

**Ejercicio 1 (Buscaminas) ★****TAD BUSCAMINAS****igualdad observacional**

$$(\forall b_1, b_2 : \text{buscaminas}) \left( b_1 =_{\text{obs}} b_2 \iff \begin{pmatrix} \text{Filas}(b_1) =_{\text{obs}} \text{Filas}(b_2) \wedge \text{Columnas}(b_1) =_{\text{obs}} \text{Columnas}(b_2) \wedge \text{Minas}(b_1) =_{\text{obs}} \text{Minas}(b_2) \wedge \text{CeldasReveladas}(b_1) =_{\text{obs}} \text{CeldasReveladas}(b_2) \end{pmatrix} \right)$$

**observadores básicos**

Filas : buscaminas  $\rightarrow$  nat  
 Columnas : buscaminas  $\rightarrow$  nat  
 Minas : buscaminas  $\rightarrow$  conj(pos)  
 CeldasReveladas : buscaminas  $\rightarrow$  conj(pos)

**generadores**

Iniciar : nat *filas*  $\times$  nat *columnas*  $\times$  conj(pos) *cj*  $\rightarrow$  buscaminas  
 $\{(\forall p: \text{Posición}) p \in \text{minas} \Rightarrow \Pi_1(p) < \text{filas} \wedge \Pi_2(p) < \text{columnas}\}$   
 RevelarCeldasposicion *p*  $\times$  buscaminas *b*  $\rightarrow$  buscaminas  
 $\{\text{posValida}(p, b) \wedge p \notin \text{CeldasReveladas}(b) \wedge \neg \text{Termino?}(b)\}$

**otras operaciones**

PosiciónVálida : pos  $\times$  buscaminas  $\rightarrow$  bool  
 Lindantes : pos *p*  $\times$  buscaminas *b*  $\rightarrow$  conj(pos)  $\{ \text{PosiciónVálida}(p, b) \}$   
 TienePista? : pos *p*  $\times$  buscaminas *b*  $\rightarrow$  bool  $\{ \text{PosiciónVálida}(p, b) \}$   
 TieneMina? : pos *p*  $\times$  buscaminas *b*  $\rightarrow$  bool  $\{ \text{PosiciónVálida}(p, b) \}$   
 DamePista : pos *p*  $\times$  buscaminas *b*  $\rightarrow$  bool  $\{ \text{PosiciónVálida}(p, b) \wedge_L \text{TienePista?}(p, b) \}$   
 Termino? : buscaminas  $\rightarrow$  bool  
 Gano? : buscaminas  $\rightarrow$  bool  
 Perdio? : buscaminas  $\rightarrow$  bool  
 SeguroLindaAMinas : pos *p*  $\times$  buscaminas *b*  $\rightarrow$  bool  
 $\{ \text{PosiciónVálida}(p, b) \wedge \text{TienePista?}(p, b) \wedge p \in \text{CeldasReveladas}(p, b) \}$

**axiomas**  $\forall$  buscaminas *b*,  $\forall$  posición *p*,  $\forall$  conj(pos) *cp*,  $\forall$  nat *n*

Filas(Iniciar(*f*, *c*, *cp*))  $\equiv$  *f*  
 Filas(RevelarCelda(*p*, *b*))  $\equiv$  Filas(*b*)

Columnas(Iniciar(*f*, *c*, *cp*))  $\equiv$  *c*  
 Columnas(RevelarCelda(*p*, *b*))  $\equiv$  Columnas(*b*)

Minas(Iniciar(*f*, *c*, *cp*))  $\equiv$  *cp*  
 Minas(RevelarCelda(*p*, *b*))  $\equiv$  Minas(*b*)

CeldasReveladas(Iniciar(*f*, *c*, *cp*))  $\equiv$   $\emptyset$

CeldasReveladas(RevelarCelda(*p*, *b*))  $\equiv$  **if** TieneMina?(*p*, *b*)  $\vee$  TienePista?(*p*, *b*) **then**  
     Ag(*p*, CeldasReveladas(*b*))  
**else**  
     ReveladasAux(Lindantes(*p*, *b*), {*p*}, *b*)  $\cup$  CeldasReveladas(*b*)  
**fi**

ReveladasAux(vecinos, vistos, *b*)  $\equiv$  **if**  $\emptyset?$ (vecinos) **then**  
     vistos  
**else**  
     **if** DameUno(vecinos)  $\in$  vistos **then**  
         ReveladasAux(SinUno(vecinos), vistos, *b*)  
     **else**  
         **if** TienePista?(DameUno(vecinos), *b*) **then**  
             ReveladasAux(SinUno(vecinos), Ag(DameUno(vecinos), vistos), *b*)  
         **else**  
             ReveladasAux(Lindantes(DameUno(vecinos))  $\cup$   
             SinUno(vecinos), Ag(DameUno(vecinos), vistos), *b*)  
         **fi**  
     **fi**  
**fi**

PosiciónVálida(*p*, *b*)  $\equiv$   $\Pi_1(p) < \text{Filas}(b) \wedge \Pi_2(p) < \text{Columnas}(b)$

TienePista?(*p*, *b*)  $\equiv$   $\neg \emptyset?(\text{Minas}(b) \cap \text{Lindantes}(p, b))$

$TieneMina?(p, b) \equiv p \in Minas(b)$   
 $DamePista(p, b) \equiv \#(Minas(b) \cap Lindantes(p, b))$   
 $Gano?(b) \equiv \#Minas(b) + \#CeldasReveladas(b) = Filas(b) * Columnas(b)$   
 $Perdió?(b) \equiv \neg \emptyset?(Minas(b) \cap CeldasReveladas(b))$   
 $Termino?(j) \equiv Gano?(b) \vee Perdió?(b)$   
 $SeguroLindaAMinas(p, b) \equiv \#(Lindantes(p, b) - CeldasReveladas(b)) = DamePista(p, b)$

**Fin TAD**

## Solucin Ej 3

- a y b)
- Toda publicacin es de un usuario vlido:  
 $claves(publicaciones) \subseteq usuarios$
  - Sin publicaciones repetidas:  
 $(\forall u_1, u_2: usuario) (def?(u_1, publicaciones) \wedge def?(u_2, publicaciones) \Rightarrow_L$   
 $(\forall t_1, t_2: tupla\langle pub, nat \rangle) (t_1 \in obtener(u_1, publicaciones) \wedge t_2 \in obtener(u_2, publicaciones)) \Rightarrow_L$   
 $\Pi_1(t_1) \neq \Pi_1(t_2))$
  - Todo dueo en *dueos* tiene su publicacin en *publicaciones*:  
 $\forall p: pub (def?(p, dueos) \Rightarrow_L (\exists n: nat) (\langle p, n \rangle \in obtener(obtener(p, dueos), publicaciones)))$
  - Toda publicacin en *publicaciones* tiene su dueo en *dueos*:  
 $\forall u: usuario \forall p: pub (def?(u, publicaciones) \wedge_L (\exists n: nat) (\langle p, n \rangle \in obtener(u, publicaciones))) \Rightarrow_L$   
 $(def?(p, dueos) \wedge_L u = obtener(p, dueos))$
  - Toda oferta es de una publicacin vlida y cada ofertas es ofertada por un usuario vlido y su monto es menor que el valor de reserva:  
 $\forall p: pub (def?(p, ofertas) \Rightarrow$   
 $def?(p, dueos) \wedge_L (\exists n: nat) (\langle p, n \rangle \in obtener(obtener(p, dueos), publicaciones) \wedge$   
 $\forall i: nat (1 \leq i \leq longitud(obtener(p, ofertas)) \Rightarrow_L$   
 $\Pi_1(obtener(p, ofertas)[i]) \in usuarios \setminus \{obtener(p, dueos)\} \wedge \Pi_2(obtener(p, ofertas)[i]) < n))$
  - Las ofertas se guardan ordenadas:  
 $\forall p: pub (def?(p, ofertas) \Rightarrow_L \forall i: nat (1 \leq i < longitud(obtener(p, ofertas)) \Rightarrow_L$   
 $\Pi_2(obtener(p, ofertas)[i]) < \Pi_2(obtener(p, ofertas)[i + 1]))$

b) *Abs*:  $estr\ e \rightarrow REMATAZOS \quad \{ Rep(e) \}$

$Abs(e) = r /$

$usuarios(c) = e.usuarios \wedge_L$

$\forall u: usuario (u \in usuarios(r) \Rightarrow_L publicaciones(r, u) = \Pi_1(obtener(u, e.publicaciones)) \wedge$

$\forall p: pub (p \in publicaciones(r, u) \Rightarrow_L (\langle p, precioReserva(r, p) \rangle \in obtener(u, e.publicaciones))$   
 $\wedge ofertas(r, p) = obtener(p, e.ofertas))$

donde

$\Pi_1 : conj(tupla\langle \alpha; \beta \rangle) \longrightarrow conj(\alpha)$

$\Pi_1(c) \equiv \text{if } c = \emptyset? \text{ then } \emptyset \text{ else } \{\Pi_1(dameUno(c))\} \cup \Pi_1(sinUno(c)) \text{ fi}$