
Έστω μια εκδοχή προβλήματος του κόσμου των κύβων, όπου διαθέτουμε έναν ρομποτικό βραχίονα για τις μετακινήσεις των κύβων. Ο βραχίονας μπορεί να μετακινεί έναν μόνο κύβο ανά πάσα στιγμή. Έχουμε τις εξής δυνατές ενέργειες:

PickUp(x): Ο ελεύθερος βραχίονας σηκώνει τον κύβο x που βρισκόταν στο τραπέζι.

Προϋποθέσεις: $Clear(x), OnTable(x), Handempty$.

Λίστα διαγραφής: $Clear(x), OnTable(x), Handempty$.

Λίστα προσθήκης: $Holding(x)$.

PutDown(x): Ο βραχίονας που κρατούσε τον κύβο x , τον αφήνει στο τραπέζι.

Προϋποθέσεις: $Holding(x)$.

Λίστα διαγραφής: $Holding(x)$.

Λίστα προσθήκης: $Clear(x), OnTable(x), Handempty$.

Stack(x,y): Ο βραχίονας που κρατούσε τον κύβο x τον αφήνει πάνω στον κύβο y .

Προϋποθέσεις: $Holding(x), Clear(y)$.

Λίστα διαγραφής: $Holding(x), Clear(y)$.

Λίστα προσθήκης: $Clear(x), On(x,y), Handempty$.

Unstack(x,y): Ο ελεύθερος βραχίονας πιάνει τον κύβο x που βρισκόταν πάνω στον κύβο y .

Προϋποθέσεις: $On(x,y), Clear(x), Handempty$.

Λίστα διαγραφής: $On(x,y), Clear(x), Handempty$.

Λίστα προσθήκης: $Holding(x), Clear(y)$.

Στις παραπάνω περιγραφές ενεργειών η σημασία των κατηγορημάτων που χρησιμοποιούνται είναι η εξής:

$Clear(x)$: Ο κύβος x δεν έχει άλλο κύβο από πάνω του.

$OnTable(x)$: Ο κύβος x είναι στο τραπέζι.

$On(x,y)$: Ο κύβος x βρίσκεται πάνω στον κύβο y .

$Handempty$: Ο βραχίονας δεν κρατά κανέναν κύβο.

$Holding(x)$: Ο βραχίονας κρατά τον κύβο x .

Εφαρμόστε τον αλγόριθμο σχεδιασμού με προς τα πίσω αναζήτηση στο χώρο των καταστάσεων (οπισθοχώρηση, regression) για να λύσετε ένα πρόβλημα με τρεις κύβους, A , B και C , και αρχική κατάσταση και στόχους ως εξής:

Αρχική κατάσταση:

$Clear(A), Clear(C), OnTable(B), On(A,B), OnTable(C), Handempty$

Στόχοι:

$On(A,C), On(B,A)$

Θεωρείστε ότι σε κάθε βήμα του αλγορίθμου επιλέγεται η σωστή ενέργεια, με αποτέλεσμα να μην χρειάζεται ποτέ να γίνει υπαναχώρηση (backtracking). Δώστε ένα-προς-ένα τα βήματα του αλγορίθμου.

Απάντηση:

Παρακάτω δίνονται εναλλάξ το τρέχον σύνολο στόχων και η εκάστοτε ενέργεια που προστίθεται στο πλάνο, από την τελευταία προς την πρώτη. Σε κάθε βήμα η ενέργεια που προστίθεται θα πρέπει να είναι τέτοια

ώστε να πετυχαίνει τουλάχιστον έναν στόχο από το τρέχον σύνολο στόχων χωρίς να διαγράφει κανέναν άλλο. Οι στόχοι που πετυχαίνει η ενέργεια αφαιρούνται από το σύνολο στόχων, ενώ οι προϋποθέσεις της ενέργειας προστίθενται στο σύνολο στόχων. Η διαδικασία ολοκληρώνεται όταν το τρέχον σύνολο στόχων γίνει υποσύνολο της αρχικής κατάστασης.

Τρέχον σύνολο στόχων:

$On(A,C), On(B,A)$

Ενέργεια: **Stack(B,A)** πετυχαίνει το $On(B,A)$ (αλλά και το αδιάφορο $Clear(B)$ και το $Handempty$)

Νέο σύνολο στόχων (με τις προϋποθέσεις $Holding(B), Clear(A)$):

$On(A,C), Holding(B), Clear(A)$

Ενέργεια: **PickUp(B)** πετυχαίνει το $Holding(B)$

Νέο σύνολο στόχων (με τις προϋποθέσεις $Clear(B), OnTable(B), Handempty$):

$On(A,C), Clear(A), Clear(B), OnTable(B), Handempty$

Ενέργεια: **Stack(A,C)** πετυχαίνει τα $Clear(A), On(A,C), Handempty$

Νέο σύνολο στόχων (με τις προϋποθέσεις $Holding(A), Clear(C)$) :

$Clear(B), OnTable(B), Holding(A), Clear(C)$.

Ενέργεια: **Unstack(A,B)** πετυχαίνει τα $Holding(A), Clear(B)$

Τρέχον σύνολο στόχων (με τις προϋποθέσεις $On(A,B), Clear(A), Handempty$):

$OnTable(B), Clear(C), On(A,B), Clear(A), Handempty$

Στο σημείο αυτό η αναζήτηση για το πλάνο ολοκληρώνεται, μιας και όλα τα γεγονότα του τρέχοντος συνόλου στόχων συμπεριλαμβάνονται στην αρχική κατάσταση. Άρα το πλάνο που βρέθηκε αποτελείται από τις ενέργειες που αναφέρονται παραπάνω, σε ανάστροφη σειρά.

Σημειώνεται ότι καμία από τις παραπάνω ενέργειες που επελέγησαν δεν διέγραφε κάτι από το αμέσως προηγούμενο σύνολο στόχων.