

Επίλυση Προβλημάτων με Αναζήτηση

Πράκτορες επίλυσης προβλημάτων

- Βασισμένοι στην αρχιτεκτονική πράκτορα με στόχο.
- **Πρόβλημα προς επίλυση:** η επίτευξη μιας συγκεκριμένης κατάστασης περιβάλλοντος (**στόχος**).
- **Επίλυση:** η ακολουθία (ή οι ακολουθίες) ενεργειών που ανήκουν στο ρεπερτόριο του πράκτορα, τέτοιες ώστε αν εκτελεστούν επιτυχώς να προκαλούν μετάβαση του περιβάλλοντος στην κατάσταση-στόχο.
- Οι πράκτορες επίλυσης προβλημάτων **βρίσκουν** την ακολουθία ενεργειών που πρέπει να εκτελέσουν, προκειμένου το περιβάλλον να μεταβεί σε κάποια προκαθορισμένη επιθυμητή κατάσταση, την κατάσταση-στόχο, μέσω μιας διαδικασίας **αναζήτησης** στο χώρο καταστάσεων του περιβάλλοντος.

Ορισμός προβλήματος

- **Αρχική κατάσταση:** η κατάσταση του περιβάλλοντος από την οποία ξεκινά ο πράκτορας.
- **Περιγραφή των δυνατών ενεργειών:** για κάθε κατάσταση x ορίζεται μια συνάρτηση διαδόχων **successor_fn(x)** η οποία επιστρέφει ένα σύνολο διατεταγμένων ζευγών $\langle \text{ενέργεια}, \text{διάδοχος} \rangle$ όπου *ενέργεια* είναι μια ενέργεια επιτρεπτή στην κατάσταση x και *διάδοχος* είναι η κατάσταση που διαδέχεται την x αν σε αυτήν εκτελέσουμε την *ενέργεια*.
- **Έλεγχος στόχου:** προσδιορίζεται εάν μια δεδομένη κατάσταση είναι κατάσταση στόχου.
- **Συνάρτηση κόστους διαδρομής:** αποδίδει ένα αριθμητικό κόστος στην μετάβαση του περιβάλλοντος από κατάσταση σε κατάσταση (και τελικά στην μετάβαση του περιβάλλοντος από την αρχική κατάσταση στην κατάσταση στόχου).

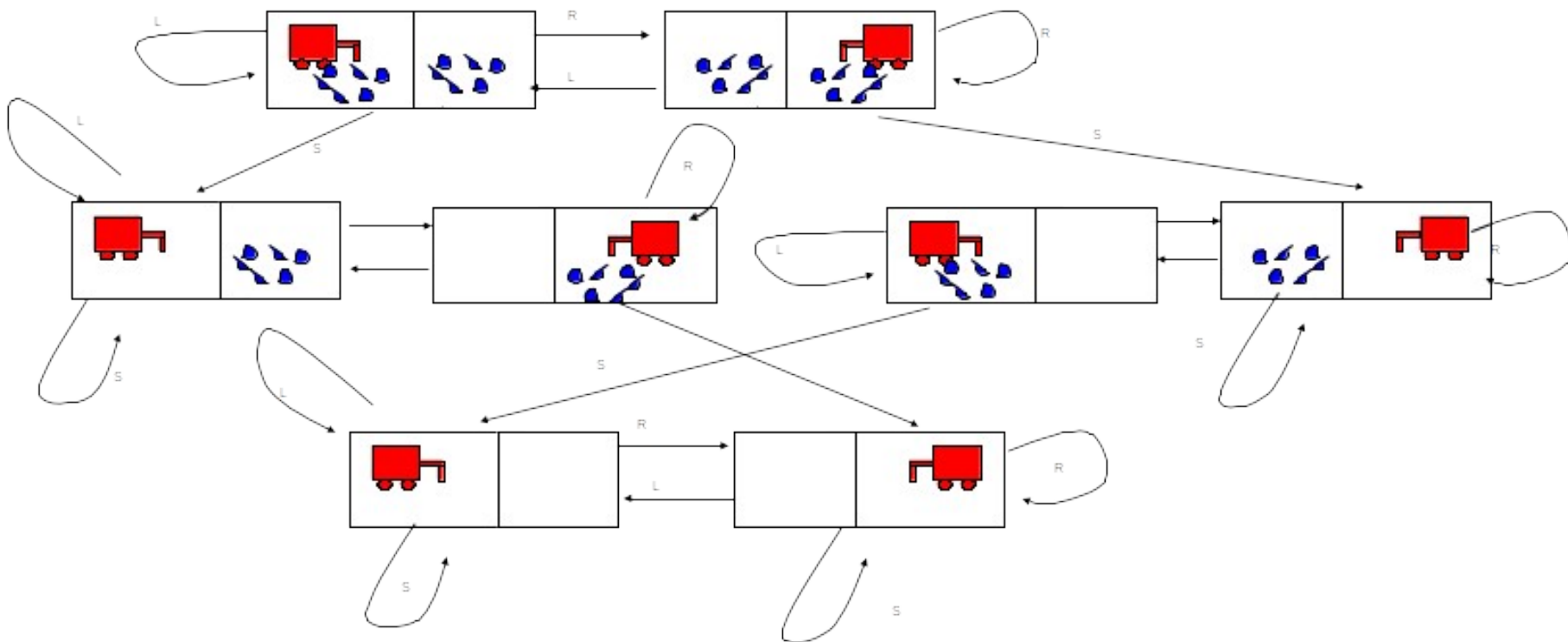
Χώρος καταστάσεων προβλήματος και λύση

- Ορίζεται έμμεσα από την αρχική κατάσταση και την περιγραφή των ενεργειών με τη χρήση της συνάρτησης διαδόχων.
- Είναι το σύνολο όλων των καταστάσεων που είναι προσπελάσιμες από την αρχική κατάσταση.
- Γράφημα (κατευθυνόμενο) όπου
 - Οι κόμβοι είναι καταστάσεις
 - Οι ακμές μεταξύ των κόμβων είναι ενέργειες.
- Λύση: μια διαδρομή στο χώρο καταστάσεων με αφετηρία την αρχική κατάσταση και κατάληξη την κατάσταση στόχο. Η ποιότητα της λύσης εξαρτάται από το κόστος της.

Παράδειγμα προβλήματος 1: ο κόσμος του πράκτορα καθαριστή

- **Καταστάσεις:** ο πράκτορας βρίσκεται σε έναν από δύο χώρους (A ή B), καθένας από τους οποίους έχει ή δεν έχει σκόνη. Οπότε 8 οι δυνατές καταστάσεις του περιβάλλοντος.
- **Αρχική κατάσταση:** μια από τις 8 της επιλογής μας.
- **Συνάρτηση διαδόχων:** για κάθε κατάσταση παράγει τις επιτρεπτές καταστάσεις που προκύπτουν αν εκτελεστεί κάθε μια από τις δυνατές ενέργειες του πράκτορα.
- **Έλεγχος στόχου:** είναι και οι δύο χώροι καθαροί;
- **Συνάρτηση κόστους:** έστω ότι το κόστος για κάθε ενέργεια του πράκτορα είναι το ίδιο, 1.

Χώρος καταστάσεων του πράκτορα-καθαριστή



Παράδειγμα προβλήματος 2: 8-puzzle

- Καταστάσεις: κάθε κατάσταση απεικονίζει τις θέσεις των 8 πλακιδίων και κενή είναι η θέση που αντιστοιχεί στο ένατο τετράγωνο.
- Αρχική: οποιαδήποτε της επιλογής μας από τις δυνατές καταστάσεις, π.χ.
- Συνάρτηση διαδόχων: παράγει τις επιτρεπτές καταστάσεις που προκύπτουν από 4 πιθανές ενέργειες: μετακίνηση της κενής θέσης Α(ριστερά), Δ(εξιά), Π(άνω), Κ(άτω).
- Έλεγχος στόχου: ταυτίζεται η κατάσταση με την κατάσταση στόχος, δεξιά;
- Συνάρτηση κόστους: έστω ίδιο κόστος για όλες τις ενέργειες, δηλαδή 1.

7	1	3
2		5
6	4	8

	2	5
1	7	4
6	3	8

Χώρος καταστάσεων για το 8-puzzle;

Άλλα παραδείγματα προβλημάτων που επιλύονται με αναζήτηση

- Εύρεση αεροπορικού δρομολογίου, όπου κάθε κατάσταση αναπαρίσταται από την τοποθεσία του αεροδρομίου και την τρέχουσα ώρα.
- Αυτόματη συναρμολόγηση αντικειμένων: το ζητούμενο είναι να βρεθεί η σειρά προτεραιότητας με την οποία συναρμολογούνται τα μέρη κάποιων αντικειμένων. Παρόμοιο πρόβλημα: σχεδίαση πρωτεϊνών.
- Αναζήτηση στο διαδίκτυο για απαντήσεις σε συγκεκριμένη ερώτηση.

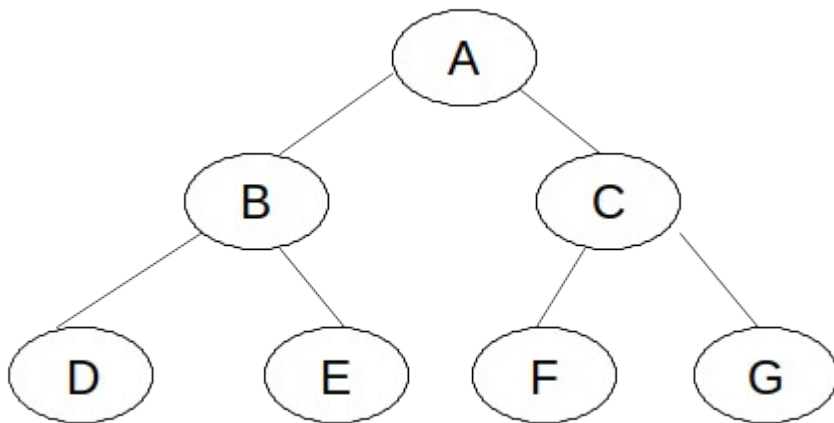
Είδη στρατηγικών αναζήτησης

- **Στρατηγικές τυφλής ή απληροφόρητης αναζήτησης:**
 - Η μόνη διαθέσιμη πληροφορία για το χώρο κατάστασης είναι αυτή που παρέχεται από τον ορισμό του προβλήματος.
 - Το περιβάλλον είναι πλήρως παρατηρήσιμο και ντετερμινιστικό.
 - Ο πράκτορας γνωρίζει ποιά είναι τα αποτελέσματα κάθε ενέργειας εκ των προτέρων και αυτά είναι εγγυημένα.
- **Στρατηγικές μερικώς πληροφορημένης αναζήτησης:**
 - Ο πράκτορας δεν διαθέτει πλήρη γνώση του χώρου κατάστασης.
 - Ο πράκτορας δεν διαθέτει πλήρη γνώση των αποτελεσμάτων των ενεργειών του.
 - Το περιβάλλον είναι καθόλου ή μόνο μερικώς παρατηρήσιμο.
- **Στρατηγικές ευρετικής ή πληροφορημένης αναζήτησης**
 - Αποδίδουν καλύτερα από τις στρατηγικές τυφλής αναζήτησης
 - Χρησιμοποιούν πρόσθετη πληροφορία, σχετική με το συγκεκριμένο πρόβλημα που επιλύεται, πέρα από αυτή που παρέχεται από τον ορισμό του προβλήματος.

Στρατηγικές τυφλής αναζήτησης

Στρατηγικές τυφλής αναζήτησης

- Οι στρατηγικές ουσιαστικά διαφέρουν ως προς τη χρονική σειρά με την οποία γίνεται επέκταση του συνόρου αναζήτησης.
- Υπάρχει ένας γενικός αλγόριθμος αναζήτησης, που εξειδικεύεται για κάθε στρατηγική της κατηγορίας αυτής.



- Αναζήτηση πρώτα κατά βάθος (BFS)
- Αναζήτηση πρώτα κατά πλάτος (DFS)
- Αναζήτηση με επαναληπτική εκβάθυνση ((ID)
- Αναζήτηση με επέκταση και οριοθέτηση (B&B)

Αναζήτηση Λύσεων -ορολογία

- **Λύση** = μια διαδρομή στο χώρο καταστάσεων που ξεκινά με την αρχική κατάσταση του προβλήματος και καταλήγει στην κατάσταση-στόχο.
- **Σύνορο αναζήτησης** = το διατεταγμένο σύνολο των καταστάσεων που έχουμε ήδη επισκεφθεί αλλά δεν έχουμε ακόμα επεκτείνει.
- **Κλειστό σύνολο** = το σύνολο των καταστάσεων που έχουμε ήδη επεκτείνει (απαιτείται για την αποφυγή βρόχων).
- **Επέκταση κατάστασης** = εφαρμογή της συνάρτησης διαδόχων, με αποτέλεσμα η κατάσταση που επεκτείνεται να εισάγεται στο κλειστό σύνολο και οι διάδοχες καταστάσεις να εισάγονται στο σύνορο αναζήτησης.

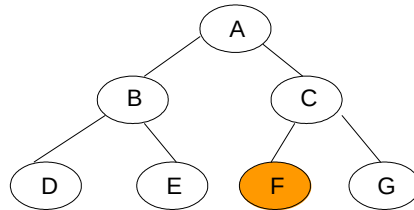
Αναζήτηση Λύσεων –βασικός αλγόριθμος

1. Βάλε την αρχική κατάσταση στο σύνορο αναζήτησης και θέσε κλειστό σύνολο = $\{\}$
2. Αν το σύνορο αναζήτησης είναι άδειο τότε σταμάτησε
3. Πάρε την πρώτη κατά σειρά κατάσταση του συνόρου αναζήτησης
4. Αν η κατάσταση αυτή ανήκει στο κλειστό σύνολο, τότε πήγαινε στο 2
5. Αν η κατάσταση αυτή είναι η τελική τότε δώσε τη λύση και πήγαινε στο 2
6. Κάνε επέκταση της κατάστασης για να πάρεις τις διάδοχες
7. Βάλε τις διάδοχες καταστάσεις στο σύνορο αναζήτησης
8. Κλάδεψε τις καταστάσεις που δεν χρειάζονται (σύμφωνα με κάποιο κριτήριο)
9. Κάνε αναδιάταξη στο σύνορο αναζήτησης (σύμφωνα με κάποιο κριτήριο)
10. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο και πήγαινε στο 2

Αναζήτηση πρώτα κατά πλάτος

1. Βάλε την αρχική κατάσταση στο σύνολο αναζήτησης και θέσε κλειστό σύνολο = $\{\}$
2. Αν το σύνολο αναζήτησης είναι άδειο τότε σταμάτησε
3. Πάρε την πρώτη κατά σειρά κατάσταση του συνόλου αναζήτησης
4. Αν η κατάσταση αυτή ανήκει στο κλειστό σύνολο, τότε πήγαινε στο 2
5. Αν η κατάσταση αυτή είναι η τελική τότε δώσε τη λύση και σταμάτησε. Αλλιώς αν θέλεις κι άλλες λύσεις πήγαινε στο 2
6. Κάνε επέκταση της κατάστασης για να πάρεις τις διάδοχες
7. Βάλε τις διάδοχες καταστάσεις στο σύνολο αναζήτησης στο τέλος
8. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
9. Πήγαινε στο 2

Αναζήτηση πρώτα κατά πλάτος-εφαρμογή



Σύνολο αναζήτησης	Κλειστό σύνολο	Κατάσταση	Διάδοχος
<A>	{}	A	<B, C>
<B, C>	{A}	B	<D, E>
<C, D, E>	{A, B}	C	<F, G>
<D, E, F, G>	{A, B, C}	D	<>
<E, F, G>	{A, B, C, D}	E	<>
<F, G>	{A, B, C, D, E}	F	ΤΕΛΙΚΗ

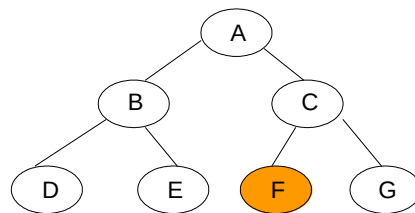
Αναζήτηση πρώτα κατά πλάτος-αξιολόγηση

- Το σύνορο αναζήτησης αντιστοιχεί σε **ουρά**.
- Βρίσκει πάντα τη μικρότερη λύση, δηλαδή αυτή με το μικρότερο αριθμό ακμών.
- Η λύση που βρίσκει είναι βέλτιστη αν όλες οι ακμές έχουν ίδιο κόστος.
- Είναι πλήρης στρατηγική: αν υπάρχει λύση τότε σίγουρα θα τη βρεί.
- Πρόβλημα: όταν το πλήθος των ενεργειών που μπορεί να εκτελεστούν σε μια κατάσταση είναι άπειρο, τότε το δέντρο αναζήτησης έχει άπειρο πλάτος.
- Απαιτεί πολλή μνήμη για να διατηρεί το σύνορο αναζήτησης το οποίο αυξάνεται εκθετικά με το βάθος της αναζήτησης.

Αναζήτηση πρώτα κατά βάθος

1. Βάλε την αρχική κατάσταση στο σύνολο αναζήτησης και θέσε κλειστό σύνολο = $\{\}$
2. Αν το σύνολο αναζήτησης είναι άδειο τότε σταμάτησε
3. Πάρε την πρώτη κατά σειρά κατάσταση του συνόλου αναζήτησης
4. Αν η κατάσταση αυτή ανήκει στο κλειστό σύνολο, τότε πήγαινε στο 2
5. Αν η κατάσταση αυτή είναι η τελική τότε δώσε τη λύση και σταμάτησε, αλλιώς αν θέλεις κι άλλες λύσεις πήγαινε στο 2
6. Κάνε επέκταση της κατάστασης για να πάρεις τις διάδοχες
7. Βάλε τις διάδοχες καταστάσεις στο σύνολο αναζήτησης στην αρχή
8. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο
9. Πήγαινε στο 2

Αναζήτηση πρώτα κατά βάθος -εφαρμογή



Σύνολο αναζήτησης	Κλειστό σύνολο	Κατάσταση	Διάδοχες
<A>	{}	A	<B, C>
<B, C>	{A}	B	<D, E>
<D, E, C>	{A, B}	D	<>
<E, C>	{A, B, D}	E	<>
<C>	{A, B, D, E}	C	<F, G>
<F, G>	{A, B, D, E, C}	F	ΤΕΛΙΚΗ

Αναζήτηση πρώτα κατά βάθος -αξιολόγηση

- Το σύνορο αναζήτησης αντιστοιχεί σε **στοίβα**.
- Έχει μικρές απαιτήσεις σε μνήμη γιατί το σύνορο αναζήτησης δεν μεγαλώνει πολύ.
- Δεν εγγυάται ότι η πρώτη λύση που θα βρεί είναι και βέλτιστη (μονοπάτι με το μικρότερο μήκος ή κόστος).
- Αν δεν υπάρχει έλεγχος βρόχων ή ο χώρος καταστάσεων έχει άπειρο βάθος τότε μπορεί να πέσει σε ατέρμονα κλαδιά ή κλαδιά μεγάλου μήκους.
- Θεωρείται μη-πλήρης γιατί υπάρχει περίπτωση να μην βρεί μια λύση αν και μπορεί να περάσει πολύ κοντά της.

Αναζήτηση με επαναληπτική εκβάθυνση

- Συνδυάζει αναζήτηση κατά βάθος με αναζήτηση κατά πλάτος.

1. Όρισε το αρχικό βάθος αναζήτησης (συνήθως 1)
2. Κάνε αναζήτηση κατά βάθος μέχρι αυτό το βάθος αναζήτησης.
3. Αν έχεις βρεί λύση σταμάτησε.
4. Αύξησε το βάθος της αναζήτησης (συνήθως κατά 1)
5. Πήγαινε στο 2.

- Αν το βάθος αυξάνεται κατά 1 σε κάθε επανάληψη τότε η λύση που βρίσκεται είναι βέλτιστη (γιατί αν υπήρχε καλύτερη θα είχε βρεθεί στην προηγούμενη επανάληψη).
- Μειονέκτημα: όταν αρχίζει αναζήτηση με διαφορετικό βάθος δεν διατηρεί πληροφορίες από την προηγούμενη αναζήτηση.

Αναζήτηση με επέκταση και οριοθέτηση (1)

- Εφαρμόζεται στα προβλήματα όπου αναζητείται βέλτιστη λύση, δηλαδή λύση με το ελάχιστο κόστος. Στα προβλήματα αυτά θεωρείται ότι οι ενέργειες που εκτελούνται σε κάθε κατάσταση έχουν διαφορετικό κόστος.
- Βασίζεται στο κλάδεμα καταστάσεων με σκοπό την ελάττωση του χώρου αναζήτησης.

Αναζήτηση με επέκταση και οριοθέτηση (2)

1. Βάλε την αρχική κατάσταση στο σύνολο αναζήτησης και θέσε κλειστό σύνολο $=\{\}$
2. Αρχική τιμή της καλύτερης λύσης είναι το $+\infty$ (όριο)
3. Αν το σύνολο αναζήτησης είναι κενό σταμάτησε. Η καλύτερη μέχρι τώρα λύση είναι και η βέλτιστη
4. βγάλε την πρώτη κατά σειρά κατάσταση από το σύνολο αναζήτησης
5. Αν η κατάσταση ανήκει στο κλειστό σύνολο τότε πήγαινε στο 3
6. Αν η κατάσταση είναι τελική τότε ανανέωσε τη λύση ως την καλύτερη μέχρι τώρα και ανανέωσε την τιμή του ορίου με την τιμή που αντιστοιχεί στην τελική κατάσταση. Πήγαινε στο 3.
7. Εφάρμοσε τη συνάρτηση διαδόχων για να παράγεις διάδοχες καταστάσεις και την τιμή που αντιστοιχεί σε αυτές
8. Βάλε τις διάδοχες καταστάσεις των οποίων η τιμή δεν υπερβαίνει το όριο μπροστά στο σύνολο αναζήτησης
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο
10. Πήγαινε στο 3.

Αναζήτηση με επέκταση και οριοθέτηση -εφαρμογή

- Εφαρμόστε τη στρατηγική αυτή για το χώρο καταστάσεων του πράκτορα-καθαριστή.
- Υποθέστε ότι κάθε μετακίνηση αριστερά-δεξιά κοστίζει 2 και κάθε αναρρόφηση κοστίζει 1.
- Θεωρείστε αρχική κατάσταση αυτή στην οποία είναι σκονισμένοι και οι δύο χώροι και ο πράκτορας βρίσκεται στον A.
- Θεωρείστε τελική κατάσταση αυτή στην οποία και οι δύο χώροι είναι καθαροί και ο πράκτορας βρίσκεται στον B.