Clase de TADs avanzada

Clase práctica AED II

29 de Agosto de 2014

La Clase

Previously on Algo2...

- Clase 1:
  - Definimos observadores e igualdad observacional
  - Definimos generadores
  - Axiomatizamos
  - Definimos y axiomatizamos otras operaciones, en caso de haberlas.
  - Aprendimos que los observadores deben ser un conjunto minimal
- Clase 2: Ejemplos de comportamiento automático
- Clase 3: Inducción estructural (no veremos nada relacionado hoy)

Objetivos de hoy

- Ver errores comúnes de modelado y de uso de los TADs básicos.
- Presentar la técnica de modelado en varios niveles de TADs. No tiene porqué ser algo novedoso.
- Analizar cuándo y porqué sirve esta técnica así como el modo de usarla.

TADs Básicos

- Bool
- Nat
- Tupla($\alpha_1, \ldots, \alpha_n$)
- Secuencia(α)
- Conjunto(α)
- Multiconjunto(α)
- Arreglo dimensionable(α)
- Pila(α)
- Cola(α)
- Árbol binario(α)
- Diccionario(clave, significado)
- Cola de prioridad(α)

Enunciado

...En una biblioteca se guarda información sobre libros y sus autores...

TAD Biblioteca

géneros biblio

...

observadores básicos

libros : biblio $\longrightarrow$ secu(libro)

autores : biblio $\longrightarrow$ secu(autor)

...

Fin TAD

Enunciado

...En una biblioteca se guarda información sobre libros y sus autores...

TAD Biblioteca

géneros biblio

...

observadores básicos

libros : biblio $\longrightarrow$ secu(libro)

autores : biblio $\longrightarrow$ secu(autor)

...

Fin TAD

¡Mal! Secu(α) determina un orden. No necesitamos definir un orden para los libros y los autores .

Versión correcta:

Enunciado

...En una biblioteca se guarda información sobre libros y sus autores...

TAD Biblioteca

géneros biblio

...

observadores básicos

libros : biblio $\longrightarrow$ conj(libro)

autores : biblio $\longrightarrow$ conj(autor)

...

Fin TAD

Enunciado

Se desea especificar el comportamiento de una comisión del Congreso. Los comisión tiene al menos ocho miembros. Los expedientes son ingresados a la comisión por cualquiera de sus miembros(..)

TAD Comision

Persona es Nat

Expediente es Nat

géneros comision

observadores básicos

miembros : comision $\longrightarrow$ conj(persona)

...

generadores

iniciar : conj(pers) *ms* $\times$ conj(tupla($exp, pers$)) *ingresados* $\longrightarrow$ comision

$$\left\{ \begin{array}{l} \#(ms) \geq 8 \wedge \forall e\text{: tupla(exp, pers) } e \in ingresados \Rightarrow \pi_2(e) \in \\ ms \end{array} \right\}$$

...

Fin TAD

Enunciado

Se desea especificar el comportamiento de una comisión del Congreso. Los comisión tiene al menos ocho miembros. Los expedientes son ingresados a la comisión por cualquiera de sus miembros(..)

TAD Comision

Persona es Nat

Expediente es Nat

géneros comision

observadores básicos

miembros : comision $\longrightarrow$ conj(persona)

...

generadores

iniciar : conj(pers) ms $\times$ conj(tupla($exp, pers$)) $ingresados$ $\longrightarrow$ comision

$$\left\{ \begin{matrix} \#(ms) \geq 8 \wedge \forall e\text{: tupla(exp, pers) } e \in ingresados \Rightarrow \pi_2(e) \in \\ ms \end{matrix} \right\}$$

...

Fin TAD

No modela el comportamiento de ingresar un expediente al sistema.

Enunciado

Se desea especificar el comportamiento de una comisión del Congreso. Los comisión tiene al menos ocho miembros. Los expedientes son ingresados a la comisión por cualquiera de sus miembros(..)

TAD Comision

Persona es Nat

Expediente es Nat

géneros comision

observadores básicos

miembros : comision $\longrightarrow$ conj(persona)

...

generadores

iniciar : conj(pers) *ms* $\longrightarrow$ comision

$\{\#(ms) \geq 8\}$

ingresarExp : exp e $\times$ pers p $\times$ comision c $\longrightarrow$ comision

$\{p \in miembros(c)\}$

...

Enunciado: Ventanilla de atención al cliente

...Cuando llega un cliente se identifica por su DNI. Si la ventanilla está libre es atendido inmediatamente, sino queda esperando. Cuando el cliente que está siendo atendido se retira se pasa a atender al siguiente en la fila.

TAD Ventanilla

géneros ventanilla

observadores básicos

estaLibre?	:	ventanilla	$\longrightarrow$	bool	
siendoAtendido	:	ventanilla v	$\longrightarrow$	DNI	$\{\neg estaLibre?(v)\}$
esperando	:	ventanilla	$\longrightarrow$	cola(DNI)	

generadores

abrir	:		$\longrightarrow$	ventanilla	
llegaCliente	:	DNI c $\times$ ventanilla v	$\longrightarrow$	ventanilla	$\{...\}$
seLibera	:	ventanilla v	$\longrightarrow$	ventanilla	
					$\{\neg estaLibre?(v)\}$
atenderSiguiente	:	ventanilla v	$\longrightarrow$	ventanilla	
					$\{estaLibre?(v) \wedge \neg vacia?(esperando(v))\}$

Fin TAD

TAD Ventanilla

géneros ventanilla

observadores básicos

estaLibre?	:	ventanilla	$\longrightarrow$	bool	
siendoAtendido	:	ventanilla v	$\longrightarrow$	DNI	$\{\neg estaLibre?(v)\}$
esperando	:	ventanilla	$\longrightarrow$	cola(DNI)	

generadores

abrir	:		$\longrightarrow$	ventanilla	
llegaCliente	:	DNI c $\times$ ventanilla v	$\longrightarrow$	ventanilla	$\{...\}$
seLibera	:	ventanilla v	$\longrightarrow$	ventanilla	
					$\{\neg estaLibre?(v)\}$
atenderSiguiente	:	ventanilla v	$\longrightarrow$	ventanilla	
					$\{estaLibre?(v) \wedge \neg vacia?(esperando(v))\}$

Fin TAD

Atender a un cliente debería ser un comportamiento automático que se produce cuando se libera la ventanilla.

TAD Ventanilla

géneros ventanilla

observadores básicos

estaLibre?	: ventanilla	$\longrightarrow$	bool	
siendoAtendido	: ventanilla v	$\longrightarrow$	DNI	$\{\neg estaLibre?(v)\}$
esperando	: ventanilla	$\longrightarrow$	cola(DNI)	

generadores

abrir	:	$\longrightarrow$	ventanilla	
llegaCliente	: DNI c $\times$ ventanilla v	$\longrightarrow$	ventanilla	$\{...\}$
seLibera	: ventanilla v	$\longrightarrow$	ventanilla	

$\{\neg estaLibre?(v)\}$

Fin TAD

TAD Ventanilla

géneros ventanilla

observadores básicos

estaLibre?	: ventanilla	$\longrightarrow$	bool	
siendoAtendido	: ventanilla v	$\longrightarrow$	DNI	$\{\neg estaLibre?(v)\}$
esperando	: ventanilla	$\longrightarrow$	cola(DNI)	

generadores

abrir	:	$\longrightarrow$	ventanilla	
llegaCliente	: DNI c $\times$ ventanilla v	$\longrightarrow$	ventanilla	$\{...\}$
seLibera	: ventanilla v	$\longrightarrow$	ventanilla	

$\{\neg estaLibre?(v)\}$

Fin TAD

Bien. Los eventos desencadenados por seLibera se reflejan en la axiomatización de los observadores.

¿Y este otro modelo?

TAD Ventanilla

géneros ventanilla

observadores básicos

estaLibre?	:	ventanilla	$\longrightarrow$	bool	
siendoAtendido	:	ventanilla v	$\longrightarrow$	DNI	$\{\neg estaLibre?(v)\}$
esperando	:	ventanilla	$\longrightarrow$	cola(DNI)	

generadores

abrir	:		$\longrightarrow$	ventanilla	
llegaCliente	:	DNI c $\times$ ventanilla v	$\longrightarrow$	ventanilla	$\{...\}$

otras operaciones

seLibera	:	ventanilla v	$\longrightarrow$	ventanilla	

$\{\neg estaLibre?(v)\}$

Fin TAD

¿Y este otro modelo?

TAD Ventanilla

géneros ventanilla

observadores básicos

estaLibre?	:	ventanilla	$\longrightarrow$	bool	
siendoAtendido	:	ventanilla v	$\longrightarrow$	DNI	$\{\neg estaLibre?(v)\}$
esperando	:	ventanilla	$\longrightarrow$	cola(DNI)	

generadores

abrir	:		$\longrightarrow$	ventanilla	
llegaCliente	:	DNI c $\times$ ventanilla v	$\longrightarrow$	ventanilla	$\{...\}$

otras operaciones

seLibera	:	ventanilla v	$\longrightarrow$	ventanilla	

$\{\neg estaLibre?(v)\}$

Fin TAD

Es discutible, pero no necesariamente está mal. Acá no hay comportamiento automático, esa axiomatización que estaba en los observadores ahora está en la Otra Operación. No hay registro de las veces que se liberó una ventanilla. Si quedan dudas sobre la diferencia entre este y el anterior CONSULTAR!

TADs multinivel: La técnica

Hablamos de TADs en varios niveles cuando definimos TADs que usan otros TADs internamente. En general, hablamos de TADs en múltiples niveles. Esto se hace principalmente para:

- Agrupar comportamientos y entidades.
- En general sirve para separar los aspectos estáticos y dinámicos del problema.
- Simplificar TADs mediante el uso de otros y sus propiedades.

En resumen, hacer que el TAD sea lo más simple y conciso posible, expresando todo lo que debe expresar.

Competencias matemáticas

En la facultad de exactas se decidieron organizar competencias de acertijos matemáticos. Las competencias se organizarían periódicamente. Los jugadores que vayan a participar deben inscribirse previo al inicio de la temporada. Los datos de la inscripción son nombre y carrera y no puede haber dos jugadores con el mismo nombre. Al se publican los acertijos. Durante el transcurso de la misma nuevos acertijos pueden ser publicados. Estos se identifican con un número y no existe distinción relacionada a cuándo fueron publicados. Los jugadores deberán encontrar respuesta a la mayor cantidad de acertijos posible.

TAD jugador es tupla(nombre, carrera)

TAD Temporada

géneros temp

observadores básicos

jugadores : temp $\longrightarrow$ conj(jugador)

acertados : temp t $\times$ jugador j $\longrightarrow$ conj(nat) $\{j \in jugadores(t)\}$

acertijos : temp $\longrightarrow$ conj(nat)

generadores

initTemp : conj(jugador) js $\times$ conj(nat) as $\longrightarrow$ temp

$\{(\forall j, j'$: jugador$)(\{j, j'\} \subseteq js \Rightarrow \neg(\prod_1(j) = \prod_1(j') \wedge \prod_2(j) \neq \prod_2(j')))\}$

anotarAcertijo : temp t $\times$ nat n $\times$ jugador j $\longrightarrow$ temp

$\{j \in jugadores(t) \wedge_L n \in (acertijos(t) - acertados(t, j))\}$

agregarAcertijo : temp t $\times$ nat n $\longrightarrow$ temp $\{\neg\, n \in acertijos(t)\}$

Fin TAD

axiomas

jugadores(initTemp(js, as)) $\equiv$ js
jugadores(agregarAcertijo(t, n)) $\equiv$ jugadores(t)
jugadores(anotarAcertijo(t, n, j)) $\equiv$ jugadores(t)

acertados(initTemp(js, as), n') $\equiv$ $\emptyset$
acertados(agregarAcertijo(t, n), n') $\equiv$ acertados(t, n')
acertados(anotarAcertijo(t, n, j), n') $\equiv$ **if** n' = j
then Ag(n, acertados(t, n'))
else acertados(t, n')
fi
acertijos(initTemp(js, as)) $\equiv$ as
acertijos(agregarAcertijo(t, n)) $\equiv$ Ag(n, acertijos(t))
acertijos(anotarAcertijo(t, n, j)) $\equiv$ acertijos(t)

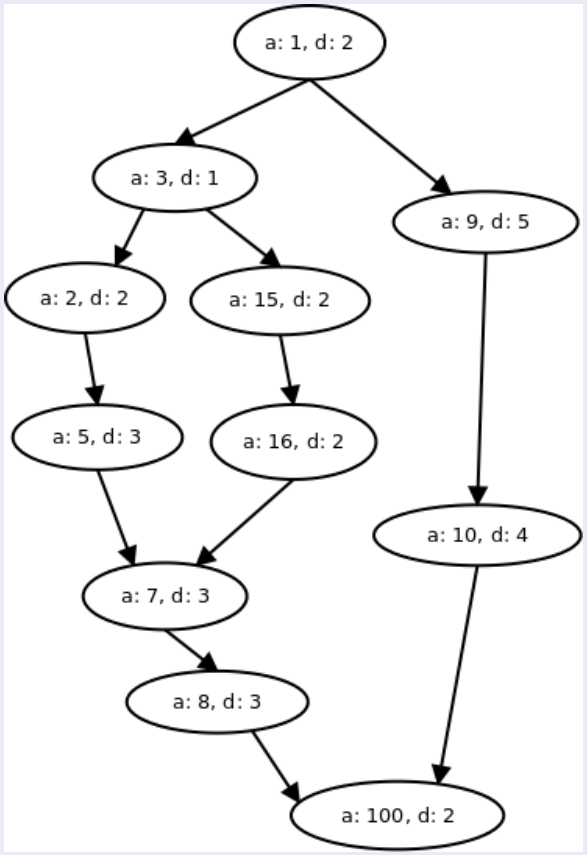
La facultad decidió agregar algunos condimentos a las competencias. Los acertijos son categorizados con un número de complejidad del 1 al 5 y los mismos se organizan de forma laberíntica. Los acertijos están fijos desde el inicio de la competencia.

El objetivo de la competencia es ser el primero en atravesar el laberinto (ver Cuadro). Todos los jugadores comienzan con un acertijo inicial. Al resolverlo tienen la posibilidad de avanzar en el laberinto. Cuando se resuelve un acertijo el jugador obtiene acceso a uno o más acertijos nuevos. No obstante, el jugador debe elegir con cuál de estos nuevos acertijos quiere enfrentarse. Esta decisión lo restringe a un camino en el laberinto. Todos los caminos posibles terminan en un acertijo final. No existen ciclos en el laberinto.

Los jugadores tienen completo conocimiento de las relaciones entre los acertijos y sus dificultades. La temporada solo termina al ser resuelto el acertijo final y el ganador es quién lo haga.

El laberinto de acertijos requerido para una temporada tiene ciertas restricciones:

- Hay un único acertijo final y un único acertijo inicial
- Siguiendo un camino del laberinto no puedo llegar a un acertijo ya resuelto



Cuadro : Ejemplo de un laberinto de acertijos

Se pide modificar el TAD Temporada para que maneje la nueva información administrativa. Se debe conocer los acertijos y la relación entre ellos, así como sus dificultades. Además debe mantener la información sobre los jugadores, en qué acertijo se encuentran y cuáles resolvieron, e imponer las restricciones del laberinto a la hora de moverse por él. La temporada debe finalizar cuando un jugador resuelve el acertijo final, luego de esto el jugador pasa a ser el ganador de la temporada y ningún otro jugador puede resolver acertijos.

TAD jugador es tupla(nombre, carrera)

TAD Laberinto

géneros lab

observadores básicos

acertijos : lab $\longrightarrow$ conj(nat)

dificultad : lab l $\times$ nat a $\longrightarrow$ nat $\{a \in acertijos(l)\}$

opciones : lab l $\times$ nat a $\longrightarrow$ conj(nat) $\{a \in acertijos(l)\}$

generadores

nuevoLab : nat a $\times$ nat d $\longrightarrow$ lab $\{1 \leq d \leq 5\}$

agAcertijo : lab l $\times$ nat acj $\times$ nat dif $\times$ conj(nat) prev $\longrightarrow$ lab

$\{prev \subseteq acertijos(l) \wedge 1 \leq dif \leq 5 \wedge \neg\, acj \in acertijos(l) \wedge \#(prev) \geq 1\}$

otras operaciones

acertijoInicial : lab $\longrightarrow$ nat

acertijosFinales : lab $\longrightarrow$ conj(nat)

filtrarAcertijosFinales : lab l $\times$ conj(nat) as $\longrightarrow$ conj(nat) $\{as \subseteq acertijos(l)\}$

Fin TAD

axiomas

acertijos(nuevoLab(a, d)) $\equiv$ Ag(a, $\emptyset$)
acertijos(agAcertijo(l, acj, dif, prev)) $\equiv$ Ag(acj, acertijos(l))

dificultad(nuevoLab(a, d), a') $\equiv$ d
dificultad(agAcertijo(l, acj, dif, prev), a') $\equiv$ **if** acj $=$ a' **then** dif **else** dificultad(l, a') **fi**

opciones(nuevoLab(a, d), a') $\equiv$ $\emptyset$
opciones(agAcertijo(l, acj, dif, prev), a') $\equiv$ **if** a' $\in$ prev **then** Ag(acj, $\emptyset$) **else** $\emptyset$ **fi**
$\cup$
if a' $\in$ acertijos(l) **then** opciones(l, a') **else** $\emptyset$ **fi**

acertijoInicial(nuevoLab(a,d)) $\equiv$ a
acertijoInicial(agAcertijo(l, acj, dif, prev)) $\equiv$ acertijoInicial(l)

acertijosFinales(l) $\equiv$ filtrarAcertijosFinales(l, acertijos(l))

TAD Temporada

géneros temp

observadores básicos

jugadores : temp $\longrightarrow$ conj(jugador)

acertados : temp t $\times$ jugador j $\longrightarrow$ conj(nat) $\{j \in jugadores(t)\}$

actual : temp t $\times$ jugador j $\longrightarrow$ nat $\{j \in jugadores(t)\}$

lab : temp $\longrightarrow$ lab

generadores

initTemp : conj(jugador) js $\times$ lab l $\longrightarrow$ temp

$$\left\{\begin{array}{l}(\forall j, j' : \text{jugador})(\{j, j'\} \subseteq js \Rightarrow (\prod_1(j) = \prod_1(j') \wedge \prod_2(j) \neq \prod_2(j'))) \wedge \\ \#(\text{acertijosFinales}(l)) = 1\end{array}\right\}$$

anotarYproxAcertijo : temp t $\times$ jugador j $\times$ nat elec $\longrightarrow$ temp

$$\left\{\begin{array}{l}j \in jugadores(t) \wedge_L \neg\ actual(t, j) \in acertijosFinales(laberinto(t)) \wedge elec \\ \in opciones(laberinto(t), actual(t, j)) \wedge \neg\ finalizada(t)\end{array}\right\}$$

anotarAcertijoFinal : temp t $\times$ jugador j $\longrightarrow$ temp

$$\left\{\begin{array}{l}j \in jugadores(t) \wedge actual(t, j) \in acertijosFinales(laberinto(t)) \wedge \neg\ finali- \\ zada(t)\end{array}\right\}$$

otras operaciones

finalizada : temp $\longrightarrow$ bool

Fin TAD

axiomas

jugadores(initTemp(js, l)) $\equiv$ js

jugadores(anotarYproxAcertijo(t, n, j)) $\equiv$ jugadores(t)

jugadores(anotarAcertijoFinal(t, j)) $\equiv$ jugadores(t)

acertados(initTemp(js, l), j') $\equiv$ $\emptyset$

acertados(anotarYproxAcertijo(t, j, elec), j') $\equiv$ **if** j' = j **then**
Ag(actual(t, j), acertados(t, j'))
else
acertados(t, j')
fi

acertados(anotarAcertijoFinal(t, j), j') $\equiv$ **if** j' = j **then**
Ag(actual(t, j), acertados(t, j'))
else
acertados(t, j')
fi

actual(initTemp(js, l), j') $\equiv$ acertijoInicial(l)

actual(anotarYproxAcertijo(t, j, elec), j') $\equiv$ **if** j = j'
then elec
else actual(t, j')
fi

actual(anotarAcertijoFinal(t, j), j') $\equiv$ actual(t, j')

axiomas

$lab(initTemp(js, l)) \equiv l$
$lab(anotarYproxAcertijo(t, j, elec)) \equiv lab(t)$
$lab(anotarAcertijoFinal(t, j)) \equiv lab(t)$

$finalizada(initTemp(js, l)) \equiv false$
$finalizada(anotarYproxAcertijo(t, j, elec)) \equiv false$
$finalizada(anotarAcertijoFinal(t, j)) \equiv true$

Axiomas de los acertijos finales del Laberinto

acertijosFinales(l) $\equiv$ filtrarAcertijosFinales(l, acertijos(l))

filtrarAcertijosFinales(l, as) $\equiv$ **if** $\emptyset$?(as)
then as
else(
if $\emptyset$?(opciones(dameUno(as), l))
then Ag(dameUno(as),$\emptyset$)
else $\emptyset$
fi
$\cup$ filtrarAcertijosFinales(l,sinUno(as))
)
fi