

# Práctica 3 - TADs en 2<sup>o</sup> niveles



Pablo Baranbaum

- Pedagogía por repetición (¡re sí! sos un capo!)

## Parte 1

Reproductor de música.

- Ilustra descomposición en 2 niveles

### Reproductor de música

Se nos pide modelar mediante TADs un servicio de música en internet. El servicio provee un catálogo de canciones. Cada canción pertenece a un género musical. Los usuarios pueden reproducir canciones del catálogo. Se nos pide saber cuantas veces se reprodujo cada canción y cuantas canciones se reprodujeron de cada género.

TAD Canción ES String

TAD Género ES String

TAD Reproductor De Música

géneros  $r_m$

observadores

#reprods :  $r_m \times \text{Canción } c \longrightarrow \text{nat}$   
 $\{ \text{def?}(c, \text{catálogo}(r)) \}$

catálogo :  $r_m \rightarrow \text{dicc}(\text{canción}, \text{género})$

## generadores

nuevo : dict (canción, género)  $\rightarrow$   $\Gamma M$

reproducir :  $\Gamma M \Gamma \times$  canción  $c \rightarrow \Gamma M$

$\{ \text{def?}(c, \text{catálogo}(r)) \}$

## Axiomas

$$\text{catálogo}(\text{nuevo}(d)) \equiv d$$

$$\text{catálogo}(\text{reproducir}(\Gamma, c)) = \text{catálogo}(\Gamma)$$

$$\# \text{reprods}(\text{nuevo}(\text{cat}), c) \equiv 0$$

$$\# \text{reprods}(\text{reproducir}(\Gamma, c'), c) \equiv$$

if  $c' = c$  then

$$1 + \# \text{reprods}(\Gamma, c)$$

else

$$\# \text{reprods}(\Gamma, c)$$

fi

-- otra forma

$$\beta(c = c') + \# \text{reprods}(\Gamma, c)$$



## Otras Operetas

-- 1ª forma

$$\# \text{reprods Género}(\Gamma, g) \equiv \text{Cuento } g\text{'s en valores del dict de } \Gamma$$

-- 2ª forma: uso generadores (lo que debería evitar, pero así no hay posibilidad de romper

$\# \text{reprods Genéro}(\text{nuevo}(d), g) \equiv 0$  congruencia pues la info. de  $\# \text{repGen}$  puede ser deducida de los observadores

$\# \text{reprods Genéro}(\text{reproducir}(r, c'), g) \equiv$  observadores

if obtener( $c'$ , catálogo( $r$ )) =  $g$  then

$\# \text{reprods Genéro}(g) + 1$

else

$\# \text{reprods Genéro}(g)$

fi

-- otra forma

$\# \text{reprods Genéro}(g) + \beta(\text{obtener}(c', \text{catálogo}(r)) = g)$

## Parte 2. Competencia de Acertijos

En la facultad de exactas se decidió organizar competencias de acertijos matemáticos. Los acertijos se categorizan con un número de complejidad del 1 al 5 y se organizan formando un laberinto.

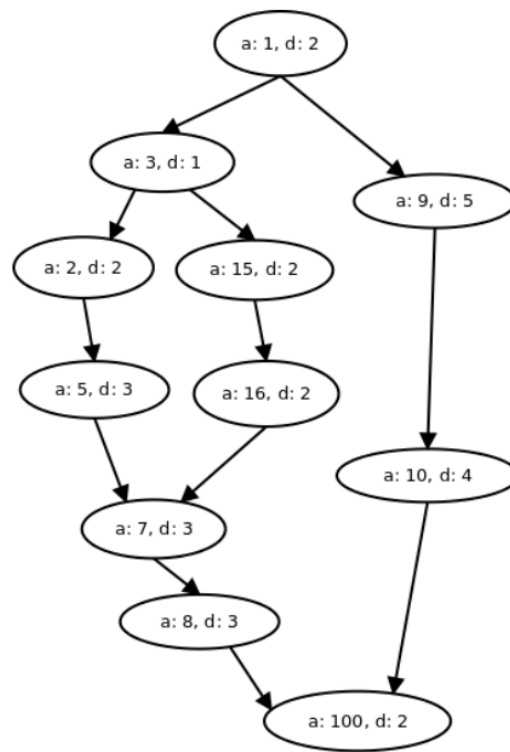
El objetivo de la competencia es ser el primero en atravesar el laberinto (ver ejemplo). Todos los jugadores comienzan en un acertijo inicial. Al resolverlo tienen la posibilidad de avanzar en el laberinto. Cuando un jugador resuelve un acertijo obtiene acceso a uno o más acertijos nuevos. No obstante, 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 termina al ser resuelto el acertijo final y el ganador es quien lo haga.

Se pide especificar el TAD Temporada para que maneje la información administrativa de cada competencia. Se deben conocer los acertijos y la relación 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.

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 se puede llegar a un acertijo ya resuelto.



Modelamos en 2 TADs:

- Laberinto
- Competencia (Temporada)

TAD Acertijo es Nat

TAD Laberinto  
géneros lab

observadores

acertijos : lab  $\rightarrow$  conj(acertijos)

dificultad : lab  $\ell$  x acertijo a  $\rightarrow$  nat

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

-- Decido que se cargue de Arriba a Abajo.

opciones :  $lab\ l \times acertijo\ a_1 \rightarrow conj(acertijo)$   
 $\{a \in acertijos(l)\}$

generadores

nuevoLab :  $acertijo \times nat\ d \rightarrow lab$   
 $\{1 \leq d \leq 5\}$

agAcertijo :  $lab\ l \times acertijo\ a \times nat\ d \times$   
 $\times conj(acertijos)\ prev \rightarrow lab$   
 $\{ \neg(a \in acertijos(l)) \wedge$   
 $1 \leq d \leq 5 \wedge$   
 $prev \in acertijos(l) \wedge$   
 $\neg \phi?(prev) \}$

otras operaciones ↙ lo conozco por opciones

acertijo Inicial :  $lab \rightarrow acertijo$

acertijos Finales :  $lab \rightarrow conj(acertijos)$

filtrar Acertijos Finales :

$lab\ l \times conj(acertijo)\ as \rightarrow conj(acertijo)$   
 $\{as \in acertijos(l)\}$

axiomas

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

$acertijos(agAcertijo(l, a, d, p)) \equiv$   
 $Ag(a, acertijos(l))$

difficulty(nuevoLab(a', d'), a)  $\equiv$  d'   
 ↙ es nuevoLab!  $a=a'$

difficultad(  $\text{agArticulo}(l, a, d, p)$ ,  $a'$  )

if  $a = a'$  then

else

$\text{difficultad}(l, a')$

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

$\text{Opciones}(\text{agArticulo}(l, a, d, p), a') \stackrel{a \neq a'}{\substack{\text{Pre} \\ \text{Post}}} \equiv \text{de opciones}$

if  $a' \in p$  then

$\text{Ag}(a, \text{opciones}(l, a'))$

else

$f_i$

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

$\text{articuloInicial}(\text{agArticulo}(l, a, d, p)) \equiv$   
 $\text{articuloInicial}(l)$

$\text{articulos Finales}(l)$

$\text{filtrar Articulos Finales}(l, \text{articulos}(l))$

filterAcertijosFinder (l, c)  $\equiv$

if  $\phi?(c)$  then

$\phi$

else --  $c =_{\text{des}} Ag(dameUno(c), sinUno(c))$

if  $\phi?(opciones(dameUno(c)))$  then

$Ag(dameUno(c), \text{filterFin}(sinUno(c)))$

else

$\text{filterFin}(sinUno(c))$

$f_i$

## Parte 3 - Competencia

TAD Jugador ES NAT

TAD Temporada

generos temp

observadores

jugadores : temp  $\rightarrow$  conj(jugador)

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

resolver : temp t  $\times$  jugador t  $\{1\}$

laberinto : temp  $\rightarrow$  lab

# generadores

$$\text{nuevaTemp} : \text{conj}(\text{jugador}) \times \text{lab} \rightarrow \text{temp} \\ \left\{ \begin{array}{l} \#(\text{artijos Finales}(l)) = 1 \end{array} \right.$$

$$\text{resolver} : \text{temp } t \times \text{jugador} \times \text{artijo elec} \rightarrow \text{temp} \\ \left\{ \begin{array}{l} \neg \text{finalizada?}(t) \wedge \\ j \in \text{jugadores}(t) \wedge \\ \text{elec} \in \text{opciones}(\text{laberinto}(t), \text{actual}(t, j)) \end{array} \right.$$

$$\text{resolverFinal} : \text{temp } t \times \text{jugador} \rightarrow \text{temp} \\ \left\{ \begin{array}{l} \neg \text{finalizada?}(t) \wedge \\ j \in \text{jugadores}(t) \wedge \\ \phi?(\text{opciones}(\text{laberinto}(t), \text{actual}(t, j))) \end{array} \right.$$

## axiomas

$$\text{jugadores}(\text{nuevaTemp}(js, l)) \equiv js$$

$$\text{jugadores}(\text{resolver}(t, j, \text{elec})) \equiv \text{jugadores}(t)$$

$$\text{jugadores}(\text{resolverFinal}(t, j)) \equiv \text{jugadores}(t)$$

$$\text{actual}(\text{nuevaTemp}(js, l), j) \equiv \text{artijoInicial}(l)$$



$$\text{actual}(\text{resolver}(t, j, \text{elec}), j') \equiv$$

if  $j = j'$  then

elec

else

actual( $t, j'$ )

fi

$$\text{actual}(\text{resolver Final}(t, j), j') \equiv \text{actual}(t, j')$$

$$\text{resul\_for}(\text{neverTemp}(j, l), j) \equiv \emptyset$$

$$\text{resul\_for}(\text{resolver}(t, j, \text{elec}), j') \equiv$$

resul\\_for( $t, j'$ )  $\cup$  if  $j = j'$  then

{actual( $t, j'$ )}

else

{ }

fi

$$\text{resul\_for}(\text{resolver Final}(t, j), j') \equiv$$

resul\\_for( $t, j'$ )  $\cup$  if  $j = j'$  then

{actual( $t, j'$ )}

else

{ }

fi

$$\text{laberinto}(\text{nuevaTemp}(js, l)) \equiv l$$

$$\text{laberinto}(\text{resolver}(t, j, \text{elec})) \equiv \text{laberinto}(t)$$

$$\text{laberinto}(\text{resolver Final}(t, j)) \equiv \text{laberinto}(t)$$

$$\text{finalizada?} : \text{temp} \rightarrow \text{bool}$$

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

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

$$\text{finalizada?}(\text{resolver Final}(t, j)) \equiv \text{true}$$

