



Αριστοτέλειο
Πανεπιστήμιο
Θεσσαλονίκης

Τεχνητή Νοημοσύνη

Περιγραφή Προβλημάτων και Αναζήτηση Λύσης

Ιώαννης Βλαχάβας

Τμήμα Πληροφορικής ΑΠΘ



Άδειες Χρήσης

Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons. Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα. Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης**» έχει χρηματοδοτήσει μόνο τη 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.



Επίλυση Προβλημάτων

Επίλυση Προβλημάτων (1/2)

- ❖ Η ικανότητα επίλυσης προβλημάτων αποτελεί ένα από τα βασικά χαρακτηριστικά γνωρίσματα της νοημοσύνης.
- ❖ Είναι το αντικείμενο της Τεχνητής Νοημοσύνης
 - Για παράδειγμα, είναι απαραίτητο για μια ευφυή οντότητα που κινείται σε ένα χώρο με εμπόδια, να μπορεί να βρίσκει τη διαδρομή προς το σημείο προορισμού της.
- ❖ η περιγραφή ενός προβλήματος (Problem description),
- ❖ η αναπαράσταση γνώσης (Knowledge representation) και
- ❖ η αναζήτηση (Search algorithms)

συνθέτουν τον πυρήνα κάθε εφαρμογής ΤΝ

Τεχνητή Νοημοσύνη = Περιγραφή Προβλήματος + Αναπαράσταση Γνώσης + Αλγόριθμοι Αναζήτησης

Επίλυση Προβλημάτων (2/2)

- ❖ **Περιγραφή Προβλήματος:** περιγραφή του χώρου καταστάσεων, ορισμός τελεστών μετάβασης, περιγραφή της αρχικής κατάστασης και των τελικών καταστάσεων.
 - Στο παράδειγμα της εύρεσης διαδρομής, η περιγραφή του προβλήματος περιλαμβάνει τον καθορισμό του ελεύθερου από εμπόδια χώρου, τις κινήσεις που μπορεί να εκτελέσει το ρομπότ και τον καθορισμό του αρχικού και του τελικού σημείου της διαδρομής.
- ❖ **Αναπαράσταση Γνώσης:** για τον περιβάλλοντα κόσμο με τέτοιο τρόπο ώστε να είναι δυνατή η επεξεργασία αυτής της γνώσης και η εξαγωγή συμπερασμάτων από ένα υπολογιστικό σύστημα.
- ❖ **Αναζήτηση:** Η λύση σε ένα πρόβλημα δίνεται με την εφαρμογή ενός **αλγορίθμου αναζήτησης**. Κάθε αλγόριθμος έχει τέτοια χαρακτηριστικά που τον κάνουν ικανό σε κάποιο πρόβλημα να αναζητήσει πιο αποδοτικά τη λύση στο χώρο αναζήτησης από κάποιον άλλο.



Περιγραφή Προβλημάτων και Αναζήτηση Λύσης

Περιγραφή Προβλημάτων

- ❖ Η έννοια του προβλήματος είναι διαισθητικά γνωστή σε όλους: υπάρχει μία δεδομένη κατάσταση (αρχική), υπάρχει μία επιθυμητή κατάσταση (τελική) και διαθέσιμες ενέργειες που πρέπει να γίνουν ώστε να φτάσουμε στην επιθυμητή.
- ❖ Πολλές φορές στην καθημερινότητα, η έννοια "πρόβλημα" χρησιμοποιείται με λάθος τρόπο. Π.χ. αν δεν είναι γνωστό ένα από τα παραπάνω τρία βασικά συστατικά, δηλαδή κάποιος δεν έχει στόχους ή δε γνωρίζει τι ενέργειες μπορεί να εκτελέσει, τότε δεν υφίσταται πρόβλημα με την τυπική έννοια του όρου.
- ❖ Καθημερινά καλούμαστε να αντιμετωπίσουμε πολλά προβλήματα σε ποικίλα περιβάλλοντα.

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

- ❖ Ο ορισμός ή η περιγραφή ενός προβλήματος πρέπει να είναι ανεξάρτητη από την πολυπλοκότητα επίλυσής του.
- ❖ Η πολυπλοκότητα καθορίζεται από την πολυπλοκότητα του αλγορίθμου αναζήτησης που εφαρμόζεται για την επίλυσή του.



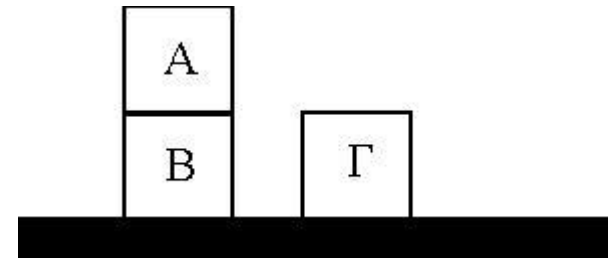
Κατηγορίες Προβλημάτων

- ❖ Πραγματικά και πολύπλοκα προβλήματα (*real world problems*):
 - σκάκι (chess),
 - πλανόδιος πωλητής (traveling salesperson),
 - N-βασίλισσες (N-queens),
 - σάκος (knapsack),
 - κλπ.
- ❖ Απλά προβλήματα (toy problems)

Απλά προβλήματα (toy problems)

κύβοι (*blocks*),

Τρεις κύβοι βρίσκονται σε τυχαία διάταξη πάνω στο τραπέζι με σκοπό να μετακινηθούν ώστε να σχηματιστεί μία άλλη διάταξη.



N-puzzle,

Ένα πλαίσιο περιέχει πλακίδια αριθμημένα από το 1 έως το 8. Τα πλακίδια μπορούν να γλιστρήσουν ανάμεσα στα άλλα, εφόσον μία θέση είναι κενή. Σκοπός είναι να έρθουν τα πλακίδια σε κάποια προκαθορισμένη διάταξη.

8	3	5
4	1	7
2		6

		O
O	X	
	X	

τρίλιζα (*tic-tac-toe*),

Δύο παίκτες εναλλάξ σημειώνουν X ή O στις εννέα θέσεις του ταμπλό με σκοπό κάποιος να έχει τρία X ή τρία O στη σειρά., οριζόντια, κατακόρυφα ή διαγώνια.

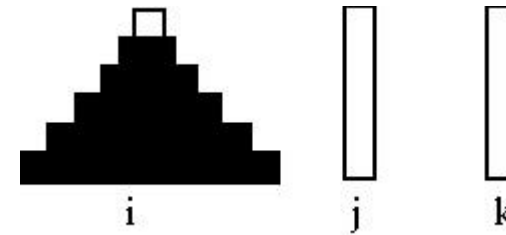
λαβύρινθος (*maze*),

Αναζήτηση μίας διαδρομής σε ένα λαβύρινθο από την είσοδο έως την έξοδο.



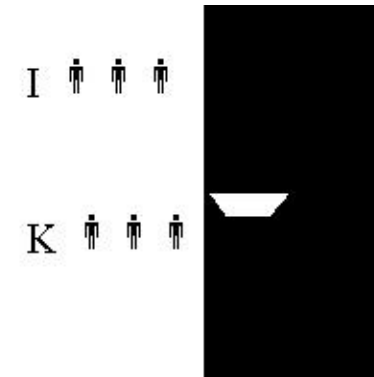
πύργοι του Ανόι (*Hanoi towers*,

Δίσκοι διαφορετικού μεγέθους πρέπει να μεταφερθούν ένας-ένας με ίδια τελική διάταξη σε ένα στύλο, χωρίς σε κάποια στιγμή ένας μεγαλύτερος να μπει πάνω από ένα μικρότερο.



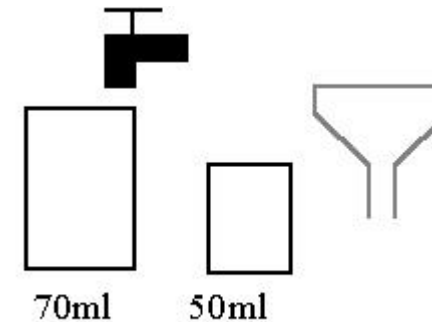
κανίβαλοι και ιεραπόστολοι (*missionaries and cannibals*,

Τρεις κανίβαλοι και τρεις ιεραπόστολοι πρέπει να περάσουν το ποτάμι με μία βάρκα που χωράει δύο άτομα, χωρίς οι κανίβαλοι σε μία από τις δύο όχθες να υπερβαίνουν σε αριθμό του ιεραπόστολους.



ποτήρια (*water glass*),

Δύο ποτήρια που χωράνε 50ml και 70ml, γεμίζουν μέχρι το χείλος από μία βρύση και αδειάζουν είτε το ένα μέσα στο άλλο είτε στον νεροχύτη με τελικό στόχο να καταλήξουν ακριβώς 40ml σε ένα από τα δύο.



❖ Η μεθοδολογία που εφαρμόζεται στην αναπαράσταση και την επίλυση είναι παρόμοια και στις δύο ακραίες περιπτώσεις προβλημάτων (Πραγματικά – Απλά).

❖ Η περιγραφή ενός προβλήματος μπορεί να γίνει με δύο βασικούς τρόπους:

- Περιγραφή με Χώρο Καταστάσεων (State Space) και
- Περιγραφή με Αναγωγή (Reduction).

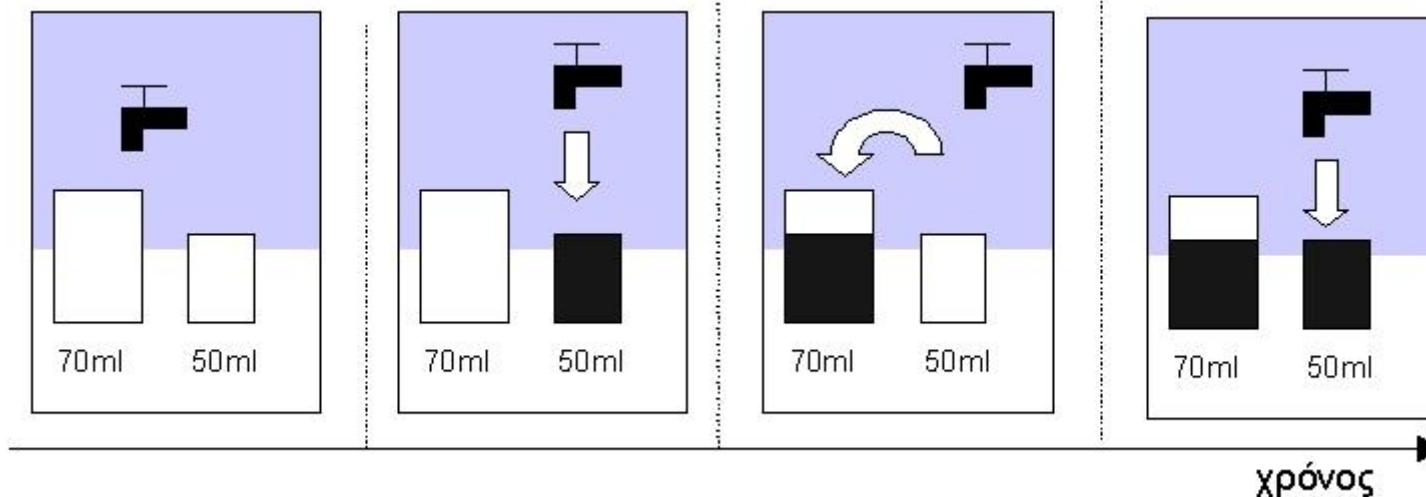
Περιγραφή Προβλημάτων με Χώρο Καταστάσεων

❖ Κόσμος προβλήματος

- αποτελείται μόνον από τα αντικείμενα που υπάρχουν σε αυτόν, τις ιδιότητες των αντικειμένων και τις σχέσεις που τα συνδέουν. Άρα, είναι υποσύνολο του πραγματικού κόσμου.
- Κλειστός κόσμος (closed world). Τίποτα δεν εισάγεται ή εξάγεται από ή προς άλλο κόσμο.
- Ανοιχτός κόσμος (open world) . Το αντίθετο του κλειστού.

Κατάσταση προβλήματος

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



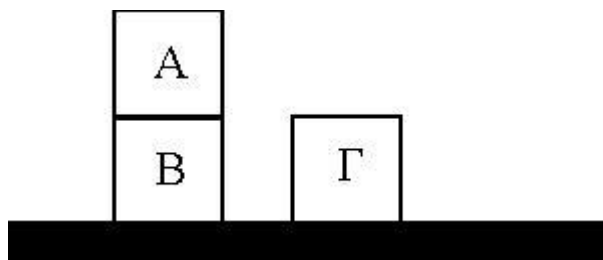


Τυπικά:

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

- Με τη λέξη "επαρκής" εννοείται ότι οι χαρακτηριστικές ιδιότητες μιας κατάστασης του κόσμου πρέπει να επιλεγούν με τέτοιο τρόπο ώστε διαφορετικές τιμές των ιδιοτήτων να αντικατοπτρίζουν τις πραγματικές διαφορές των στιγμιότυπων του κόσμου.
- Αυτό επιτυγχάνεται με τη λειτουργία της αφαίρεσης (abstraction), σύμφωνα με την οποία για να περιγραφεί μία κατάσταση αφαιρούνται όλες εκείνες οι λεπτομέρειες οι οποίες δεν είναι απαραίτητες για τη μετέπειτα επίλυση ενός προβλήματος.

Παράδειγμα



Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος Α	Κύβος Α είναι ελεύθερος	Κύβος Α πάνω στον κύβο Β
Κύβος Β	Κύβος Γ είναι ελεύθερος	Κύβος Β πάνω στο Τ
Κύβος Γ	Τ έχει αρκετό ελεύθερο χώρο	Κύβος Γ πάνω στο Τ
Τ είναι Τραπέζι	Κύβος Β δεν είναι ελεύθερος	

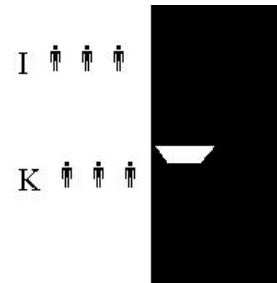
- ❖ Κόσμος του προβλήματος: Τρεις κύβοι και ένα τραπέζι (ή πάτωμα).
- ❖ Μια Κατάσταση (αυτή του σχήματος):

Κύβος Α πάνω στον κύβο Β
Κύβος Β πάνω στο Τ
Κύβος Γ πάνω στο Τ
Κύβος Α ελεύθερος
Κύβος Γ ελεύθερος

- ❖ Άλλα χαρακτηριστικά (π.χ. χρώμα, βάρος, κλπ) είναι αδιάφορα για το συγκεκριμένο πρόβλημα.



Άλλο παράδειγμα:



Αντικείμενα

3 Ιεραπόστολοι

3 Κανίβαλοι

Βάρκα δύο ατόμων

Αριστερή Όχθη

Δεξιά Όχθη

Μία Κατάσταση

Ιεραπόστολοι στην αριστερή όχθη

Κανίβαλοι στην αριστερή όχθη

Βάρκα στην αριστερή όχθη

Τελεστές μετάβασης

- ❖ Οι καταστάσεις ενός κόσμου συνδέονται μεταξύ τους.
- ❖ **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
- ❖ Επειδή δεν μπορεί να υπάρχει ένας τελεστής για κάθε περίπτωση, χρησιμοποιούμε μεταβλητές.
 - Παράδειγμα**: Βάλε κάποιον κύβο X πάνω σε κάποιον κύβο Y.
- ❖ Επειδή όλες οι μεταβάσεις δεν είναι δυνατές (π.χ. στο προηγούμενο παράδειγμα, ο κύβος B δεν μπορεί να πάει επάνω στον κύβο Γ) υπάρχουν **Προϋποθέσεις** εφαρμογής (*preconditions*) που πρέπει να τηρούνται για να εφαρμοστεί ένας τελεστής.
- ❖ Μια βασική προϋπόθεση είναι η κατάσταση που προκύπτει να είναι **Έγκυρη** (*valid*).

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.

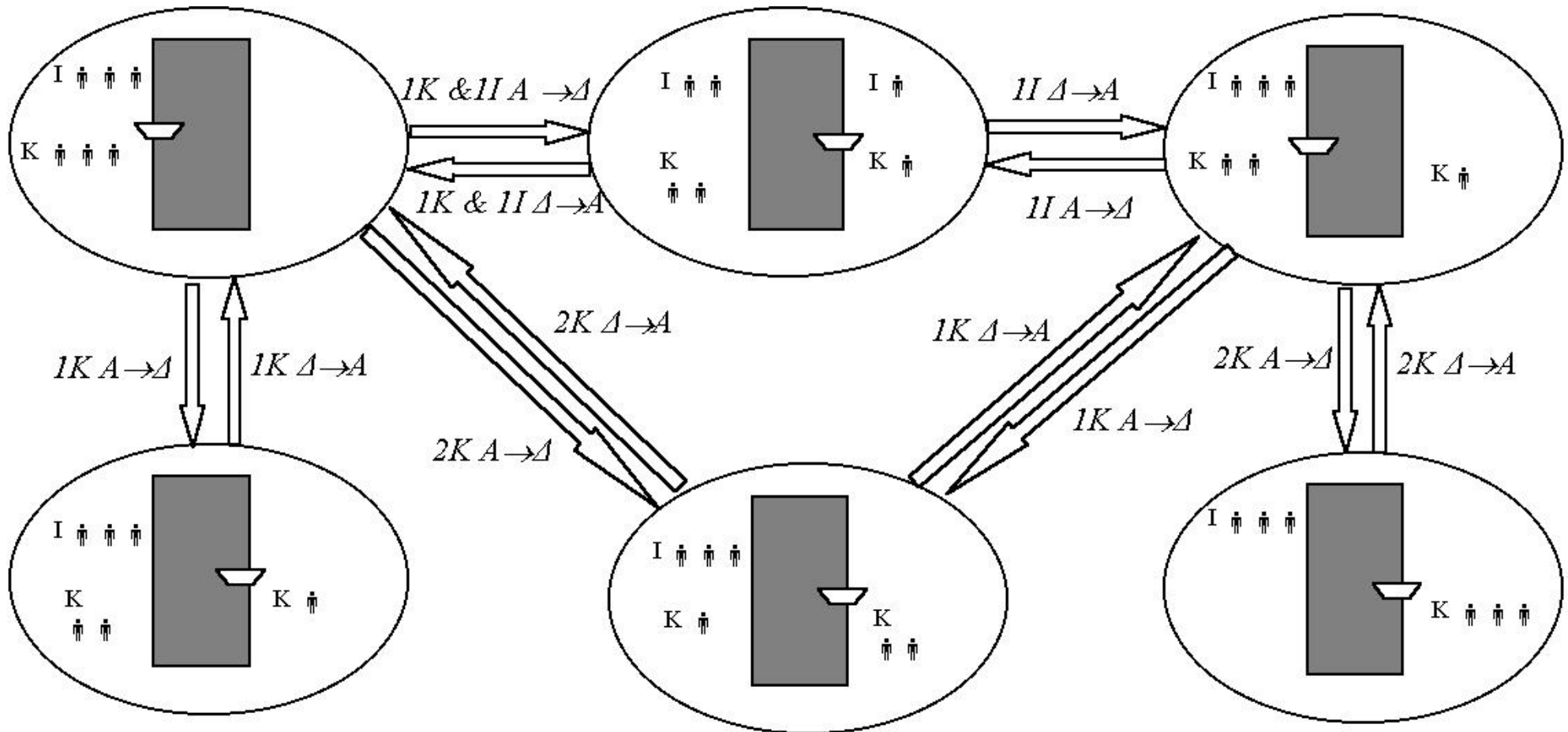
Ο αριθμός των ιεραποστόλων στην δεξιά όχθη αυξάνεται κατά 2.

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.

❖ Απεικονίζεται συνήθως διαισθητικά με ένα γράφο



Αρχικές και Τελικές καταστάσεις

- ❖ Σε κάθε πρόβλημα αρχίζουμε από κάπου και καταλήγουμε κάπου.
- ❖ Η αρχική (*initial state*) και τελική (*final* ή *goal state*) ή τελικές καταστάσεις εκφράζουν το δεδομένο και το ζητούμενο (ζητούμενα) αντίστοιχα.

8	3	5
4	1	7
2		6

Αρχική Κατάσταση

1	2	3
4	5	6
7	8	

Τελική Κατάσταση

Αρχική και Τελική Κατάσταση στο 8 Puzzle πρόβλημα.

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, $T: S \rightarrow S$
 - 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 $g \in G$

- ❖ **Δηλαδή** λύση σε ένα πρόβλημα είναι η ακολουθία τελεστών που εφαρμόζονται στην αρχική κατάσταση για να προκύψει μια τελική κατάσταση.
- ❖ **Παράδειγμα:** Η λύση στο πρόβλημα των κανιβάλων και ιεραποστόλων:
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)

- ❖ Τα προβλήματα μπορεί να κατηγοριοποιηθούν σε τέσσερις (4) κατηγορίες, ανάλογα με την ερμηνεία που δίνεται στον όρο "λύση".
- ❖ Υπάρχουν προβλήματα στα οποία είναι πλήρως γνωστές οι τελικές καταστάσεις και επιδιώκεται η εύρεση μίας σειράς ενεργειών, η εκτέλεση των οποίων προκαλεί τη μετάβαση από την αρχική κατάσταση σε μία τελική,
 - όπως τα **προβλήματα σχεδιασμού ενεργειών** (*planning*) και τα προβλήματα πλοήγησης, στρατηγικής, εφοδιαστικής, κτλ.
 - Για παράδειγμα, στο πρόβλημα 8-puzzle λύση είναι οι κινήσεις που έγιναν ώστε να διαμορφωθεί το τελικό puzzle.
- ❖ Σε άλλα προβλήματα, είναι γνωστές κάποιες ιδιότητες μόνο της τελικής κατάστασης και επιδιώκεται η εύρεση ενός πλήρους στιγμιότυπου της τελικής κατάστασης,
 - όπως σε **προβλήματα χρονοπρογραμματισμού** (*scheduling*), παζλ, σταυρόλεξα, κρυπτογραφικά, κτλ.
 - Για παράδειγμα, στο ωρολόγιο πρόγραμμα ενός σχολείου, ξεκινώντας από ένα κενό πρόγραμμα αναζητείται το πλήρες πρόγραμμα.
 - Το ίδιο ισχύει και με το σταυρόλεξο, όπου στο πλήρες στιγμιότυπο της τελικής κατάστασης όλες οι λέξεις είναι τοποθετημένες στη σωστή τους θέση και αυτό αποτελεί τη λύση του.
 - Αυτά τα προβλήματα είναι γνωστά ως **προβλήματα ικανοποίησης περιορισμών** (*constraint satisfaction problems*) και περιγράφονται σε σχετικό κεφάλαιο.

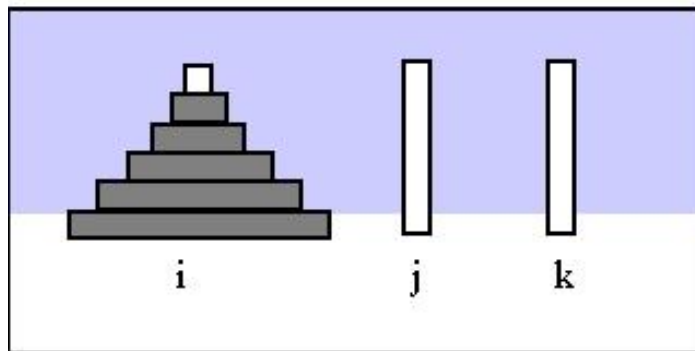
Κατηγορίες προβλημάτων (2/2)

- ❖ Συνδυασμό των παραπάνω κατηγοριών αποτελούν τα 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, όπως είναι τα **προβλήματα διαμόρφωσης** (*configuration*).
 - Για παράδειγμα, στην κατασκευή ενός αυτοκινήτου βάσει κάποιων προδιαγραφών, ζητούμενο είναι όχι μόνο τα εξαρτήματα που θα συνθέσουν το αυτοκίνητο αλλά και η σειρά με την οποία θα τοποθετηθούν.
- ❖ Τέλος, υπάρχουν τα προβλήματα όπου είναι σχετικά εύκολο να βρεθούν λύσεις, αλλά το ζητούμενο είναι η βέλτιστη από αυτές.
 - Για παράδειγμα, στο πρόβλημα του πλανόδιου πωλητή αναζητείται η συντομότερη διαδρομή που περνά από όλες τις πόλεις.
 - Αυτά είναι τα λεγόμενα **προβλήματα βελτιστοποίησης**, στα οποία και πάλι η τελική κατάσταση δεν είναι πλήρως γνωστή αλλά είναι γνωστά κάποια χαρακτηριστικά της.
 - Υπάρχει και μια κατηγορία αλγορίθμων, οι **γενετικοί αλγόριθμοι**, οι οποίοι αποτελούν ένα ανάλογο της θεωρίας της εξέλιξης, προσομοιάζοντας την εύρεση μίας καλύτερης λύσης σε ένα πρόβλημα με τη συνεχή εξέλιξη άλλων λύσεων, μέσω κατάλληλων διαδικασιών αναπαραγωγής και μετάλλαξης.

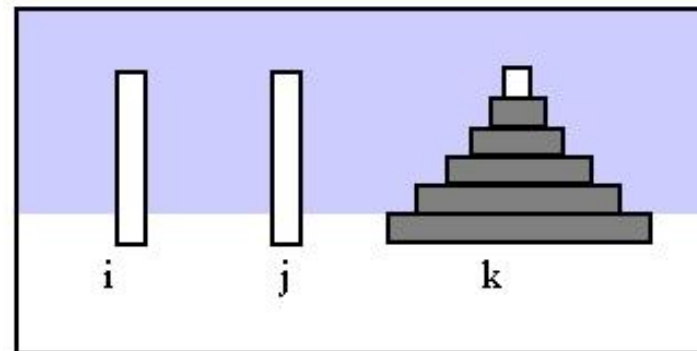
Περιγραφή με Αναγωγή (1/2)

- ❖ 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)

- ❖ Η βασική δομή δεν είναι η κατάσταση αλλά η ίδια η περιγραφή του προβλήματος.
- ❖ Αντί για αρχική κατάσταση έχουμε **Αρχική Περιγραφή**.
- ❖ Αντί για τελεστή μετάβασης έχουμε τελεστή αναγωγής.
- ❖ Ένας **Τελεστής Αναγωγής** (reduction operator) ανάγει ένα πρόβλημα σε υποπροβλήματα.
- ❖ Αν δοθεί μια αρχική περιγραφή, μια ακολουθία από τελεστές αναγωγής μπορούν να το ανάγουν σε απλούστερα (αρχέγονα), δηλαδή άμεσα επιλύσιμα.

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.
 - Η τυποποίηση της περιγραφής ενός προβλήματος διευκολύνει την επίλυσή του.
- ❖ Κατ'αντιστοιχία, η τυποποίηση των βημάτων επίλυσης διευκολύνει την υλοποίηση του τρόπου επίλυσης σε ένα υπολογιστικό σύστημα.
 - Η τυποποίηση αυτή επιτυγχάνεται μέσω αυστηρά προκαθορισμένων βημάτων, δηλαδή αλγορίθμων, που πρέπει να εφαρμοστούν για να επιλυθεί ένα πρόβλημα.
- ❖ Επειδή οι αλγόριθμοι αυτοί αναζητούν τη λύση στο 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.
 - Ο λόγος για την ύπαρξη τόσων αλγορίθμων είναι ότι κάθε ένας έχει διαφορετικά χαρακτηριστικά που τον καθιστούν περισσότερο ή λιγότερο αποδοτικό από άλλους, σε απαιτήσεις μνήμης ή/και χρόνου εκτέλεσης.

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

Ευριστικοί

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

Παιχνιδιών 2 ατόμων

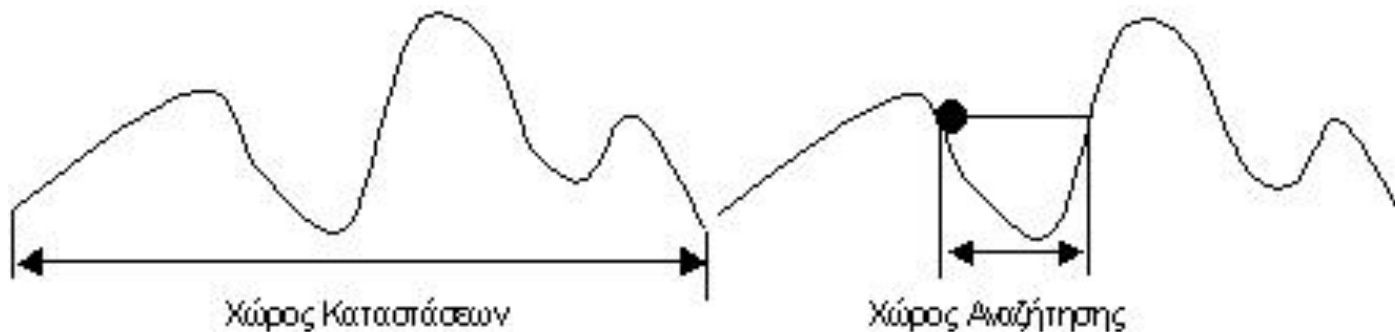
Minimax	Minimax	Αναζήτηση Μεγίστου-Ελαχίστου
Alpha-Beta	AB	Άλφα-Βήτα

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 $s = t_k(\dots(t_2(t_1(I))))$.

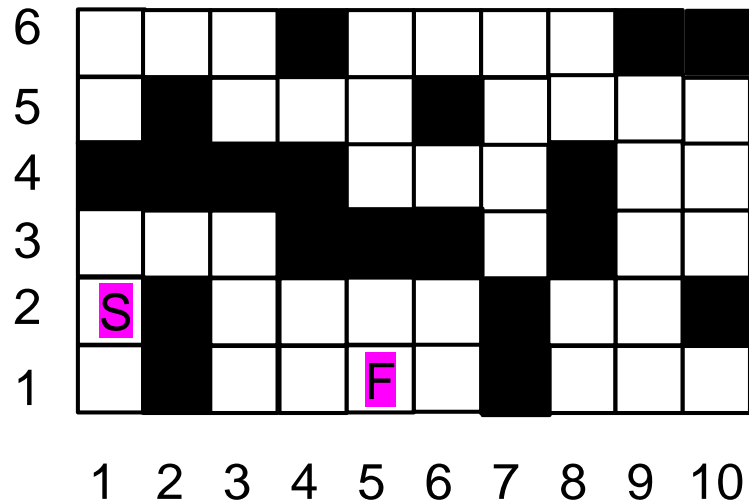
- ❖ Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων, δηλαδή $SP \subseteq S$.
- ❖ Μπορεί και να συμπίπτει αλλά συνήθως είναι μικρότερος γιατί ο χώρος αναζήτησης εξαρτάται από την αρχική κατάσταση ενώ ο χώρος καταστάσεων όχι.



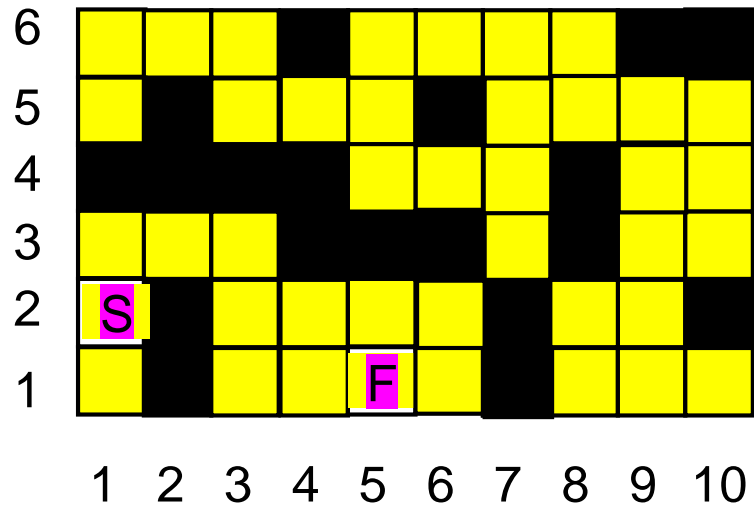
❖ Ένας αλγόριθμος αναζήτησης δεν μειώνει τον χώρο αναζήτησης (που είναι δεδομένος) αλλά καθορίζει τον αριθμό των καταστάσεων που επισκέπτεται.

Παράδειγμα 2.1

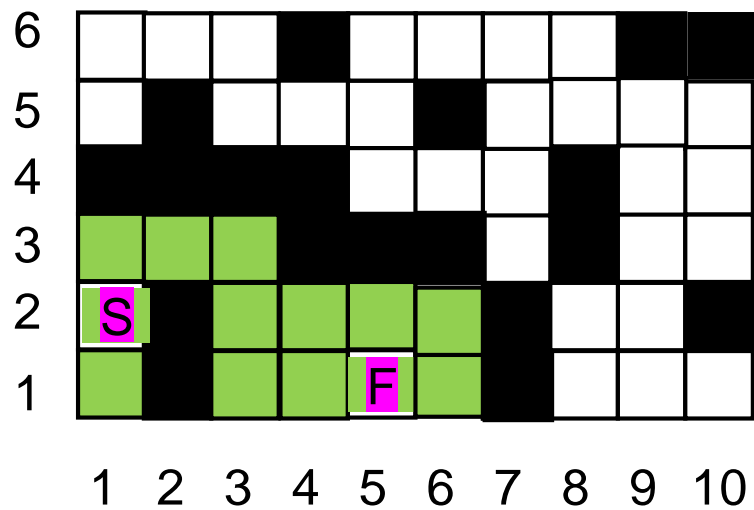
- ❖ Να ορίσετε τον χώρο καταστάσεων και τον χώρο αναζήτησης στον ακόλουθο λαβύρινθο



Χώρος καταστάσεων



Χώρος Αναζήτησης



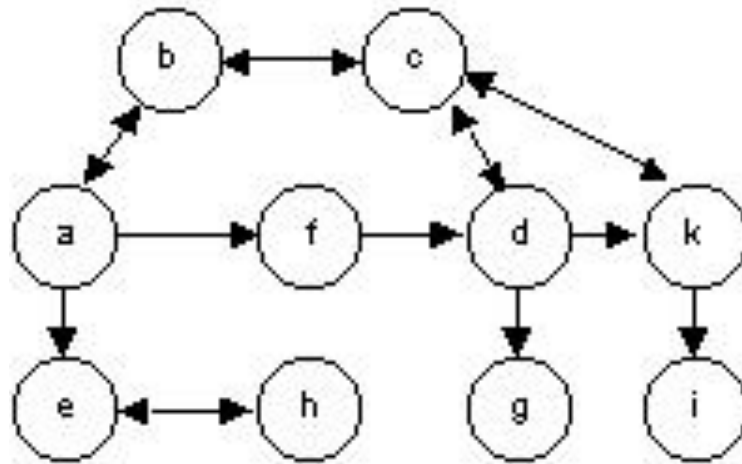
Χώρος Αναζήτησης ως Δένδρο Αναζήτησης (1/2)

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

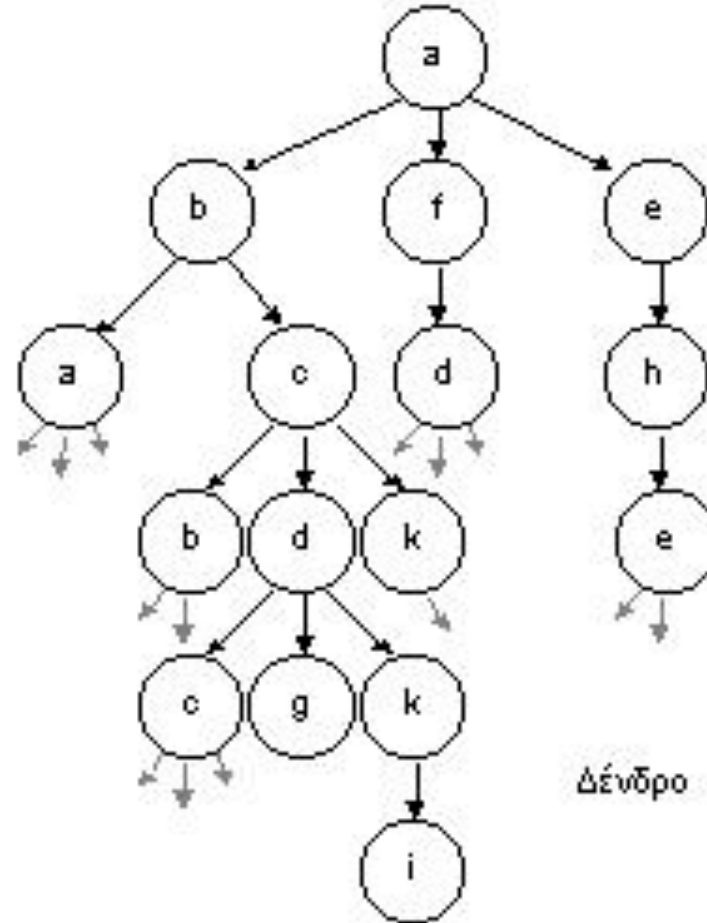
Τμήμα Δένδρου	Αναπαράσταση
Κόμβος (Node)	Κατάσταση
Ρίζα (Root)	Αρχική Κατάσταση
Φύλλο (Tip, Leaf) ή Τερματικός κόμβος	Τελική Κατάσταση ή Αδιέξοδο (Dead Node), δηλαδή κατάσταση στην οποία δεν μπορεί να εφαρμοστεί κανένας τελεστής μετάβασης.
Κλαδί (Branch)	Τελεστής Μετάβασης που μετατρέπει μια κατάσταση-Γονέα (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).

- ❖ Τα δένδρα είναι Η'-δένδρα (Or-trees) γιατί σε κάθε κόμβο έχουμε δυνατότητα επιλογής

Χώρος Αναζήτησης ως Δένδρο Αναζήτησης (2/2)



Γράφος

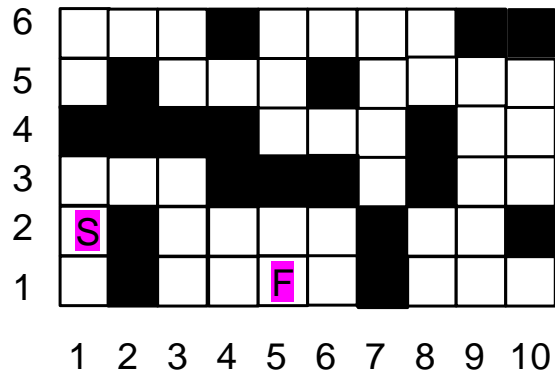


Δένδρο

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

Παράδειγμα 2.2

❖ Να αναπαραστήσετε τον χώρο αναζήτησης του παραδείγματος 2.1 με δένδρο



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 πληθάριθμος του V (ο αριθμός των καταστάσεων που περιέχει) και η σχέση του με το χώρο αναζήτησης SP ενός προβλήματος, 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
- ❖ Στα κριτήρια αυτά εντάσσεται και η έννοια του *κλαδέματος* ή *αποκοπής καταστάσεων* (*pruning*) του χώρου αναζήτησης.

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:

❖ Ο αλγόριθμος είναι γενικός με την έννοια ότι μπορεί να χρησιμοποιηθεί για να υλοποιήσει τους περισσότερους αλγορίθμους 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 ← ∅;
  Frontier ← <InitialState>;
  CurrentState ← First(Frontier);
  while CurrentState ≠ FinalState do
    Frontier ← delete(CurrentState, Frontier);
    if CurrentState ∉ ClosedSet then
      begin
        Next ← Expand(CurrentState);
        Frontier ← insert(Next, Frontier);
        Frontier ← prune(Frontier);
        Frontier ← reorder(Frontier);
        Closed ← Closed ∪ {CurrentState};
      end;
    if Frontier = ∅ then return failure;
    CurrentState ← First(Frontier);
  endwhile;
end.
```