

Algoritmos y Estructuras de datos II - Trabajo Práctico 1

Grupo BlackMesa

TAD posición es tupla(int × int)

1. TAD Sokoban

TAD SOKOBAN

géneros sokoban

exporta observadores, generadores

usa PARTIDA, CONJ

igualdad observacional

$$(\forall s_1, s_2 : \text{sokoban}) \left(s_1 =_{\text{obs}} s_2 \iff \left(\begin{array}{l} \text{partidaActual}(s_1) =_{\text{obs}} \text{partidaActual}(s_2) \wedge \\ \text{niveles}(s_1) =_{\text{obs}} \text{niveles}(s_2) \wedge \\ \text{nivelesCompletados}(s_1) =_{\text{obs}} \text{nivelesCompletados}(s_2) \end{array} \right) \right)$$

observadores básicos

partidaActual : sokoban \longrightarrow partida

niveles : sokoban \longrightarrow conj(nivel)

nivelesCompletados : sokoban \longrightarrow conj(nivel)

generadores

sokoban : conj(nivel) \longrightarrow sokoban

axiomas $\forall s: \text{sokoban}, \forall pa: \text{partida}, \forall cn: \text{conj}(\text{nivel})$

partidaActual(sokoban(cn)) \equiv iniciarPartida(dameUno(cn))

niveles(sokoban(cn)) \equiv cn

nivelesCompletados(sokoban(cn)) \equiv \emptyset

Fin TAD

2. TAD Partida

TAD PARTIDA

géneros partida

exporta observadores, generadores

usa BOOL, NIVEL, POSICION, DIRECCION, NAT, PILA

igualdad observacional

$$(\forall pa_1, pa_2 : \text{partida}) \left(pa_1 =_{\text{obs}} pa_2 \iff \begin{pmatrix} \text{nivelActual}(pa_1) =_{\text{obs}} \text{nivelActual}(pa_2) \wedge \\ \text{posicionDelJugador}(pa_1) =_{\text{obs}} \text{posicionDelJugador}(pa_2) \wedge \\ (\forall p: \text{posicion}) \\ (\text{esTransitable}(pa_2, p) =_{\text{obs}} \text{esTransitable}(pa_1, p)) \wedge \\ \text{hayCaja}(pa_1, p) =_{\text{obs}} \text{hayCaja}(pa_2, p) \wedge \\ \text{estadosAnteriores}(pa_1) =_{\text{obs}} \text{estadosAnteriores}(pa_2) \end{pmatrix} \right)$$

observadores básicos

`posicionDelJugador` : partida \longrightarrow posicion

`nivelActual` : partida \longrightarrow nivel

`#bombas` : partida \longrightarrow nat

`esTransitable` : partida \times posicion \longrightarrow bool

`estadosAnteriores` : partida \longrightarrow pila(partida)

`hayCaja` : partida \times posicion \longrightarrow bool

generadores

`iniciarPartida` : nivel \longrightarrow partida

`moverJugador` : partida $pa \times$ direccion $d \longrightarrow$ partida

`ponerBomba` : partida $p \longrightarrow$ partida $\{\#bombas(pa) \neq 0\}$

`Undo` : partida \longrightarrow partida $\{\neg vacia?estadosAnteriores(pa)\}$

$$\left\{ \begin{array}{l} \text{esTransitable}(pa, \text{siguientePosicion}(d, \text{posicionDelJugador}(pa))) \wedge \\ ((\neg \text{hayCaja}(pa, \text{siguientePosicion}(d, \text{posicionDelJugador}(pa)))) \vee \\ (\text{esTransitable}(pa, \text{siguientePosicion}(d, \text{siguientePosicion}(d, \text{posicionDelJugador}(pa)))) \wedge \\ \neg \text{hayCaja}(pa, \text{siguientePosicion}(d, \text{siguientePosicion}(d, \text{posicionDelJugador}(pa)))))) \end{array} \right\}$$

otras operaciones

`siguientePosicion` : direccion $d \times$ posicion \longrightarrow posicion

axiomas

$\forall pa: \text{partida}, \forall n: \text{nivel}, \forall p: \text{posicion}, \forall d: \text{direccion}$

`posicionDelJugador(iniciarPartida(n))` \equiv `posicionInicial(n)`

`posicionDelJugador(moverJugador(pa, d))` \equiv `siguientePosicion(d, posicionDelJugador(pa))`

`posicionDelJugador(ponerBomba(pa))` \equiv `posicionDelJugador(pa)`

`posicionDelJugador(Undo(pa))` \equiv `posicionDelJugador(tope(estadosAnteriores(pa)))`

`nivelActual(iniciarPartida(n))` \equiv `n`

`nivelActual(moverJugador(pa, d))` \equiv `nivelActual(pa)`

`nivelActual(ponerBomba(pa))` \equiv `nivelActual(pa)`

nivelActual(Undo(pa))	\equiv nivelActual(pa)
#bombas(iniciarPartida(n))	\equiv bombasIniciales(n)
#bombas(moverJugador(pa, d))	\equiv #bombas(pa)
#bombas(ponerBomba(pa))	\equiv pred(#bombas(pa))
#bombas(Undo(pa))	\equiv #bombas(tope(estadosAnteriores(pa)))
esTransitable(iniciarPartida(n), p)	\equiv \neg tienePared(n, p)
esTransitable((moverJugador(pa, d), p))	\equiv esTransitable(pa, p)
esTransitable(ponerBomba(pa), p)	\equiv ($\pi_1(p) =_{\text{obs}} \pi_1(\text{posicionDelJugador}(pa))$) \vee ($\pi_2(p) =_{\text{obs}} \pi_2(\text{posicionDelJugador}(pa))$) \vee esTransitable(pa, p)
esTransitable(Undo(pa), p)	\equiv esTransitable(tope(estadosAnteriores(pa)), p)
hayCaja(iniciarPartida(n), p)	\equiv tieneCaja(n, p)
hayCaja(moverJugador(pa,d), p)	\equiv if $p =_{\text{obs}} \text{siguientePosicion}(d, \text{siguientePosicion}(d, \text{posicionDelJugador}(pa)))$ then hayCaja(pa, siguientePosicion(d, posicionDelJugador(pa))) else hayCaja(pa, p) fi
hayCaja(ponerBomba(pa), p)	\equiv hayCaja(pa, p)
hayCaja(Undo(pa), p)	\equiv hayCaja(tope(estadosAnteriores(pa)), p)
estadosAnteriores(iniciarPartida(n))	\equiv vacia
estadosAnteriores(moverJugador(pa, d), p)	\equiv apilar(pa, estadosAnteriores(pa))
estadosAnteriores(ponerBomba(pa))	\equiv apilar(pa, estadosAnteriores(pa))
estadosAnteriores(Undo(pa))	\equiv desapilar(estadosAnteriores(pa))
siguientePosicion(d, p)	\equiv if $\text{char?}(d) =_{\text{obs}} \text{'n'} \vee \text{char?}(d) =_{\text{obs}} \text{'s'}$ then $< \pi_1(p), \text{if } \text{char?}(d) =_{\text{obs}} \text{'n'} \text{ then } \pi_2(p) + 1 \text{ else } \pi_2(p) - 1$ fi $>$ else $< \text{if } \text{char?}(d) =_{\text{obs}} \text{'e'} \text{ then } \pi_1(p) + 1 \text{ else } \pi_1(p) - 1$ fi $, \pi_2(p) >$ fi

Fin TAD

3. TAD Nivel

TAD NIVEL

géneros nivel

exporta observadores, generadores

usa BOOL, POSICION

igualdad observacional

$$(\forall n_1, n_2 : \text{nivel}) \left(n_1 =_{\text{obs}} n_2 \iff \left((\forall p : \text{posicion}) \left(\begin{aligned} &\{ \text{tienePared}(n_1, p) =_{\text{obs}} \text{tienePared}(n_2, p) \} \wedge \\ &\text{esDeposito}(n_1, p) =_{\text{obs}} \text{esDeposito}(n_1, p) \wedge \\ &\text{tieneCaja}(n_1, p) =_{\text{obs}} \text{tieneCaja}(n_1, p) \} \wedge \\ &\text{posicionInicial}(n_1) =_{\text{obs}} \text{posicionInicial}(n_2) \wedge \\ &\text{bombasIniciales}(n_1) =_{\text{obs}} \text{bombasIniciales}(n_2) \end{aligned} \right) \right) \right)$$

observadores básicos

tienePared : nivel \times posicion \longrightarrow bool

esDeposito : nivel \times posicion \longrightarrow bool

tieneCaja : nivel \times posicion \longrightarrow bool

posicionIncial : nivel \longrightarrow posicion

bombasIniciales: nivel \longrightarrow nat

generadores

nivel : posicion \times nat \longrightarrow nivel

ponerDeposito : posicion p \longrightarrow nivel $\{\neg \text{tienePared}(p)\}$

ponerPared : posicion p \longrightarrow nivel $\{\neg \text{esDeposito}(p) \wedge \neg \text{tieneCaja}(p) \wedge \neg \text{tienePared}(p) \wedge p \neq \text{posicionIncial}(n)\}$

ponerCaja : posicion p \longrightarrow nivel $\{\neg \text{tienePared}(p) \wedge \neg \text{tieneCaja}(p) \wedge p \neq \text{posicionIncial}(n)\}$

axiomas $\forall n: \text{nivel}, \forall p, p': \text{posicion}$

tienePared(nivel(p, n), p') \equiv False

tienePared(ponerDeposito(p), p') \equiv **if** $p = p'$ **then** false **else** **tienePared**(p') **fi**

tienePared(ponerPared(p), p') \equiv **if** $p = p'$ **then** true **else** **tienePared**(p') **fi**

tienePared(ponerCaja(p), p') \equiv **if** $p = p'$ **then** false **else** **tienePared**(p') **fi**

esDeposito(nivel(p, n), p) \equiv false

esDeposito(ponerDeposito(p), p') \equiv **if** $p = p'$ **then** true **else** **esDeposito**(p') **fi**

esDeposito(ponerPared(p), p') \equiv **if** $p = p'$ **then** false **else** **esDesposito**(p') **fi**

esDeposito(ponerCaja(p), p') \equiv **esDeposito**(p')

tieneCaja(nivel(p, n), p) \equiv false

tieneCaja(ponerDepostio(p), p') \equiv **tieneCaja**(p')

tieneCaja(ponerPared(p), p') \equiv **if** $p = p'$ **then** false **else** **tieneCaja**(p') **fi**

tieneCaja(ponerCaja(p), p') \equiv **if** p = p' **then** true **else** tieneCaja(p') **fi**

bombasIniciales(nivel(p ,n)) \equiv n

Fin TAD

4. TAD Direccion

TAD Direccion

géneros dir

usa CHAR

exporta observadores, generadores

igualdad observacional

$(\forall dir_1, dir_2 : \text{dir}) \ (dir_1 =_{\text{obs}} dir_2 \iff (\text{char?}(dir_1) =_{\text{obs}} \text{char?}(dir_2)))$

observadores básicos

char? : dir \longrightarrow char

generadores

Norte : \longrightarrow dir

Sur : \longrightarrow dir

Este : \longrightarrow dir

Oeste : \longrightarrow dir

axiomas $\forall d: \text{dir}$

char?(Norte) \equiv 'n'

char?(Sur) \equiv 's'

char?(Este) \equiv 'e'

char?(Oeste) \equiv 'o'

Fin TAD