

TADs en 2 niveles

Clase práctica

Francisco “Pachi” Gómez Fernández

Algoritmos y Estructuras de Datos II

19 de agosto de 2015

Poniéndonos en contexto

La práctica hasta ahora...

- Clase 1:
 - ▶ Definimos observadores e igualdad observacional.
 - ▶ Definimos generadores.
 - ▶ Axiomatizamos recursivamente.
 - ▶ Definimos y axiomatizamos otras operaciones, en caso de haberlas.
- Clase 2: Ejemplos de comportamiento automático.

La clase de hoy

- Presentar técnicas de modelado con TADs
- Introducir buenas prácticas de especificación
- Analizar críticamente cuándo y por qué conviene aplicar estas técnicas

Con qué contamos al especificar?

- TADs básicos
- Apunte de TADs
- Ejercicios ya resueltos (y chequeados!) de la práctica

Ejercicio 1

Competencia de acertijos

En la facultad de exactas se decidieron organizar competencias de acertijos matemáticos. Nos fue encargada la tarea de especificar mediante TADs, como funciona el sistema informático que maneja la parte administrativa del mismo. Las competencias se organizan 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 comenzar la temporada se cierran las inscripciones a la competencia y 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 cuando fueron publicados. Los jugadores deberán encontrar respuesta a la mayor cantidad de acertijos posible.

Se pide especificar un TAD que modele el comportamiento de una temporada de esta competencia. Como solo nos incumben las cuestiones administrativas, solo nos importará registrar cuando un jugador acierta un acertijo, no así los intentos fallidos. Se pide además conocer en cada momento la cantidad de acertijos acertados por jugador.

axiomas

$\text{jugadores}(\text{nuevaTemp}) \equiv \emptyset$
 $\text{jugadores}(\text{inscribirJugador}(n, c, t)) \equiv \text{Ag}(n, \text{jugadores}(t))$
 $\text{jugadores}(\text{initTemp}(t)) \equiv \text{jugadores}(t)$
 $\text{jugadores}(\text{agregarAcertijo}(t, n)) \equiv \text{jugadores}(t)$
 $\text{jugadores}(\text{anotarAcertijo}(t, n, j)) \equiv \text{jugadores}(t)$

$\text{acertados}(\text{initTemp}(t), n') \equiv \emptyset$
 $\text{acertados}(\text{agregarAcertijo}(t, n), n') \equiv \text{acertados}(t, n')$
 $\text{acertados}(\text{anotarAcertijo}(t, n, j), n') \equiv (\text{if } n' = j \text{ then } \text{Ag}(n, \emptyset) \text{ else } \emptyset \text{ fi}) \cup \text{acertados}(t, n')$

$\text{iniciada}(\text{nuevaTemp}) \equiv \text{false}$
 $\text{iniciada}(\text{inscribirJugador}(n, c, t)) \equiv \text{false}$
 $\text{iniciada}(\text{initTemp}(t)) \equiv \text{true}$
 $\text{iniciada}(\text{agregarAcertijo}(t, n)) \equiv \text{iniciada}(t)$
 $\text{iniciada}(\text{anotarAcertijo}(t, n, j)) \equiv \text{true}$

axiomas

$\text{acertijos}(\text{nuevaTemp}) \equiv \emptyset$

$\text{acertijos}(\text{inscribirJugador}(n, c, t)) \equiv \text{acertijos}(t)$

$\text{acertijos}(\text{initTemp}(t)) \equiv \text{acertijos}(t)$

$\text{acertijos}(\text{agregarAcertijo}(t, n)) \equiv \text{Ag}(n, \text{acertijos}(t))$

$\text{acertijos}(\text{anotarAcertijo}(t, n, j)) \equiv \text{acertijos}(t)$

$\text{carrera}(\text{inscribirJugador}(n, c, t), n') \equiv \text{if } n = n' \text{ then } c \text{ else } \text{carrera}(t, n') \text{ fi}$

$\text{carrera}(\text{initTemp}(t), n') \equiv \text{carrera}(t, n')$

$\text{carrera}(\text{agregarAcertijo}(t, n), n') \equiv \text{carrera}(t, n')$

$\text{carrera}(\text{anotarAcertijo}(t, n, j), n') \equiv \text{carrera}(t, n')$

Critiquemos la solución...

- Cuál es la relación entre nombre y carrera?
- Cómo ven un ejemplo de uso del TAD? Agrego jugadores, acertijos, etc, INICIO, resuelvo.
 - ▶ El TAD modela dos momentos o etapas (una tiene que terminar para que empiece la otra), en las que se realizan mayormente operaciones diferentes (con la excepción de agregarAcertijo).
 - ▶ Prestemos un poco de atención a la primera etapa. De qué otra forma se podría especificar?
- Qué otras se podrían modificar o mejorar?

axiomas

$$\text{jugadores}(\text{initTemp}(\text{js}, \text{as})) \equiv \text{js}$$
$$\text{jugadores}(\text{agregarAcertijo}(t, n)) \equiv \text{jugadores}(t)$$
$$\text{jugadores}(\text{anotarAcertijo}(t, n, j)) \equiv \text{jugadores}(t)$$
$$\text{acertados}(\text{initTemp}(\text{js}, \text{as}), n') \equiv \emptyset$$
$$\text{acertados}(\text{agregarAcertijo}(t, n), n') \equiv \text{acertados}(t, n')$$

```

acertados(anotarAcertijo(t, n, j), n')  $\equiv$  if n' = j
                                     then Ag(n, acertados(t, n'))
                                     else acertados(t, n')
                                     fi

```

$$\text{acertijos}(\text{initTemp}(\text{js}, \text{as})) \equiv \text{as}$$
$$\text{acertijos}(\text{agregarAcertijo}(t, n)) \equiv \text{Ag}(n, \text{acertijos}(t))$$
$$\text{acertijos}(\text{anotarAcertijo}(t, n, j)) \equiv \text{acertijos}(t)$$

Critiquemos la solución... (otra vez!)

- Cuál es la relación entre nombre y carrera dentro del conjunto de tuplas?
- Estamos haciendo un diccionario con un conjunto de tuplas!
- Se podría encapsular en **otro** TAD ?

Algunas conclusiones hasta el momento

- Usar TADs básicos puede simplificar bastante el modelado del problema pero...
- Uno a veces en pos de usar los TADs básicos tiende a “forzar” su uso en donde no encajan.
- Moraleja: usar los TADs básicos a consciencia.
- Una especificación tiene que ser clara, sino no sirve.
- **Iterar la solución.** Primero proponer una solución y luego mejorarla, teniendo en cuenta las buenas prácticas de especificación.

Ejercicio 2

A pesar del éxito que fueron las competencias matemáticas y su excelente organización, la facultad decidió condimentar las mismas con un nuevo sistema. En esta nueva competencia, los acertijos además son categorizados con un número de complejidad del 1 al 5 y los mismos se organizan de forma laberíntica. En este caso, los acertijos están fijos desde el inicio de la temporada.

El objetivo de la competencia es ser el primero en atravesar el laberinto (ver ejemplo). Todos los jugadores comienzan en un acertijo inicial. Al resolverlo tienen la posibilidad de avanzar en el laberinto. Cuando un jugador resuelve un acertijo 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.

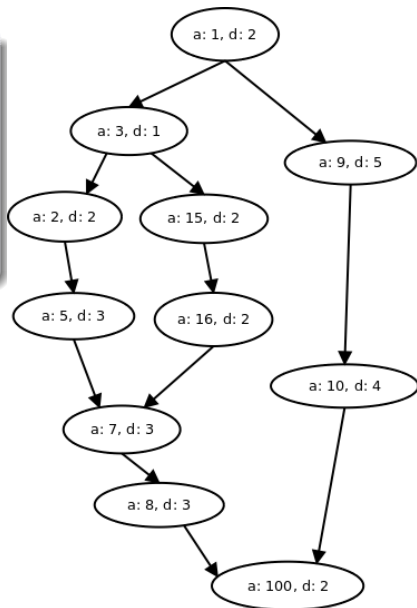
Los jugadores tienen completo conocimiento de las relaciones entre los acertijos y sus dificultades. La temporada solo termina al ser resuelto el acertijo final y el ganador es quién lo haga.

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.

Ejemplo de laberinto de acertijos

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



Resolución

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 $\{a \in \text{acertijos}(l)\}$

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

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

$$\left\{ \begin{array}{l} \text{prev} \subseteq \text{acertijos}(l) \wedge 1 \leq \text{dif} \leq 5 \wedge \neg \text{acj} \in \text{acertijos}(l) \wedge \#(\text{prev}) \\ \geq 1 \end{array} \right\}$$

otras operaciones

acertijoInicial : lab \longrightarrow nat

acertijosFinales : lab \longrightarrow conj(nat)

filtrarAcertijosFinales : lab l \times conj(nat) as \longrightarrow conj(nat)

$\{as \subseteq \text{acertijos}(l)\}$

axiomas

$\text{acertijos}(\text{nuevoLab}(a, d)) \equiv \text{Ag}(a, \emptyset)$

$\text{acertijos}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev})) \equiv \text{Ag}(\text{acj}, \text{acertijos}(l))$

$\text{dificultad}(\text{nuevoLab}(a, d), a') \equiv d$

$\text{dificultad}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev}), a') \equiv \text{if } \text{acj} = a' \text{ then } \text{dif} \text{ else } \text{dificultad}(l, a') \text{ fi}$

$\text{opciones}(\text{nuevoLab}(a, d), a') \equiv \emptyset$

$\text{opciones}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev}), a') \equiv \text{if } a' \in \text{prev} \text{ then } \text{Ag}(\text{acj}, \emptyset) \text{ else } \emptyset \text{ fi}$
 \cup
 $\text{if } a' \in \text{acertijos}(l) \text{ then } \text{opciones}(l, a') \text{ else } \emptyset \text{ fi}$

$\text{acertijoInicial}(\text{nuevoLab}(a, d)) \equiv a$

$\text{acertijoInicial}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev})) \equiv \text{acertijoInicial}(l)$

$\text{acertijosFinales}(l) \equiv \text{filtrarAcertijosFinales}(l, \text{acertijos}(l))$

TAD Temporada

géneros temp

observadores básicos

jugadores : temp \longrightarrow conj(jugador)

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

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

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

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

anotarAcertijoFinal : temp t \times jugador j \longrightarrow temp

$$\{j \in \text{jugadores}(t) \wedge \text{actual}(t, j) \in \text{acertijosFinales}(\text{laberinto}(t))\}$$

otras operaciones

finalizada : temp \longrightarrow bool

axiomas

$$\text{jugadores}(\text{initTemp}(\text{js}, \text{l})) \equiv \text{js}$$
$$\text{jugadores}(\text{anotarYproxAcertijo}(t, n, j)) \equiv \text{jugadores}(t)$$
$$\text{jugadores}(\text{anotarAcertijoFinal}(t, j)) \equiv \text{jugadores}(t)$$
$$\text{acertados}(\text{initTemp}(js, l), j') \equiv \emptyset$$

```

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

```

$$\text{actual}(\text{initTemp}(\text{js}, l), j') \equiv \text{acertijolnicial}(l)$$
[illegible]
$$\text{actual}(\text{anotarAcertijoFinal}(t, j), j') \equiv \text{actual}^{\mathbb{A}}(t, j')$$

axiomas

$\text{lab}(\text{initTemp}(\text{js}, \text{l})) \equiv \text{l}$

$\text{lab}(\text{anotarYproxAcertijo}(\text{t}, \text{j}, \text{elec})) \equiv \text{lab}(\text{t})$

$\text{lab}(\text{anotarAcertijoFinal}(\text{t}, \text{j})) \equiv \text{lab}(\text{t})$

$\text{finalizada}(\text{initTemp}(\text{js}, \text{l})) \equiv \text{false}$

$\text{finalizada}(\text{anotarYproxAcertijo}(\text{t}, \text{j}, \text{elec})) \equiv \text{false}$

$\text{finalizada}(\text{anotarAcertijoFinal}(\text{t}, \text{j})) \equiv \text{true}$

Axiomas de los acertijos finales del Laberinto

$\text{acertijosFinales}(l) \equiv \text{filtrarAcertijosFinales}(l, \text{acertijos}(l))$

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

Cerrando

TADs en 2 niveles

- Identificar la **parte estática** de la **parte dinámica** del problema
- Analizar si conviene partir en 2 niveles para sumar claridad a la especificación
- Las partes tienen que tener comportamiento marcadamente distinto y separable

Algunas consejos y conclusiones finales...

- Cuándo conviene usar los TAD básicos y cuándo TADs más específicos.
- Analizar las “partes” o “etapas” de un problema y como impactan en un TAD.
- Verificar si hay partes inmutables o siempre están cambiando.
- Crear posibles instancias y como se deberían observar
- Puede ocurrir que estemos modelando comportamiento ya conocido o haga falta delegar “responsabilidades” en otros TADs.
- Identificar en el problema “entidades” con comportamiento/responsabilidades/propiedades asociadas y como se relacionan con otras “entidades”.

La especificación tiene que ser CLARA y lo más concisa posible.

Es todo por hoy...

Preguntas ???