

Recuperatorio del trabajo práctico 2: Diseño

Normativa

Límite de entrega: martes 25 de noviembre *hasta las 22:00 hs.* Enviar a algo2.dc@gmail.com

Normas de entrega: Ver “Información sobre la cursada” en el sitio Web de la materia.

(<http://www.dc.uba.ar/materias/aed2/2014/2c/informacion>)

Versión: 1.2 del 22 de noviembre de 2014 (ver RTP2_Changelog)

Enunciado

El objetivo de este trabajo práctico es, en primer lugar, corregir todos los errores marcados en la corrección del TP2, manteniendo el contexto de uso y los órdenes de complejidad requeridos en el enunciado del mismo¹. En segundo lugar, para este recuperatorio se modificó el TAD CIUDADROBÓTICA con respecto al TP original para poder conocer:

- Cuál es la característica que estuvo involucrada en la mayor cantidad de infracciones a lo largo de toda la historia de la ciudad. Más precisamente, decimos que una característica “estuvo involucrada” en una infracción si pertenece al conjunto de características del robot que cometió la infracción. Por ejemplo, si un robot tiene las características {**función_ambulancia**, **modelo_XZ-18**} y comete una infracción, ambas características resultan involucradas.
- Cuántas inspecciones se realizaron en cada estación.

Estos cambios están especificados como son deseados, y debe realizarse el diseño teniendo en cuenta estos agregados. Sugerimos que confíen en la axiomatización más que en el castellano.

Contexto de uso y complejidades requeridas

Se requiere que las operaciones exportadas de los TADs tengan una contraparte en los módulos diseñados. Además, se debe cumplir con los mismos requerimientos de complejidad que en el TP2 original, pero con los siguientes agregados según la modificación del TAD CIUDADROBÓTICA:

- $\text{tagMáslInvolucrado}(c)$ es $\mathcal{O}(1)$.
- $\#inspecciones(e, c)$ es $\mathcal{O}(|e|)$, donde $|e|$ es la longitud del nombre de la estación e .
- $\text{mover}(u, e_2, c)$ es $\mathcal{O}(|e_1| + |e_2| + \log N_{e_1} + \log N_{e_2} + CT_u)$, donde:
 - e_1 es la estación donde se encuentra el robot antes de moverse, y e_2 es la estación a la que se mueve.
 - CT_u es el cardinal del conjunto de tags del robot con u en la ciudad c , es decir, $CT_u = \#(tags(u, c))$

¹Con excepción de la función **mover**

Requisitos y consideraciones

- Todas las operaciones auxiliares deben ser especificadas formalmente según las herramientas vistas en clase. Agregar comentarios necesarios para entender la forma en la cual deben ser utilizadas para su correcto funcionamiento.
- Todos los algoritmos *deben* tener su desarrollo que justifique los órdenes de complejidad. Si algún paso es no trivial pueden hacer notas a continuación del mismo.
- Cuando se formalicen los invariantes y funciones de abstracción, *deben* identificar cada parte de la fórmula del Rep y comentar en castellano lo que describe.
- Tener en cuenta que las complejidades son en peor caso. Soluciones más eficientes serán bien recibidas.
- Tengan en cuenta que hay estructuras que pueden servir para más de una finalidad, sobre todos los contenedores.
- Pueden crear módulos adicionales si así lo necesitan.
- Cuentan con los siguiente TADs y módulos:
 - CHAR que representan los posibles caracteres. Siendo un tipo enumerado de 256 valores. con funciones ord y ord^{-1} para la correspondencia de cada *Char* a *Nat*.
 - STRING como sinónimo de $Vector(Char)$ y su comparación:
 - $<$ • (**in** $s_1 : String$, **in** $s_2 : String$) $\rightarrow bool$ con costo $\mathcal{O}(\min(|s_1|, |s_2|))$.
 - Todos los definidos en el apunte de TADs básicos.
 - Todos los definidos en el apunte de módulos básicos.

Especificación

Aclaraciones de nomenclatura:

- Las características de los robots se llaman *tags*.
- Se usa la notación $\{x_1, x_2, \dots, x_n\}$ para denotar el conjunto:

$$\text{Ag}(x_1, \text{Ag}(x_2, \dots, \text{Ag}(x_n, \emptyset) \dots))$$

- Se respeta la siguiente convención para los nombres de las variables:

e	\mapsto	estaciones
t	\mapsto	tags
ts	\mapsto	conjuntos de tags
u	\mapsto	unidades robóticas (RURs)
us	\mapsto	conjuntos de unidades robóticas
r	\mapsto	restricciones
m	\mapsto	mapas
c	\mapsto	ciudades

TAD LETRA es ENUMERADO ('_', 'A', 'B', ..., 'Z', '0', '1', ..., '9')

TAD STRING es SECU(LETRA)

TAD ESTACIÓN es STRING

TAD TAG es STRING

TAD RUR es NAT

TAD RESTRICCIÓN

exporta todos los generadores, todos los observadores

géneros restricción

igualdad observacional

$$(\forall r1, r2 : \text{restricción}) \left(\begin{array}{l} r1 =_{\text{obs}} r2 \iff \\ (\forall ts : \text{conj}(\text{tag})) \\ \text{verifica?}(ts, r1) =_{\text{obs}} \text{verifica?}(ts, r2) \end{array} \right)$$

observadores básicos

$\text{verifica?} : \text{conj}(\text{tag}) \times \text{restricción} \longrightarrow \text{bool}$

generadores

$\langle \bullet \rangle : \text{tag} \longrightarrow \text{restricción}$

$\bullet \text{ AND } \bullet : \text{restricción} \times \text{restricción} \longrightarrow \text{restricción}$

$\bullet \text{ OR } \bullet : \text{restricción} \times \text{restricción} \longrightarrow \text{restricción}$

$\text{NOT } \bullet : \text{restricción} \longrightarrow \text{restricción}$

axiomas $(\forall t : \text{tag}, \forall ts : \text{conj}(\text{tag}), \forall r, r1, r2 : \text{restricción})$

$\text{verifica?}(ts, \langle t \rangle) \equiv t \in ts$

$\text{verifica?}(ts, r1 \text{ AND } r2) \equiv \text{verifica?}(ts, r1) \wedge \text{verifica?}(ts, r2)$

$\text{verifica?}(ts, r1 \text{ OR } r2) \equiv \text{verifica?}(ts, r1) \vee \text{verifica?}(ts, r2)$

$\text{verifica?}(ts, \text{NOT } r) \equiv \neg \text{verifica?}(ts, r)$

Fin TAD

TAD MAPA

exporta todos los generadores, todos los observadores

géneros mapa

igualdad observacional

$$(\forall m1, m2 : \text{mapa}) \left(\begin{array}{l} m1 =_{\text{obs}} m2 \iff \\ \wedge_L \left(\begin{array}{l} \text{estaciones}(m1) =_{\text{obs}} \text{estaciones}(m2) \\ (\forall e1, e2 : \text{estación}) \\ \left(\begin{array}{l} \{e1, e2\} \subseteq \text{estaciones}(m1) \\ \Rightarrow_L \text{conectadas?}(e1, e2, m1) =_{\text{obs}} \text{conectadas?}(e1, e2, m2) \end{array} \right) \end{array} \right) \\ \wedge_L \left(\begin{array}{l} (\forall e1, e2 : \text{estación}) \\ \left(\begin{array}{l} \{e1, e2\} \subseteq \text{estaciones}(m1) \wedge_L \text{conectadas?}(e1, e2, m1) \\ \Rightarrow_L \text{restricción}(e1, e2, m1) =_{\text{obs}} \text{restricción}(e1, e2, m2) \end{array} \right) \end{array} \right) \end{array} \right)$$

observadores básicos

estaciones	: mapa	$\rightarrow \text{conj}(\text{estación})$
conectadas?	: estación $e1 \times$ estación $e2 \times$ mapa m	$\rightarrow \text{bool}$ $\{\{e1, e2\} \subseteq \text{estaciones}(m)\}$
restricción	: estación $e1 \times$ estación $e2 \times$ mapa m	$\rightarrow \text{restricción}$ $\{\{e1, e2\} \subseteq \text{estaciones}(m) \wedge_L \text{conectadas?}(e1, e2, m)\}$

generadores

vacío	:	$\rightarrow \text{mapa}$
agregar	: estación $e \times$ mapa m	$\rightarrow \text{mapa}$ $\{e \notin \text{estaciones}(m)\}$
conectar	: estación $e1 \times$ estación $e2 \times$ restricción $r \times$ mapa m	$\rightarrow \text{mapa}$ $\{\{e1, e2\} \subseteq \text{estaciones}(m) \wedge_L \neg \text{conectadas?}(e1, e2, m)\}$

axiomas $(\forall m : \text{mapa}, \forall r : \text{restricción}, \forall e, e1, e2, e1', e2' : \text{estación})$

estaciones(vacío)	$\equiv \emptyset$
estaciones(agregar(e, m))	$\equiv \text{Ag}(e, \text{estaciones}(m))$
estaciones(conectar($e1, e2, m$))	$\equiv \text{estaciones}(m)$
conectadas?($e1, e2, \text{agregar}(e, m)$)	$\equiv \neg(e1 = e \wedge e2 = e) \vee_L \text{conectadas?}(e1, e2, m)$
conectadas?($e1, e2, \text{conectar}(e1', e2', r, m)$)	$\equiv \{e1, e2\} = \{e1', e2'\} \vee_L \text{conectadas?}(e1, e2, m)$
restricción($e1, e2, \text{agregar}(e, m)$)	$\equiv \text{restricción}(e1, e2, m)$
restricción($e1, e2, \text{conectar}(e1', e2', r, m)$)	$\equiv \text{if } \{e1, e2\} = \{e1', e2'\} \text{ then } r \text{ else } \text{restricción}(e1, e2, m) \text{ fi}$

Fin TAD**TAD CIUDADROBÓTICA**

exporta todos los generadores, todos los observadores, estaciones, tagMásInvolucrado

géneros ciudad

igualdad observacional

$$(\forall c1, c2 : \text{ciudad}) \left(\begin{array}{l} c1 =_{\text{obs}} c2 \iff \\ \wedge \text{mapa}(c1) =_{\text{obs}} \text{mapa}(c2) \wedge \text{robots}(c1) =_{\text{obs}} \text{robots}(c2) \\ \wedge \text{tagsHistóricos}(c1) =_{\text{obs}} \text{tagsHistóricos}(c2) \\ \wedge_L \left(\begin{array}{l} (\forall e : \text{estación}) \\ \left(\begin{array}{l} e \in \text{estaciones}(c1) \\ \Rightarrow_L \# \text{inspecciones}(e, c1) =_{\text{obs}} \# \text{inspecciones}(e, c2) \end{array} \right) \end{array} \right) \\ \wedge \left(\begin{array}{l} (\forall t : \text{tag}) \\ \left(\begin{array}{l} t \in \text{tagsHistóricos}(c1) \\ \Rightarrow_L \# \text{infraccionesXtag}(t, c1) =_{\text{obs}} \# \text{infraccionesXtag}(t, c2) \end{array} \right) \end{array} \right) \\ \wedge \left(\begin{array}{l} (\forall u : \text{rur}) (u \in \text{robots}(c1) \Rightarrow_L \\ \left(\begin{array}{l} \text{estación}(u, c1) =_{\text{obs}} \text{estación}(u, c2) \\ \wedge \text{tags}(u, c1) =_{\text{obs}} \text{tags}(u, c2) \\ \wedge \# \text{infracciones}(u, c1) =_{\text{obs}} \# \text{infracciones}(u, c2) \end{array} \right) \end{array} \right) \end{array} \right)$$

observadores básicos

próximoRUR	: ciudad c	\rightarrow rur	
mapa	: ciudad c	\rightarrow mapa	
robots	: ciudad c	\rightarrow conj(rur)	
estación	: rur $u \times$ ciudad c	\rightarrow estación	$\{u \in \text{robots}(c)\}$
tags	: rur $u \times$ ciudad c	\rightarrow conj(tag)	$\{u \in \text{robots}(c)\}$
#infracciones	: rur $u \times$ ciudad c	\rightarrow nat	$\{u \in \text{robots}(c)\}$
#infraccionesXtag	: tag $t \times$ ciudad c	\rightarrow nat	$\{t \in \text{tagsHistóricos}(c)\}$
#inspecciones	: estación $e \times$ ciudad c	\rightarrow nat	$\{e \in \text{estaciones}(c)\}$
tagsHistóricos	: ciudad c	\rightarrow conj(tag)	

generadores

crear	: mapa m	\rightarrow ciudad	
entrar	: conj(tag) $ts \times$ estación $e \times$ ciudad c	\rightarrow ciudad	$\{e \in \text{estaciones}(c)\}$
mover	: rur $u \times$ estación $e \times$ ciudad c	\rightarrow ciudad	
	$\{u \in \text{robots}(c) \wedge e \in \text{estaciones}(c) \wedge_{\text{L}} \text{conectadas?}(\text{estación}(u, c), e, \text{mapa}(c))\}$		
inspección	: estación $e \times$ ciudad c	\rightarrow ciudad	$\{e \in \text{estaciones}(c)\}$

otras operaciones

estaciones	: ciudad c	\rightarrow conj(estación)	
robotsEn	: estación $e \times$ ciudad c	\rightarrow conj(rur)	$\{e \in \text{estaciones}(c)\}$
filtrarRobotsEn	: conj(rur) $us \times$ estación $e \times$ ciudad c	\rightarrow conj(rur)	$\{e \in \text{estaciones}(c) \wedge us \subseteq \text{robots}(c)\}$
elMásInfractor	: conj(rur) $us \times$ ciudad c	\rightarrow rur	$\{-\emptyset?(us) \wedge us \subseteq \text{robots}(c)\}$
másInfractor	: rur $u1 \times$ rur $u2 \times$ ciudad c	\rightarrow rur	$\{\{u1, u2\} \subseteq \text{robots}(c)\}$
tagMásInvolucrado	: ciudad c	\rightarrow tag	$\{\exists t \in \text{tagsHistóricos}(c) \wedge_{\text{L}} \# \text{infraccionesXtag}(t, c) > 0\}$
filtrarTagMásInvolucrado	: ciudad $c \times$ conj(tag) $ts \times$ tag t	\rightarrow tag	

axiomas $(\forall c : \text{ciudad}, \forall m : \text{mapa}, \forall u, u' : \text{rur}, \forall ts : \text{conj}(\text{tag}), \forall us : \text{conj}(\text{rur}))$

próximoRUR(crear(m))	$\equiv 0$
próximoRUR(entrar(ts, e, c))	$\equiv \text{próximoRUR}(c) + 1$
próximoRUR(mover(u, e, c))	$\equiv \text{próximoRUR}(c)$
próximoRUR(inspección(e, c))	$\equiv \text{próximoRUR}(c)$
mapa(crear(m))	$\equiv m$
mapa(entrar(ts, e, c))	$\equiv \text{mapa}(c)$
mapa(mover(u, e, c))	$\equiv \text{mapa}(c)$
mapa(inspección(e, c))	$\equiv \text{mapa}(c)$
robots(crear(m))	$\equiv \emptyset$
robots(entrar(ts, e, c))	$\equiv \text{Ag}(\text{próximoRUR}(c), \text{robots}(c))$
robots(mover(u, e, c))	$\equiv \text{robots}(c)$
robots(inspección(e, c))	$\equiv \text{robots}(e, c) \setminus \text{if } \emptyset?(\text{robotsEn}(e, c))$ $\vee_{\text{L}} \# \text{infracciones}(\text{elMásInfractor}(\text{robotsEn}(e, c), c)) = 0 \text{ then } \emptyset$ else $\{\text{elMásInfractor}(\text{robotsEn}(e, c), c)\}$ fi
estación($u, \text{entrar}(ts, e, c)$)	$\equiv \text{if } u = \text{próximoRUR}(c) \text{ then } e \text{ else estación}(u, c) \text{ fi}$
estación($u, \text{mover}(u', e, c)$)	$\equiv \text{if } u = u' \text{ then } e \text{ else estación}(u, c) \text{ fi}$
estación($u, \text{inspección}(e, c)$)	$\equiv \text{estación}(u, c)$
tags($u, \text{entrar}(ts, e, c)$)	$\equiv \text{if } u = \text{próximoRUR}(c) \text{ then } ts \text{ else tags}(u, c) \text{ fi}$
tags($u, \text{mover}(u', e, c)$)	$\equiv \text{tags}(u, c)$

```

tags(u, inspección(e, c)) ≡ tags(u, c)

#infracciones(u, entrar(ts, e, c)) ≡ if u = próximoRUR(c) then 0 else #infracciones(u, c) fi
#infracciones(u, mover(u', e, c)) ≡ if u = u' then
    β(¬ (verifica?(tags(u, c),
        restricción(estación(u, c), e, mapa(c))))))
    else
        0
    fi + #infracciones(u, c)
#infracciones(u, inspección(u', e, c)) ≡ #infracciones(u, c)

#infraccionesXtag(t, entrar(ts, e, c)) ≡ if t ∈ ts ∧ ¬(t ∈ tagsHistóricos(c)) then
    0
    else
        #infraccionesXtag(t, c)
    fi
#infraccionesXtag(t, mover(u, e, c)) ≡ if t ∈ tags(u, c) then
    β(¬ (verifica?(tags(u, c),
        restricción(estación(u, c), e, mapa(c))))))
    else
        0
    fi + #infraccionesXtag(t, c)
#infraccionesXtag(t, inspección(u, e, c)) ≡ #infraccionesXtag(t, c)

#inspecciones(e, crear(m)) ≡ 0
#inspecciones(e, entrar(ts, e', c)) ≡ #inspecciones(e, c)
#inspecciones(e, mover(u, e', c)) ≡ #inspecciones(e, c)
#inspecciones(e, inspección(u, e', c)) ≡ if e = e' then 1 else 0 fi + #inspecciones(e, c)

tagsHistóricos(crear(m)) ≡ ∅
tagsHistóricos(entrar(ts, e, c)) ≡ tagsHistóricos(c) ∪ ts
tagsHistóricos(mover(u, e, c)) ≡ tagsHistóricos(c)
tagsHistóricos(inspección(u, e, c)) ≡ tagsHistóricos(c)

estaciones(c) ≡ estaciones(mapa(c))
robotsEn(e, c) ≡ filtrarRobotsEn(robots(c), e, c)
filtrarRobotsEn(us, e, c) ≡ if ∅?(us) then
    ∅
    else
        if estación(dameUno(us), c) = e then {dameUno(us)} else ∅ fi ∪
        filtrarRobotsEn(sinUno(us), e, c)
    fi
elMásInfractor(us, c) ≡ if #(us) = 1 then
    dameUno(us)
    else
        másInfractor(dameUno(us),
            elMásInfractor(sinUno(us, c)),
            c)
    fi
másInfractor(u1, u2, c) ≡ if #infracciones(u1, c) > #infracciones(u2, c)
    ∨ (#infracciones(u1, c) = #infracciones(u2, c) ∧ u1 < u2)
    then
        u1
    else
        u2
    fi
tagMásInvolucrado(c) ≡ filtrarTagMásInvolucrado(c,
    sinUno(tagsHistóricos(c)),
    dameUno(tagsHistóricos(c)))

```

```
filtrarTagMásInvolucrado( $c, ts, t$ )  $\equiv$  if  $\emptyset?(ts)$  then  
     $t$   
    else  
        if  $\#infraccionesXtag(dameUno(ts), c) > \#infraccionesXtag(t, c)$   
             $\vee$   
             $(\#infraccionesXtag(dameUno(ts), c) = \#infraccionesXtag(t, c)$   
                 $\wedge dameUno(ts) < t)$   
            then  
                filtrarTagMásInvolucrado( $c, sinUno(ts), dameUno(ts)$ )  
            else  
                filtrarTagMásInvolucrado( $c, sinUno(ts), t$ )  
        fi  
    fi  
  
Fin TAD
```