

Sava
Dragos
342C4

Predicate:

location(Room)
spheres(Color, Room, N)
color(Room, Color)
color(Color)
door(Room1, Room2)
carries(Color, N)
succ(N1, N2)
greater(N1, N2)
positive(N)

Voi defini urmatorul predicat ajutorator:

total_carries(N) //reprezinta numarul total de bile incarcate de robot
 //N = N1+N2+N3 , unde
 //carries(red,N1)
 //carries(blue,N2)
 //carries(gray,N3)

Operatori

Move(Room1, Room2)

LP: location(Room1) AND
 door(Room1,Room2) AND
 carries(Color,N1) AND
 greater(3,N) AND //N trebuie sa fie 0 sau 2 asa ca l-am definit ca
 not succ(N,2) //N < 3 si succesorul lui nu poate fi 2(deci N != 1)
LE: location(Room2)
LA: location(Room1)

Load(Color)

LP: location(Room) AND
total_carries(N1) AND //nu trebuie sa avem mai mult de o bila incarcata
greater(2,N1) AND
spheres(Color,Room1,N2) AND
positive(N2) AND //exista in camera cel putin o bila de culoarea
dorita care poate fi incarcata
not color(Room,Color) //nu putem incarca bile din depozit(warehouse)
ce au culoarea asemenea lui
LE: total_carries(N1) AND //acum exista posibilitatea de a avea 2 bile
incarcate
greater(2,N1)
LA: carries(Color,N) AND
positive(N) AND //acum stim cu siguranta in robotul are incarcata cel
putin o bila de culoare Color

Unload(Color)

LP: location(Room1) AND
carries(Color,N) AND
positive(N)
LE: carries(Color,N) AND positive(N) //nu mai avem garantia ca mai este vreo
bila incarcata
LA: spheres(Color,Room1,N2) AND
positive(N2) AND //acum stim ca in camera se va afla cel putin o bila
pozitiva
carries(Color,N1),
greater(2,N1) //deoarece am descarcata o sfera , suntem siguri ca
numarul de sfere incarcate
//este < 2 (adica 0 sau 1)

Legenda:

LP = Lista Preconditiilor ce contine formulele care trebuie sa fie adevarate intr-o stare a problemei pentru ca operatorul sa poata fi aplicat

LA = Lista Adaugarilor ce contine formulele care vor deveni adevarate dupa aplicarea operatorului

LE = Lista Eliminarilor ce contine formulele care vor deveni false dupa aplicarea operatorului