

Algoritmos y Estructuras de datos II - Trabajo Práctico 1

Grupo BlackMesa

posición es tupla(int × int)

dirección es { “n”, “s”, “e”, “o” }

1. TAD Sokoban

TAD SOKOBAN

géneros sokoban

exporta generadores, observadores

usa CELDA, BOOL

igualdad observacional

$$(\forall s_1, s_2 : \text{sokoban}) \left(s_1 =_{\text{obs}} s_2 \iff \left(\begin{array}{l} (\forall p : \text{posicion}) \quad (\text{hayCeldaEn}(s_1, p) \Rightarrow \text{hayCeldaEn}(s_2, p)) \wedge_{\text{L}} (\text{celdaEn}(s_1, p) =_{\text{obs}} \text{celdaEn}(s_2, p)) \\ (\text{existeJugador}(s_1) \Rightarrow \text{existeJugador}(s_2)) \wedge_{\text{L}} (\text{posicionDelJugador}(s_1) =_{\text{obs}} \text{posicionDelJugador}(s_2)) \end{array} \right) \right)$$

observadores básicos

hayCeldaEn	: sokoban × posicion	→ bool	
celdaEn	: sokoban s × posicion p	→ celda	{hayCeldaEn(s,p)}
posicionDelJugador	: sokoban s × posicion p	→ celda	{existeJugador(s)}
existeJugador	: sokoban	→ bool	

generadores

Sokoban	:	→ sokoban	
PonerCelda	: sokoban s × posicion p × celda	→ sokoban	{¬hayCelda(s, p)}
PonerJugador	: sokoban s × posicion p	→ sokoban	{hayCeldaEn(s,p) ∧ esTransitable(celdaEn(s, p) ∧ ¬tieneCaja(celdaEn(s, p)) ∧ ¬existeJugador(s)}
moverJugador	: sokoban s × direccion p	→ sokoban	{esTransitable(celdaEn(siguientePosicion(s, posicionDelJugador(s), d))) ∧ (tieneCaja(siguientePosicion(s, posicionDelJugador(s), d)) ⇒ esTransitablePorCaja(siguientePosicion(s, siguientePosicion(s, posicionDelJugador(s), d), d)))}

otras operaciones

siguientePosicion	: sokoban × posicion × direccion	→ posicion
ganó	: sokoban	→ bool

axiomas $\forall s$: sokoban, $\forall p$: posicion, $\forall d$: direccion, $\forall c$: celda

hayCeldaEn(Sokoban, p)	\equiv	false
hayCeldaEn(PonerCelda(s, p_1 , c), p_2)	\equiv	if ($p_1 = p_2$) then true else hayCeldaEn(s, p_2) fi

$\text{hayCeldaEn}(\text{PonerJugador}(s, p_1), p_2) \equiv \text{if } (p_1 = p_2) \text{ then } true \text{ else } \text{hayCeldaEn}(s, p_2) \text{ fi}$
 $\text{hayCeldaEn}(\text{moverJugador}(s, d), p) \equiv \text{hayCelda}(s, p)$
 $\text{celdaEn}(\text{PonerCelda}(s, p_2, c), p_2) \equiv \text{if } (p_1 = p_2) \text{ then } c \text{ else } \text{celdaEn}(s, p_2) \text{ fi}$
 $\text{celdaEn}(\text{PonerJugador}(s, p_1), p_2) \equiv \text{celdaEn}(s, p_2)$
 $\text{celdaEn}(\text{moverJugador}(s, d), p) \equiv \text{celdaEn}(s, p)$
 $\text{posicionDelJugador}(\text{PonerCelda}(s, p, c)) \equiv \text{posicionDelJugador}(s)$
 $\text{posicionDelJugador}(\text{PonerJugador}(s, p)) \equiv p$
 $\text{posicionDelJugador}(\text{moverJugador}(s, d)) \equiv \text{siguientePosicion}(s, \text{posicionDelJugador}(s), d)$

Fin TAD

2. TAD Celda

TAD CELDA

géneros Celda

exporta celda, observadores, generadores, esTransitablePorCaja

usa Bool

igualdad observacional

$$(\forall c_1, c_2 : \text{celda}) \left(c_1 =_{\text{obs}} c_2 \iff \left(\begin{array}{l} \text{esTransitable}(c_1) =_{\text{obs}} \text{esTransitable}(c_2) \wedge \\ \text{esDeposito}(c_1) =_{\text{obs}} \text{esDeposito}(c_2) \wedge \\ \text{tieneCaja}(c_1) =_{\text{obs}} \text{tieneCaja}(c_2) \end{array} \right) \right)$$

observadores básicos

$\text{esTransitable} : \text{celda} \longrightarrow \text{bool}$

$\text{esDeposito} : \text{celda} \longrightarrow \text{bool}$

$\text{tieneCaja} : \text{celda} \longrightarrow \text{bool}$

generadores

$\text{generar} : \longrightarrow \text{celda}$

$\text{ponerDeposito} : \text{celda } c \longrightarrow \text{celda} \quad \{\text{esTransitable}(c)\}$

$\text{ponerPared} : \text{celda } c \longrightarrow \text{celda} \quad \{\neg \text{esDeposito}(c) \wedge \neg \text{tieneCaja}(c) \wedge \text{esTransitable}(c)\}$

$\text{ponerCaja} : \text{celda } c \longrightarrow \text{celda} \quad \{\text{esTransitable}(c) \wedge \neg \text{tieneCaja}(c)\}$

otras operaciones

$\text{esTransitablePorCaja} : \text{celda} \longrightarrow \text{bool}$

axiomas $\forall c: \text{celda}$

$\text{esTransitable}(\text{generar}) \equiv true$

$\text{esTransitable}(\text{ponerPared}(c)) \equiv false$

$\text{esTransitable}(\text{ponerDeposito}(c)) \equiv true$

<code>esTransitable(ponerCaja(c))</code>	\equiv <i>true</i>
<code>esDeposito(generar)</code>	\equiv <i>false</i>
<code>esDeposito(ponerPared(c))</code>	\equiv <i>false</i>
<code>esDeposito(ponerDeposito(c))</code>	\equiv <i>true</i>
<code>esDeposito(ponerCaja(c))</code>	\equiv <code>esDeposito(c)</code>
<code>tieneCaja(generar)</code>	\equiv <i>false</i>
<code>tieneCaja(ponerPared(c))</code>	\equiv <i>false</i>
<code>tieneCaja(PonerDeposito(c))</code>	\equiv <code>tieneCaja(c)</code>
<code>tieneCaja(PonerCaja(c))</code>	\equiv <i>true</i>
<code>esTransitablePorCaja(generar)</code>	\equiv <i>true</i>
<code>esTransitablePorCaja(ponerPared(c))</code>	\equiv <i>false</i>
<code>esTransitablePorCaja(ponerDeposito(c))</code>	\equiv <code>esTransitablePorCaja(c)</code>
<code>esTransitablePorCaja(ponerCaja(c))</code>	\equiv <i>false</i>

Fin TAD