

Esercizi sulla pianificazione

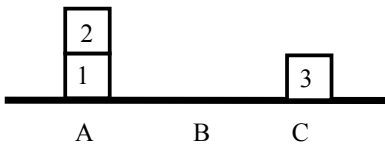
Esercizio 1

Si supponga di avere un mondo a blocchi così strutturato: abbiamo 3 blocchi etichettati 1, 2 e 3 che possono essere posizionati in 3 posizioni sul tavolo A, B e C. I blocchi 1 e 3 non possono MAI toccarsi, quindi non sono ammissibili quelle configurazioni in cui il blocco 1 è immediatamente sopra al 3 e viceversa.

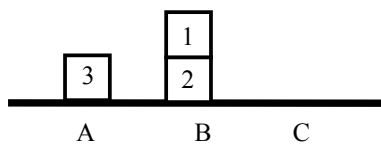
Inoltre, nella posizione C del tavolo si può piazzare al MASSIMO UN BLOCCO.

Si disegnino i nodi aperti esplorando in AMPIEZZA lo spazio di ricerca del problema partendo dal nodo I fino alla prima soluzione G.

I:



G:

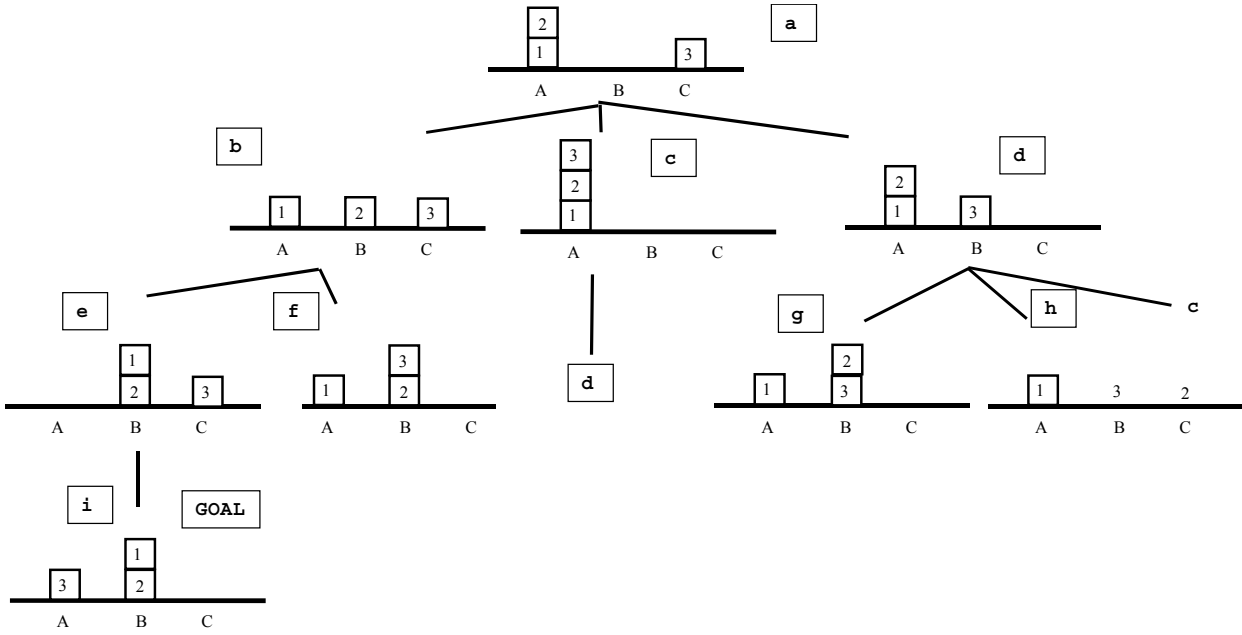


Le possibili mosse sono quelle del mondo a blocchi. Possono essere mossi solo i pezzi liberi (ossia quelli che non hanno pezzi sovrastanti) e la posizione finale può essere o il tavolo, se esiste una posizione libera su di esso, oppure un altro pezzo. L'ordine in cui vengono generati i nodi è il seguente: se possibile, prima viene mosso il pezzo 1, poi il 2 e poi il 3. Inoltre, se possibile vengono prima esplorati i movimenti che portano sulla posizione A poi B e poi C.

Si supponga inoltre di non poter visitare un nodo già visitato SULLO STESSO RAMO per evitare loop infiniti.

Infine, per comodità si possono etichettare i nodi con label uniche in modo da indicare con la label nodi già visitati precedentemente (su rami diversi).

SOLUZIONE



Esercizio 2

È dato lo stato iniziale descritto dalle seguenti formule atomiche:

[ontable(a,p1), ontable(d,p3), on(c,d), clear(a), clear(c), handempty]

(a,c,d rappresentano dei blocchi e p1,p2,p3 sono le uniche 3 posizioni occupabili del tavolo

- le azioni sono modellate opportunamente come segue:

pickup(X,Pos)

PRECOND: ontable(X,Pos), clear(X), handempty

DELETE: ontable(X,Pos), clear(X), handempty

ADD: holding(X), empty(Pos)

putdown(X,Pos)

PRECOND: holding(X), empty(Pos)

DELETE: holding(X), empty(Pos)

ADD: ontable(X,Pos), clear(X), handempty

stack(X,Y)

PRECOND: holding(X), clear(Y)

DELETE: holding(X), clear(Y)

ADD: handempty, on(X,Y), clear(X)

unstack(X,Y)

PRECOND: handempty, on(X,Y), clear(X)

DELETE: handempty, on(X,Y), clear(X)

ADD: holding(X), clear(Y)

(Pos è una variabile che rappresenta la posizione del tavolo, X,Y rappresentano blocchi)
e il goal **ontable(a,p2)**

si descriva come l'algoritmo lineare backward STRIPS trova un piano per questo goal (quindi si mostri SOLO una strada nell'albero di ricerca considerando come ordinamento tra precondizioni in and nello stack quello che considera prima le precondizioni che unificano direttamente con un letterale dello stato e poi quelle che per essere soddisfatte hanno bisogno di una azione). Si descriva lo stato e lo stack dei goal passo passo.