

Algo 2 - TP 1



3.1. TP 1 — fecha de entrega: domingo 27 de septiembre

Este trabajo consiste en especificar el juego de ingenio Sokoban³, diseñado en 1981 por Hiroyuki Imabayashi.

Reglas del juego. El juego tiene lugar en una grilla infinita dividida en celdas de 1×1 . El jugador controla una persona que puede moverse de a una celda por vez en cualquiera de las cuatro direcciones (Norte, Este, Sur y Oeste). Algunas celdas de la grilla tienen paredes por las que la persona no puede pasar. Las celdas que no tienen paredes son transitables. Algunas de las celdas transitables están marcadas como depósitos. En las celdas transitables de la grilla puede haber cajas. Si la persona se mueve en dirección hacia una celda en la que hay una caja, se mueve hacia esa dirección y además *empuja* la caja hacia dicha dirección. Esta acción de *empujar* una caja solamente se puede realizar si la celda a la que debe ir a parar la caja no tiene una pared ni otra caja. En un nivel del juego suponemos que hay una cierta cantidad k de cajas y el mismo número k de depósitos. El objetivo del juego es ubicar cada caja sobre un depósito.

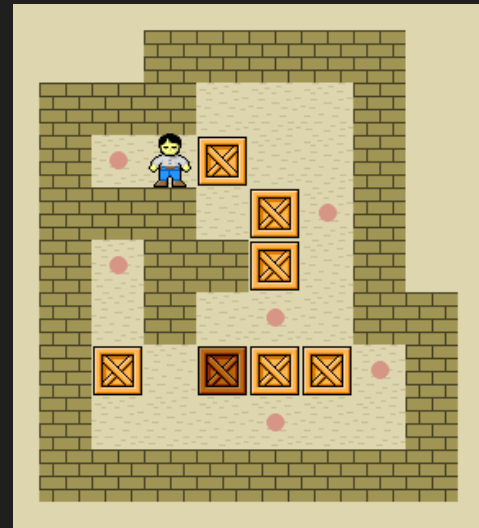
Algunas aclaraciones: (1) la grilla es infinita, pero sólo puede haber un número finito de paredes, cajas y depósitos; (2) no puede haber dos cajas, ni dos paredes, ni dos depósitos en una misma celda; (3) una caja no puede estar en la misma celda que una pared; (4) un depósito no puede estar en la misma celda que una pared; (5) la persona no puede estar en la misma celda que una caja ni que una pared.

Grilla Infinita


- Celdas : 1×1
- Acciones : 4 {N, S, E, O}
- Paredes : no transitables


Celdas sin paredes : Transitables

- Depósito : celda transitable
- Cajas : ubicadas en las celdas transitables



Empujar caja

sin pared 

no otra caja 

Nivel Es Diccionario

Estado Es Diccionario

TAD Física (Nivel)

-- Estado con propiedades físicas
igualdad obs

param formater
géneros

géneros físicos

exporta

usa

Observadores Básicos

computar Estado : física \rightarrow estado

Otras Operaciones

personaje : física \rightarrow personaje

cajas : física \rightarrow conj (caja)

depósitos : física \rightarrow conj (depósito)

paredes : física \rightarrow conj (pared)

Generadores

crear Física : Nivel $n \rightarrow$ física $\{ \text{nivel Válido?}(n) \}$
 \uparrow Estado inicial

simular Acción : física \times Acción \rightarrow física

\uparrow simulador de física sobre el escenario dado

Axiomas

$$\begin{aligned} \text{personaje} \left(\text{crear F\u00edsica} \left(\overset{\text{nivel}}{n} \right) \right) &\equiv n[\text{"personaje"}] \\ \text{cajas} \left(\text{crear F\u00edsica} \left(n \right) \right) &\equiv n[\text{"caja"}] \\ \text{dep\u00f3sitos} \left(\text{crear F\u00edsica} \left(n \right) \right) &\equiv n[\text{"dep\u00f3sito"}] \\ \text{paredes} \left(\text{crear F\u00edsica} \left(n \right) \right) &\equiv n[\text{"pared"}] \end{aligned}$$

$$\text{paredes} \left(\text{sim Acci\u00f3n} \left(f, a \right) \right) \equiv \text{paredes} (f)$$

$$\text{dep\u00f3sitos} \left(\text{sim Acci\u00f3n} \left(f, a \right) \right) \equiv \text{dep\u00f3sitos} (f)$$

↑ nunca se mueven

$$\text{personaje} \left(\text{sim Acci\u00f3n} \left(f, a \right) \right) \equiv$$

if $a = \text{Arriba}$ then

$$\text{accionar Personaje} \left(f, (0, -1) \right)$$

else if $a = \text{Abajo}$ then

$$\text{accionar Personaje} \left(f, (0, 1) \right)$$

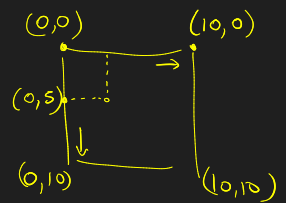
else if $a = \text{Izquierda}$ then

$$\text{accionar Personaje} \left(f, (-1, 0) \right)$$

else if $a = \text{Derecha}$ then

$$\text{accionar Personaje} \left(f, (1, 0) \right)$$

else $f; f; f; f;$



[sigue axiomas]

$\text{cajas}(\text{sim Acción}(f, a)) \equiv$

if $a = \text{Arriba}$ then

 accionarCaja($f, (0, -1)$)

else if $a = \text{Abajo}$ then

 accionarCaja($f, (0, 1)$)

else if $a = \text{Izquierda}$ then

 accionarCaja($f, (-1, 0)$)

else if $a = \text{Derecha}$ then

 accionarCaja($f, (1, 0)$)

else $f; f; f; f;$


accionar Caja : $f_{\text{size}} \times \text{tupla}(\text{nat}, \text{nat}) \rightarrow \text{conj}(\text{caja})$

accionar Caja (f , peso) \equiv

if esCaja? ($\text{personaje}(f) + \text{peso}$) then

if estsLibre? ($\text{personaje}(f) + \text{peso} + \text{peso}$) then

-- Tambores ~~...~~

-- Nuevo la caja! 


moverCaja($\text{personaje}(f) + \text{peso}$, $\text{cajar}(f)$, peso)

else

cajas(f)

fi

else

fi -- solo me importan las cajas... 

accionar Personaje : $física \times tuple(nat, nat)$

accionar Personaje (f , $paso$) \equiv

if $est\acute{a} Libre? (personaje(f) + paso)$ then

-- me muevo

$personaje(f) + paso$

else

if $es Pared? (personaje(f) + paso)$ then

-- no se mueve, yo tampoco

$personaje(f)$

else

-- es Caja! veo si se puede mover

if $est\acute{a} Libre? (personaje(f) + paso + paso)$ then

$personaje(f) + paso$ -- no muevo la caja!

else

-- no se mueve, yo tampoco

$personaje(f)$

f;

f,

f;

← nivel \equiv estado
pero nivel es siempre el
estado inicial

$$\text{Dict} \left(\begin{array}{l} \text{"personaje"} \\ \text{"cajas"} \\ \text{"paredes"} \\ \text{"depósitos"} \end{array} \right), \left(\begin{array}{l} \text{personaje (sim Acción (f, a))} \\ \text{cajas (sim Acción (f, a))} \\ \text{paredes (sim Acción (f, a))} \\ \text{depósitos (sim Acción (f, a))} \end{array} \right),$$

$\text{moverCaja} : \text{Posición} \times \text{conj}(\text{Caja}) \times \text{tupla}(\text{nat}, \text{nat})$

$\text{moverCaja}(\text{pos}, \text{cajas}, \text{peso}) \equiv$

if $\text{demeUno}(\text{cajas}) = \text{pos}$ then

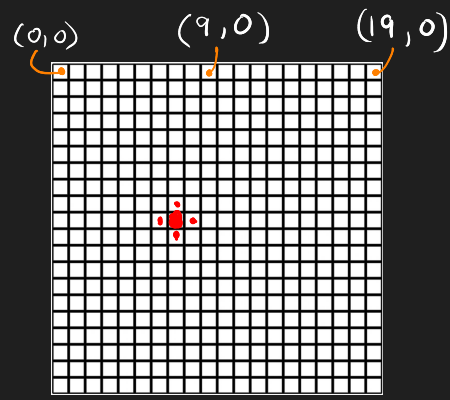
$\text{Ag}(\text{pos} + \text{peso}, \text{sinUno}(\text{cajas}))$
else

$\text{Ag}(\text{demeUno}(\text{cajas}),$

$\text{moverCaja}(\text{pos}, \text{sinUno}(\text{cajas}), \text{peso}))$

fi

TAD Juego (Nivel)



Igualdad obar
()
parámetros formales
generar

generos
juego

exporte

usa bool, personaje, caja, depósito, pared, física

Observadores básicos

estado : juego \rightarrow Estado

Generadores

reset : Nivel $n \rightarrow$ juego $\{ \text{nivel Válido?}(n) \}$

step : Juego \times Acción \rightarrow juego

Axiomas

estado (reset(n)) \equiv
Física.computarEstado (Física.crearFísica(n))

estado (step(j , a)) \equiv
Física.computarEstado (Física.simularAcción(a))

Otras Operaciones

completado? : juego \rightarrow bool

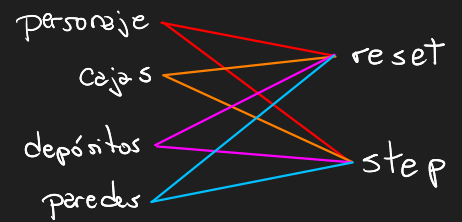
↙ si ganó o no

completados : juego \rightarrow conj(depósitos)

estado Válida? : Juego x Grilla \rightarrow bool

está Libre? : Juego x Posicion \rightarrow bool

Recordo:



Decisión de modelado

- Como todas las cajas son iguales y solo se diferencian en su posición:

↳ Renombro a **Posición** como

Donde **Posición** \equiv TUPLA (NAT, NAT)

- Lo mismo con **Depósito**
Pared

Renombrar

TAD **Posición** ES TUPLA (NAT, NAT)

TAD **Caja** ES **Posición**

TAD **Depósito** ES **Posición**

TAD **Pared** ES **Posición**

TAD **Personaje** ES **Posición**

? TAD (**Acción**) ES STRING $S \in \left\{ \begin{array}{l} \text{"Arriba", Abajo,} \\ \text{Izq., Derecha} \end{array} \right\}$

Arriba ES "Arriba"

Enum = if e = "Arriba", :

TAD Posición ES TUPLA (net, net)

TAD CAJA

igualdad obs

param formater
géneros

géneros caja

exporta

usa

Observador Básico

posición : juego x caja \rightarrow Posición