

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		
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
{ $(p \in \text{Puestos}(f) \wedge \text{vende?}(p, \text{ítem})) \wedge_L$
 $\text{estáEnElFestival?}(f, \text{per}) \wedge \text{hayStock?}(p, \text{ítem}, \text{unidades})$ }

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,  $per_2$ , ítem, unidades),  $per_1$ )  $\equiv$  Personas(f)
Compras(ponerPuesto(f, p), per)  $\equiv$  Compras(f, per)
Compras(agregarPersona(f,  $per_2$ ),  $per_1$ )  $\equiv$  if  $per_1 =_{\text{obs}} per_2$  then  $\emptyset$  else Compras(f,  $per_1$ ) fi
Compras(comprarÍtem(f, p,  $per_2$ , ítem, unidades),  $per_1$ )  $\equiv$  if  $per_1 =_{\text{obs}} per_2$ 
then Ag(<p, ítem, unidades>, Compras(f,  $per_1$ ))
else Compras(f,  $per_1$ )
fi

estáEnElFestival?(f, per)  $\equiv$  per  $\in$  Personas(f)
stock(ponerPuesto(f,  $p_2$ ),  $p_1$ , ítem)  $\equiv$  if  $p_1 =_{\text{obs}} p_2$  then stockDe( $p_1$ , ítem) else stock(f,  $p_1$ , ítem) fi
stock(agregarPersona(f, per), p, ítem)  $\equiv$  stock(f, p, ítem)
stock(comprarÍtem(f, p, per,  $ítem_2$ , unidades), p,  $ítem_1$ )  $\equiv$  if  $ítem_1 =_{\text{obs}} ítem_2$ 
then stock(f, p,  $ítem_1$ ) - unidades
else stock(f, p,  $ítem_1$ )
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  $\#(\text{personas}) =_{\text{obs}} 1$  then dameUno(personas)
else if cuántoGastó(f, dameUno(personas))  $\geq$ 
cuántoGastó(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  $=_{\text{obs}} \emptyset$ 
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,  $per_2$ , ítem),  $per_1$ )  $\equiv$  estáEnElFestival?(f,  $per_1$ )
Compras(hackear(f,  $per_2$ , ítem),  $per_1$ )  $\equiv$  if  $per_2 \neq_{\text{obs}} per_1$  then Compras(f,  $per_1$ ) else Compras(f,  $per_1$ ) -
{compraAHackear(f,  $per_1$ , Compras(f,  $per_1$ , ítem), ítem)} fi
stock(hackear(f, per,  $ítem_2$ ), p,  $ítem_1$ )  $\equiv$  if  $ítem_1 =_{\text{obs}} ítem_2 \wedge$  puestoDe(compraAHackear(f, per,  $ítem_1$ ))  $=_{\text{obs}} p$ 
then stock(f, p,  $ítem_1$ ) + cantidadDe(compraAHackear(f, per,  $ítem_1$ ))
else stock(f, p,  $ítem_1$ )
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)  $=_{\text{obs}} ítem \wedge_L$  cuántoGastóEnCompra(f, per, compra)  $=_{\text{obs}}$ 
precio(puestoDe(compra), ítem)  $\times$  cantidadDe(compra)

puestoDe(compra)  $\equiv$   $\pi_1(\text{compra})$ 
ítemDe(compra)  $\equiv$   $\pi_2(\text{compra})$ 
cantidadDe(compra)  $\equiv$   $\pi_3(\text{compra})$ 

```

Fin TAD


```

//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

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

stockDe(p, ítem)  $\equiv \text{obtener}(\text{ítem}, \text{Stock}(p))$ 

```

Fin TAD