

Κεφάλαιο 2

Περιγραφή Προβλημάτων και Αναζήτηση Λύσης

Τεχνητή Νοημοσύνη - Β' Έκδοση

Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου



Περιγραφή Προβλημάτων

- ❖ Διαισθητικά: υπάρχει μία δεδομένη κατάσταση (*αρχική*), υπάρχει μία επιθυμητή κατάσταση (*τελική*) και διαθέσιμες *ενέργειες* που πρέπει να γίνουν ώστε να φτάσουμε στην επιθυμητή.

Η επίλυση προβλημάτων που επιδιώκεται από την TN απαιτεί τον τυποποιημένο και σαφή ορισμό τους.

- ❖ Ο ορισμός ενός προβλήματος είναι ανεξάρτητη από την πολυπλοκότητα επίλυσής του η οποία αφορά τον αλγόριθμο αναζήτησης.

Κατηγορίες Προβλημάτων

❖ Πραγματικά και πολύπλοκα προβλήματα (*real world problems*):

- σκάκι (*chess*),
- πλανόδιος πωλητής (*traveling salesperson*),
- N*-βασίλισσες (*N-queens*),
- σάκος (*knapsack*),

κλπ.

❖ Απλά προβλήματα (*toy problems*)

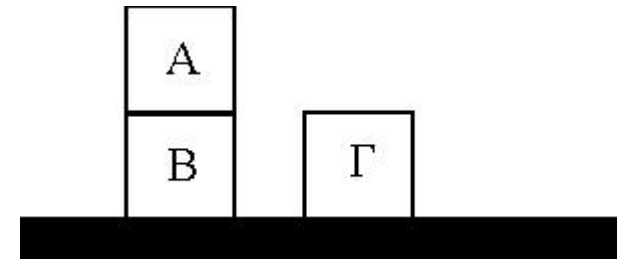
- κύβοι (*blocks*),
- N*-puzzle,
- τρίλιζα (*tic-tac-toe*),
- λαβύρινθος (*maze*),
- πύργοι του Ανόι (*Hanoi towers*),
- κανίβαλοι και ιεραπόστολοι (*missionaries and cannibals*),
- ποτήρια (*water glass*)

κλπ.

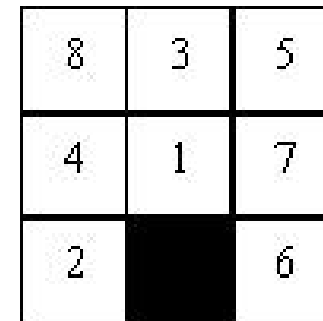
❖ Η μεθοδολογία που εφαρμόζεται στην αναπαράσταση και την επίλυση είναι παρόμοια και στις δύο ακραίες περιπτώσεις προβλημάτων.

Παραδείγματα Απλών Προβλημάτων (1/2)

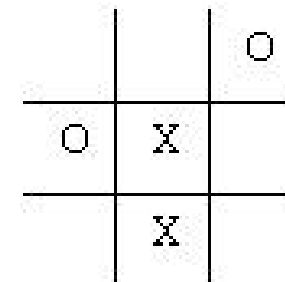
κύβοι (*blocks*)



N-puzzle



τρίλιζα (*tic-tac-toe*)

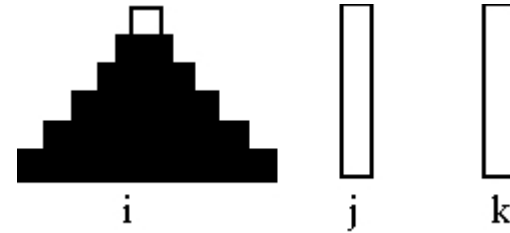


λαβύρινθος (*maze*)

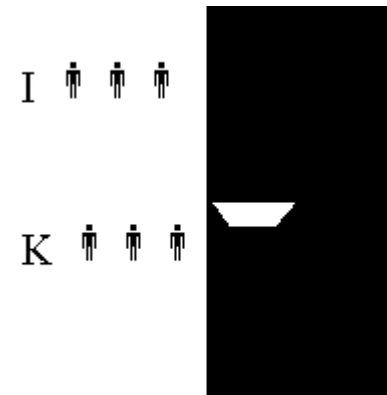


Παραδείγματα Απλών Προβλημάτων (2/2)

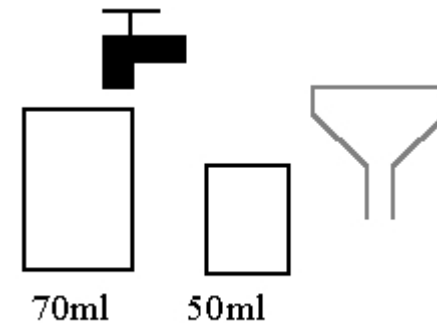
πύργοι του Ανόι (Hanoi towers)



κανίβαλοι και ιεραπόστολοι (missionaries and cannibals)



ποτήρια (water glass)





Περιγραφή Προβλημάτων

- ❖ Η περιγραφή ενός προβλήματος μπορεί να γίνει με δύο βασικούς τρόπους:
 - ❑ Περιγραφή με Χώρο Καταστάσεων (*State Space*) και
 - ❑ Περιγραφή με Αναγωγή (*Reduction*).

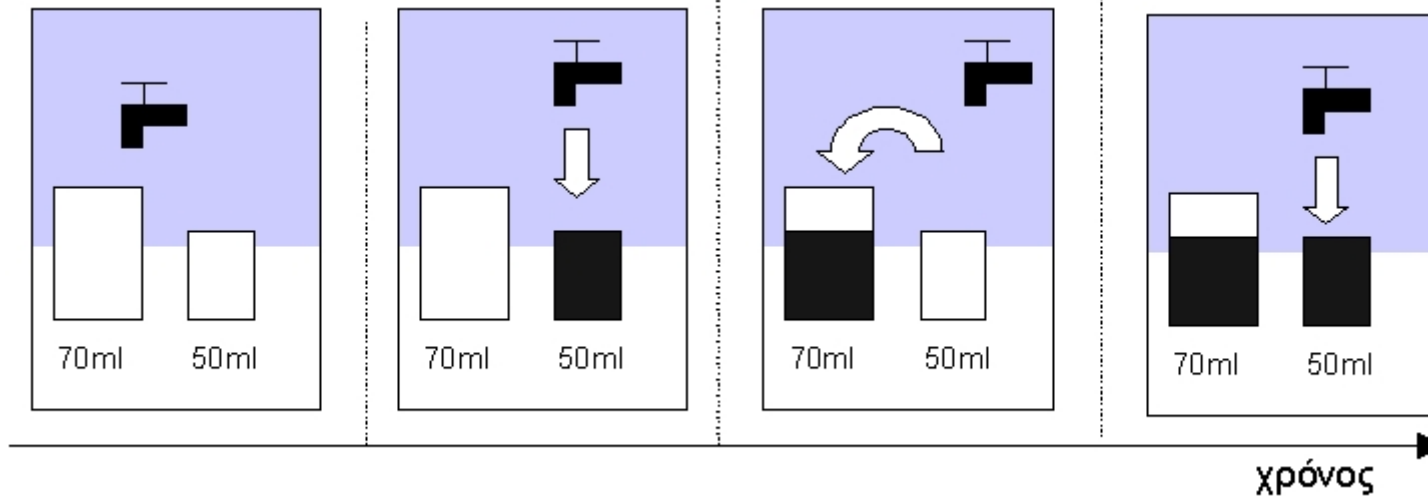
Περιγραφή Προβλημάτων με Χώρο Καταστάσεων (1/2)

❖ Ο κόσμος ενός προβλήματος αποτελείται από τα αντικείμενα, τις ιδιότητες των αντικειμένων και τις σχέσεις που τα συνδέουν

- ❑ Κλειστός κόσμος (*closed world*).
- ❑ Ανοιχτός κόσμος (*open world*).

Κατάσταση προβλήματος

Κατάσταση ενός κόσμου είναι ένα *στιγμιότυπο (instance)* ή *φωτογραφία (snapshot)* μίας συγκεκριμένης χρονικής στιγμής της εξέλιξης του κόσμου.



Περιγραφή Προβλημάτων με Χώρο Καταστάσεων (2/2)

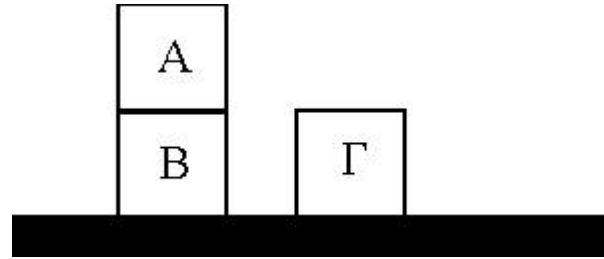
Κατάσταση (state) ενός κόσμου είναι μία επαρκής αναπαράσταση του κόσμου σε μία δεδομένη χρονική στιγμή.

- *"Επαρκής": ιδιότητες μιας κατάστασης πρέπει να επιλεγούν με τέτοιο τρόπο ώστε διαφορετικές τιμές των ιδιοτήτων να αντικατοπτρίζουν τις πραγματικές διαφορές των στιγμιότυπων του κόσμου (αφαίρεσης, abstraction).*



Παράδειγμα

Ο κόσμος των κύβων



Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος A	Κύβος A είναι ελεύθερος	Κύβος A πάνω στον κύβο B
Κύβος B	Κύβος Γ είναι ελεύθερος	Κύβος B πάνω στο T
Κύβος Γ	T έχει αρκετό ελεύθερο χώρο	Κύβος Γ πάνω στο T
T είναι Τραπέζι	Κύβος B δεν είναι ελεύθερος	

❖ Κόσμος του προβλήματος: Τρεις κύβοι και ένα τραπέζι.

❖ Μια Κατάσταση:

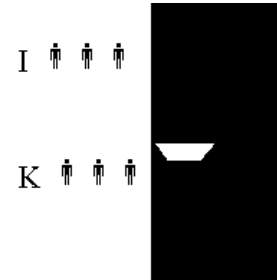
Κύβος A πάνω στον κύβο B
Κύβος B πάνω στο T
Κύβος Γ πάνω στο T
Κύβος A ελεύθερος
Κύβος Γ ελεύθερος

❖ Άλλα χαρακτηριστικά (π.χ. χρώμα, βάρος, κλπ) είναι αδιάφορα.



Παράδειγμα

Ιεραπόστολοι και Κανίβαλοι



Αντικείμενα	Ιδιότητες	Σχέσεις
3 Ιεραπόστολοι 3 Κανίβαλοι Βάρκα Αριστερή Όχθη Δεξιά Όχθη	Βάρκα δύο ατόμων	Ιεραπόστολοι στην αριστερή όχθη Κανίβαλοι στην αριστερή όχθη Βάρκα στην αριστερή όχθη

Τελεστές Μετάβασης

- ❖ **Τελεστής μετάβασης** (*transition operator*) ή **ενέργεια** (*action*) είναι μια αντιστοίχιση μίας κατάστασης του κόσμου σε νέες καταστάσεις.
- ❖ **Παράδειγμα:** Στον κόσμο των κύβων, οι τελεστές μετάβασης είναι:
 - ❑ *Βάλε τον κύβο A πάνω στον κύβο Γ.*
 - ❑ *Βάλε τον κύβο A πάνω στον κύβο B.*
 - ❑ *κλπ*
- ❖ Στους τελεστές χρησιμοποιούμε και μεταβλητές.
 - ❑ **Παράδειγμα:** *Βάλε κάποιον κύβο X πάνω σε κάποιον κύβο Y.*
- ❖ Οι **Προϋποθέσεις** εφαρμογής (preconditions) που πρέπει να τηρούνται για να εφαρμοστεί ένας τελεστής.
- ❖ Η κατάσταση που προκύπτει πρέπει να είναι **Έγκυρη** (*valid*).

Τελεστές Μετάβασης

Παράδειγμα

Τελεστής:

Μετέφερε δύο ιεραπόστολους από την αριστερή όχθη στη δεξιά

Προϋποθέσεις:

Υπάρχουν τουλάχιστον 2 ιεραπόστολοι στην αριστερή όχθη.

Η βάρκα είναι στην αριστερή όχθη.

Ο αριθμός των ιεραποστόλων που θα προκύψει στην αριστερή όχθη να μην είναι μικρότερος από τον αριθμό των κανιβάλων ή να μην υπάρχει άλλος ιεραπόστολος στην αριστερή όχθη.

Αποτελέσματα:

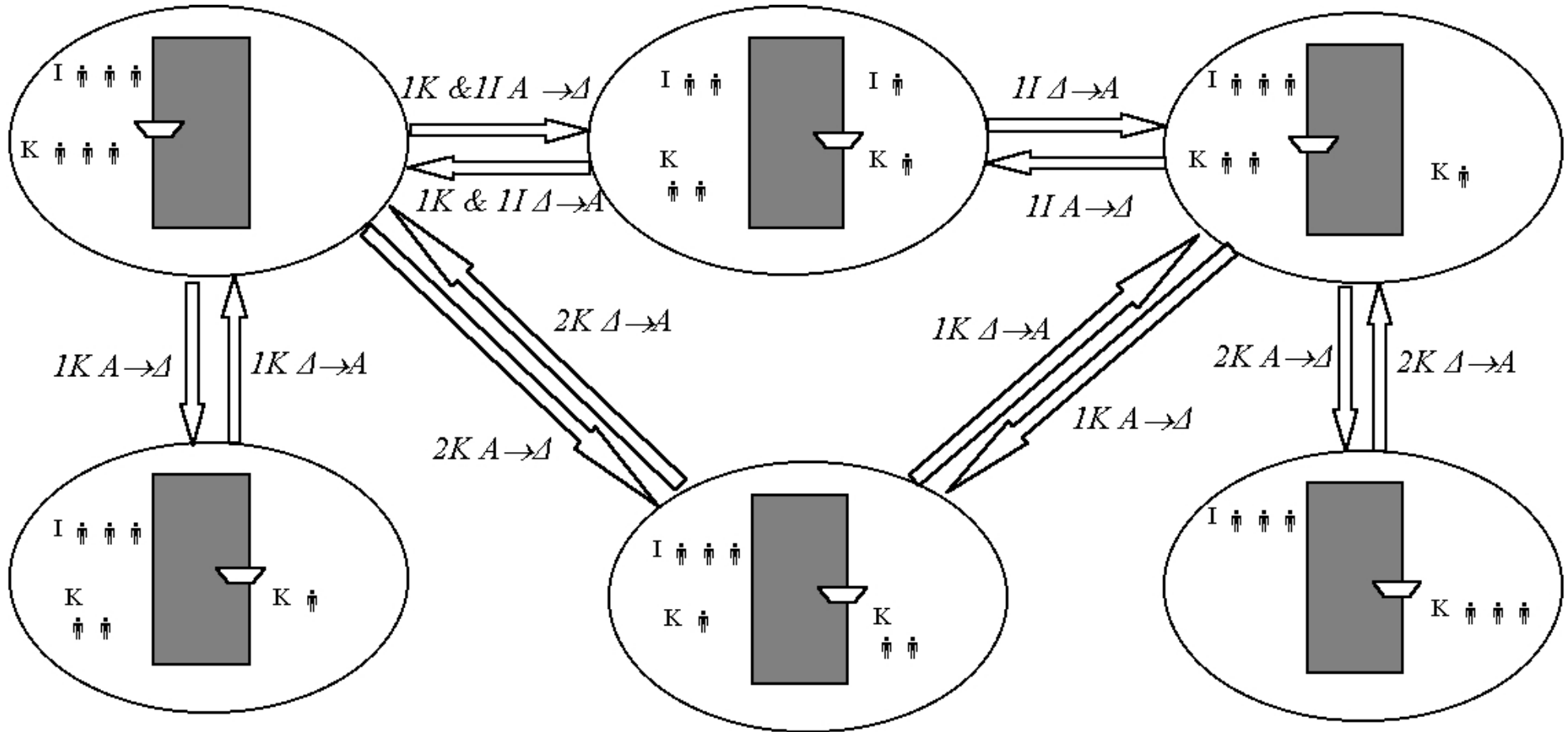
Ο αριθμός των ιεραποστόλων στην αριστερή όχθη μειώνεται κατά 2.

Ο αριθμός των ιεραποστόλων στην δεξιά όχθη αυξάνεται κατά 2.

Η βάρκα είναι πλέον δεξιά και όχι αριστερά

Χώρος Καταστάσεων

Χώρος καταστάσεων (*state space* ή *domain space*) ενός προβλήματος ονομάζεται το σύνολο όλων των έγκυρων καταστάσεων.



Αρχικές και Τελικές καταστάσεις

- ❖ Η αρχική (*initial state*) και τελική (*final* ή *goal state*) ή τελικές καταστάσεις εκφράζουν το δεδομένο και το ζητούμενο αντίστοιχα.

8	3	5
4	1	7
2		6

Αρχική Κατάσταση

1	2	3
4	5	6
7	8	

Τελική Κατάσταση

Ορισμός προβλήματος

- ❖ Ένα πρόβλημα (*Problem*) ορίζεται ως η τετράδα $P = (I, G, T, S)$ όπου:
 - ❑ I είναι η αρχική κατάσταση, $I \in S$
 - ❑ G είναι το σύνολο των τελικών καταστάσεων, $G \subseteq S$
 - ❑ T είναι το σύνολο των τελεστών μετάβασης, $T: S \leftrightarrow S$
 - ❑ S είναι ο χώρος καταστάσεων.

Λύση προβλήματος

Λύση (Solution) σε ένα πρόβλημα (I, G, T, S) , είναι μία ακολουθία από τελεστές μετάβασης $t_1, t_2, \dots, t_n \in T$ με την ιδιότητα $g = t_n(\dots(t_2(t_1(I))))$, όπου $g \in G$

❖ Λύση σε ένα πρόβλημα είναι η ακολουθία τελεστών που εφαρμόζονται στην αρχική κατάσταση για να προκύψει η τελική κατάσταση.

❖ **Παράδειγμα:**

Μετέφερε 1 ιεραπόστολο και 1 κανίβαλο από την αριστερή στη δεξιά όχθη

Μετέφερε 1 ιεραπόστολο από τη δεξιά στην αριστερή όχθη

Μετέφερε 2 κανίβαλους από την αριστερή στη δεξιά όχθη

Μετέφερε 1 κανίβαλο από τη δεξιά στην αριστερή όχθη

Μετέφερε 2 ιεραπόστολους από την αριστερή στη δεξιά όχθη

Μετέφερε 1 ιεραπόστολο και 1 κανίβαλο από τη δεξιά στην αριστερή όχθη

Μετέφερε 2 ιεραπόστολους από την αριστερή στη δεξιά όχθη

Μετέφερε 1 κανίβαλο από τη δεξιά στην αριστερή όχθη

Μετέφερε 2 κανίβαλους από την αριστερή στη δεξιά όχθη

Μετέφερε 1 ιεραπόστολο από τη δεξιά στην αριστερή όχθη

Μετέφερε 1 ιεραπόστολο και 1 κανίβαλο από την αριστερή στη δεξιά όχθη

Κατηγορίες προβλημάτων (1/2)

- ❖ Κατηγοριοποίηση ανάλογα με την ερμηνεία του όρου "λύση".
- ❖ Προβλήματα όπου είναι πλήρως γνωστές οι τελικές καταστάσεις και επιδιώκεται η εύρεση μίας σειράς ενεργειών:
 - ❑ **προβλήματα σχεδιασμού ενεργειών** (*planning*) και *προβλήματα πλοήγησης, στρατηγικής, εφοδιαστικής, κτλ.*
- ❖ Προβλήματα όπου είναι γνωστές κάποιες ιδιότητες μόνο της τελικής κατάστασης και επιδιώκεται η εύρεση ενός πλήρους στιγμιότυπου της τελικής κατάστασης,
 - ❑ *προβλήματα χρονοπρογραμματισμού* (*scheduling*), *σταυρόλεξα, κρυπτογραφικά, κτλ.*
 - ❑ *τα προβλήματα είναι γνωστά ως* **προβλήματα ικανοποίησης περιορισμών** (*constraint satisfaction problems*).

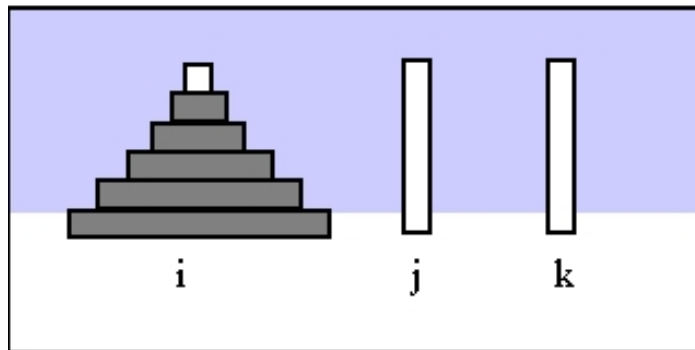
Κατηγορίες προβλημάτων (2/2)

- ❖ Προβλήματα στα οποία είναι γνωστές κάποιες ιδιότητες μόνο της τελικής κατάστασης και επιδιώκεται η εύρεση μίας πλήρως γνωστής τελικής κατάστασης και η σειρά ενεργειών που θα οδηγήσουν σε αυτή:
 - ❑ **προβλήματα διαμόρφωσης** (*configuration*).
- ❖ Προβλήματα όπου είναι σχετικά εύκολο να βρεθούν λύσεις, αλλά το ζητούμενο είναι η βέλτιστη από αυτές.
 - ❑ **προβλήματα βελτιστοποίησης**, στα οποία και πάλι η τελική κατάσταση δεν είναι πλήρως γνωστή αλλά είναι γνωστά κάποια χαρακτηριστικά της.

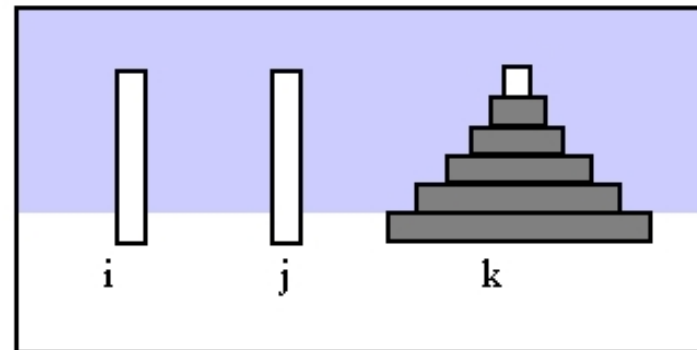
Περιγραφή με Αναγωγή (1/2)

- ❖ Μία ακολουθία από τελεστές ανάγουν την περιγραφή ενός προβλήματος σε υποπροβλήματα τα οποία είναι άμεσα επιλύσιμα, **αρχέγονα** (Primitive Problems).
- ❖ **Παράδειγμα:** Για να μεταφερθούν $n > 1$ δίσκοι από τον στύλο i στο στύλο k , πρέπει:
 - ❑ να μεταφερθούν $n-1$ δίσκοι από το i στο j ,
 - ❑ να μεταφερθεί 1 δίσκος από το i στο k ,
 - ❑ να μεταφερθούν $n-1$ δίσκοι από το j στο k .

Αρχική και τελική περιγραφή προβλήματος



Αρχική Περιγραφή



Τελική Περιγραφή

Περιγραφή με Αναγωγή (2/2)

- ❖ Αρχική Περιγραφή.
- ❖ Ένας **Τελεστής Αναγωγής** (reduction operator) ανάγει ένα πρόβλημα σε υποπροβλήματα.
- ❖ Τελική Περιγραφή.

Ορισμός προβλήματος

- ❖ Ένα πρόβλημα ορίζεται τυπικά ως η τετράδα $P = (ID, GD, TR, PP)$
 - ❑ όπου ID είναι η αρχική περιγραφή,
 - ❑ GD είναι ένα σύνολο από τελικές περιγραφές,
 - ❑ TR είναι ένα σύνολο τελεστών αναγωγής και
 - ❑ PP είναι ένα σύνολο από αρχέγονα προβλήματα.



Αλγόριθμοι Αναζήτησης

- ❖ Δοθέντος ενός προβλήματος με περιγραφή στο χώρο καταστάσεων ή με αναγωγή, στόχος είναι να βρεθεί η λύση του.
- ❖ Οι αλγόριθμοι που αναζητούν τη λύση σε ένα πρόβλημα ονομάζονται **αλγόριθμοι αναζήτησης** (*search algorithms*)
- ❖ Η επιλογή ενός αλγορίθμου αναζήτησης για ένα συγκεκριμένο πρόβλημα είναι σημαντική, διότι οι αλγόριθμοι αυτοί διαφέρουν μεταξύ τους σε αρκετά χαρακτηριστικά.

Αλγόριθμοι Αναζήτησης

Τυφλοί

Όνομα Αλγορίθμου	Συντομογραφία	Ελληνική Ορολογία
Depth-First Search	DFS	Αναζήτηση Πρώτα σε Βάθος
Breadth-First Search	BFS	Αναζήτηση Πρώτα σε Πλάτος
Iterative Deepening	ID	Επαναληπτική Εκβάθυνση
Bi-directional Search	BiS	Αναζήτηση Διπλής Κατεύθυνσης
Branch and Bound	B&B	Επέκταση και Οριοθέτηση
Beam Search	BS	Ακτινωτή Αναζήτηση

Ευριστικοί

Hill Climbing	HC	Αναρρίχηση Λόφων
Best-First Search	BestFS	Αναζήτηση Πρώτα στο Καλύτερο
A* (A-star)	A*	A* (Άλφα Άστρο)

Παιχνιδιών 2 ατόμων

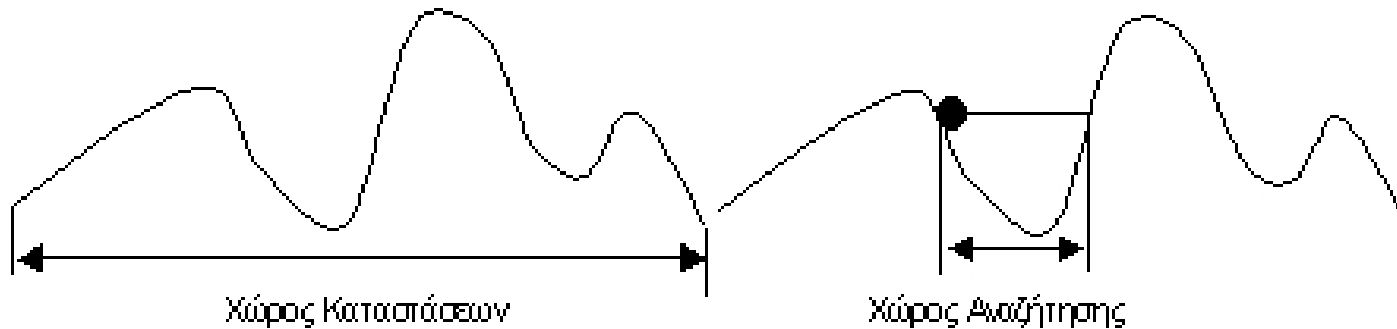
Minimax	Minimax	Αναζήτηση Μεγίστου-Ελαχίστου
Alpha-Beta	AB	Άλφα-Βήτα

Χώρος Αναζήτησης

Δοθέντος ενός προβλήματος (I, G, T, S) , *χώρος αναζήτησης (search space) SP* είναι το σύνολο όλων των καταστάσεων που είναι προσβάσιμες από την αρχική κατάσταση.

Μία κατάσταση s ονομάζεται *προσβάσιμη (accessible)* αν υπάρχει μια ακολουθία τελεστών μετάβασης $t_1, t_2, \dots, t_k \in T$ τέτοια ώστε $s = t_k(\dots(t_2(t_1(I))))$.

- ❖ Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων, δηλαδή $SP \subseteq S$.



- ❖ Ένας αλγόριθμος αναζήτησης δεν μειώνει τον χώρο αναζήτησης (που είναι δεδομένος) αλλά καθορίζει τον αριθμό των καταστάσεων που επισκέπτεται.

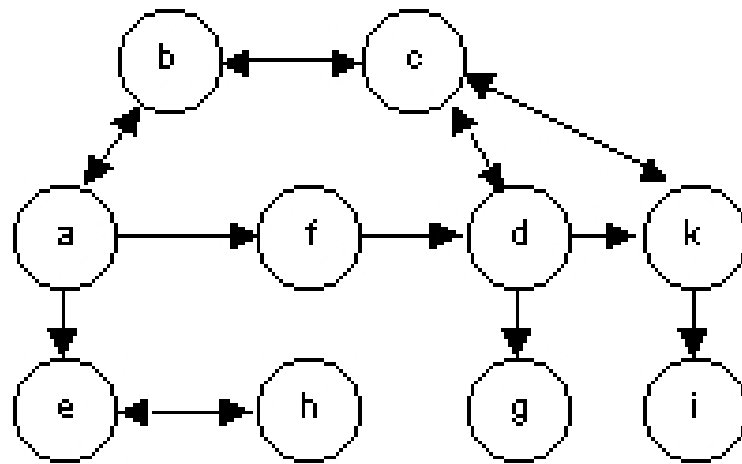
Χώρος Αναζήτησης ως Δένδρο Αναζήτησης (1/2)

- ❖ Ο χώρος αναζήτησης μπορεί να αναπαρασταθεί με γράφο.

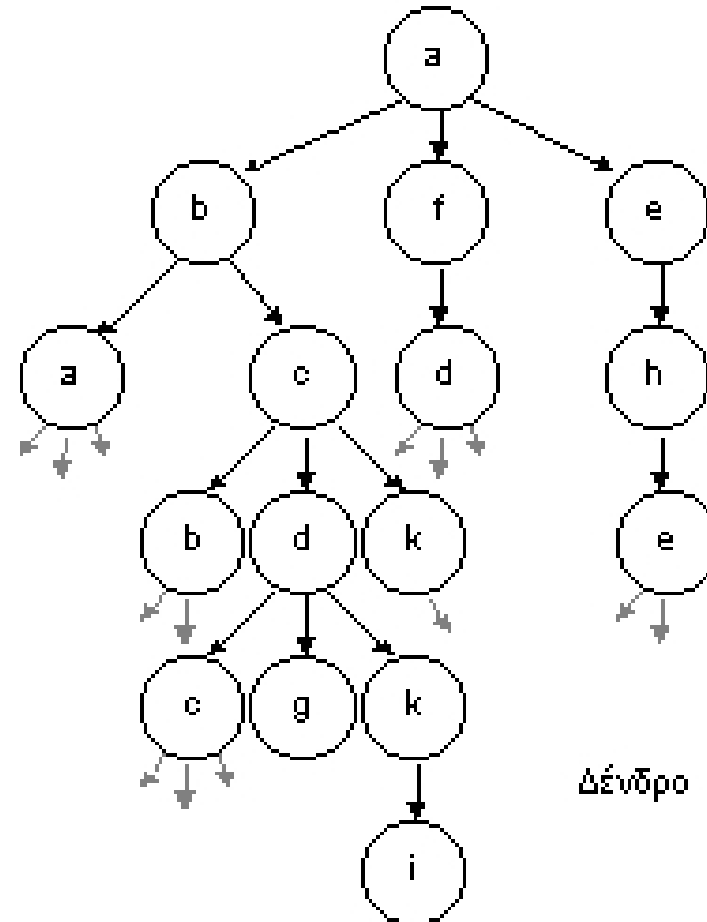
Τμήμα Δένδρου	Αναπαράσταση
Κόμβος (Node)	Κατάσταση
Ρίζα (Root)	Αρχική Κατάσταση
Φύλλο (Tip, Leaf) ή Τερματικός κόμβος	Τελική Κατάσταση ή Αδιέξοδο (Dead Node), δηλαδή κατάσταση στην οποία δεν μπορεί να εφαρμοστεί κανένας τελεστής μετάβασης.
Κλαδί (Branch)	Τελεστής Μετάβασης που μετατρέπει μια κατάσταση-Γονέα (Parent State) σε μία άλλη κατάσταση-Παιδί (Child State).
Λύση (Solution)	Μονοπάτι (Path) που ενώνει την αρχική με μία τελική κατάσταση
Επέκταση (Expansion)	Η διαδικασία παραγωγής όλων των καταστάσεων-παιδιών ενός κόμβου.
Παράγοντας Διακλάδωσης (Branching Factor)	Ο αριθμός των καταστάσεων-Παιδιών που προκύπτουν από την επέκταση μιας κατάστασης. Επειδή δεν είναι σταθερός αριθμός, αναφέρεται και ως Μέσος Παράγοντας Διακλάδωσης (Average Branching Factor).

Χώρος Αναζήτησης ως Δένδρο Αναζήτησης (2/2)

- ❖ Είναι πάντα εφικτό να μετατραπεί ο γράφος σε **δένδρο αναζήτησης** (*search tree*), το οποίο όμως μπορεί να έχει μονοπάτια απείρου μήκους.



Γράφος



Δένδρο

Το φαινόμενο της εκθετικής αύξησης του αριθμού των κόμβων του δένδρου ονομάζεται συνδυαστική έκρηξη (*combinatorial explosion*).

Χαρακτηριστικά Αλγορίθμων (1/2)

- ❖ Ένας αλγόριθμος είναι μία αυστηρά καθορισμένη ακολουθία βημάτων-εντολών που επιδιώκει να λύσει ένα πρόβλημα.
- ❖ Δοθέντος ενός προβλήματος $P=(I,G,T,S)$ και μετά την εφαρμογή κάποιου αλγορίθμου στο χώρο αναζήτησής του, προκύπτει το **επιλυμένο πρόβλημα** (*solved problem*), το οποίο ορίζεται ως μία τετράδα $P_s=(V,A,F,G_s)$, όπου:
 - ❑ V είναι το σύνολο των καταστάσεων που εξέτασε ο αλγόριθμος αναζήτησης,
 - ❑ A είναι ο αλγόριθμος που χρησιμοποιήθηκε,
 - ❑ F είναι το σύνολο των λύσεων που βρέθηκαν, και
 - ❑ G_s είναι το σύνολο των τελικών καταστάσεων που εξετάστηκαν.
- ❖ Ο αριθμός των καταστάσεων που περιέχει το V και η σχέση του με το χώρο καταστάσεων S ενός προβλήματος και τον χώρο αναζήτησης SP , είναι ένα από τα χαρακτηριστικά της αποδοτικότητας του αλγορίθμου.

Χαρακτηριστικά Αλγορίθμων (2/2)

- ❖ Ένας αλγόριθμος ονομάζεται **εξαντλητικός** (*exhaustive*) όταν το σύνολο των καταστάσεων που εξετάζει ο αλγόριθμος για να βρει τις απαιτούμενες λύσεις είναι ίσο με το χώρο αναζήτησης, δηλαδή $V=SP$.
- ❖ Ένας αλγόριθμος δεν λύνει πάντα κάποιο πρόβλημα, έστω και αν υπάρχει κάποια λύση. Τότε τα σύνολα G_s και F είναι κενά.
- ❖ Ένας αλγόριθμος αναζήτησης ονομάζεται **πλήρης** (*complete*) αν εγγυάται ότι θα βρει μία λύση για οποιαδήποτε τελική κατάσταση, αν τέτοια λύση υπάρχει. Σε αντίθετη περίπτωση, ο αλγόριθμος ονομάζεται μη-πλήρης (*incomplete*).
- ❖ Μία λύση ονομάζεται **βέλτιστη** (*optimal*) αν οδηγεί στην καλύτερη, σύμφωνα με τη διάταξη, τελική κατάσταση. Όταν δεν υπάρχει διάταξη, μία λύση ονομάζεται βέλτιστη αν είναι η συντομότερη (*shortest*).
- ❖ Ένας αλγόριθμος αναζήτησης καλείται **αποδεκτός** (*admissible*) αν εγγυάται ότι θα βρει τη βέλτιστη λύση, αν μια τέτοια λύση υπάρχει.

Διαδικασία Επιλογής Αλγορίθμου Αναζήτησης

- ❖ Η επιλογή ενός αλγορίθμου βασίζεται στα εξής κριτήρια:
 - αριθμός των καταστάσεων που αυτός επισκέπτεται
 - δυνατότητα εύρεσης λύσεων εφόσον αυτές υπάρχουν
 - αριθμός των λύσεων
 - ποιότητα των λύσεων
 - αποδοτικότητά του σε χρόνο
 - αποδοτικότητά του σε χώρο (μνήμη)
 - ευκολία υλοποίησής του
 - κλάδεμα

Κλάδεμα ή αποκοπή καταστάσεων (pruning) του χώρου αναζήτησης είναι η διαδικασία κατά την οποία ο αλγόριθμος απορρίπτει, κάτω από ορισμένες συνθήκες, κάποιες καταστάσεις και μαζί με αυτές όλο το υποδένδρο που εκτυλίσσεται κάτω από τις καταστάσεις αυτές.

Γενικός Αλγόριθμος Αναζήτησης

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι άδειο τότε σταμάτησε.
3. Πάρε την πρώτη σε σειρά κατάσταση του μετώπου της αναζήτησης.
4. Αν είναι η κατάσταση αυτή μέρος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
5. Αν είναι η κατάσταση αυτή τελική κατάσταση τότε τύπωσε τη λύση και πήγαινε στο βήμα 2.
6. Εφάρμοσε τους τελεστές μετάβασης για να παράγεις τις καταστάσεις-παιδιά.
7. Βάλε τις νέες καταστάσεις-παιδιά στο μέτωπο της αναζήτησης.
8. Κλάδεψε τις καταστάσεις που δε χρειάζονται (σύμφωνα με κάποιο κριτήριο), βγάζοντάς τες από το μέτωπο της αναζήτησης.
9. Κάνε αναδιάταξη στο μέτωπο της αναζήτησης (σύμφωνα με κάποιο κριτήριο).
10. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
11. Πήγαινε στο βήμα 2.

Γενικός Αλγόριθμος (ψευδοκώδικας)

```
algorithm general(InitialState, FinalState)
begin
  Closed $\leftarrow$   $\emptyset$ ;
  Frontier $\leftarrow$  <InitialState>;
  CurrentState $\leftarrow$  First(Frontier);
  while CurrentState  $\notin$  FinalState do
    Frontier $\leftarrow$  delete(CurrentState, Frontier);
    if CurrentState  $\notin$  ClosedSet then
      begin
        Next $\leftarrow$  Expand(CurrentState);
        Frontier $\leftarrow$  insert(Next, Frontier);
        Frontier $\leftarrow$  prune(Frontier);
        Frontier $\leftarrow$  reorder(Frontier);
        Closed $\leftarrow$  Closed  $\cup$  {CurrentState};
      end;
    if Frontier =  $\emptyset$  then return failure;
    CurrentState $\leftarrow$  First(Frontier);
  endwhile;
end.
```