

```

Fin TAD

2. TAD PUESTODeCOMIDA

TAD PuestoDeComida

géneros Puesto

usa Descuento, Menu, Oferta, Item, Bool

igualdad observacional

$$(\forall n, m : \text{Puesto}) \left(n =_{\text{obs}} m \iff \left(\text{verMenu}(n) =_{\text{obs}} \text{verMenu}(m) \wedge \text{verOfertas}(n) =_{\text{obs}} \text{verOfertas}(m) \wedge \bigwedge_L (\forall i : \text{Items}) (\text{verStockItem}(n, i) =_{\text{obs}} \text{verStockItem}(m, i)) \right) \right)$$

generadores

crearPuesto : Menu \longrightarrow Puesto

definirStock : Puesto $p \times$ Item $i \times$ Cantidad \longrightarrow Puesto $\{i \in \text{claves}(\text{verMenu}(p))\}$

crearOferta : Puesto $p \times$ Item $i \times$ Cantidad $c \times$ Descuento $d \longrightarrow$ Puesto $\{i \in \text{claves}(\text{verMenu}(p)) \wedge c > 0 \wedge d < 100\}$

observadores básicos

verMenu : Puesto \longrightarrow Menu

verOfertas : Puesto \longrightarrow Ofertas

verStockItem : Puesto \times Item \longrightarrow Cantidad

otras operaciones

hayItem? : Puesto \times Item \longrightarrow Bool

recargarStock : Puesto $p \times$ Item $i \times$ Cantidad \longrightarrow Puesto $\{i \in \text{claves}(\text{verMenu}(p))\}$

descontarStock : Puesto $p \times$ Item $i \times$ Cantidad $c \longrightarrow$ Puesto $\{i \in \text{claves}(\text{verMenu}(p)) \wedge c \leq \text{verStockItem}(p, i)\}$

cuantoDescuento? : Puesto $p \times$ Item $i \times$ Cantidad \longrightarrow Descuento $\{i \in \text{def?}(i, \text{verMenu}(p))\}$

axiomas

$\text{verMenu}(\text{crearPuesto}(m)) \equiv m$

$\text{verMenu}(\text{definirStock}(p, i, c)) \equiv \text{verMenu}(p)$

$\text{verMenu}(\text{crearOferta}(p, i, c, d)) \equiv \text{verMenu}(p)$

$\text{verOfertas}(\text{crearPuesto}(m)) \equiv \text{vacío}$

$\text{verOfertas}(\text{definirStock}(p, i, c)) \equiv \text{verOfertas}(p)$

$\text{verOfertas}(\text{crearOferta}(p, i, c, d)) \equiv \text{definir}(\langle i, c \rangle, d, \text{verOfertas}(p))$

$\text{verStockItem}(\text{crearPuesto}(m), i) \equiv 0$

$\text{verStockItem}(\text{definirStock}(p, r, c), i) \equiv \text{if } r =_{\text{obs}} i \text{ then } c \text{ else } \text{verStockItem}(p, i) \text{ fi}$

$\text{verStockItem}(\text{crearOferta}(p, o), i) \equiv \text{verStockItem}(p, i)$

$\text{hayItem?}(p, i) \equiv i \in \text{claves}(\text{verMenu}(p))$

$\text{recargarStock}(p, i, c) \equiv \text{definirStock}(p, i, \text{verStockItem}(p, i) + c)$

$\text{descontarStock}(p, i, c) \equiv \text{descontarStock}(p, i, \text{verStockItem}(p, i) - c)$

```
cuantoDescuento?(p,i,c)  $\equiv$  if c = 0 then 0 else if def?(<i,c>, verOfertas(p)) then  
    obtener(<i,c>, verOfertas(p))  
else  
    cuantoDescuento?(p, i, c-1) fi  
fi
```

Fin TAD

