sdi1900085

Project 1

Πρόβλημα 2: Από την αναζήτηση με αλγόριθμο πρώτα σε βάθος με επαναληπτική εκβάθυνση σε ~~[illegible]~~ πεπερασμένο δένδρο με βάθος d και παράγοντα διακλάδωσης b, δημιουργούνται κόμβοι:

$$(d+1) + db + (d-1)b^2 + \dots + 2b^{d-1} + 1b^d$$

Άρα για κατάσταση στόχου σε βάθος $g \le d$ έχουμε:

• Για $g=0$ τον μικρότερο δυνατό αριθμό κόμβων:

$$1 - b^2 + \dots + 2b^{-1} + 1 = -b^2 + \dots + 2b^{-1} + 2$$

• Για $g=d$ τον μεγαλύτερο δυνατό αριθμό κόμβων:

$$(d+1) + db + (d-1)b^2 + \dots + 2b^{d-1} + b^d$$

Πρόβλημα 3: α) Έχουμε: $h(n) \le c(n,a,n') + h(n')$, όπου n' απόγονος του n που παράγεται από ενέργεια a

• o103: $\rightarrow$ ts: $h(o103) \le c(o103, a, ts) + h(ts)$

$\Rightarrow 21 \le 8 + 23 = 31$ που ισχύει

$\rightarrow$ b3: $h(o103) \le c(o103, a, b3) + h(b3)$

$\Rightarrow 21 \le 4 + 17 = 21$ που ισχύει

$\rightarrow$ o109: $h(o103) \le c(o103, a, o109) + h(o109)$

$21 \le 12 + 24 = 36$ που ισχύει

• ts: $\rightarrow$ mail: $h(ts) \le c(ts, a, mail) + h(mail)$

$23 \le 6 + 26$ που ισχύει

• b3: $\rightarrow$ b1: $h(b3) \le c(b3, a, b1) + h(b1)$

$17 \le 4 + 13 = 17$ που ισχύει

$\rightarrow b4: \ h(b3) \leq c(b3, a, b4) + h(b4)$

$17 \leq 7 + 18$ που ισχύει

$\cdot\, b1: \rightarrow c2: \ h(b1) \leq c(b1, a, c2) + h(c2)$

$13 \leq 3 + 10$ που ισχύει

$\rightarrow b2: \ h(b1) \leq c(b1, a, b2) + h(b2)$

$13 \leq 6 + 15$ που ισχύει

$\cdot\, c2: \rightarrow c1: \ h(c2) \leq c(c2, a, c1) + h(c1)$

$10 \leq 4 + 6$ που ισχύει

$\rightarrow c3: \ h(c2) \leq c(c2, a, c3) + h(c3)$

$10 \leq 6 + 12$ που ισχύει

$\cdot\, c1: \rightarrow c3: \ h(c1) \leq c(c1, a, c3) + h(c3)$

$6 \leq 8 + 12$ που ισχύει

$\cdot\, b2: \rightarrow b4: \ h(b2) \leq c(b2, a, b4) + h(b4)$

$15 \leq 3 + 18$ που ισχύει

$\cdot\, b4: \rightarrow o109: \ h(b4) \leq c(b4, a, o109) + h(o109)$

$18 \leq 7 + 29$ που ισχύει

$\cdot\, o109: \rightarrow o119: \ h(o109) \leq c(o109, a, o119) + h(o119)$

$29 \leq 16 + 11$ που ισχύει

$\rightarrow o111: \ h(o109) \leq c(o109, a, o111) + h(o111)$

$29 \leq 4 + 27$ που ισχύει

$\cdot\, o119: \rightarrow o123: \ h(o119) \leq c(o119, a, o123) + h(123)$

$11 \leq 9 + 4$ που ισχύει

$\longrightarrow$ storage: $h(o119) \leq c(o119, a, storage) + h(storage)$

$11 \leq 7 + 12$ που ισχύει

• o123: $\longrightarrow$ o125: $h(o123) \leq c(o123, a, o125) + h(o125)$

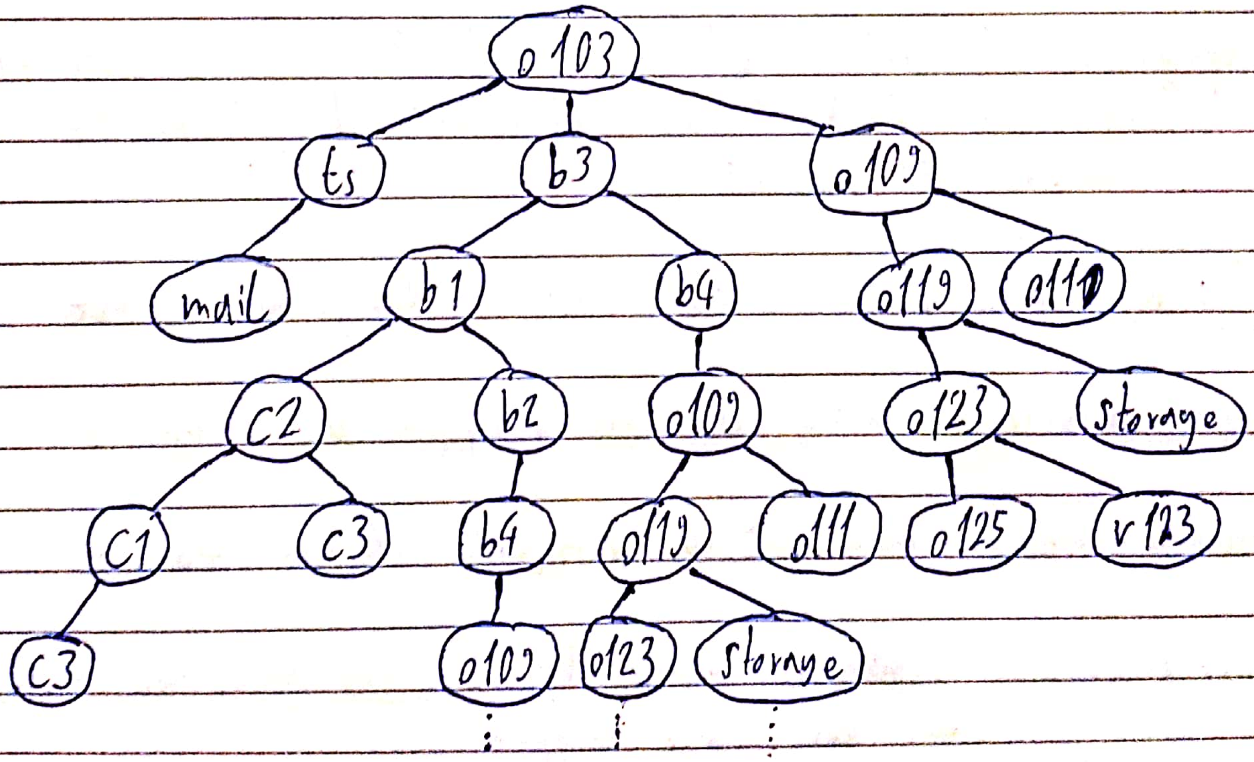
$4 \leq 4 + 6$ που ισχύει

$\longrightarrow$ r123: $h(o123) \leq c(o123, a, r123) + h(r123)$

$4 \leq 4$ που ισχύει

Άρα ισχύει η σχέση $h(n) \leq c(n,a,n') + h(n')$ για όλους τους κόμβους. Επομένως η ευρετική συνάρτηση είναι συνεπής και εφόσον είναι συνεπής είναι και παραδεκτή

β)



state	h(n)
mail	26
ts	23
o103	21
o109	24
o111	27
o119	11
o123	4
o125	6
r123	0
b1	13
b2	15
b3	17
b4	18
c1	6
c2	10
c3	12
storage	12

• Πρώτα σε πλάτος: o103, ts, b3, o109, mail, b1, b4, o119, o111, c2, b2, o123, storage, c1, c3, o125, r123

• Πρώτα σε βάθος: o103, ts, mail, b3, b1, c2, c1, c3, b2, b4, o109, o119, o123, o125, r123

• Πρώτα σε βάθος με επαναληπτική εκβάθυνση: · όριο βάθους 0: o103
· όριο βάθους 1: ts, b3, o109
· όριο βάθους 2: mail, b1, b4, o119, o111
· όριο βάθους 3: c2, b2, o123, storage
· όριο βάθους 4: c1, c3, o125, r123

• Αναζήτηση πρώτα στον καλύτερο: o103, b3, b1, c2, c1, c3, b2, b4, ts, o109, o119, o123, r123
κόστη: ts = 23, b3 = 17, o109 = 24, b1 = 13, b4 = 18, c2 = 10, b2 = 15, c1 = 6, c3 = 12
mail = 26, o119 = 11, o111 = 27, o123 = 4, storage = 12, o125 = 6, r123 = 0

• A*: o103, b3, b1, b4, c2, c1, b2, c3, ts, o109, o119, mail, o123, r123

κόστη: ts = 31, b3 = 21, o109 = 36, b1 = 17, b4 = 18, c2 = 21, b2 = 29, o109 = 42
c1 = 21, c3 = 29, c3 = 35, b4 = 35, mail = 40, o119 = 39, o111 = 43, o123 = 41,
storage = 47, o125 = 47, r123 = 41

Πρόβλημα 4: α) Καταστάσεις: ορίζονται από το δωμάτιο που βρίσκεται το ρομπότ

· Αρχική κατάσταση: mail: Δεν έχει μεταφερθεί κανένα πακέτο

· Κατάσταση στόχου: mail: Έχουν μεταφερθεί όλα τα πακέτα

· Τωρινή κατάσταση: το ρομπότ βρίσκεται σε οποιοδήποτε δωμάτιο και κάνει μια ενέργεια

· Ενέργειες: Το ρομπότ μετακινείται κατά ένα δωμάτιο και μπορεί να μεταφέρει ένα πακέτο

· Κόστος: Απόσταση που θα διανύσει το ρομπότ.

Πρόβλημα 5: Ο αλγόριθμος αμφίδρομης αναζήτησης θεωρείται πλήρης όταν ο παράγοντας διακλάδωσης είναι πεπερασμένος και όταν οι 2 αναζητήσεις χρησιμοποιούν BFS. Θεωρείται βέλτιστος όταν είναι πλήρης και χρησιμοποιεί BFS.

Άρα εφόσον η αναζήτηση περιορισμένου βάθους δεν είναι ξεκάθαρα BFS α, β και γ δεν είναι πλήρης. Η δ είναι πλήρης αφού ο A^* χρησιμοποιεί BFS μόνο εφόσον ο παράγοντας διακλάδωσης είναι πεπερασμένος.

Αφού δεν είναι πλήρης α, β, γ δεν είναι και βέλτιστες.

Ο δ είναι βέλτιστος εφόσον είναι πλήρης εάν ο BFS την χρησιμοποιούν και οι 2 αναζητήσεις έχει το ίδιο μη αρνητικό κόστος για όλες τις ενέργειες.

Μπορεί να γίνει αποδοτικός έλεγχος ότι οι 2 αναζητήσεις συναντιούνται χρησιμοποιώντας έναν πίνακα κατακερματισμού στον οποίο θα αποθηκεύονται οι καταστάσεις που έχουν ήδη ~~...~~ παραχθεί.