

Trabajo práctico 2: AED2-TEG

Normativa

Fecha de entrega: 3 de Junio de 2011.

Normas de entrega: Las contenidas en la página web de la materia.

Enunciado

El objetivo de este trabajo práctico consiste en realizar el diseño completo del módulo que se explica con el TAD AED2-TEG (ver la sección Especificación de este documento), así como todos aquellos módulos necesarios para la tarea.

En un nuevo relevamiento se determinó que, en una primera etapa, los movimientos serán simplemente palabras o frases. Sin embargo, el cliente solicitó que el tablero esté preparado para soportar otro tipo de movimientos. Asimismo, se determinó la política para elegir la ficha que se elimina de un casillero disputado, cada vez que se produce un movimiento (ver operación Perdedor). En pocas palabras, se elimina una ficha de aquel jugador que posee menor cantidad de fichas en la casilla. En caso de empate, se elimina la ficha del jugador correspondiente al número máximo. Finalmente, se limaron algunas asperezas, obteniéndose la especificación que se brinda al final del documento. Notar que dicha especificación no resuelve el TP1 necesariamente.

Complejidades requeridas

El diseño debe respetar las siguientes restricciones de complejidad (siempre de **peor caso**). Las operaciones que no están mencionadas no tienen restricción de complejidad.

Tablero

- La operación **Contener**(t, c, m_i, m_v) debe tomar tiempo

$$O(f(\text{ProxCasilla}(t)) + \text{copy}(m_i) + \text{copy}(m_v)),$$

donde $\text{copy}(m)$ es el costo de copiar m , para todo $m \in \mu$, y

$$f(n) = \begin{cases} 1 & \text{si } n \text{ no es potencia de dos} \\ n & \text{en caso contrario.} \end{cases}$$

- La operación **Agregar**(t, c, m_i, m_v) debe tomar tiempo

$$O(f(\text{ProxCasilla}(t)) + f(\text{ProxCasilla}(t)) + \text{copy}(m_i) + \text{copy}(m_v)),$$

donde $\text{copy}(m)$ y f se definen como en la operación anterior.

- Las operaciones **Salidas** y **CasillasDe** debe tomar tiempo $O(1)$. En caso de devolver un iterador, las funciones del iterador que devuelven el valor actual y avanzan al siguiente elemento deben tomar tiempo $O(1)$.
- HINT para el teg:** se sugiere agregar una operación **Destinos**(c) que, dada una casilla c , retorne un iterador que permita recorrer las tuplas $\langle c', m \rangle$ tales que $c \in \text{Origenes}(c', m)$. Dicho de otra forma, **Destinos**(c) permite recorrer todos los pares $\langle c', m \rangle$ tales que el movimiento m desde c lleva a c' . La complejidad de dicha operación debe ser de $O(1)$, y las funciones del iterador que devuelven el valor actual y avanzan al siguiente elemento deben tomar tiempo $O(1)$.

AED2-TEG

- La operación **Iniciar**(t, cs, ms) debe tomar tiempo

$$O\left(\sum_{c \in C(t)} \left(\text{Long}(cs) + \sum_{m \in S(t,c)} |m|\right)\right),$$

donde $|m|$ es la longitud de m , $C(t) = \text{Casillas}(t)$ y $S(t, c) = \text{Salidas}(t, c)$. El tablero t de entrada debe ser una instancia de un módulo **Tablero** que satisfaga las complejidades mencionadas anteriormente, mientras que cs y ms deben ser instancias del módulo **Vector**.

- Las operaciones **Dominada?**, **Vacia?**, **Disputada?**, **Dominador**, **TieneFichas?**, **MisionCumplida?**, **PorDominar**, **FichasAgregadas**, **Mision**, **Tablero**, y **AgregarFicha** deben tomar tiempo $O(1)$.
- Las operaciones **Dominadas**, **Disputan**, **Fichas** y **ConFichas** deben tomar tiempo $O(1)$. En caso de devolver un iterador, las funciones del iterador que devuelven el valor actual y avanzan al siguiente elemento deben tomar tiempo $O(1)$.
- La operación **Mover**(t, j, m, n) debe tomar tiempo

$$O\left(\sum_{c \in D(t)} (\log \#Disputan(t, c)) + |m| + \sum_{c \in C(t)} (\#Origenes(t, c, m))\right),$$

donde $|m|$ es la longitud de m , $D(t) = \{c \in \text{Casillas}(t) \mid \text{Disputada?}(t, c)\}$ y $C(t) = \text{Casillas}(t)$.

Condiciones de aprobación y/o promoción

Para la confección de este trabajo práctico se deberán tener en cuenta las siguientes pautas, además de las que se encuentran en la página de la materia:

- El TP debe estar presentado con la carátula oficial de la materia, dentro de una carpeta de la cual se sujeten apropiadamente sus hojas, y debe contener un índice o tabla de contenidos.
- El formato del TP debe ser homogéneo, y debe estar escrito en un procesador de textos apropiado a tal efecto. El código debe estar claramente indentado, y las funciones tienen que ser suficientemente concisas y descriptivas.
- El TP debe contener la definición de todos los módulos utilizados, a excepción de los descriptos en los apuntes de diseño y de módulos básicos de la materia.
- La interfaz de cada módulo debe contener para cada operación: descripción, pre y postcondición, complejidad temporal e información sobre aliasing.
- Se pueden agregar operaciones auxiliares que no pertenezcan a la interfaz del módulo. Para cada una de estas funciones, debe agregarse un comentario acerca de su propósito en lenguaje coloquial. Asimismo, se debe calcular la complejidad temporal y se debe indicar los aspectos de aliasing.
- La complejidad de cada operación deberá estar justificada adecuadamente en función de su algoritmo y de la complejidad de las operaciones de los módulos utilizados. Esto puede hacerse directamente sobre el pseudocódigo o en un texto separado.
- Para cada módulo, deberán consignarse adecuadamente los módulos utilizados y la complejidad requerida para cada una de sus operaciones.
- Todos los invariantes de representación deben estar escritos en lenguaje coloquial y en lenguaje formal.
- Para recorrer linealmente los elementos de una estructura se deberá utilizar iteradores. Los mismos deberán ser diseñados junto al módulo correspondiente.
- Para los módulos que se expliquen con TADs básicos sólo es obligatorio incluir en el diseño un conjunto mínimo de operaciones que permitan utilizar de manera razonable el tipo.
- Se evaluará el uso de iteradores y la modularidad de la solución, tanto a nivel de estructuras como de algoritmos.
- En el caso de tener que devolver una colección de elementos que resulten de proyectar alguna componente de una colección de tuplas (o alguna operación equivalente), es posible devolver un iterador a esa colección que se ocupe de generar la proyección correspondiente. Dicho iterador debe ser implementado especialmente para dicha estructura y no debe realizar otro cálculo más que recorrer y proyectar. De esta manera se evita tener que copiar la estructura y el costo computacional asociado.

Especificación

TAD TABLERO(μ)

parámetros formales

géneros μ

géneros $\text{tablero}(\mu)$

exporta generadores, observadores, otras operaciones (exportadas).

igualdad observacional

$$(\forall t, t' : \text{tablero}(\mu)) \left(t =_{\text{obs}} t' \iff \left(\begin{array}{l} \text{Casillas}(t) = \text{Casillas}(t') \wedge \\ (\forall c: \text{casilla}) (c \in \text{Casillas}(t) \Rightarrow \\ \text{Continente}(t, c) = \text{Continente}(t', c) \wedge \\ (\forall m: \mu) (\text{Origenes}(t, c, m) = \text{Origenes}(t', c, m)) \end{array} \right) \right)$$

observadores básicos

Casillas	: $\text{tablero}(\mu)$	$\longrightarrow \text{conj}(\text{casilla})$	
Continente	: $\text{tablero}(\mu) \ t \times \text{casilla } c$	$\longrightarrow \text{continente}$	$\{c \in \text{Casillas}(t)\}$
Origenes	: $\text{tablero}(\mu) \ t \times \text{casilla } c \times \mu \ m$	$\longrightarrow \text{conj}(\text{casilla})$	$\{c \in \text{Casillas}(t)\}$

generadores

Informalmente, el objetivo de cada generador es el siguiente:

Inicial: crea el tablero inicial que sólo la casilla 0 que pertenece al continente 0.

Contener(t, c, m_i, m_v): agrega una nueva casilla c' cuyo continente es el mismo que el de c . Asimismo, se agregan dos flechas, una de c' a c y otro en dirección opuesta. El movimiento correspondiente a $c' \rightarrow c$ es m_i , mientras que el movimiento correspondiente a $c \rightarrow c'$ es m_v .

Agregar(t, c, m_i, m_v): similar a Contener; la única diferencia es que c' se agrega en un continente nuevo.

Movilizar(t, c_1, c_2, m): agrega una flecha de c_1 a c_2 con movimiento m .

Conectar(t, c_1, c_2, m_i, m_v): conecta dos casillas previamente desconectadas, agregando una flecha de c_1 a c_2 y otra de c_2 a c_1 , cuyos movimientos son m_i y m_v , respectivamente.

Inicial	: $\longrightarrow \text{tablero}(\mu)$	
Contener	: $\text{tablero}(\mu) \ t \times \text{casilla } c \times \mu \ m_i \times \mu \ m_v$	$\longrightarrow \text{tablero}(\mu) \quad \{c \in \text{Casillas}(t) \wedge m_v \notin \text{Salidas}(t, c)\}$
Agregar	: $\text{tablero}(\mu) \ t \times \text{casilla } c \times \mu \ m_i \times \mu \ m_v$	$\longrightarrow \text{tablero}(\mu) \quad \{c \in \text{Casillas}(t) \wedge m_v \notin \text{Salidas}(t, c)\}$
Movilizar	: $\text{tablero}(\mu) \ t \times \text{casilla } c_1 \times \text{casilla } c_2 \times \mu \ m$	$\longrightarrow \text{tablero}(\mu) \quad \{\{c_1, c_2\} \subseteq \text{Casillas}(t) \wedge c_1 \neq c_2 \wedge \text{Conectadas}(t, c_1, c_2) \wedge m \notin \text{Salidas}(t, c_1)\}$
Conectar	: $\text{tablero}(\mu) \ t \times \text{casilla } c_1 \times \text{casilla } c_2 \times \mu \ m_i \times \mu \ m_v$	$\longrightarrow \text{tablero}(\mu) \quad \left\{ \begin{array}{l} \{c_1, c_2\} \subseteq \text{Casillas}(t) \wedge c_1 \neq c_2 \wedge \neg \text{Conectadas}(t, c_1, c_2) \wedge \\ m_i \notin \text{Salidas}(t, c_1) \wedge m_v \notin \text{Salidas}(t, c_2) \end{array} \right\}$

otras operaciones (exportadas)

Continentes	: $\text{tablero}(\mu)$	$\longrightarrow \text{conj}(\text{continente})$	
Conectadas	: $\text{tablero}(\mu) \ t \times \text{casilla } c_1 \times \text{casilla } c_2$	$\longrightarrow \text{bool}$	$\{\{c_1, c_2\} \subseteq \text{Casillas}(t)\}$
Salidas	: $\text{tablero}(\mu) \ t \times \text{casilla } c$	$\longrightarrow \text{conj}(\mu)$	$\{c \in \text{Casillas}(t)\}$
OrigenesR	: $\text{tablero}(\mu) \ t \times \text{casilla } c \times \mu \ m$	$\longrightarrow \text{conj}(\text{casilla})$	$\{c \in \text{Casillas}(t)\}$
CasillasDe	: $\text{tablero}(\mu) \ t \times \text{continente } cc$	$\longrightarrow \text{conj}(\text{casilla})$	$\{cc \in \text{Continentes}(t)\}$

otras operaciones (no exportadas)

ProxCasilla	: $\text{tablero}(\mu)$	$\longrightarrow \text{casilla}$
ProxContinente	: $\text{tablero}(\mu)$	$\longrightarrow \text{continente}$

axiomas

4/9

$$\text{CasillasDe}(\text{Conectar}(t, _, _, _, _), cc) \equiv \text{CasillasDe}(t, cc)$$
$$\text{ProxCasilla}(t) \equiv \# \text{Casillas}(t)$$
$$\text{ProxContinente}(t) \equiv \# \text{Continentes}(t)$$
Fin TAD

TAD AED2-TEG**géneros** teg**exporta** generadores, observadores, otras operaciones (exportadas).

Para simplificar un poco el TP, vamos a suponer que se desea conocer la misión y la cantidad de fichas agregadas de los jugadores que ya perdieron, además de aquellas agregadas por los que aún juegan.

igualdad observacional

$$(\forall t, t' : \text{teg}) \left(t =_{\text{obs}} t' \iff \left(\begin{array}{l} \text{Tablero}(t) = \text{Tablero}(t') \wedge \\ \text{Jugadores}(t) = \text{Jugadores}(t') \wedge \\ (\forall c : \text{casilla}) (c \in \text{Casillas}(\text{Tablero}(t)) \Rightarrow \\ \quad \text{Fichas}(t, c) = \text{Fichas}(t', c)) \wedge \\ (\forall j : \text{jugador}) (j \in \text{Jugadores}(t) \Rightarrow \\ \quad \text{Mision}(t, j) = \text{Mision}(t', j) \wedge \\ \quad \text{FichasAgregadas}(t, j) = \text{FichasAgregadas}(t', j)) \end{array} \right) \right)$$

observadores básicos

Tablero	: teg	→	tablero	
Jugadores	: teg	→	conj(jugador)	
Fichas	: teg $t \times$ casilla c	→	multiconj(ficha)	$\{c \in \text{Casillas}(t)\}$
Mision	: teg $t \times$ jugador j	→	mision	$\{j \in \text{Jugadores}(t)\}$
FichasAgregadas	: teg $t \times$ jugador j	→	nat	$\{j \in \text{Jugadores}(t)\}$

generadores

Iniciar	: tablero $t \times$ secu(casilla) $cs \times$ secu(mision) ms	→	teg	
	$\{cs \neq \emptyset \wedge \text{Long}(cs) = \text{Long}(ms) \wedge \text{Conj}(cs) \subseteq \text{Casillas}(t) \wedge \text{Conj}(ms) \subseteq \text{Continentes}(t) \wedge \text{SinRepetidos}(cs)\}$			
AgregarFicha	: teg $t \times$ jugador $j \times$ casilla c	→	teg	
	$\{c \in \text{Casillas}(t) \wedge j \in \text{Jugadores}(t) \wedge \text{PuedeAgregar?}(t, j, c)\}$			
Mover	: teg $t \times$ jugador $j \times$ movimiento \times nat	→	teg	$\{\neg \text{Finalizado?}(t) \wedge j \in \text{ConFichas}(t)\}$

otras operaciones (exportadas)

Dominada?	: teg $t \times$ casilla c	→	bool	$\{c \in \text{Casillas}(t)\}$
Vacia?	: teg $t \times$ casilla c	→	bool	$\{c \in \text{Casillas}(t)\}$
Disputada?	: teg $t \times$ casilla c	→	bool	$\{c \in \text{Casillas}(t)\}$
Disputan	: teg $t \times$ casilla c	→	conj(jugador)	$\{c \in \text{Casillas}(t) \wedge \text{Disputada?}(t, c)\}$
Dominador	: teg $t \times$ casilla c	→	jugador	$\{c \in \text{Casillas}(t) \wedge \text{Dominada?}(t, c)\}$
Dominadas	: teg $t \times$ jugador j	→	conj(casilla)	
PorDominar	: teg $t \times$ jugador j	→	nat	$\{j \in \text{ConFichas}(t)\}$
MisionCumplida?	: teg $t \times$ jugador j	→	bool	$\{j \in \text{Jugadores}(t)\}$
TieneFichas?	: teg $t \times$ jugador j	→	bool	$\{j \in \text{Jugadores}(t)\}$
ConFichas	: teg t	→	conj(jugador)	
Eliminados	: teg t	→	conj(jugador)	$\{\neg \text{Finalizado?}(t)\}$
CumplieronMision	: teg	→	conj(jugador)	
Ganadores	: teg t	→	conj(jugador)	$\{\text{Finalizado?}(t)\}$
Finalizado?	: teg	→	bool	
Casillas	: teg	→	conj(casilla)	
Continentes	: teg	→	conj(continente)	
PuedeAgregar?	: teg $t \times$ jugador $j \times$ casilla c	→	bool	$\{c \in \text{Casillas}(t) \wedge j \in \text{Jugadores}(t)\}$

otras operaciones (no exportadas)

Entran	: teg $t \times$ jugador \times casilla $c \times$ movimiento \times nat	→	multiconj(ficha)	$\{c \in \text{Casillas}(t)\}$
Salen	: teg $t \times$ jugador \times casilla $c \times$ nat	→	multiconj(ficha)	$\{c \in \text{Casillas}(t)\}$
Perdedor	: teg $t \times$ casilla c	→	ficha	$\{c \in \text{Casillas}(t) \wedge \text{Disputada?}(t, c)\}$
Movidas	: teg $t \times$ jugador \times casilla $c \times$ nat	→	multiconj(ficha)	$\{c \in \text{Casillas}(t) \wedge \neg \text{Disputada?}(t, c)\}$
Movidas*	: teg $t \times$ jugador \times conj(casilla) $cs \times$ nat	→	multiconj(ficha)	

Dominadas*	: teg $t \times$ jugador $j \times$ conj(casilla) $cs \longrightarrow$ conj(casilla)	$\{cs \subseteq \text{Casillas}(t) \wedge \neg \text{Disputada?}(t, c)\}$
TieneFichas?*	: teg $t \times$ jugador $j \times$ conj(casilla) $cs \longrightarrow$ bool	$\{j \in \text{Jugadores}(t) \wedge cs \subseteq \text{Casillas}(t)\}$
ConFichas*	: teg $t \times$ conj(jugador) $js \longrightarrow$ conj(jugador)	$\{js \subseteq \text{Jugadores}(t)\}$
OrigenesR	: teg $t \times$ casilla $c \times$ movimiento \longrightarrow conj(casilla)	$\{c \in \text{Casillas}(t)\}$
CasillasDe	: teg $t \times$ mision $m \longrightarrow$ conj(casilla)	$\{m \in \text{Continentes}(t)\}$
CumplieronMision*	: teg $t \times$ conj(jugador) $js \longrightarrow$ conj(jugador)	$\{js \subseteq \text{Jugadores}(t)\}$

axiomas

Tablero(Iniciar($t, _, _$))	$\equiv t$
Tablero(AgregarFicha($t, _, _$))	$\equiv \text{Tablero}(t)$
Tablero(Mover($t, _, _, _$))	$\equiv \text{Tablero}(t)$
Jugadores(Iniciar($_, cs, _$))	$\equiv \text{Rango}(0, \text{Long}(cs))$
Jugadores(AgregarFicha($t, _, _$))	$\equiv \text{Jugadores}(t)$
Jugadores(Mover($t, _, _, _$))	$\equiv \text{Jugadores}(t)$
Fichas(Iniciar($_, cs, _$), c)	$\equiv \text{if Esta?}(cs, c) \text{ then } \{\text{Indice}(cs, c)\} \text{ else } \emptyset \text{ fi}$
Fichas(AgregarFicha(t, j, c'), c)	$\equiv \text{if } c = c' \text{ then } \{j\} \text{ else } \emptyset \text{ fi} \cup \text{Fichas}(t, c)$
Fichas(Mover(t, j, m, k), c)	$\equiv (\text{Fichas}(t, c) - \text{Salen}(t, j, c, k)) \cup \text{Entran}(t, j, c, m, k)$
Mision(Iniciar($_, _, ms$), j)	$\equiv ms[j]$
Mision(AgregarFicha($t, _, _$), j)	$\equiv \text{Mision}(t, j)$
Mision(Mover($t, _, _, _$), j)	$\equiv \text{Mision}(t, j)$
FichasAgregadas(Iniciar($_, _, _$), j)	$\equiv 0$
FichasAgregadas(AgregarFicha($t, j', _$), j)	$\equiv \text{if } j = j' \text{ then } 1 \text{ else } 0 \text{ fi} + \text{FichasAgregadas}(t, j)$
FichasAgregadas(Mover($t, _, _, _$), j)	$\equiv \text{FichasAgregadas}(t, j)$
Dominada?(t, c)	$\equiv \# \text{Conj}(\text{Fichas}(t, c)) = 1$
Vacia?(t, c)	$\equiv \emptyset?(\text{Fichas}(t, c))$
Disputada?(t, c)	$\equiv \neg \text{Dominada?}(t, c) \wedge \neg \text{Vacia?}(t, c)$
Disputan(t, c)	$\equiv \text{Conj}(\text{Fichas}(t, c))$
Dominador(t, c)	$\equiv \text{DameUno}(\text{Fichas}(t, c))$
Dominadas(t, j)	$\equiv \text{Dominadas}^*(t, j, \text{Casillas}(t))$
PorDominar(t, j)	$\equiv \#(\text{CasillasDe}(t, \text{Mision}(t, j)) \setminus \text{Dominadas}(t, j))$
MisionCumplida?(t, j)	$\equiv \text{PorDominar}(t, j) = 0$
Casillas(t)	$\equiv \text{Casillas}(\text{Tablero}(t))$
Continentes(t)	$\equiv \text{Continentes}(\text{Tablero}(t))$
PuedeAgregar?(t, j, c)	$\equiv \neg \text{Finalizado?}(t) \wedge \text{TieneFichas?}(t, j) \wedge$ $(\text{Vacia?}(t, c) \vee (\text{Dominada?}(t, c) \wedge j = \text{Dominador}(t, c)))$
TieneFichas?(t, j)	$\equiv \text{TieneFichas}^*(t, j, \text{Casillas}(t))$
ConFichas(t)	$\equiv \text{ConFichas}^*(t, \text{Jugadores}(t))$
Finalizado?(t)	$\equiv \# \text{ConFichas}(t) = 1 \vee \neg \emptyset?(\text{CumplieronMision}(t))$
Eliminados(t)	$\equiv \text{Jugadores}(t) \setminus \text{ConFichas}(t)$
CumplieronMision(t)	$\equiv \text{CumplieronMision}^*(t, \text{Jugadores}(t))$
Ganadores(t)	$\equiv \text{if } \# \text{ConFichas}(t) = 1 \text{ then } \text{ConFichas}(t) \text{ else } \text{CumplieronMision}(t) \text{ fi}$
Salen(t, j, c, k)	$\equiv \text{if } \text{Disputada?}(t, c) \text{ then } \{\text{Perdedor}(t, c)\} \text{ else } \text{Movidas}(t, j, c, k) \text{ fi}$
Entran(t, j, c, m, k)	$\equiv \text{Movidas}^*(t, j, \text{Dominadas}(t, c) \cap \text{OrigenesR}(t, c, m), k)$
Perdedor(t, c)	$\equiv \text{máx}\{\text{minAps}(\text{Fichas}(t, c))\}$
Movidas(t, j, c, k)	$\equiv \text{Reps}(j, \text{mín}\{k, \#(\text{Fichas}(t, c), j)\})$

```

Movidas*(t, j, cs, k)  ≡ if  $\emptyset?(cs)$  then  $\emptyset$  else Movidas(t, j, DameUno(cs), k)  $\cup$  Movidas*(t, j, SinUno(cs), k) fi
Dominadas*(t, j, cs)  ≡ if  $\emptyset?(cs) \vee \neg \text{Dominada?}(t, \text{DameUno}(cs)) \vee \text{Dominador}(t, \text{DameUno}(cs)) \neq j$  then
                         $\emptyset$ 
                        else
                          {DameUno(cs)}
                        fi  $\cup$ 
if  $\emptyset?(cs)$  then  $\emptyset$  else Dominadas*(t, j, SinUno(cs)) fi

CumplieronMision*(t, js) ≡ if  $\emptyset?(js) \vee \neg \text{MisionCumplida?}(t, \text{DameUno}(js))$  then  $\emptyset$  else {DameUno(js)} fi  $\cup$ 
                        if  $\emptyset?(js)$  then  $\emptyset$  else CumplieronMision*(t, SinUno(js)) fi

TieneFichas?(t, j, cs)  ≡  $\neg \emptyset?(cs) \wedge (j \in \text{Fichas}(t, \text{DameUno}(cs)) \vee \text{TieneFichas?}^*(t, j, \text{SinUno}(cs)))$ 

ConFichas*(t, js)       ≡ if  $\emptyset?(js) \vee \neg \text{TieneFichas?}(t, \text{DameUno}(js))$  then  $\emptyset$  else {DameUno(js)} fi  $\cup$ 
                        if  $\emptyset?(js)$  then  $\emptyset$  else ConFichas*(t, SinUno(js)) fi

OrigenesR(t, c, m)      ≡ OrigenesR(Tablero(t), c, m)
CasillasDe(t, c, m)     ≡ CasillasDe(Tablero(t), c, m)

```

Fin TAD

CASILLA es renombre de NAT con género casilla.

CONTINENTE es renombre de NAT con género continente.

JUGADOR es renombre de NAT con género jugador.

MISION es renombre de CONTINENTE con género mision.

MOVIMIENTO es renombre de STRING con género movimiento.

TABLERO es renombre de TABLERO(STRING) con género tablero.

TAD SECUENCIA EXTENDIDA(α)**extiende** SECUENCIA(α)**otras operaciones**

Conj	:	secu(α)	\longrightarrow	conj(α)	
$\bullet[\bullet]$:	secu(α) $s \times \text{Nat } n$	\longrightarrow	α	$\{0 \leq n < \text{Long}(s)\}$
Indice	:	secu(α) $s \times \alpha a$	\longrightarrow	Nat	$\{\text{Esta?}(a, s)\}$
SinRepetidos	:	secu(α)	\longrightarrow	bool	

axiomas

Conj(s) \equiv **if** Vacía?(s) **then** \emptyset **else** Ag(Prim(s), Conj(Fin(s))) **fi**
 $s[n]$ \equiv **if** $n = 0$ **then** Prim(s) **else** $s[n - 1]$ **fi**
 Indice(s, a) \equiv **if** $a = \text{Prim}(s)$ **then** 0 **else** $1 + \text{Indice}(s, a)$ **fi**
 SinRepetidos(s) \equiv Long(s) = #Conj(s)

Fin TAD**TAD CONJUNTO EXTENDIDO(NAT)****extiende** CONJUNTO(NAT)**otras operaciones**Rango : nat \times nat \longrightarrow conj(Nat)**axiomas**Rango(ℓ, r) \equiv **if** $\ell \geq r$ **then** \emptyset **else** Ag(ℓ , Rango($\ell + 1, r$)) **fi****Fin TAD****TAD MULTICONJUNTO EXTENDIDO(α)****extiende** MULTICONJUNTO(α)**otras operaciones**

$\bullet - \bullet$:	multiconj(α) \times multiconj(α)	\longrightarrow	multiconj(α)	
Reps	:	$\alpha \times \text{Nat}$	\longrightarrow	multiconj(α)	
Conj	:	multiconj(α)	\longrightarrow	conj(α)	
$\bullet \setminus \{\bullet\}$:	multiconj(α) $\times \alpha$	\longrightarrow	multiconj(α)	
QuitarUno	:	multiconj(α) m	\longrightarrow	multiconj(α)	$\{\neg\emptyset?(m)\}$
#Uno	:	multiconj(α) m	\longrightarrow	nat	$\{\neg\emptyset?(m)\}$
minAps	:	multiconj(α)	\longrightarrow	conj(α)	
mín #	:	multiconj(α) m	\longrightarrow	nat	$\{\neg\emptyset?(m)\}$

axiomas

$m - c$ \equiv **if** $\emptyset?(c)$ **then** m **else** $(m - \{\text{DameUno}(c)\}) - \text{SinUno}(c)$ **fi**
 Conj(m) \equiv **if** $\emptyset?(m)$ **then** \emptyset **else** Ag(DameUno(m), Conj(SinUno(m))) **fi**
 Reps(a, n) \equiv **if** $n = 0$ **then** \emptyset **else** Ag(a , Reps($a, n - 1$)) **fi**
 $m \setminus \{e\}$ \equiv $m - \text{Reps}(e, \#(m, e))$
 QuitarUno(m) \equiv $m \setminus \{\text{DameUno}(m)\}$
 #Uno(m) \equiv $\#(m, \text{DameUno}(m))$
 minAps(m) \equiv **if** #Conj(m) ≤ 1 **then**
 Conj(ms)
 else
 if #Uno(m) \leq mín # (QuitarUno(m)) **then** {DameUno(m)} **else** \emptyset **fi** \cup
 if #Uno(m) $<$ mín # (QuitarUno(m)) **then** \emptyset **else** minAps(QuitarUno(m)) **fi**
 fi
 mín #(m) \equiv #Uno(mínAps(m))

Fin TAD