



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Τεχνητή Νοημοσύνη I

Ενότητα 3: Επίλυση Προβλημάτων με Αναζήτηση

Μουστάκας Κωνσταντίνος
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών

Σκοποί ενότητας

- Επίλυση προβλημάτων με Αναζήτηση



Περιεχόμενα ενότητας

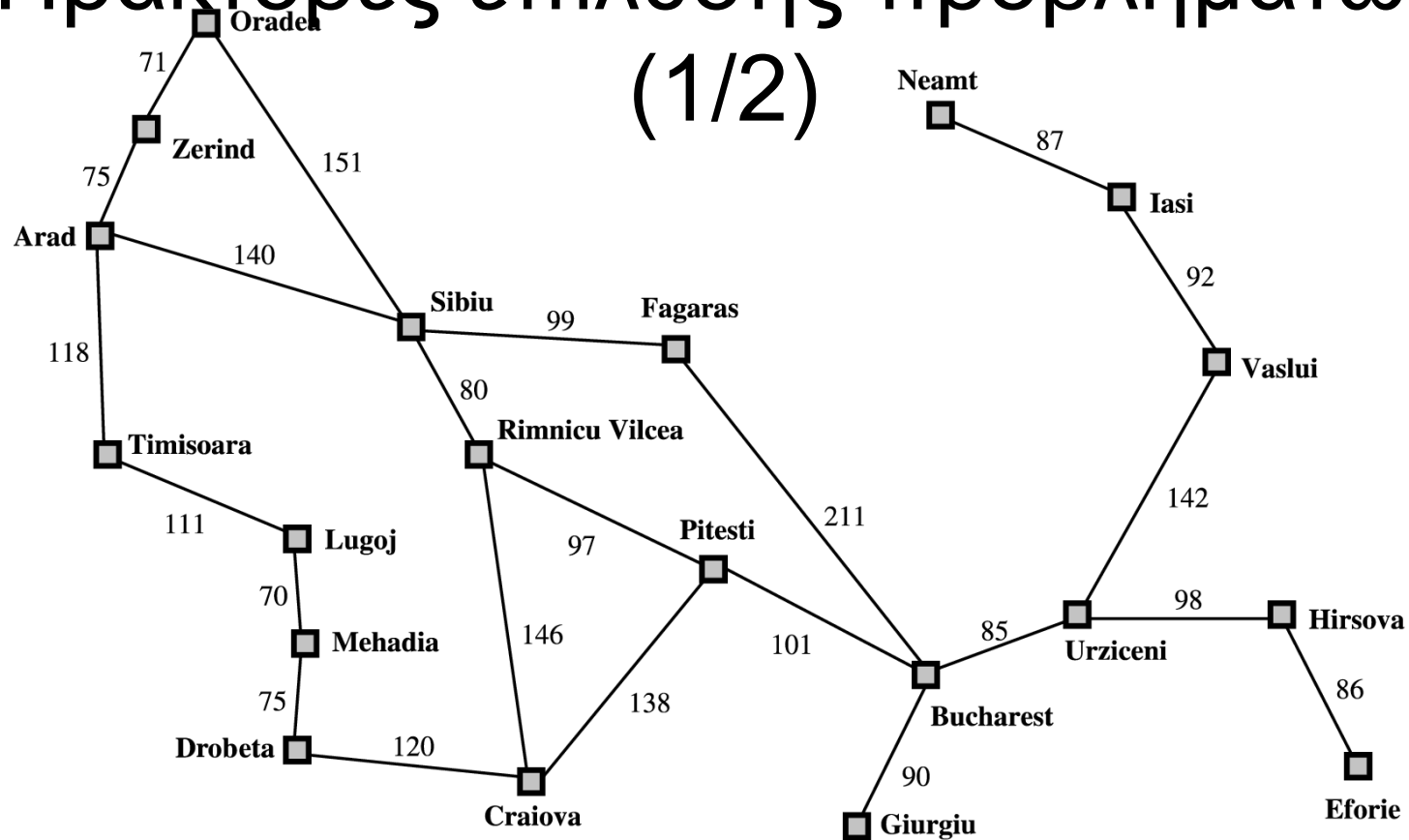
- Επίλυση προβλημάτων με Αναζήτηση



Επίλυση προβλημάτων με αναζήτηση



Πράκτορες επίλυσης προβλημάτων (1/2)



- Διατύπωση στόχου: Σύνολο καταστάσεων του κόσμου
- Διατύπωση προβλήματος
 - Επιλογή επιπέδου λεπτομέρειας (αφαίρεση)



Πράκτορες επίλυσης προβλημάτων (2/2)

- Ένας πράκτορας που έχει στη διάθεσή του πολλές άμεσες επιλογές άγνωστης αξίας μπορεί να αποφασίζει τι να κάνει εξετάζοντας πρώτα διάφορες δυνατές ακολουθίες ενεργειών που οδηγούν σε καταστάσεις γνωστής αξίας και μετά επιλέγοντας την καλύτερη ακολουθία.
- Διατύπωση – Αναζήτηση – Λύση – Εκτέλεση



Απλός πράκτορας επίλυσης προβλημάτων

- **function** Simple-Problem-Solving-Agent(*αντίληψη*) **returns** μια ενέργεια
inputs: *αντίληψη*, μια αντίληψη
static: *ακολουθία*, ακολουθία αντιλήψεων, αρχικά κενή
κατάσταση, περιγραφή της τρέχουσας κατάστασης του κόσμου
στόχος, ένας στόχος, αρχικά κενός
πρόβλημα, μια διατύπωση προβλήματος

κατάσταση ← Update-State(*κατάσταση*, *αντίληψη*)

if *ακολουθία* είναι κενή **then do**

στόχος ← Formulate-Goal(*κατάσταση*)

πρόβλημα ← Formulate-Problem(*κατάσταση*, *στόχος*)

ακολουθία ← Search(*πρόβλημα*)

ενέργεια ← First(*ακολουθία*)

ακολουθία ← Rest(*ακολουθία*)

return *ενέργεια*

- Παραδοχές: Στατικό, παρατηρήσιμο, διακριτό, αιτιοκρατικό περιβάλλον
- Πράκτορας ανοικτού βρόχου



Προβλήματα και Λύσεις

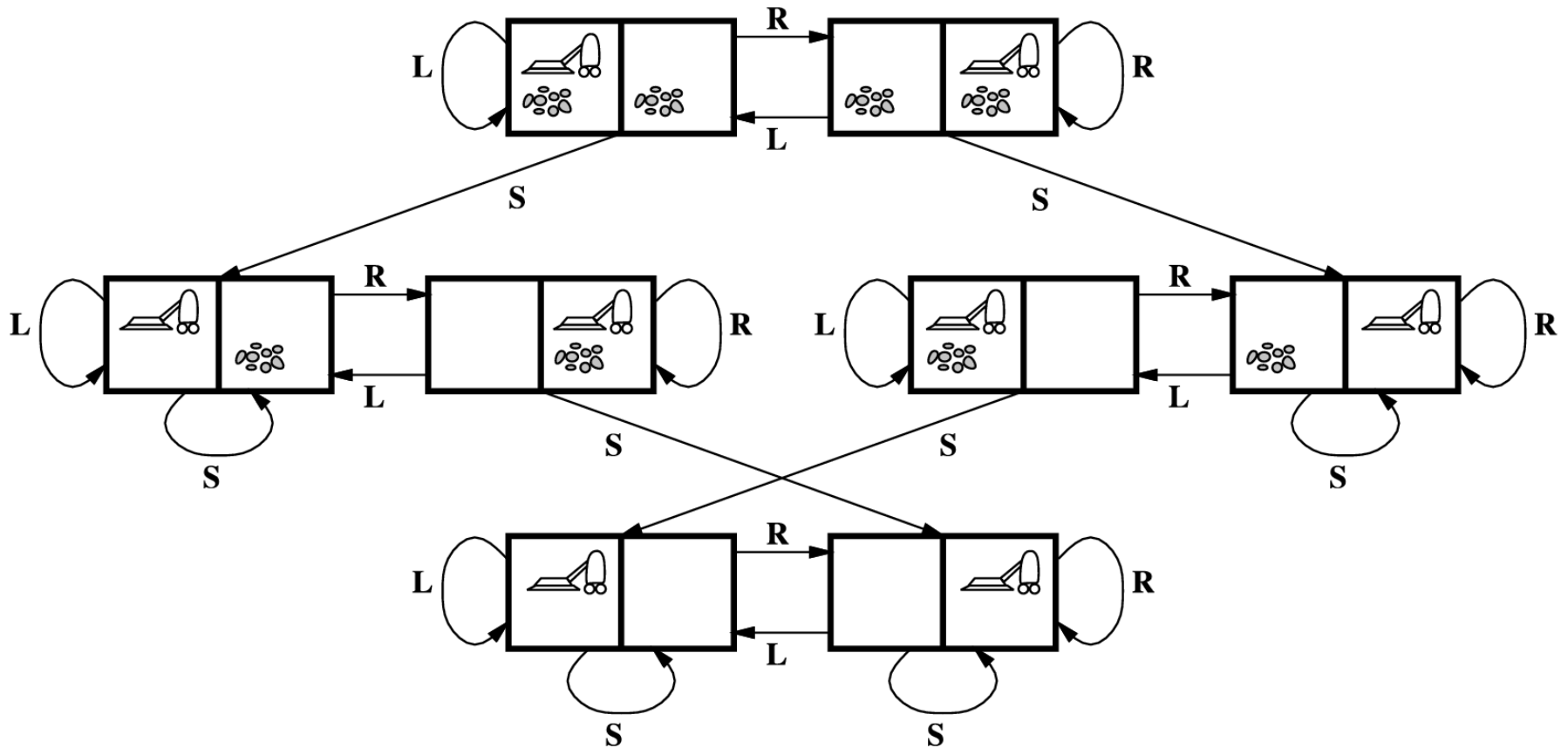
- Συνιστώσες προβλήματος:
 - Αρχική κατάσταση
 - π.χ. *Εντός(Arad)*
 - Ενέργειες, συνάρτηση διαδόχων $\text{Successor-Fn}(x)$
 - π.χ. $\text{Successor-Fn}(\text{Arad}) = \{ \langle \text{Μετάβαση}(\text{Sibiu}), \text{Εντός}(\text{Sibiu}) \rangle, \langle \text{Μετάβαση}(\text{Timisoara}), \text{Εντός}(\text{Timisoara}) \rangle, \langle \text{Μετάβαση}(\text{Zerind}), \text{Εντός}(\text{Zerind}) \rangle \}$
 - Χώρος καταστάσεων, αναπαράσταση με γράφημα, διαδρομές
 - Έλεγχος στόχου
 - Ρητή απαρίθμηση καταστάσεων ή περιγραφή ιδιοτήτων
 - Κόστος διαδρομής
 - Κόστος βήματος, $c(x,a,y)$



Λύση: Διαδρομή από αρχική κατάσταση σε κατάσταση στόχου

- Βέλτιστη λύση

Κόσμος της ηλεκτρικής σκούπας



L=Αριστερά, R=Δεξιά, S=Αναρρόφηση



Το παζλ των 8 πλακιδίων

7	2	4
5		6
8	3	1

Αρχική κατάσταση

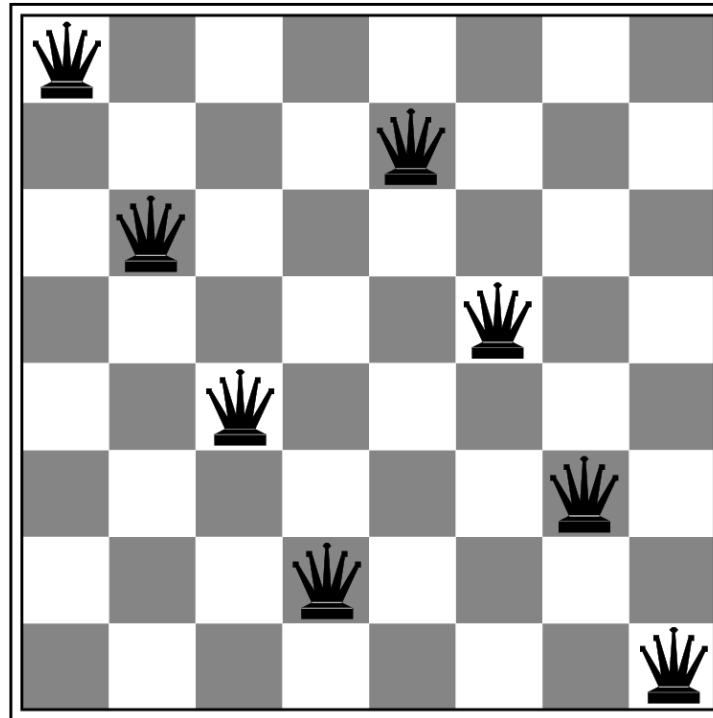
	1	2
3	4	5
6	7	8

Κατάσταση στόχου

- Παζλ 15 πλακιδίων: Λύνεται εύκολα βέλτιστα
- Παζλ 24 πλακιδίων: Δεν λύνεται βέλτιστα (ακόμη)



Το πρόβλημα των 8 βασίλισσών



- Δύο εναλλακτικές διατυπώσεις:
 - Αυξητική (παραλλαγές)
 - Πλήρεις καταστάσεις



Προβλήματα του πραγματικού κόσμου

- Εύρεση δρομολογίου
- Προβλήματα περιήγησης
 - Πλανόδιος πωλητής
- Διάταξη κυκλωμάτων VLSI
 - Διάταξη κελιών, διάταξη καναλιών
- Πλοήγηση ρομπότ
- Αυτόματη ακολουθία συναρμολόγησης
 - Σχεδίαση πρωτεϊνών
- Αναζήτηση στο διαδίκτυο (κείμενο, εικόνα 3D)

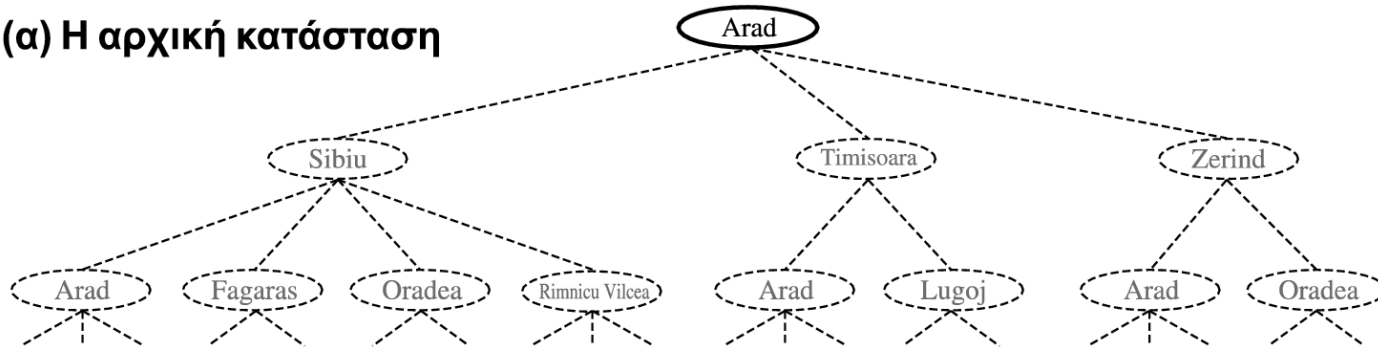


Αναζήτηση λύσεων

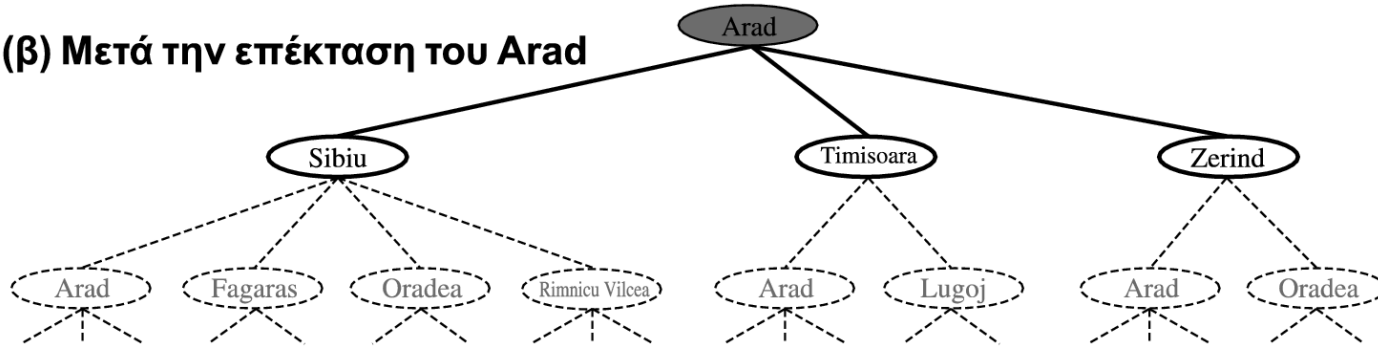


Δένδρο αναζήτησης (1/3)

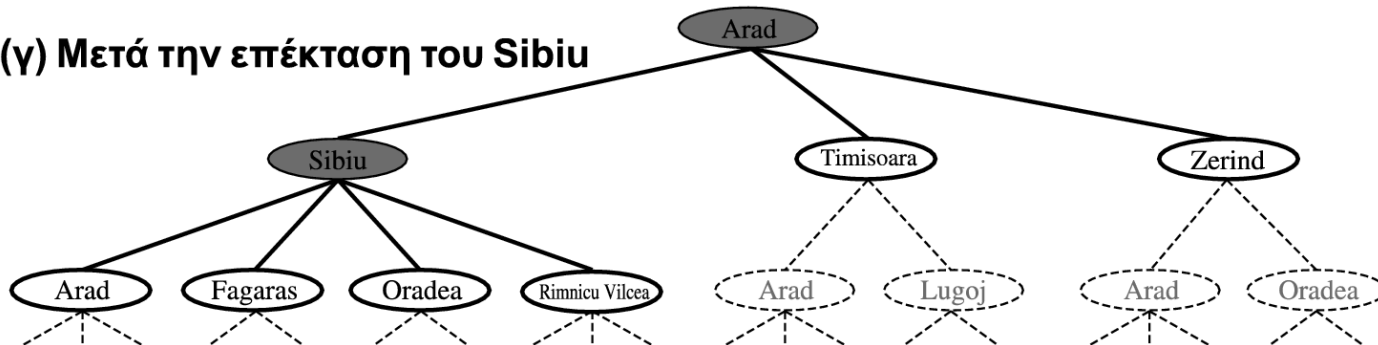
(α) Η αρχική κατάσταση



(β) Μετά την επέκταση του Arad

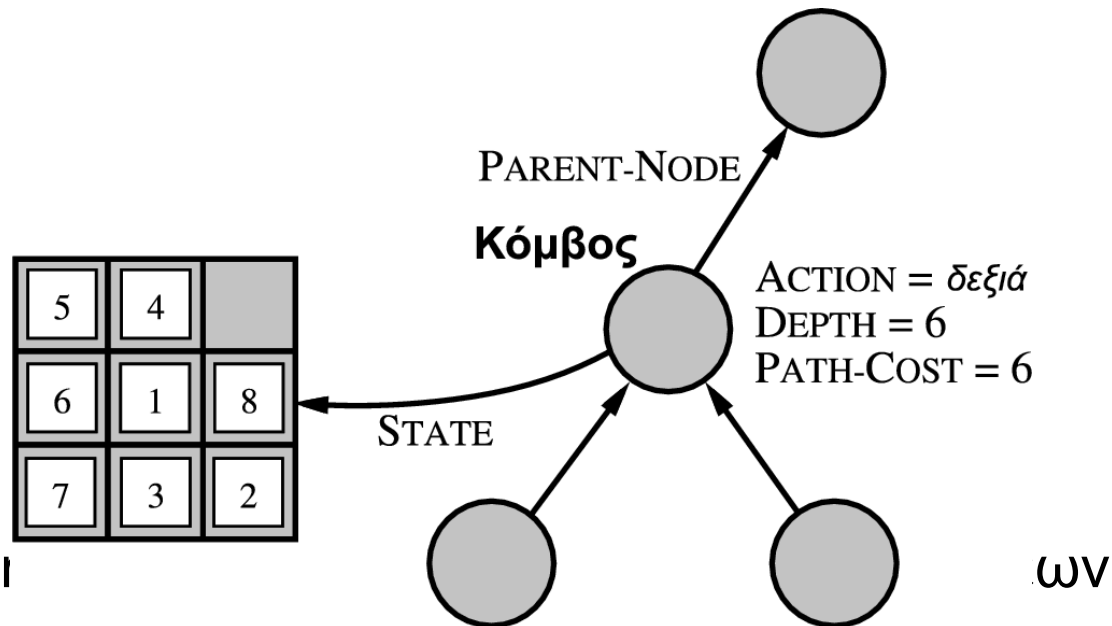


(γ) Μετά την επέκταση του Sibiu



Δένδρο αναζήτησης (2/3)

- Κόμβος αναζήτησης
⟨State, Parent-Node, Action, Path-Cost, Depth⟩



- Επέκτασι
καταστάσεων
- Στρατηγική αναζήτησης



Δένδρο αναζήτησης (3/3)

- Κόμβος φύλλο (leaf node)
- Σύνορο (fringe)
 - Υλοποίηση με ουρά
- Βασικές λειτουργίες στο σύνορο:
 - void Make-Queue(*στοιχείο*, ...).
 - bool Empty?(*ουρά*).
 - node First(*ουρά*)
 - node Remove-First(*ουρά*)
 - void Insert(*στοιχείο*, *ουρά*)
 - void Insert-All(*στοιχεία*, *ουρά*)



Άτυπος (γενικός) αλγόριθμος αναζήτησης

function Tree-Search(*πρόβλημα, στρατηγική*) **returns** μια λύση ή αποτυχία

αρχικοποίηση του δένδρου αναζήτησης με χρήση της αρχικής κατάστασης του *προβλήματος*

loop do

if δεν υπάρχουν υποψήφιοι για επέκταση **then**

return αποτυχία

επιλογή ενός κόμβου-φύλλου για να επεκταθεί, σύμφωνα με τη *στρατηγική*

if ο κόμβος περιέχει μια κατάσταση στόχου **then**

return την αντίστοιχη λύση

else ο κόμβος επεκτείνεται και οι κόμβοι που

προκύπτουν προστίθενται στο δένδρο

αναζήτησης



Τυπικός αλγόριθμος αναζήτησης (1/2)

function Tree-Search(*πρόβλημα*, *σύνορο*) **returns** μια λύση ή αποτυχία

σύνορο ← Insert(Make-Node(Initial-State[*πρόβλημα*]), *σύνορο*)

loop do

if Empty?(*σύνορο*) **then return** αποτυχία

κόμβος ← Remove-First(*σύνορο*)

if Goal-Test[*πρόβλημα*] που εφαρμόστηκε στην State[*κόμβος*] επιτύχει
 then return Solution(*κόμβος*)

σύνορο ← Insert-All(Expand(*κόμβος*, *πρόβλημα*), *σύνορο*)



Τυπικός αλγόριθμος αναζήτησης (2/2)

- **function** Expand(*κόμβος*, *πρόβλημα*) **returns** ένα σύνολο κόμβων

διάδοχοι \leftarrow κενό σύνολο

for each \langle *ενέργεια*, *αποτέλεσμα* \rangle **in** Successor-Fn[*πρόβλημα*](State[*κόμβος*]) **do**

s \leftarrow νέος κόμβος Node

State[*s*] \leftarrow *αποτέλεσμα*

Parent-Node[*s*] \leftarrow *κόμβος*

Action[*s*] \leftarrow *ενέργεια*

Path-Cost[*s*] \leftarrow Path-Cost[*κόμβος*] + Step-Cost(*κόμβος*, *ενέργεια*, *s*)

Depth[*s*] \leftarrow Depth[*κόμβος*] + 1

προσθήκη του *s* στο σύνολο *διάδοχοι*

return *διάδοχοι*



Μέτρηση απόδοσης αλγορίθμων αναζήτησης

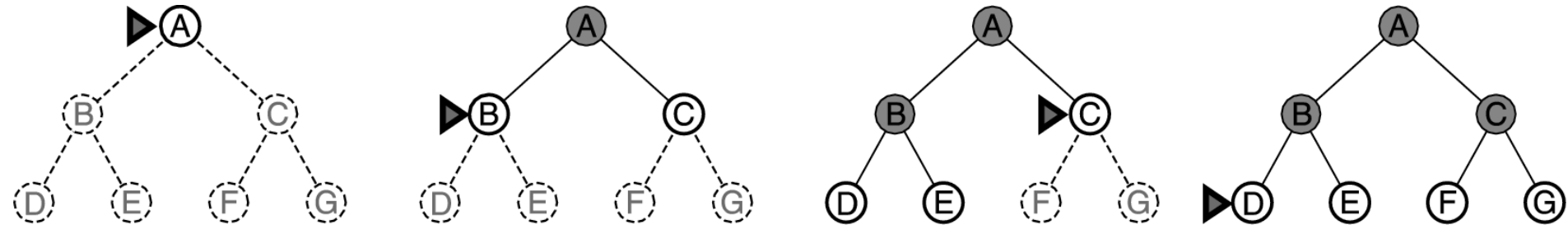
- Πληρότητα (Completeness) = Να βρίσκει πάντα λύση, αν υπάρχει
- Βέλτιστη συμπεριφορά (Optimality)
- Χρονική πολυπλοκότητα (Time complexity)
- Χωρική πολυπλοκότητα (Space complexity)
- Μέγεθος προβλήματος:
 - Παράγοντας διακλάδωσης, b (branching factor)
 - Βάθος d (depth) του πιο ρηχού κόμβου στόχου
 - Μέγιστο μήκος, m , οποιασδήποτε διαδρομής του χώρου καταστάσεων.
- Κόστος αναζήτησης (χρονικό)
- Ολικό κόστος = Κόστος αναζήτησης + κόστος εκτέλεσης πλάνου



Στρατηγικές απληροφόρητης αναζήτησης



Αναζήτηση πρώτα κατά πλάτος (Breadth-first search)



Βάθος	Κόμβοι	Χρόνος	Μνήμη
2	1100 (10+100+990)	0.11 δευτερόλεπτα	1 megabyte
4	111.100	11 δευτερόλεπτα	106 megabytes
6	10^7	19 λεπτά	10 gigabytes
8	10^9	31 ώρες	1 terabyte
10	10^{11}	129 ημέρες	101 terabytes
12	10^{13}	35 χρόνια	10 petabytes
14	10^{15}	3.523 χρόνια	1 exabyte

- Πλήρης, βέλτιστος (υπό προϋποθέσεις)
- Χωρική, χρονική πολυπλοκότητα $O(b^{d+1})$
 - π.χ. για $b=10$, 10.000 κόμβοι/sec, 1000 bytes/κόμβο



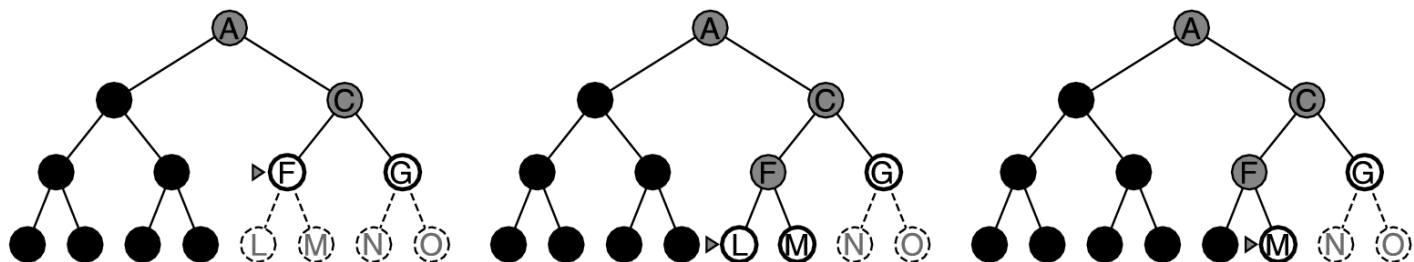
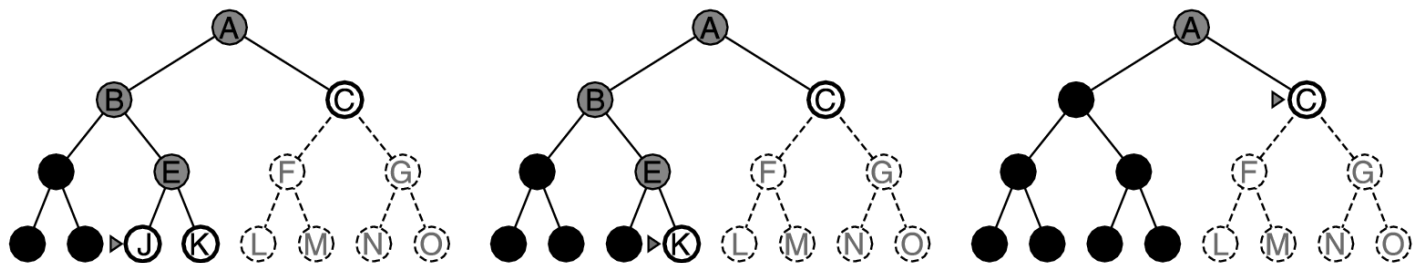
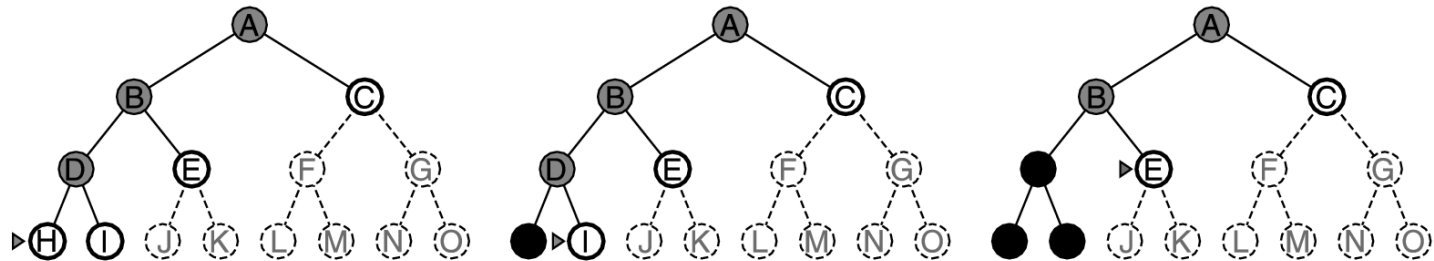
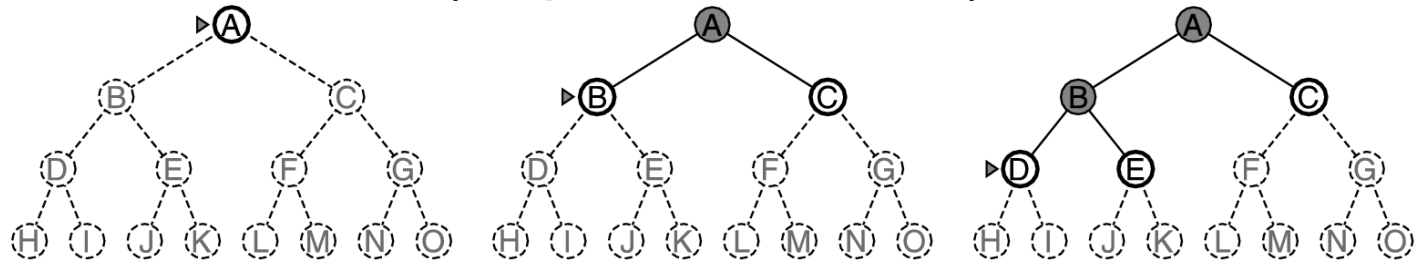
Αναζήτηση ομοιόμορφου κόστους (Uniform-cost search)

- Παραλλαγή της αναζήτησης πρώτα κατά πλάτος:
 - Επεκτείνει πάντα τον κόμβο με το μικρότερο κόστος διαδρομής
- Προϋπόθεση πληρότητας: Κόστος βήματος > 0
- Εγγυάται ότι θα βρει τη βέλτιστη λύση
- Ο έλεγχος στόχου γίνεται κατά την έξοδο ενός κόμβου από το σύνορο.



Αναζήτηση πρώτα κατά βάθος (1/2)

(Depth-first search)



Αναζήτηση πρώτα κατά βάθος (2/2)

(Depth-first search)

- Ελάχιστες απαιτήσεις σε χώρο: $b \cdot m + 1$
- Δεν είναι βέλτιστη
- Δεν είναι πλήρης (ατέρμονες βρόχοι)
- Πολυπλοκότητα χρόνου: $O(b^m)$
 - Προσοχή: $m \geq d$



Αναζήτηση περιορισμένου βάθους (Depth-limited search)

- Τίθεται όριο βάθους $L < m$.
- Λύνεται το πρόβλημα των άπειρων διαδρομών
- Απώλεια πληρότητας αν $L < d$.
- Όχι βέλτιστες λύσεις αν $L > d$
- Χρονική πολυπλοκότητα: $O(b^L)$
- Χωρική πολυπλοκότητα: $O(bL)$

- Επιλογή L με γνώση του προβλήματος (π.χ. διάμετρος του χώρου καταστάσεων)



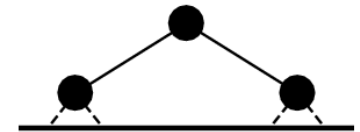
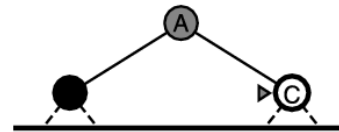
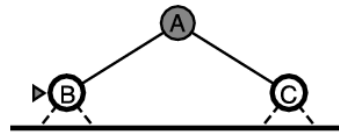
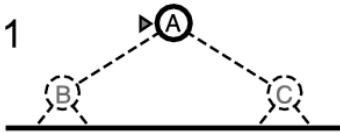
Επαναληπτική εμφάθυνση (1/3)

(Iterative-deepening search)

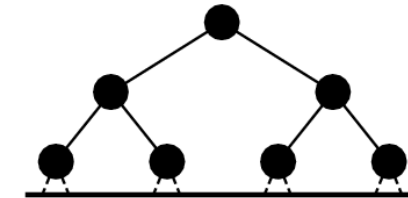
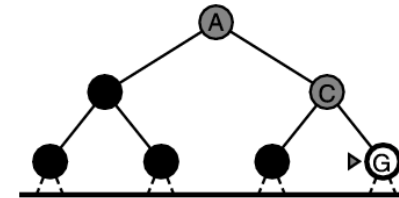
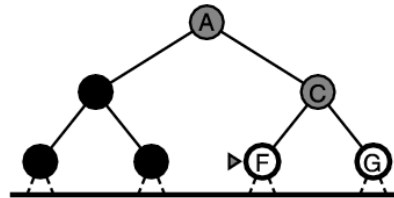
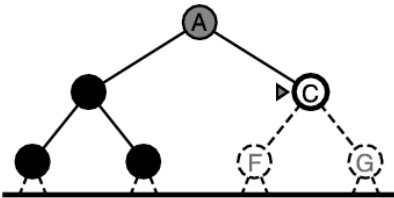
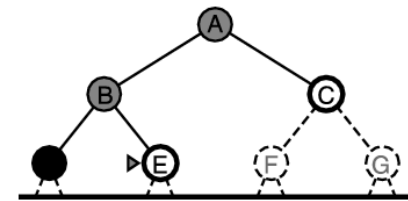
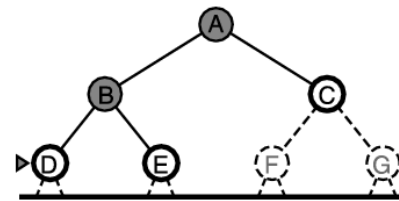
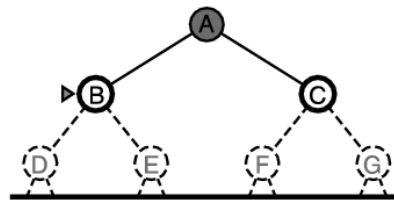
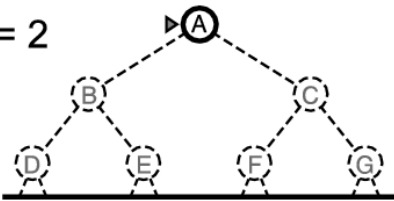
Όριο = 0



Όριο = 1



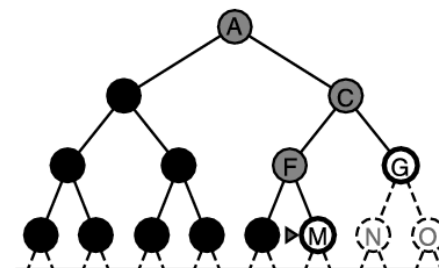
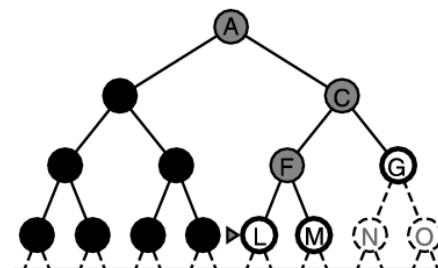
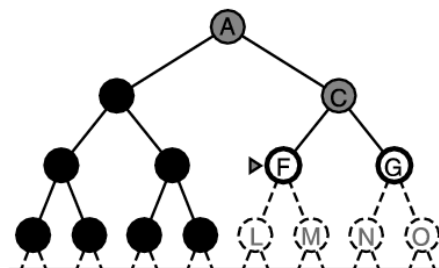
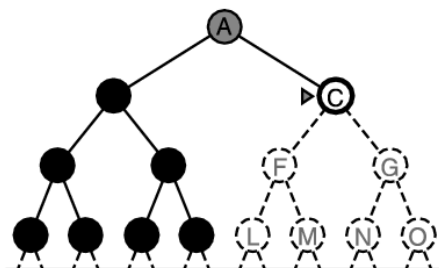
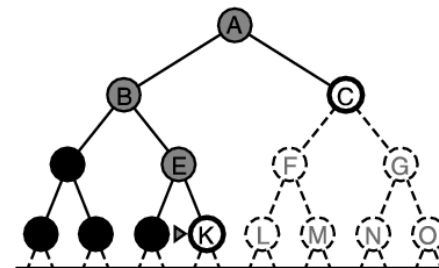
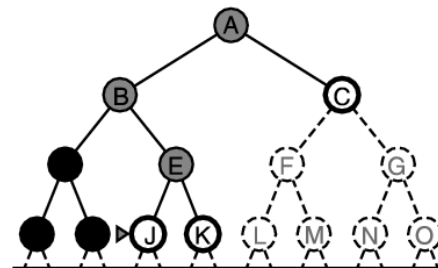
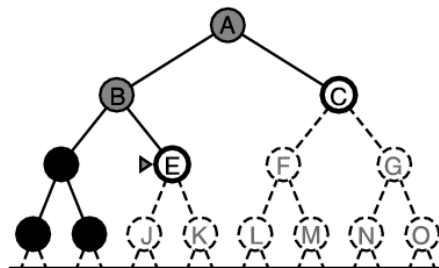
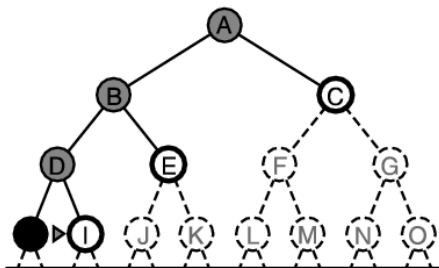
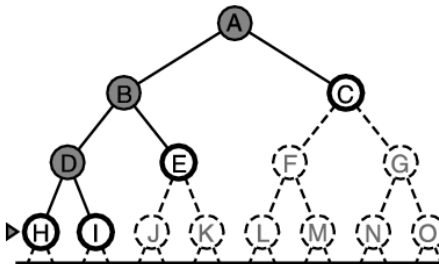
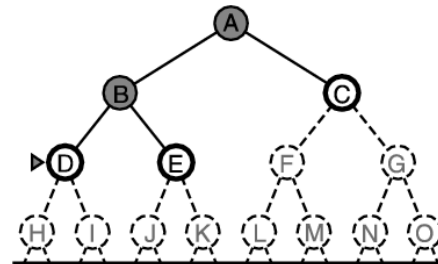
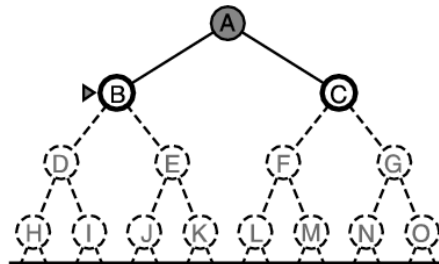
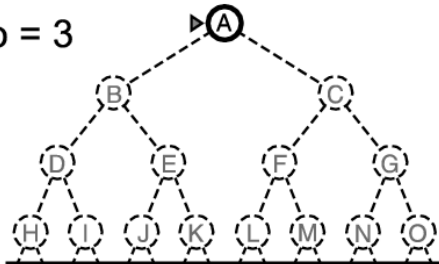
Όριο = 2



Επαναληπτική εμφάθυνση (2/3)

(Iterative-deepening search)

Όριο = 3



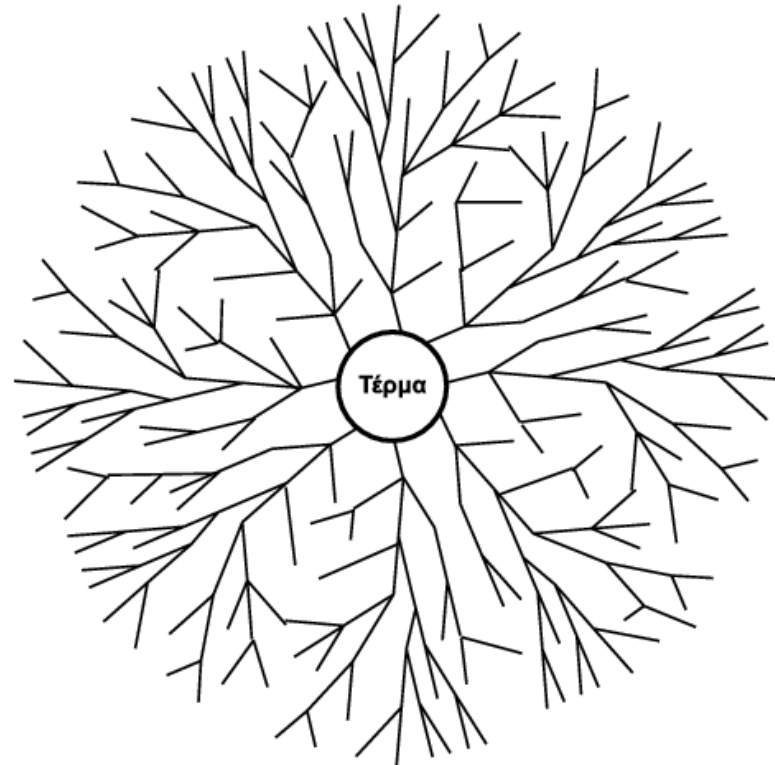
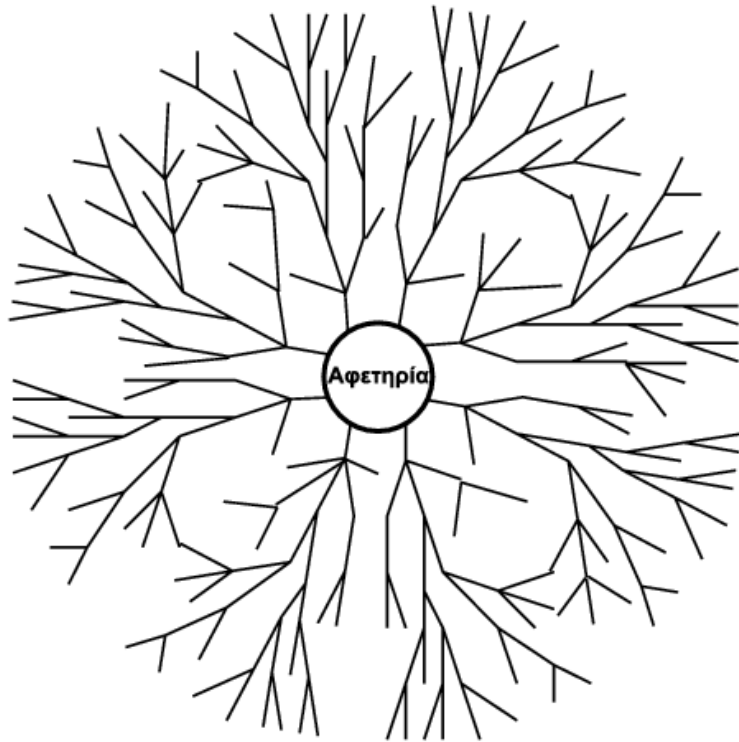
Επαναληπτική εμβάθυνση (3/3)

(Iterative-deepening search)

- Χωρική πολυπλοκότητα: $O(bd)$
- Χρονική πολυπλοκότητα:
 - $N(\text{IDS}) = (d)b + (d - 1)b^2 + \dots + (1)b^d = O(b^d)$!!!!
 - Μικρότερη από τη χωρική πολυπλοκότητα της αναζήτησης πρώτα κατά πλάτος!!! Γιατί;
 - Η BFS παράγει κάποιους κόμβους σε βάθος $d+1$ ενώ η IDS όχι
- Γενικά, η επαναληπτική εμβάθυνση είναι η προτιμότερη μέθοδος απληροφόρητης αναζήτησης όταν ο χώρος αναζήτησης είναι μεγάλος και το βάθος της λύσης δεν είναι γνωστό.
- Αναζήτηση επαναληπτικής επιμήκυνσης (με κόστη αντί για βήματα)



Αμφίδρομη αναζήτηση (1/2)



- $b^{d/2} + b^{d/2} \ll b^d$



Αμφίδρομη αναζήτηση (2/2)

- Χρονική πολυπλοκότητα: $O(b^{d/2})$
- Χωρική πολυπλοκότητα: $O(b^{d/2})$ (μειονέκτημα)
- Προβλήματα:
 - Εύρεση προκατόχων καταστάσεων
 - Έλλειψη καλά καθορισμένων στόχων (π.χ. σκάκι)



Σύγκριση στρατηγικών απληροφόρητης αναζήτησης

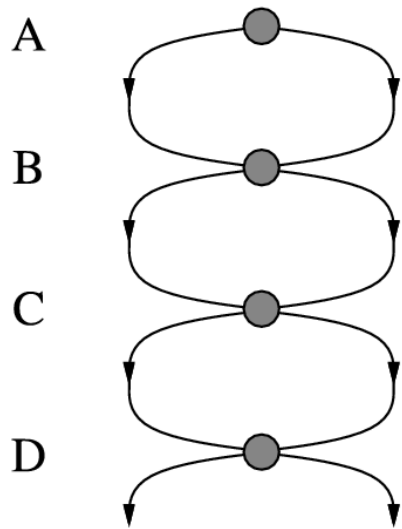
Στον πίνακα της Εικόνας 3.17 συγκρίνονται οι στρατηγικές αναζήτησης με βάση τα τέσσερα κριτήρια αξιολόγησης που θέσαμε στην Ενότητα 3.4.

Κριτήριο	Πρώτα σε πλάτος	Ομοιόμορφου κόστους	Πρώτα σε βάθος	Περιορισμένου βάθους	Επαναληπτικής εκβάθυνσης	Αμφίδρομη (αν εφαρμόζεται)
Πλήρης:	Ναι ^a	Ναι ^{a,b}	Όχι	Όχι	Ναι ^a	Ναι ^{a,d}
Χρόνος	$O(b^{d+1})$	$O(b^{1+cn/\epsilon})$	$O(b^m)$	$O(b^l)$	$O(b^d)$	$O(b^{d/2})$
Χώρος	$O(b^{d+1})$	$O(b^{1+cn/\epsilon})$	$O(bm)$	$O(bl)$	$O(bd)$	$O(b^{d/2})$
Βέλτιστη:	Ναι ^c	Ναι	Όχι	Όχι	Ναι ^c	Ναι ^{c,d}

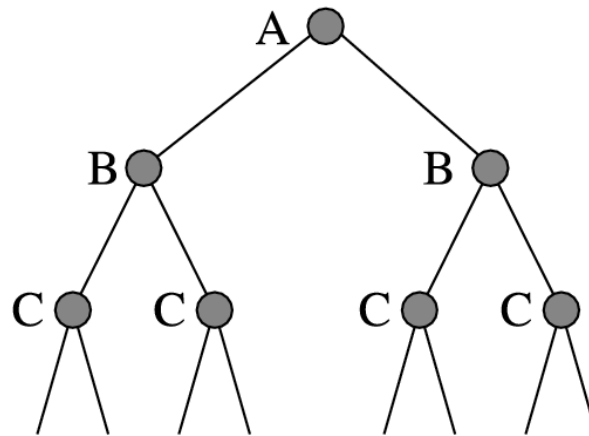
Εικόνα 3.17 Αξιολόγηση των στρατηγικών αναζήτησης. b είναι ο παράγοντας διακλάδωσης· d είναι το βάθος της ρηχότερης λύσης· m είναι το μέγιστο βάθος του δένδρου αναζήτησης· l είναι το όριο βάθους. Οι επιφυλάξεις είναι οι εξής: ^a πλήρης αν το b είναι πεπερασμένο· ^b πλήρης αν τα κόστη βημάτων είναι $\geq \epsilon$ για ϵ θετικό· ^c βέλτιστη αν τα κόστη βημάτων είναι όλα ίδια· ^d αν και οι δύο κατευθύνσεις χρησιμοποιούν αναζήτηση πρώτα σε πλάτος.



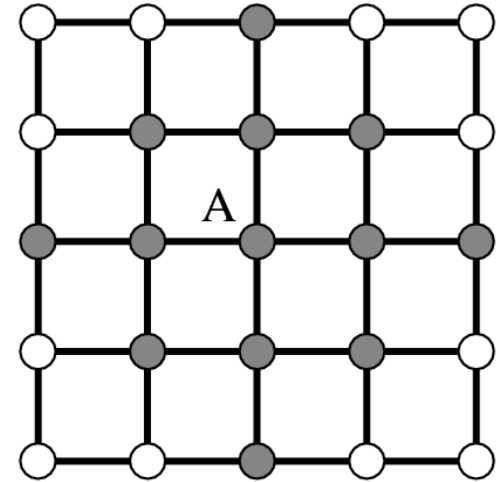
Αποφυγή επαναλαμβανόμενων καταστάσεων (1/2)



(α)



(β)



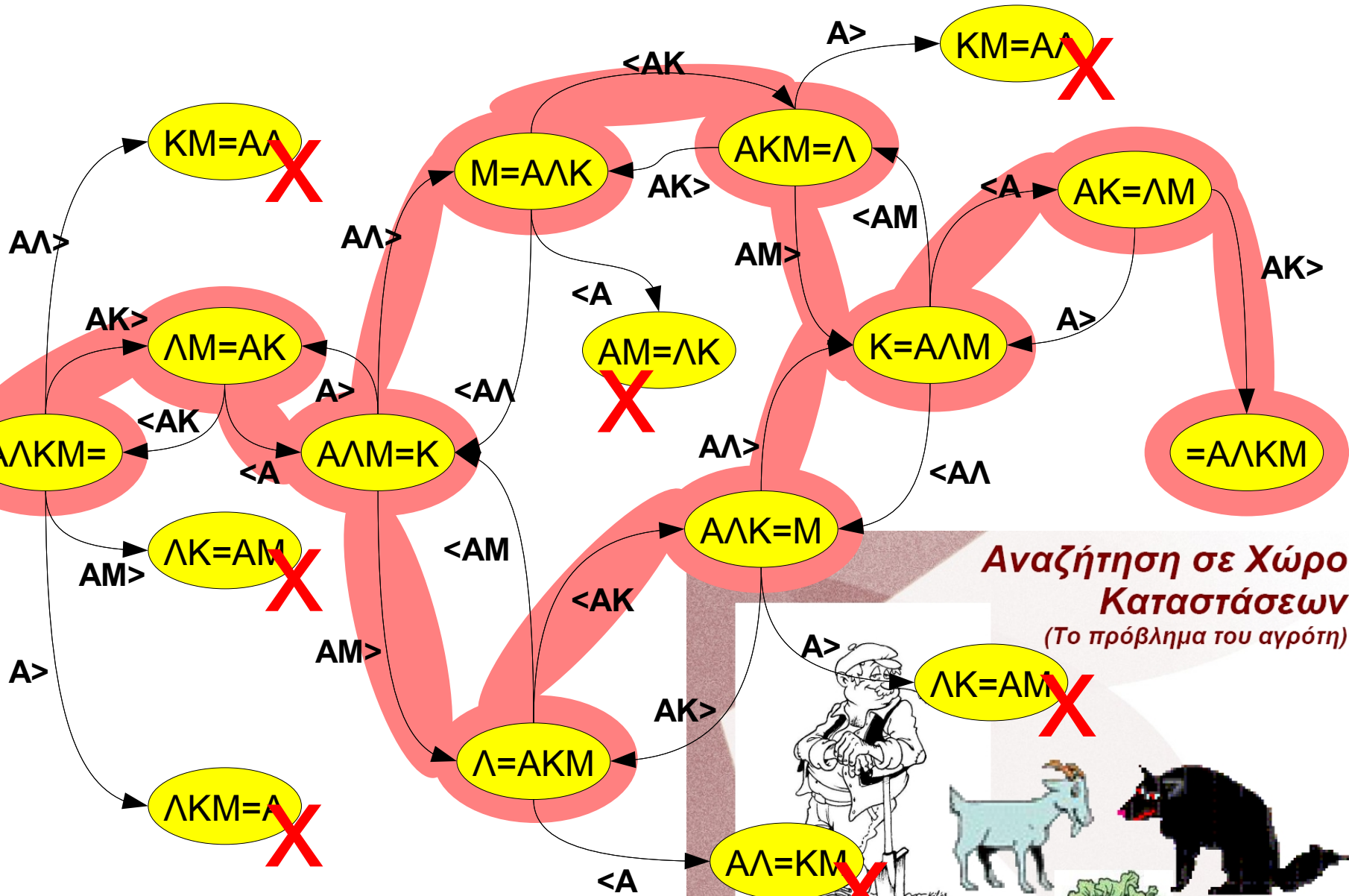
(γ)



Αποφυγή επαναλαμβανόμενων καταστάσεων (2/2)

- *«Οι αλγόριθμοι που ξεχνούν την ιστορία τους είναι καταδικασμένοι να την επαναλαμβάνουν !»*
- Έλεγχος μόνο τρέχουσας διαδρομής
 - π.χ. αλγόριθμος πρώτα κατά βάθος
- Έλεγχος όλων των καταστάσεων
 - Κλειστή λίστα
 - (σύνоро = ανοικτή λίστα)
- Πρόβλημα: Ποια από τις δύο καταστάσεις κρατάμε;
 - Φυσικά αυτή με το μικρότερο κόστος





Αναζήτηση σε Χώρο Καταστάσεων
(Το πρόβλημα του αγρότη)

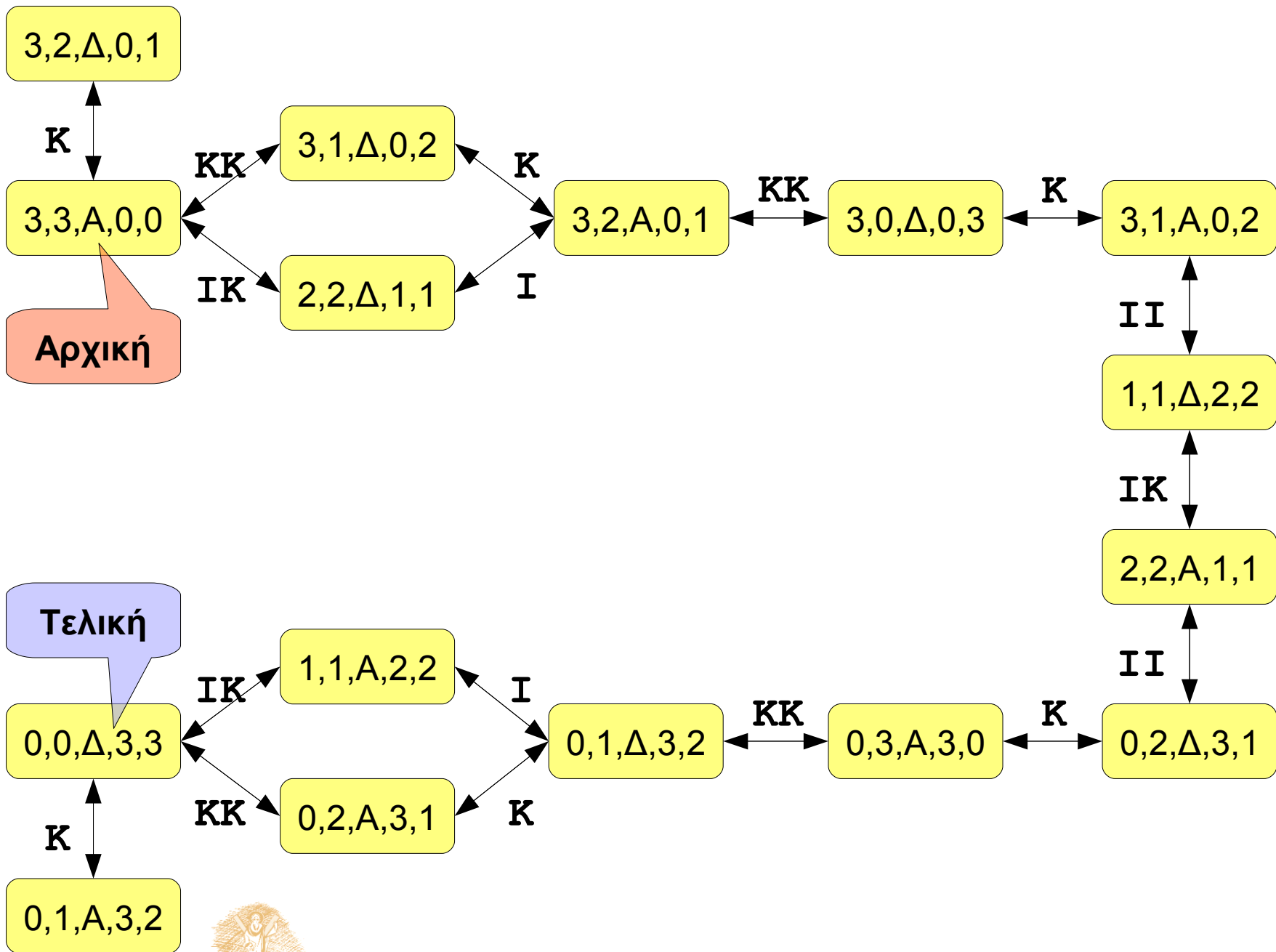


Άσκηση

3.9 Το πρόβλημα των ιεραποστόλων και των κανίβαλων διατυπώνεται συνήθως με τον εξής τρόπο: Τρεις ιεραπόστολοι και τρεις κανίβαλοι βρίσκονται στη μια όχθη ενός ποταμού, με μια βάρκα που χωρά ένα ή δύο άτομα. Βρείτε έναν τρόπο για να περάσουν όλοι στην απέναντι όχθη, χωρίς ποτέ να μείνει σε ένα μέρος μια ομάδα ιεραποστόλων αριθμητικά μικρότερη από τους κανίβαλους στο ίδιο μέρος. Αυτό το πρόβλημα έγινε διάσημο στην ΤΝ επειδή ήταν το θέμα της πρώτης επιστημονικής δημοσίευσης που προσέγγισε τη διατύπωση προβλημάτων από αναλυτική άποψη (Amarel, 1968).

- α. Διατυπώστε το πρόβλημα με ακρίβεια, κάνοντας μόνο τις διακρίσεις που είναι απαραίτητες για να εξασφαλιστεί μια έγκυρη λύση. Σχεδιάστε ένα διάγραμμα ολόκληρου του χώρου καταστάσεων.
- β. Υλοποιήστε και λύστε το πρόβλημα με βέλτιστο τρόπο χρησιμοποιώντας έναν κατάλληλο αλγόριθμο αναζήτησης. Είναι καλή ιδέα να ελέγχετε για επαναλαμβανόμενες καταστάσεις;
- γ. Γιατί νομίζετε ότι οι άνθρωποι έχουν δυσκολία να λύσουν αυτό το παζλ, αφού ο χώρος καταστάσεων είναι τόσο απλός;



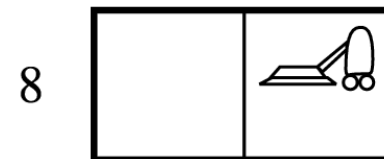
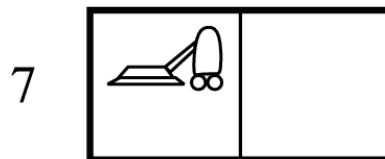
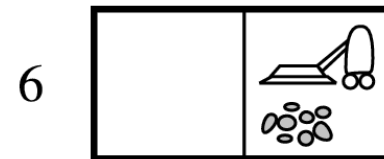
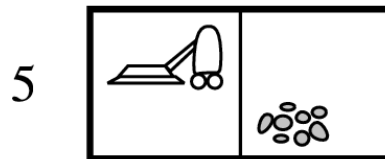
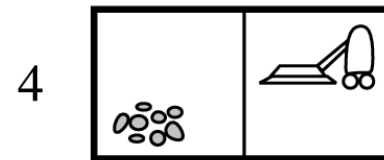
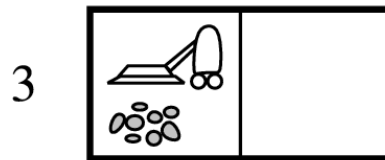
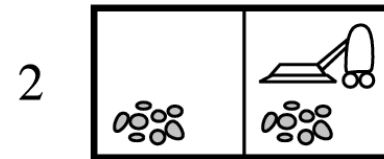
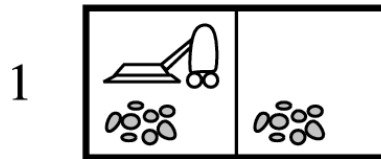


Αναζήτηση με μερική πληροφόρηση (με ατελή αισθητηριακή πληροφόρηση)

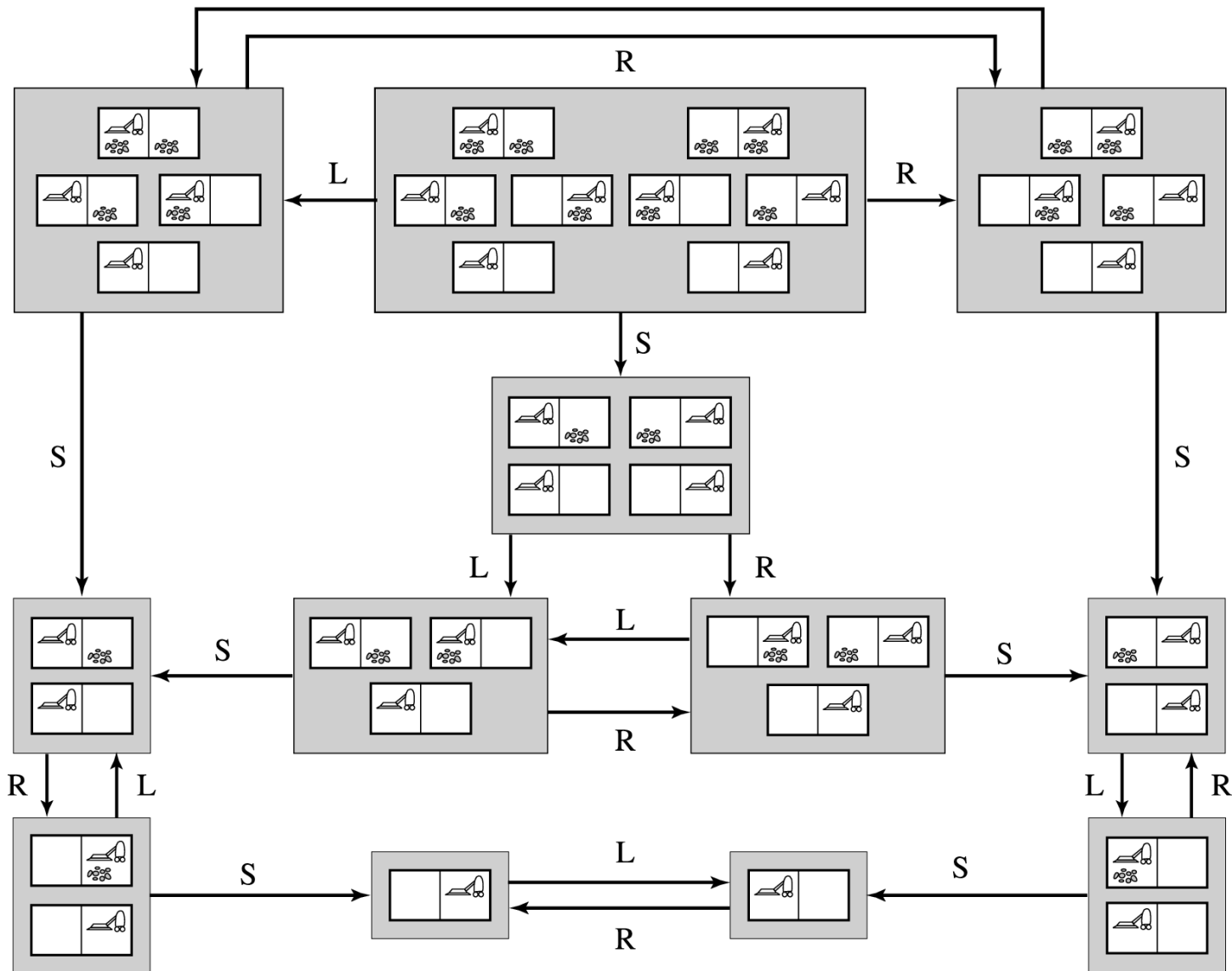


Κατηγορίες προβλημάτων

1. Προβλήματα χωρίς αισθητήρες (sensorless problems) ή σύμμορφα προβλήματα (conformant problems)
2. Προβλήματα ενδεχομένων (contingency problems) – μερικώς παρατηρήσιμο (δυναμικό) περιβάλλον, ο πράκτορας λαμβάνει νέες πληροφορίες μετά από κάθε ενέργεια (Κεφ.12)
3. Προβλήματα εξερεύνησης (exploration problems) – τελείως άγνωστο περιβάλλον (Κεφ.4)



Προβλήματα χωρίς αισθητήρες (1/2)



Προβλήματα χωρίς αισθητήρες (2/2)

- Κατάσταση πεποίθησης
 - Αρχική $\rightarrow \{1,2,3,4,5,6,7,8\}$
- Εξαναγκασμός
 - [Δεξιά] $\rightarrow \{2,4,6,8\}$
 - [Δεξιά, Αναρρόφηση] $\rightarrow \{4,8\}$
 - [Δεξιά, Αναρρόφηση, Αριστερά] $\rightarrow \{3,7\}$
 - [Δεξιά, Αναρρόφηση, Αριστερά, Αναρρόφηση] $\rightarrow \{7\}$
- Χώρος καταστάσεων πεποίθησης: Δυναμοσύνολο του χώρου καταστάσεων.
 - Δεν είναι πάντα όλες προσπελάσιμες.
- Η προσέγγιση καλύπτει και μη αιτιοκρατικές ενέργειες.
- Δεν έχουν πάντα λύση αυτά τα προβλήματα.



Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**.

Έχουν προηγηθεί οι κάτωθι εκδόσεις:

- Έκδοση **1.0** διαθέσιμη [εδώ](#).

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, **Σγάρμπας Κυριάκος**. «**Τεχνητή Νοημοσύνη Ι, Επίλυση Προβλημάτων με Αναζήτηση**». Έκδοση: **1.0**. Πάτρα **2014**. Διαθέσιμο από τη δικτυακή διεύθυνση:

https://eclass.upatras.gr/modules/course_metadata/opencourses.php?fc=15

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων (1)

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων: