

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

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

Trabajo Práctico 1: Especificación

Lollapatuza

Integrante	LU	Correo electrónico
Marziano, Franco	849/19	franco.marziano.96@gmail.com
Gonzalez Correas, Alvaro	233/22	alvarogonzalezc4@gmail.com
Vazquez, Martin Ignacio	327/17	vazquez.martin.ignacio@gmail.com
Sturm, Candelaria	244/20	sturmcande@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. Lollapatuza

1.1. Festival

TAD PRODUCTO es **STRING**

TAD CANTIDAD es **NAT**

TAD DESCUENTO es **NAT**

TAD CONSUMO es **TUPLA**(**PRODUCTO** × **CANTIDAD** × **DESCUENTO**)

TAD Festival

igualdad observacional

$$(\forall f, f' : \text{Festival}) \left(f =_{\text{obs}} f' \iff \left(\begin{array}{c} \text{Puestos}(f) =_{\text{obs}} \text{Puestos}(f') \wedge_{\text{L}} \\ \text{Personas}(f) =_{\text{obs}} \text{Personas}(f') \end{array} \right) \right)$$

usa Diccionario, Puesto, Persona, Conjunto, Bool, Nat, Tupla

exporta observadores, generadores, operación adicional

géneros Festival

observadores básicos

puestos : Festival $f \rightarrow \text{Conj}(\text{Puesto})$

personas : Festival $f \rightarrow \text{conj}(\text{persona})$

generadores

abrirFestival : $\text{Conj}(\text{Puesto}) \times \text{Conj}(\text{Persona}) \rightarrow \text{Festival } f$

$$\left\{ \begin{array}{l} (\forall p1, p2 : \text{puesto}) (p1 \in cp \wedge p2 \in cp \Rightarrow (\nexists k : \text{producto}) (k \in \text{claves}(\text{menu}(p1)) \wedge k \in \\ \text{claves}(\text{menu}(p2)) \wedge_{\text{L}} \text{obtener}(k, \text{menu}(p1)) \neq \text{obtener}(k, \text{menu}(p2))) \end{array} \right\}$$

comprar : festival $f \times \text{Persona } p \times \text{Puesto } pu \times \text{Conj}(<\text{Producto} \times \text{Nat}>) \times cp \rightarrow \text{Festival}$

$$\{p \in \text{personas}(f) \wedge pu \in \text{puestos}(f) \wedge (\forall k : <\text{producto}, \text{nat}>) (k \in cp \Rightarrow_{\text{L}} \pi_1(k) \in \text{menu}(pu))\}$$

hackeo : $\text{Persona } p \times \text{festival} \rightarrow \text{Festival}$

$$\left\{ \begin{array}{l} (p \in \text{personas}(f)) \wedge (\exists k : \text{puesto}) (k \in \text{claves}(\text{historialDeCompras}(pe)) \wedge_{\text{L}} \\ \text{hayCompraSinDescuentoEnPuesto}(\text{obtener}(k, \text{historialDeCompras}(pe))) \end{array} \right\}$$

otras operaciones

personaQueMasGasto : Festival $\rightarrow \text{persona}$

personaQueMasGastoDadaLista : festival $f \times \text{conj}(\text{persona}) \times p \rightarrow \text{persona}$ $\{p \in \text{personas}(f)\}$

axiomas

$\forall pu : \text{puesto}, f : \text{festival}, pe : \text{persona}, co : \text{conj}(<\text{producto}, \text{nat}>), pers : \text{conj}(\text{persona})$

$\text{Puestos}(\text{abrirFestival}(pu, pers)) \equiv pu$

$\text{Puestos}(\text{comprar}(f, pe, pu, co)) \equiv \text{if } \text{vacio}(co) \text{ then } \text{puestos}(f)$

else

if sePuedeVender($pu, \pi_1(\text{dameUno}(co)), \pi_2(\text{dameUno}(co))$) **then**
 $\text{Vender}(pu, \pi_1(\text{dameUno}(co)), \pi_2(\text{dameUno}(co))) \wedge$

$\text{Puestos}(\text{comprar}(f, pe, pu, \text{sinUno}(co)))$

else

$\text{Puestos}(\text{Comprar}(f, pe, pu, \text{sinUno}(co)))$

fi

fi

$\text{Personas}(\text{abrirFestival}(pu, pers)) \equiv \text{pers}(f)$

```

Personas(Comprar(f,pe,pu,co)  $\equiv$  if vacio(co) then
    personas(f)
else
    if sePuedeVender(pu, $\pi_1$ (dameUno(co), $\pi_2$ (dameUno(co)))) then
        comprar(pu, $\pi_1$ (dameUno(co), $\pi_2$ (dameUno(co))) $\wedge$ 
            Personas(comprar(f,pe,pu,sinUno(co)))
    else
        Personas(comprar(f,pe,pu,sinUno(co)))
    fi
fi

Puestos(hackeo(pe,f))  $\equiv$  Ag(hackeoPuesto(puestoDeCompraSinDescuento(pe), $\pi_1$ (compraSinDescuento(pe))),
    puestos(f)-puestoDeCompraSinDescuento(pe))

Personas(hackeo(pe,f))  $\equiv$  Ag(hackeoPersona(pe),personas(f)-pe)

personaQueMasGasto(f)  $\equiv$  personaQueMasGastoDadaLista(f,personas(f))

personaQueMasGastoDadaLista(f,pers)  $\equiv$  if vacio(pers) then
     $\emptyset$ 
else
    max(plataGastadaDe(dameUno(pers)),
        personaQueMasGastoDadaLista(f,sinUno(pers))
    fi

```

Fin TAD

1.2. Puesto de Comida

TAD Puesto de Comida

igualdad observacional

$$(\forall p, p' : \text{Puesto}) \left(p =_{\text{obs}} p' \iff \left(\text{menu}(p) =_{\text{obs}} \text{menu}(p') \wedge_{\text{L}} \text{descuentos}(p) =_{\text{obs}} \text{descuentos}(p') \right) \right)$$

usa Diccionario, Conjunto, Bool, Nat

exporta observadores, generadores, Puesto, operación adicional

géneros Puesto

observadores básicos

$\text{menu} : \text{Puesto} \rightarrow \text{Dicc}(\text{producto}:\text{nat})$

$\text{descuento} : \text{Puesto} \rightarrow \text{conj}(\langle \text{producto}, \text{cantidad}, \text{descuento} \rangle)$

$\text{stock} : \text{Puesto} \rightarrow \text{Dicc}(\text{producto}:\text{cantidad})$

generadores

$\text{abrirPuesto} : \text{Dicc}(\text{producto}:\text{nat}) \times \text{conj}(\text{consumo}) \times \text{Dicc}(\text{producto}:\text{cantidad}) \times s \rightarrow \text{Puesto}$
 $\left\{ \begin{array}{l} (\forall p : \text{consumo})(p \in d) \Rightarrow_{\text{L}} (\nexists c : \text{consumo})((c \in d) \wedge_{\text{L}} (\pi_1(c) = \pi_1(p) \wedge \pi_2(c) = \pi_2(p)) \wedge_{\text{L}} \pi_3(c)) \\ \neq \pi_3(p) \end{array} \right\}$

$\text{vender} : \text{puesto } pu \times \text{producto } p \times \text{cantidad } c \rightarrow \text{Puesto} \quad \{c \leq \text{obtener}(p, \text{stock}(pu))\}$

$\text{hackeoPuesto} : \text{Puesto} \times \text{Producto} \rightarrow \text{Puesto}$

otras operaciones

$\text{sePuedeVender?} : \text{Puesto} \times \text{Producto} \times \text{Cantidad} \rightarrow \text{bool}$

$\text{tieneDescuento?} : \text{Puesto} \times \text{Producto} \times \text{Cantidad} \rightarrow \text{bool}$

$\text{tieneDescuento?Aux} : \text{conj}(\text{consumo}) \times \text{producto} \times \text{cantidad} \rightarrow \text{bool}$

$\text{queDescuento?} : \text{Puesto } Pu \times \text{Producto } p \times \text{Cantidad } c \rightarrow \text{descuento} \quad \{\text{tieneDescuento?}(pu, p, c)\}$

$\text{queMejorDescuento} : \text{conj}(\text{descuento}) \times \text{producto} \times \text{cantidad} \rightarrow \text{descuento}$

$\text{queDescuentoExacto} : \text{conj}(\text{descuento}) \times \text{producto} \times \text{cantidad} \rightarrow \text{descuento}$

axiomas

$\forall m : \text{Dicc}(\text{producto} : \text{nat}), d : \text{conj}(\text{consumo}), s : \text{Dicc}(\text{producto} : \text{cantidad}), pu : \text{puesto}, p : \text{producto}, c : \text{cantidad}$

$\text{menu}(\text{abrirPuesto}(m, d, s)) \equiv m$

$\text{menu}(\text{vender}(pu, p, c)) \equiv \text{menu}(pu)$

$\text{menu}(\text{hackeoPuesto}(pu, p, c)) \equiv \text{menu}(pu)$

$\text{descuentos}(\text{abrirPuesto}(m, d, s)) \equiv d$

$\text{descuentos}(\text{vender}(pu, p, c)) \equiv \text{descuentos}(pu)$

$\text{descuentos}(\text{hackeoPuesto}(pu, p)) \equiv \text{descuentos}(pu)$

$\text{stock}(\text{abrirPuesto}(m, d, s)) \equiv s$

$\text{stock}(\text{vender}(pu, p, c)) \equiv \text{def}(p, \text{obtener}(p, \text{stock}(pu)) - c, \text{stock}(pu))$

$\text{stock}(\text{hackeoPuesto}(pu, p)) \equiv \text{definir}(p, \text{obtener}(p, \text{stock}(pu)) + 1, \text{stock}(pu))$

$\text{sePuedeVender?}(pu, p, c) \equiv \text{obtener}(p, \text{stock}(pu)) \geq c$

$\text{tieneDescuento}(pu, p, c) \equiv \text{tieneDescuento?Aux}(\text{descuentos}(pu), p, c)$

```

tieneDescuento?Aux(d,p,c)  $\equiv$  if vacio(d) then
    false
else
    (if  $\pi_1(\text{dameUno}(d)) = p \wedge \pi_2(\text{dameUno}(d)) \leq c$  then
        true
    else
        tieneDesceunto?Aux(sinUno(d),p,c)
    fi)
fi
queDescuento(pu,p,c)  $\equiv$  queMejorDescuento(descuentos(pu),p,c)
queMejorDescuento(co,p,c)  $\equiv$  if queDescuentoExacto(co,p,c)  $\neq 0$  then
    queDescuentoExacto(co,p,c)
else
    queDescuentoExacto(co,p,c-1)
fi
queDescuentoExacto(co,p,c)  $\equiv$  if vacio(co) then 0 else if  $\pi_1(\text{dameUno}(co)) = p \wedge c = \pi_2(\text{dameUno}(co))$ 
then
     $\pi_3(\text{dameUno}(co))$ 
else
    queDescuentoExacto(sinUno(co),p,c) fi
fi

```

Fin TAD

1.3. Persona

TAD Persona

igualdad observacional

$$(\forall pe, pe' : \text{Persona}) \left(pe =_{\text{obs}} pe' \iff \left(\begin{array}{c} \text{historialDeCompras}(pe) =_{\text{obs}} \\ \text{historialDeCompras}(pe') \end{array} \right) \right)$$

usa Diccionario, Conjunto, Bool, Nat

exporta observadores, generadores, Persona, operación adicional, Puesto

géneros Persona

observadores básicos

$\text{historialDeCompras} : \text{persona} \longrightarrow \text{dicc} \{ \text{puesto} : \text{multiconj}(\text{consumo}) \}$

generadores

$\text{llegar} : \longrightarrow \text{persona}$

$\text{comprar} : \text{Puesto } Pu \times \text{Producto } p \times \text{Cantidad } c \times \text{Persona } pe \longrightarrow \text{persona}$
 $\{ (p \in \text{claves}(\text{menu}(pu)) \wedge \text{sePuedeVender}(pu, p, c)) \}$

$\text{hackeoPersona} : \text{persona} \longrightarrow \text{persona}$

otras operaciones

$\text{plataGastadaDe} : \text{persona} \longrightarrow \text{nat}$

$\text{plataGastadaTotal} : \text{dicc}(\text{puesto} : \text{multiconj}(\text{consumo})) \text{ di} \times \text{conj}(\text{puesto}) \text{ co} \longrightarrow \text{nat}$ $\{ \text{co} = \text{claves}(\text{di}) \}$

$\text{plataGastadaPorPuesto} : \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo})) \text{ di} \times \text{puesto } pu \times \text{multiconj}(\text{consumo}) \text{ mu} \longrightarrow \text{nat}$
 $\{ pu \in \text{claves}(\text{di}) \wedge \text{obtener}(pu, \text{di}) = mu \}$

$\text{compraSinDescuento} : \text{persona} \longrightarrow \text{consumo}$

$\text{compraSinDescuentoEnHistorial} : \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo})) \text{ di} \times \text{conj}(\text{puesto}) \text{ cp} \longrightarrow \text{consumo}$
 $\{ cp \in \text{claves}(\text{di}) \}$

$\text{hayCompraSinDescuentoEnPuesto} : \text{mult}(\text{consumo}) \longrightarrow \text{bool}$

$\text{compraSinDescuentoEnPuesto} : \text{mult}(\text{consumo}) \text{ mu} \longrightarrow \text{consumo}$
 $\{ \text{hayCompraSinDescuentoEnPuesto}(\text{mu}) \}$

$\text{puestoDeCompraSinDescuento} : \text{persona} \longrightarrow \text{puesto}$

$$\left\{ (\exists pu : \text{puesto}, mu : \text{multiconj}(\text{consumo})) ((pu \in \text{claves}(\text{historialDeCompras}(pe)) \wedge mu \in \{ \text{obtener}(pu, \text{historialDeCompras}(pe)) \} \wedge_L \text{hayCompraSinDescuentoEnPuesto}(mu)) \right\}$$

$\text{puestoDeCompraSinDescuentoDadoHistorial} : \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo})) \text{ di} \times \text{conj}(\text{puesto}) \text{ co} \longrightarrow \text{puesto}$
 $\{ \text{co} = \text{claves}(\text{di}) \}$

$\text{borrarConsumo} : \text{persona } p \times \text{puesto } pu \times \text{consumo } c \longrightarrow \text{persona}$
 $\{ c \in \text{obtener}(pu, \text{historialDeCompras}(pe)) \}$

$\text{borrarConsumoDadoHistorial} : \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo})) \text{ di} \longrightarrow \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo}))$
 $\times \text{puesto } pu \times \text{consumo } c$
 $\{ pu \in \text{claves}(\text{di}) \wedge c \in \text{obtener}(pu, \text{di}) \}$

$\text{borrarConsumoDadoHistorialAux} : \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo})) \text{ di} \times \longrightarrow \text{dict}(\text{puesto} : \text{multiconj}(\text{consumo}))$
 $\text{producto } p \times \text{multiconj}(\text{consumo}) \text{ mu}$
 $\times \text{consumo } c$
 $\{ p \in \text{claves}(\text{di}) \wedge mu = \text{obtener}(pu, \text{di}) \wedge c \in mu \}$

$\text{restarConsumo} : \text{consumo} \longrightarrow \text{consumo}$

axiomas $\forall pu, pa$: puesto, $\forall pe, pi$: persona, $\forall p, q$: producto, $\forall di$: dict(puesto:multiconj(consumo)), $\forall cl$: conj(puesto), $\forall mu$: multiconj(consumo), $\forall con$: consumo ,
 $\forall c, k$: cantidad

historialDeCompras(llegar()) $\equiv \emptyset$

historialDeCompras(comprar(pu,p,c,pe)) \equiv **if** def?(pu,historialDeCompras(pe)) \wedge tieneDescuento(pu,p,c)
then
 definir(pu,Ag($\langle p,c,queDescuento(pu,p,c) \rangle$,
 obtener(pu,historialDeCompras(pe)))
 ,historialDeCompras(pe))
else
 if def?(pu,historialDeCompras(pe))
 $\wedge \neg$ tieneDescuento(pu,p,c) **then**
 definir(pu,Ag($\langle p,c,0 \rangle$,
 obtener(pu,historialDeCompras(pe))),historialDeCompras(pe))
 else
 if \neg def?(pu,historialDeCompras(pe)) \wedge tieneDescuen-
 to(pu,p,c) **then**
 definir(p,Ag($\langle p,c,queDescuento(pu,p,c) \rangle$), \emptyset)
 ,historialDeCompras(pe))
 else
 definir(pu,Ag($\langle p,c,0 \rangle$), historialDeCompras(pe))
 fi
 fi
fi

plataGastadaDe(pe) \equiv plataGastadaAux(historialDeCompras(pe),claves(historialDeCompras(pe)))

plataGastadaTotal(di,cl) \equiv **if** vacio(cl) **then**
 0
else
 plataGastadaPorPuesto(di,dameUno(cl),obtener(dameUno(cl),di) + plataGas-
 tadaTotal(di,sinUno(cl))
fi

plataGastadaPorPuesto(di,pu,mu) \equiv **if** vacio(mu) **then**
 0
else
 aplicarDesc(obtener(π_1 (dameUno(mu)),menu(pu))* π_2 (dameUno(mu)),
 π_3 (dameUno(mu))) + plataGastadaPorPuesto(di,pu,sinUno(mu))
fi

historialDeCompras(hackeoPersona(pe)) \equiv **if** π_2 (compraSinDescuento(pe)) = 1 **then**
 borrarConsumo(
 historialDeCompras(pe),
 puestoDeCompraSinDescuento(pe),
 compraSinDescuento(pe))
else
 definir(
 puestoDeCompraSinDescuento(pe),
 Ag(restarConsumo(compraSinDescuento(pe)),
 obtener(puestoDeCompraSinDescuento(pe),
 borrarConsumo(pe,puestoDeCompraSinDescuento(pe),
 compraSinDescuento(pe))),historialDeCompras(pe))
 fi
fi

borrarConsumo(pe,pu,con) \equiv borrarConsumoDadoHistorial(historialDeCompras(pe),pu,con)

borrarConsumoDadoHistorial(di,pu,con) \equiv borrarConsumoDadoHistorialAux(di,obtener(pu,di),con)

```

borrarConsumoDadoHistorialAux(di,mu,con)  $\equiv$  if dameUno(mu) = con then
    definir(pu,mu - dameUno(mu), di)
else
    borrarConsumoDadoHistorialAux(di,sinUno(mu),con)
fi

puestoDeCompraSinDescuento(pe)  $\equiv$  puestoDeCompraSinDescuentoDadoHistorial
    (historialDeCompras(pe),claves(historialDeCompras(pe)))

puestoDeCompraSinDescuentoDadoHistorial(di,cl)  $\equiv$  if
    hayCompraSinDescuentoEnPuesto(obtener(dameUno(cl),di))
then
    dameUno(cl)
else
    puestoDeCompraSinDescuentoDadoHistorial(di,sinUno(cl))
fi

hayCompraSinDescuentoEnPuesto(mu)  $\equiv$  if vacio(mu) then false else if  $\pi_3$ (dameUno(mu))=0 then
    true
else
    hayCompraSinDescuento(sinUno(mu)) fi

compraSinDescuento(pe)  $\equiv$  compraSinDescuentoEnHistorial(historialDeCompras(pe),claves(historialDeCompras(pe)))

compraSinDescuentoEnHistorial(di,cl)  $\equiv$  if    hayCompraSinDescuentoEnPuesto(obtener(dameUno(cl),di))
then
    compraSinDescuentoEnPuesto(obtener(dameUno(cl),di))
else
    compraSinDescuentoEnHistorial(di,sinUno(cl))
fi

compraSinDescuentoEnPuesto(mu)  $\equiv$  if  $\pi_3$ (dameUno(mu)) = 0 then
    dameUno(mu)
else
    CompraSinDescuento(sinUno(mu))
fi

restarConsumo(con)  $\equiv$   $\langle \pi_1(\text{consumo}), \pi_2(\text{consumo})-1, \pi_3(\text{consumo}) \rangle$ 

```

Fin TAD