

Algoritmos y Estructuras de Datos II

Especificación

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

Trabajo Práctico 1: Especificación

Alias del grupo:
WOWOANERNCTYDOHJTMEA

Integrante	LU	Correo electrónico
Barragán, Jerónimo	1472/21	barragan.jeronimo123@gmail.com
Cardozo, Alfredo	820/19	mail@alfredo.com
García Alurralde, Jorge	437/22	jalurralde@dc.uba.ar
Sarkissian, Ralph	1698/21	ralphsarkissian.b@gmail.com



Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega	Fermín	A
Segunda entrega		

1. Renombres de tipos

TAD Persona es STRING

TAD Compra es TUPLA(Puesto, STRING, NAT)

TAD Promo es TUPLA(NAT, NAT)

2. TAD Lollapatuza

TAD Lollapatuza

géneros festival

exporta festival, generadores, observadores, estáEnElFestival, stock, hayStock?, másGastó, cuantoGastó

usa BOOL, NAT, SECUENCIA(α), CONJUNTO(α), MULTICONJUNTO(α), DICCIONARIO(CLAVE, SIGNIFICADO), STRING, PUESTO, PERSONA, COMPRA

igualdad observacional

$$(\forall f_1, f_2: \text{festival}) (f_1 =_{\text{obs}} f_2 \iff \left(\begin{array}{l} \text{Puestos}(f_1) =_{\text{obs}} \text{Puestos}(f_2) \wedge_L \\ \text{Personas}(f_1) =_{\text{obs}} \text{Personas}(f_2) \wedge_L \\ (\forall \text{per} : \text{persona}) (\text{per} \in \text{Personas}(f_1) \Rightarrow_L \\ \text{Compras}(f_1, \text{per}) =_{\text{obs}} \text{Compras}(f_2, \text{per})) \end{array} \right))$$

observadores básicos

Puestos : festival \rightarrow conj(puesto)

Personas : festival $f \rightarrow$ conj(persona)

Compras : festival $f \times$ persona $\text{per} \rightarrow$ multiconj(compra) {estáEnElFestival?(f, per)}

generadores

abrirFestival : \rightarrow festival

ponerPuesto : festival $f \times$ puesto $p \rightarrow$ festival { $p \notin \text{Puestos}(f) \wedge (\forall q : \text{puesto}) (q \in \text{Puestos}(f) \Rightarrow_L$
($\forall \text{ítem} : \text{string}) (\text{vende?}(p, \text{ítem}) \wedge \text{vende?}(q, \text{ítem}) \Rightarrow_L$
 $\text{precio}(p, \text{ítem}) =_{\text{obs}} \text{precio}(q, \text{ítem}))$)}

agregarPersona : festival $f \times$ persona $\text{per} \rightarrow$ festival { $\neg \text{estáEnElFestival?}(f, \text{per})$ }

comprarItem : festival $f \times$ puesto $p \times$ persona $\text{per} \times$ string $\text{ítem} \times$ nat $\text{unidades} \rightarrow$ festival
{ $(p \in \text{Puestos}(f) \wedge \text{vende?}(p, \text{ítem})) \wedge_L$
 $\text{estáEnElFestival?}(f, \text{per}) \wedge \text{hayStock?}(p, \text{ítem}, \text{unidades})$ }

Podría no ser gen.

hackear : festival $f \times$ persona $p \times$ string $\text{ítem} \rightarrow$ festival
{ $\text{estáEnElFestival?}(f, \text{per}) \wedge_L (\exists c : \text{compras}) (c \in \text{Compras}(f, \text{per}) \wedge_L (\text{ítemDe}(c) =_{\text{obs}} \text{ítem} \wedge_L$
 $\text{cuántoGastóEnCompra}(f, \text{per}, c) =_{\text{obs}} \text{precio}(\text{puestoDe}(c), \text{ítem}) \times \text{cantidadDe}(c)))$ }

otras operaciones

estáEnElFestival? : festival $f \times$ persona $\text{per} \rightarrow$ bool

stock : festival $f \times$ puesto $p \times$ string $\text{ítem} \rightarrow$ nat { $p \in \text{Puestos}(f) \wedge \text{vende?}(p, \text{ítem})$ }

hayStock? : festival $f \times$ puesto $p \times$ string $\text{ítem} \times$ nat $\text{unidades} \rightarrow$ bool
{ $p \in \text{Puestos}(f) \wedge \text{vende?}(p, \text{ítem})$ }

másGastó : festival $f \rightarrow$ persona { $\neg \text{vacío?}(\text{Personas}(f))$ }

másGastóAux : festival $f \times$ conj(persona) $c \rightarrow$ persona
{ $\neg \text{vacío?}(c) \wedge_L (\forall \text{per} : \text{persona}) (\text{per} \in c \Rightarrow_L \text{estáEnElFestival?}(f, \text{per}))$ }

cuántoGastó : festival $f \times$ persona $\text{per} \rightarrow$ nat { $\text{estáEnElFestival?}(f, \text{per})$ }

cuántoGastóAux : festival $f \times$ persona $\text{per} \times$ multiconj(compra) \rightarrow nat { $\text{estáEnElFestival?}(f, \text{per})$ }

cuántoGastóEnCompra : festival $f \times$ persona $\text{per} \times$ compra $c \rightarrow$ nat { $\text{estáEnElFestival?}(f, \text{per})$ }

compraAHackear : festival $f \times$ persona $\text{per} \times$ multiconj(compra) $\text{compras} \times$ string $\text{ítem} \rightarrow$ compra
{ $\text{estáEnElFestival?}(f, \text{per}) \wedge_L (\exists c : \text{compra}) (c \in \text{Compras}(f, \text{per}) \wedge_L (\text{ítemDe}(c) =_{\text{obs}} \text{ítem} \wedge_L$
 $\text{cuántoGastóEnCompra}(f, \text{per}, c) = \text{precio}(\text{puestoDe}(c), \text{ítem}) \times \text{cantidadDe}(c)))$ }

puestoDe : compra \rightarrow puesto

ítemDe : compra \rightarrow string

cantidadDe : compra \rightarrow nat

axiomas $\forall f: \text{festival}, \forall p, p_1, p_2: \text{puesto}, \forall \text{per}, \text{per}_1, \text{per}_2: \text{persona}, \forall \text{ítem}, \text{ítem}_1, \text{ítem}_2: \text{string},$
 $\forall \text{unidades}: \text{nat}, \forall \text{compras}: \text{multiconj}(\text{compra}), \forall \text{compra}: \text{compra}$

Puestos(abrirFestival) $\equiv \emptyset$

Puestos(ponerPuesto(f, p)) $\equiv \text{Ag}(p, \text{Puestos}(f))$

Puestos(agregarPersona(f, per)) $\equiv \text{Puestos}(f)$

Puestos(comprarItem(f, p, per, ítem, unidades)) $\equiv \text{Puestos}(f)$

Personas(abrirFestival) $\equiv \emptyset$

Personas(ponerPuesto(f, p)) $\equiv \text{Personas}(f)$

Personas(agregarPersona(f, per)) \equiv Ag(per, Personas(f))
 Personas(comprarÍtem(f, p, per₂, ítem, unidades), per₁) \equiv Personas(f) ✓
 Compras(ponerPuesto(f, p), per) \equiv Compras(f, per)
 Compras(agregarPersona(f, per₂), per₁) \equiv **if** per₁ =_{obs} per₂ **then** \emptyset **else** Compras(f, per₁) **fi** ✓
 Compras(comprarÍtem(f, p, per₂, ítem, unidades), per₁) \equiv **if** per₁ =_{obs} per₂ **then** Ag(<p, ítem, unidades>, Compras(f, per₁)) **else** Compras(f, per₁) **fi** ✓
 estáEnElFestival?(f, per) \equiv per \in Personas(f)
 stock(ponerPuesto(f, p₂), p₁, ítem) \equiv **if** p₁ =_{obs} p₂ **then** stockDe(p₁, ítem) **else** stock(f, p₁, ítem) **fi** ✓
 stock(agregarPersona(f, per), p, ítem) \equiv stock(f, p, ítem)
 stock(comprarÍtem(f, p, per, ítem₂, unidades), p, ítem₁) \equiv **if** ítem₁ =_{obs} ítem₂ **then** stock(f, p, ítem₁) - unidades **else** stock(f, p, ítem₁) **fi** ✓
 hayStock?(f, p, ítem, unidades) \equiv unidades \leq stock(f, p, ítem) ✓
 másGastó(f) \equiv másGastóAux(f, Personas(f))
 másGastóAux(f, personas) \equiv **if** #(personas) =_{obs} 1 **then** dameUno(personas) **else if** cuántoGastó(f, dameUno(personas)) \geq cuántoGastó(f, dameUno(sinUno(personas))) **then** másGastó(f, Ag(dameUno(personas), sinUno(sinUno(personas)))) **else** másGastó(f, sinUno(personas)) **fi fi** ✓
 cuántoGastó(f, per) \equiv cuántoGastóAux(f, per, Compras(f, per))
 cuántoGastóAux(f, per, compras) \equiv **if** compras =_{obs} \emptyset **then** 0 **else** cuántoGastóEnCompra(f, per, dameUno(compras)) + cuántoGastóAux(f, per, sinUno(compras)) **fi** ✓
 cuántoGastóEnCompra(f, per, compra) \equiv **if** hayPromo?(puestoDe(compra), ítemDe(compra), cantidadDe(compra)) **then** aplicarDescuento(cantidadDe(compra) \times precio(puestoDe(compra), descuento(puestoDe(compra), ítemDe(compra), cantidadDe(compra))) **else** cantidadDe(compra) \times precio(puestoDe(compra), ítemDe(compra)) **fi** ✓
 Puestos(hackear(f, per, ítem)) \equiv Puestos(f)
 estáEnElFestival?(hackear(f, per₂, ítem), per₁) \equiv estáEnElFestival?(f, per₁)
 Compras(hackear(f, per₂, ítem), per₁) \equiv **if** per₂ \neq_{obs} per₁ **then** Compras(f, per₁) **else** Compras(f, per₁) - {compraAHackear(f, per₁, Compras(f, per₁, ítem), ítem)} **fi** ✓
 stock(hackear(f, per, ítem₂), p, ítem₁) \equiv **if** ítem₁ =_{obs} ítem₂ \wedge puestoDe(compraAHackear(f, per, ítem₁)) =_{obs} p **then** stock(f, p, ítem₁) + cantidadDe(compraAHackear(f, per, ítem₁)) **else** stock(f, p, ítem₁) **fi** ✓
 compraAHackear(f, per, compras, ítem) \equiv **if** esHackeable(f, per, dameUno(compras), ítem) **then** dameUno(compras) **else** compraAHackear(per, sinUno(compras), ítem) **fi** ✓
 esHackeable(f, per, compra, ítem) \equiv ítemDe(compra) =_{obs} ítem \wedge_L cuántoGastóEnCompra(f, per, compra) =_{obs} precio(puestoDe(compra), ítem) \times cantidadDe(compra) ✓
 puestoDe(compra) \equiv π_1 (compra)
 ítemDe(compra) \equiv π_2 (compra)
 cantidadDe(compra) \equiv π_3 (compra)

Fin TAD

3. TAD Puesto

TAD Puesto

géneros puesto

exporta puesto, generadores, observadores, vende?, precio, hayPromo?, descuento, stockDe

usa BOOL, NAT, SECUENCIA(α), DICCIONARIO(CLAVE, SIGNIFICADO), STRING, PROMO

igualdad observacional

$$(\forall p_1, p_2: \text{puesto}) \ (p_1 =_{\text{obs}} p_2 \iff \left(\begin{array}{l} \text{Menú}(p_1) =_{\text{obs}} \text{Menú}(p_2) \wedge \text{Stock}(p_1) =_{\text{obs}} \text{Stock}(p_2) \\ \wedge \text{Promos}(p_1) =_{\text{obs}} \text{Promos}(p_2) \end{array} \right))$$

observadores básicos

Menú : puesto \longrightarrow dicc(string, nat)

Stock : puesto \longrightarrow dicc(string, nat)

Promos : puesto \longrightarrow dicc(string, secu(promo))

generadores

abrirPuesto : \longrightarrow puesto

$$\text{agregarÍtem} : \text{puesto } p \times \text{string ítem} \times \text{nat precio} \times \text{nat stock} \longrightarrow \text{puesto} \quad \{ \neg \text{def?}(\text{ítem}, \text{Stock}(p)) \}$$
$$\text{agregarPromo} : \text{puesto } p \times \text{string } \text{ítem} \times \text{nat } \text{cantidad} \times \text{nat } \text{descuento} \rightarrow \text{puesto}$$

$$\left\{ \begin{array}{l} \text{descuento} > 0 \wedge (\forall c : \text{string}) ((c \in \text{claves}(\text{Promos}(p)) \wedge c =_{\text{obs}} \text{ítem}) \Rightarrow_{\text{L}} \\ (\forall \text{prom} : \text{promo}) (\text{está?}(\text{prom}, \text{obtener}(c, \text{Promos}(p)) \Rightarrow_{\text{L}} \pi_1(\text{prom}) \neq_{\text{obs}} \text{cantidad}))) \end{array} \right\}$$

otras operaciones

$$\text{vende?} : \text{puesto } p \times \text{string } \textit{ítem} \longrightarrow \text{bool}$$
$$\text{precio} : \text{puesto } p \times \text{string } \textit{ítem} \longrightarrow \text{nat}$$
$$\text{hayPromo?} : \text{puesto } p \times \text{string } \textit{ítem} \times \text{nat } \textit{cantidad} \longrightarrow \text{bool}$$
$$\text{descuento} : \text{puesto } p \times \text{string ítem} \times \text{nat cantidad} \longrightarrow \text{nat}$$
$$\{\text{vende? (p, ítem)} \wedge_L \text{hayPromo? (p, ítem, cantidad)}\}$$
$$\text{stockDe} : \text{puesto } p \times \text{string } \textit{ítem} \longrightarrow \text{nat}$$
$$\text{cantMínDePrim} : \text{secu}(\text{promo})\ s \longrightarrow \text{nat}$$
$$\text{cantMínDeSeg} : \text{secu}(\text{promo})\ s \longrightarrow \text{nat}$$
$$\text{hayPromoAux} : \text{puesto } p \times \text{string } \textit{ítem} \times \text{nat } \textit{cantidad} \times \text{secu}(\text{promo}) \textit{s} \longrightarrow \text{bool}$$
$$\{\text{def?}(\text{ítem}, \text{Promos}(p)) \wedge \neg \text{vacía?}(s)\}$$
$$\text{descuentoAux} : \text{puesto } p \times \text{string } \textit{ítem} \times \text{nat } \textit{cantidad} \times \text{secu}(\text{promo}) \, s \longrightarrow \text{promo}$$
$$\{\text{hayPromoAux}(p, \text{ítem}, \text{cantidad}, \text{Promos}(p)) \wedge \neg \text{vacía?}(s)\}$$

axiomas $\forall p$: puesto, $\forall item$: string, $\forall precio, stock, cantidad, descuento$: nat, $\forall promos$: secu(promo)

$$\text{Menú}(\text{abrirPuesto}) \equiv \text{vacío}$$
$$\text{Menú}(\text{agregarÍtem}(p, \text{ítem}, \text{precio}, \text{stock})) \equiv \text{definir}(\text{ítem}, \text{precio}, \text{Menú}(p))$$
$$\text{Menú}(\text{agregarPromo}(p, \text{ítem}, \text{cantidad}, \text{descuento})) \equiv \text{Menú}(p)$$
$$\text{Stock}(\text{abrirPuesto}) \equiv \text{vacío}$$
$$\text{Stock}(\text{agregarÍtem}(p, \text{ítem}, \text{precio}, \text{stock})) \equiv \text{definir}(\text{ítem}, \text{stock}, \text{Stock}(p))$$
$$\text{Stock}(\text{agregarPromo}(p, \text{ítem}, \text{cantidad}, \text{descuento})) \equiv \text{Stock}(p)$$
$$\text{Promos}(\text{abrirPuesto}) \equiv \text{vacío}$$
$$\text{Promos}(\text{agregarÍtem}(p, \text{ítem}, \text{precio}, \text{stock})) \equiv \text{Promos}(p)$$
Promos(agregarPromo(p, ítem, cantidad, descuento)) \equiv if def?(ítem, Promos(p))

```

then definir(ítem, <cantidad, descuento> •
    obtener(ítem, Promos(p)), Promos(p))

```

```

else definir(item, <cantidad, descuento> • <> ,
             Promos(p))

```

fi

$$\text{vende?}(p, \text{ítem}) \equiv \text{def?}(\text{ítem}, \text{Menú}(p))$$

```
precio(p, ítem) ≡ obtener(ítem, Menú(p))
```

```

hayPromo?(p, ítem, cantidad) ≡ if def?(ítem, Promos(p))
    then hayPromoAux(p, ítem, cantidad, obtener(ítem, Promos(p)))
    else false

```

fi

$$\text{cantMínDePrim}(\text{promos}) \equiv \pi_1(\text{prim}(\text{promos}))$$
$$\text{cantMínDeSeg}(\text{promos}) \equiv \pi_1(\text{prim}(\text{fin}(\text{promos})))$$

//Cabe observar que la secuencia promos nunca es vacía
hayPromoAux(p, ítem, cantidad, promos) \equiv

```

if long(promos) =obs 1
  then cantidad  $\geq$  cantMínDePrim(promos)
  else if cantMínDePrim(promos) > cantidad
    then hayPromoAux(p, ítem, cantidad, fin(promos))
    else if cantMínDeSeg(promos)  $\leq$  cantidad
      then if cantMínDeSeg(promos) > cantMínDePrim(promos)
        then hayPromoAux(p, ítem, cantidad, fin(promos))
        else hayPromoAux(p, ítem, cantidad, prim(promos) • fin(fin(promos)))
      fi
    else hayPromoAux(p, ítem, cantidad, prim(promos) • fin(fin(promos)))
  fi
fi

```

*Para qué todo esto?
Por qué no fijarse si alguna promo es 15 cent?
(es decir, dejar solo las primeras tres líneas)*

descuento(p, ítem, cantidad) $\equiv \pi_2(\text{promoAux}(p, \text{ítem}, \text{cantidad}, \text{obtener}(\text{ítem}, \text{Promos}(p))))$
descuentoAux(p, ítem, cantidad, promos) \equiv

```

if long(promos) =obs 1
  then prim(promos)
  else if cantMínDePrim(promos) > cantidad
    then descuentoAux(p, ítem, cantidad, fin(promos))
    else if cantMínDeSeg(promos)  $\leq$  cantidad
      then if cantMínDeSeg(promos) > cantMínDePrim(promos)
        then descuentoAux(p, ítem, cantidad, fin(promos))
        else descuentoAux(p, ítem, cantidad, prim(promos) • fin(fin(promos)))
      fi
    else descuentoAux(p, ítem, cantidad, prim(promos) • fin(fin(promos)))
  fi
fi

```

descuentoAux?

stockDe(p, ítem) \equiv obtener(ítem, Stock(p))

Fin TAD