<u>ΑΣΚΗΣΕΙΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ</u>

Αναζήτηση και Προβλήματα Ικανοποίησης Περιορισμών

- Υποθέστε έναν χώρο αναζήτησης όπου η αρχική κατάσταση είναι το νούμερο 1, και η συνάρτηση μετάβασης για μια κατάσταση n επιστρέφει τρεις καταστάσεις, την 2n, την 2n+1, και την 2n+2. Υποθέτοντας ότι η κατάσταση στόχου είναι το 11, δείξτε τη σειρά με την οποία γίνεται επίσκεψη των κόμβων του δέντρου αναζήτησης από τους αλγόριθμους: **a)** (0.5) αναζήτηση κατά πλάτος, **b)** (1) αναζήτηση επαναληπτικής εκβάθυνσης.

- Υποθέστε ότι έχετε την παρακάτω συνάρτηση κόστους για ευριστική αναζήτηση:

  f(node) = Wgt × g(node) + (1 – Wgt) × h(node)   όπου $0 \le Wgt \le 1$

Ποιόν αλγόριθμο αναζήτησης παίρνουμε με Wgt = 0; Ποιόν αλγόριθμο αναζήτησης παίρνουμε με Wgt = 1;

- Στο πρόβλημα του πλανόδιου πωλητή (traveling salesperson problem) πρέπει ένας πράκτορας ξεκινώντας από μια συγκεκριμένη πόλη να επισκεφτεί όλες τις πόλεις σε ένα σύνολο πόλεων, ακριβώς μια φορά την καθεμία, και στο τέλος να επιστρέψει στην αρχική πόλη έχοντας καλύψει τη μικρότερη δυνατή απόσταση. Θέλουμε να λύσουμε ένα τέτοιο πρόβλημα χρησιμοποιώντας τον αλγόριθμο Α*. Υποθέστε ότι ο πράκτορας βρίσκεται σε μια πόλη n και απομένουν Μ πόλεις για επίσκεψη. Για καθένα από τα παρακάτω heuristics υπολογισμού της απόστασης από την πόλη n ως τη λύση, προσδιορίστε αν είναι αποδεκτό ή όχι και εξηγήστε την απάντηση σας.
  - h(n) = (M) x (τη μέγιστη απόσταση μεταξύ δύο πόλεων στις οποίες δεν έχει γίνει ακόμα επίσκεψη)
  - h(n) = (M) x (τη μέση απόσταση μεταξύ όλων των ζευγαριών πόλεων στις οποίες δεν έχει γίνει ακόμα επίσκεψη)
  - h(n) = (M) x (την ελάχιστη απόσταση μεταξύ δύο πόλεων στις οποίες δεν έχει γίνει ακόμα επίσκεψη)

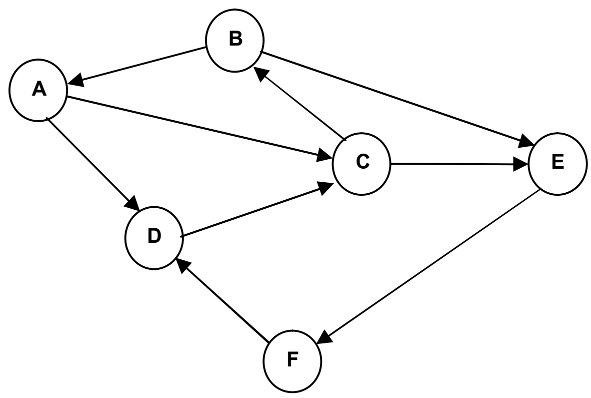
- Έστω ένα πρόβλημα ικανοποίησης περιορισμών με 5 μεταβλητές x1,x2,x3,x4,x5 και τους περιορισμούς x1≤x2, x2≤x3, x3≤x5, x4>x5, |x4-x1|>1, x2+x5>1. Όλες οι μεταβλητές έχουν πεδίο τιμών {0,1,2}. Εφαρμόστε συνέπεια τόξου (arc consistency) σε αυτό το πρόβλημα. Για κάθε διαγραφή τιμής, εξηγήστε το λόγο για τη διαγραφή. Ποιά είναι τα πεδία τιμών των μεταβλητών στο τέλος της εφαρμογής της συνέπειας τόξου?

- Απαντήστε με ναι ή όχι στο παρακάτω ερώτημα κι εξηγήστε την απάντηση σας: Η επίλυση ενός προβλήματος ικανοποίησης περιορισμών με τον αλγόριθμο Α* είναι καλή ιδέα;

Ένας πράκτορας βρίσκεται στην πόλη Α του διπλανού σχήματος και θέλει να βρει μονοπάτι για την πόλη F χρησιμοποιώντας τον αλγόριθμο αναρρίχησης λόφων (hill climbing). Δώστε κατάλληλες τιμές της ευριστικής συνάρτησης για κάθε πόλη ώστε:

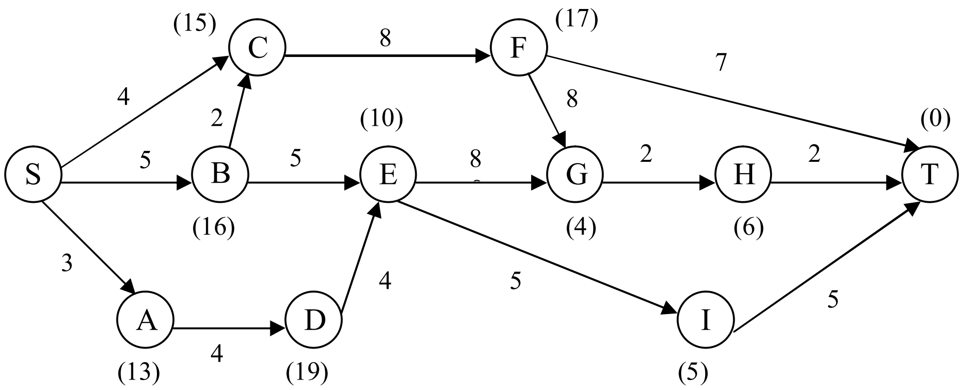
A) Ο αλγόριθμος να “κολλήσει” σε τοπικό βέλτιστο.

B) Ο αλγόριθμος να βρει λύση.



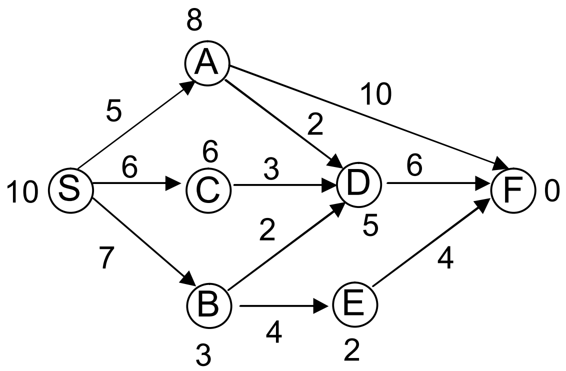
- Αναπαραστήστε τα παρακάτω ως προβλήματα αναζήτησης. Δηλ. προσδιορίστε την αρχική κατάσταση, την κατάσταση στόχου, τις ενέργειες μετάβασης μεταξύ των καταστάσεων, κι αν είναι απαραίτητο το κόστος των ενεργειών. **A)** Ένας πράκτορας διαθέτει έναν τηλεφωνικό κατάλογο και θέλει να βρει το νούμερο τηλεφώνου ενός συγκεκριμένου ατόμου. Ο πράκτορας μπορεί να γυρίζει σελίδες, να ψάχνει σε μια σελίδα για ένα όνομα, και γνωρίζει το αλφάβητο. **B)** Το ίδιο πρόβλημα με τη διαφορά ότι ο πράκτορας ψάχνει το όνομα ενός άγνωστου ατόμου γνωρίζοντας το τηλέφωνο του.

- Στο παρακάτω διάγραμμα οι κύκλοι αναπαριστούν πόλεις και οι κατευθυνόμενες ακμές δρόμους μονής κατεύθυνσης. Η αφετηρία ενός πράκτορα είναι η πόλη S και ο προορισμός η T. Ο αριθμός που συνοδεύει την κάθε ακμή δηλώνει το μήκος του δρόμου σε χιλιόμετρα. Ο αριθμός σε παρένθεση δίπλα σε κάθε κόμβο αποτελεί την τιμή της ευριστικής συνάρτησης για τον συγκεκριμένο κόμβο (δηλ. την εκτιμούμενη απόσταση ως την πόλη T). Προσδιορίστε το μονοπάτι θα επιλεχθεί ως λύση, την συνολική απόσταση της λύσης, και το δέντρο αναζήτησης που κατασκευάζεται για **A)** τον αλγόριθμο αναζήτησης ενιαίου κόστους (UCS), και **B)** τον A*. Είναι το heuristic που χρησιμοποιείται αποδεκτό?



- Απαντήστε στο εξής ερώτημα εξηγώντας με σαφήνεια την απάντηση σας. Η αναζήτηση κατά πλάτος και η αναζήτηση οριοθετημένου βάθους με όριο x επεκτείνουν το ίδιο πλήθος κόμβων όταν ο στόχος βρίσκεται στον τέρμα δεξιά κόμβο του δέντρου αναζήτησης στο επίπεδο x?

- Έχετε τον παρακάτω χώρο αναζήτησης, όπου S είναι ο αρχικός κόμβος και F ο κόμβος στόχου. Σε κάθε ακμή αναγράφεται το κόστος διάσχισης της. Δίπλα σε κάθε κόμβο αναγράφεται το εκτιμούμενο κόστος μονοπατιού από τον συγκεκριμένο κόμβο ως τον στόχο (δηλ. η τιμή της ευριστικής συνάρτησης). Για κάθε μια από τις παρακάτω στρατηγικές αναζήτησης, σχεδιάστε επακριβώς το δέντρο αναζήτησης που δημιουργεί για να λύσει το πρόβλημα. Προσδιορίστε και τη σειρά με την οποία επεκτείνουν κόμβους μέχρι να τερματίσουν. Σε περιπτώσεις «ισοπαλιών» οι κόμβοι επεκτείνονται τυχαία (δηλ. όπως θέλετε εσείς). **A)** (1.5) A*, **B)** (0.5) Απλή αναρρίχηση λόφων (hill climbing).



- Κατασκευάστε ένα πρόβλημα ικανοποίησης περιορισμών, δηλ. προσδιορίστε τις μεταβλητές, τα πεδία τιμών τους, και τους περιορισμούς, ώστε να ισχύει το εξής: Το πρόβλημα αυτό επιλύεται από τον αλγόριθμο MAC χωρίς καθόλου αναζήτηση (δηλ. στο δέντρο αναζήτησης υπάρχει μόνο η ρίζα) ενώ ο αλγόριθμος χρονολογικής οπισθοδρόμησης (BT) κάνει τουλάχιστον 2 οπισθοδρομήσεις. Για τον BT μπορείτε να χρησιμοποιήσετε οποιαδήποτε σειρά επίσκεψης των μεταβλητών και δοκιμής των τιμών σας βολεύει.

- Έχετε κατασκευάσει έναν πράκτορα που επιλύει σταυρόλεξα. Χαρακτηρίστε το περιβάλλον όπου λειτουργεί. Δηλ. είναι πλήρως ή μερικώς παρατηρήσιμο? Αιτιοκρατικό ή στοχαστικό? Επεισοδιακό ή ακολουθιακό? Στατικό ή δυναμικό? Διακριτό ή συνεχές? Εξηγήστε συνοπτικά. **B)** (1) Αναφέρετε έναν αλγόριθμο αναζήτησης που είναι πλήρης κι έχει εκθετική χρονική πολυπλοκότητα κι έναν άλλο που είναι πλήρης κι έχει πολυωνυμική χρονική πολυπλοκότητα. **Γ)** (0.5) Γιατί η επίλυση ενός προβλήματος ικανοποίησης περιορισμών με αναζήτηση κατά πλάτος δεν είναι καλή ιδέα?

- Υποθέτουμε ότι έχουμε ένα παζλ με την ακόλουθη αρχική κατάσταση: |Μ|Μ|Μ|Λ|Λ|Λ|Κ| όπου υπάρχουν τρία μάυρα κομμάτια (Μ), τρία λευκά κομμάτια (Λ) και μία κενή θέση (Κ). Επιτρέπονται οι ακόλουθες κινήσεις: Ένα κομμάτι μπορεί να μετακινηθεί στην κενή θέση εφόσον είναι δίπλα του. Η κίνηση αυτή έχει κόστος μία μονάδα. Ένα κομμάτι μπορεί να μετακινηθεί στην κενή θέση πηδώντας πάνω από το πολύ δύο άλλα κομμάτια. Η κίνηση αυτή έχει κόστος ίσο με τον αριθμό των κομματιών πάνω από τα οποία πήδηξε το κομμάτι. Για παράδειγμα, το κόστος της μετακίνησης του δεύτερου ή του τρίτου Λ στη θέση του κενού είναι 1 ενώ το κόστος της μετακίνησης του πρώτου Λ στη θέση του κενού είναι 2. Ο στόχος του παζλ είναι να βρεθούν όλα τα λευκά κομμάτια αριστερά όλων των μαύρων κομματιών (η κενή θέση δεν παίζει ρόλο). **Α)** (1.5) Προτείνετε δύο ευρετικές συναρτήσεις αξιολόγησης (heuristics) για αυτό το πρόβλημα εκ των οποίων η μια τουλάχιστον να είναι αποδεκτή. Για κάθε ένα από τα heuristics εξηγήστε γιατί είναι αποδεκτό, αν είναι, ή γιατί δεν είναι αποδεκτό, αν δεν είναι. **Β)** (1.5) Κατασκευάστε το δέντρο αναζήτησης που παράγεται από τον αλγόριθμο αναζήτησης Α* μέχρι να επεκταθούν οι 6 πρώτοι κόμβοι χρησιμοποιώντας το αποδεκτό heuristic σας. Αν είναι και τα δύο αποδεκτά χρησιμοποιήστε το ισχυρότερο.

- Θεωρήστε ένα πρόβλημα ικανοποίησης περιορισμών με 5 μεταβλητές x1,x2,x3,x4,x5 όπου κάθε μια έχει πεδίο τιμών {0,1} και τους περιορισμούς $x1 \neq x2$, $x1 \neq x3$, $x2 \neq x4$, $x2 \neq x5$, $x3 \neq x5$, $x4 \neq x5$. **Α)** (1.5) Δείξτε το δέντρο αναζήτησης που εξερευνά ο αλγόριθμος χρονολογικής οπισθοδρόμισης (simple backtracking) μέχρι να αποδείξει ότι δεν υπάρχει λύση. Υποθέστε ότι η σειρά επίσκεψης των μεταβλητών είναι x1,x2,x3,x4,x5 και η σειρά επιλογής των τιμών είναι 0,1 για όλες τις μεταβλητές. Κάντε το ίδιο για τον αλγόριθμο forward checking. **Β)** (0.5) Προτείνετε μια σειρά επίσκεψης των μεταβλητών που να οδηγεί σε δέντρο αναζήτησης ελάχιστου μεγέθους για τον forward checking. Δείξτε αυτό το δέντρο. **Γ)** (0.5) Υπάρχει κάποια σειρά επιλογής τιμών που να είναι η προτιμότερη? Εξηγήστε.

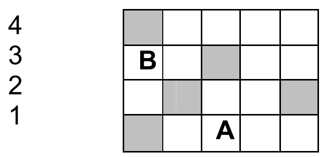
- Σε ένα πρόβλημα 8-puzzle στόχος είναι να μετασχηματιστεί η δοθείσα αρχική κατάσταση στην τελική όπου το κενό βρίσκεται στο κέντρο και οι αριθμοί, ξεκινώντας από το πάνω αριστερά τετράγωνο, είναι τοποθετημένοι σε διάταξη σύμφωνα με τη φορά των δεικτών του ρολογιού. Οι κινήσεις που επιτρέπονται είναι οι μετακινήσεις του κενού πάνω, κάτω, αριστερά και δεξιά, και οι ταυτόχρονες μετακινήσεις των αντίστοιχων αριθμών στη θέση του κενού. Για την παρακάτω αρχική κατάσταση ενός 8-puzzle δείξτε: **Α)** (2) το δέντρο αναζήτησης που προκύπτει αφού ο αλγόριθμος αναζήτησης πρώτα στο καλύτερο (best-first search) έχει επεκτείνει τους 5 πρώτους κόμβους. Για την κατάσταση που αντιστοιχεί σε κάθε κόμβο δείξτε και την τιμή της ευρετικής συνάρτησης. **Β)** (1) τη σειρά των κόμβων που επεκτείνει ο αλγόριθμος αναρίχησης λόφων (hill-climbing) μέχρι τον τερματισμό του. Η ευρετική που χρησιμοποιούν και οι δύο αλγόριθμοι για να εκτιμήσουν τις καταστάσεις είναι το άθροισμα των αποστάσεων Manhattan όλων των αριθμών από τις σωστές τους θέσεις. Σε περίπτωση ισοβαθμίας δύο ή και περισσοτέρων καταστάσεων, επιλέγεται αυτή στην οποία όσο το δυνατόν περισσότεροι αριθμοί είναι στη σωστή τους θέση. Σε περιπτώση περαιτέρω ισοβαθμίας επιλέγεται μια κατάσταση στην τύχη. Υποθέστε ότι οι αλγόριθμοι ποτέ δεν επισκέπτονται καταστάσεις που έχουν ήδη επισκεφτεί.

4	7	2
3	8	1
6	5	

- Έστω ένα πρόβλημα ικανοποίησης περιορισμών με 5 μεταβλητές x1,x2,x3,x4,x5 και τους περιορισμούς $x1 \leq x2$, $x1 = x4$, $x2 \leq x3$, $x3 \leq x5$, $x4 > x5$. Όλες οι μεταβλητές έχουν πεδίο τιμών {0,1,2}. Εφαρμόστε συνέπεια τόξου (arc consistency) σε αυτό το πρόβλημα. Για κάθε διαγραφή τιμής, εξηγήστε το λόγο για τη διαγραφή. Ποιά είναι τα πεδία τιμών των μεταβλητών στο τέλος της εφαρμογής της συνέπειας τόξου?

- Έχετε κατασκευάσει έναν πράκτορα που λειτουργεί ως βοηθός απόδειξης θεωρημάτων. Χαρακτηρίστε το περιβάλλον όπου λειτουργεί. Δηλ. είναι πλήρως ή μερικώς παρατηρήσιμο? Αιτιοκρατικό ή στοχαστικό? Επεισοδιακό ή ακολουθιακό? Στατικό ή δυναμικό? Διακριτό ή συνεχές? Δικαιολογήστε συνοπτικά κάθε απάντηση σας.

- Ένα ρομπότ αναζητεί μια διαδρομή από το σημείο Α στο σημείο Β. Ο χώρος αναζήτησης αναπαριστάται με ένα πλέγμα 5x4 όπου τα εμπόδια φαίνονται σκιασμένα. Οι κινήσεις που μπορεί να κάνει το ρομπότ (με τη σειρά που τις εξετάζει) είναι κάτω, δεξιά, πάνω και αριστερά. Ποιοί από τους παρακάτω αλγόριθμους μπορούν να βρουν λύση; Α) Αναζήτηση Πρώτα σε Βάθος (DFS), Β) Αναζήτηση Αναρρίχησης Λόφων (HC), Γ) Αναζήτηση Α*. Για τους δύο τελευταίους αλγόριθμους θεωρήστε ως ευρετική συνάρτηση την απόσταση Manhattan. Για τον DFS γράψτε τους 7 πρώτους κόμβους που επισκέπτεται κατά τη διάρκεια της αναζήτησης (δηλ. γράψτε τη σειρά συντεταγμένων). Για τους HC και Α* γράψτε όλη την ακολουθία κόμβων που επισκέπτονται μέχρι να τερματίσουν. Θεωρείστε ότι υπάρχει έλεγχος αποφυγής ατέρμονων βρόγχων.



1 2 3 4 5

- Έχετε ένα πρόβλημα ικανοποίησης περιορισμών με τις εξής 5 μεταβλητές x1, x2, x3, x4, x5. Τα πεδία ορισμού των μεταβλητών είναι τα εξής: D(x1)={0,1,2,3,4,5}, D(x2)={2,3,4,5}, D(x3)={0,1}, D(x4)={0,1,2,3,4}, D(x1)={0,1,2}. Υπάρχουν και κάποιοι περιορισμοί τους οποίους δε γνωρίζετε.
  Ποια σειρά επιλογής των μεταβλητών από έναν αλγόριθμο οπισθοδρόμησης θα προτείνατε και γιατί?
  Αν γνωρίζατε τους περιορισμούς υπάρχει περίπτωση να άλλαζε η προτεινόμενη σειρά? Εξηγήστε την απάντηση σας.

- Κατασκευάστε παραδείγματα προβλημάτων αναζήτησης όπου να φαίνονται τα εξής: **Α)** Η αναζήτηση κατά πλάτος (BFS) είναι καλύτερη (δηλ. επισκέπτεται λιγότερους κόμβους στο δέντρο αναζήτησης) από την αναζήτηση επαναληπτικής εκβάθυνσης (IDS), **Β)** Το αντίστροφο, **Γ)** Η αναρρίχηση λόφων (hill climbing) είναι καλύτερη από την αναζήτηση ενιαίου κόστους (UCS), **Δ)** Το αντίστροφο. **Ε)** Κατασκευάστε κι ένα παράδειγμα προβλήματος αναζήτησης όπου δεν είναι δυνατό κάποιος αλγόριθμος να είναι καλύτερος από την αναζήτηση κατά βάθος (DFS). Για κάθε παράδειγμα εξηγήστε γιατί ο ένας αλγόριθμος είναι καλύτερος από τον άλλο.

- Εξηγήστε γιατί ισχύει το καθένα από τα παρακάτω:
  - Η αναζήτηση κατά πλάτος (BFS) είναι ειδική περίπτωση της αναζήτησης ενιαίου κόστους (UCS)
  - Η αναζήτηση κατά βάθος (DFS), η αναζήτηση κατά πλάτος, και η αναζήτηση ενιαίου κόστους είναι ειδικές περιπτώσεις της αναζήτησης πρώτα στο καλύτερο (best-first search)
  - Η αναζήτηση ενιαίου κόστους είναι ειδική περίπτωση του αλγόριθμου A*

- Στο πρόβλημα κατασκευής (όχι επίλυσης) σταυρολέξων πρέπει να μπουν λέξεις σε ένα πλαίσιο. Στο πλαίσιο, που δίνεται ως μέρος του προβλήματος, υπάρχουν κενά και μαύρα τετράγωνα. Υποθέστε ότι έχετε στη διάθεση σας ένα σύνολο λέξεων (δηλ. ένα λεξικό) και ο στόχος είναι να γεμίσετε τα κενά τετράγωνα χρησιμοποιώντας λέξεις από το λεξικό. Κατασκευάστε μοντελοποίηση του προβλήματος ως 1) δυαδικού προβλήματος ικανοποίησης περιορισμών, 2) μη-δυαδικού (n-αδικού) προβλήματος ικανοποίησης περιορισμών. Δηλαδή σε κάθε μια περίπτωση προσδιορίστε ποιες είναι οι μεταβλητές, ποια είναι τα πεδία ορισμού τους, και ποιοι είναι οι περιορισμοί.

- Έχετε ένα πρόβλημα ικανοποίησης περιορισμών με 4 μεταβλητές x1,x2,x3,x4 και τους περιορισμούς x1=x2, x1≠x3, x1≥x4, x3=x4. Σχεδιάστε το δέντρο αναζήτησης που εξερευνά ο καθένας από του παρακάτω αλγόριθμους μέχρι να τερματίσει.
  - Χρονολογική οπισθοδρόμηση (chronological backtracking)
  - Έλεγχος προς εμπρός (forward checking)
  - Διατήρηση συνέπειας τόξου (MAC)

Λογική και Σχεδιασμός Ενεργειών

- Θεωρήστε ένα λεξιλόγιο με μόνο τέσσερις ατομικές προτάσεις, A, B, C, D. Για κάθε μια από τις παρακάτω προτάσεις γράψτε όλα τα μοντέλα που έχει: **1)** $(A \Rightarrow B) \Rightarrow C$, **2)** $(A \Leftrightarrow B) \vee (\neg C \wedge D)$

- Αποφανθείτε αν κάθε μια από τις παρακάτω προτάσεις είναι έγκυρη, μη ικανοποιήσιμη, ή τίποτα από τα δύο. Εξηγήστε την απάντηση σας. **1)** $(A \Rightarrow B) \Rightarrow ((A \wedge C) \Rightarrow B)$, **2)** $A \vee B \vee (A \Rightarrow B)$

- Θεωρήστε τις εξής προτάσεις προτασιακής λογικής σε CNF: $(A \vee \neg C \vee D)$, $(A \vee B)$, $(\neg B \vee C)$, $(\neg A)$. Δείξτε τη σειρά εφαρμογών του κανόνα μοναδιαίας ανάλυσης (unit resolution) ώστε από αυτό το σύνολο προτάσεων να προκύψει η πρόταση D.

- Θεωρήστε το παρακάτω πρόβλημα σχεδιασμού ενεργειών. Μερικά κιβώτια βρίσκονται σε μια αποθήκη και θέλουμε να μεταφερθούν σε ένα λιμάνι χρησιμοποιώντας ένα φορτηγό. Για να γίνει αυτό πρέπει τα κιβώτια να φορτωθούν στο φορτηγό, το οποίο πρέπει έπειτα να μεταβεί από την αποθήκη στο λιμάνι, όπου τέλος τα κιβώτια πρέπει να εκφορτωθούν. Για να φορτωθεί ένα κιβώτιο πρέπει να βρίσκεται δίπλα στο φορτηγό, να λειτουργεί το μηχάνημα φόρτωσης και να υπάρχει ο απαραίτητος χώρος στο φορτηγό. Το φορτηγό μπορεί να μεταβεί από την αποθήκη στο λιμάνι μόνο αν είναι γεμάτο. Για να εκφορτωθεί ένα κιβώτιο πρέπει να είναι στο φορτηγό και να μην βρίσκεται μπροστά του κανένα άλλο κιβώτιο. Γράψτε τελεστές STRIPS για τις ενέργειες φόρτωσης ενός κιβωτίου, μετάβασης του φορτηγού, και εκφόρτωσης ενός κιβωτίου.

- Ο τελεστής STRIPS Ride(x,e,f1,f2) περιγράφει την ενέργεια μεταφοράς ενός ατόμου x με έναν ανελκυστήρα e από τον όροφο f1 στον όροφο f2, και ορίζεται ως εξής:

  Op(ACTION: Ride(x,e,f1,f2), PRECOND: On(x,f1) $\wedge$ On(e,f1) $\wedge$ Working(e),
  EFFECT: $\neg$On(x,f1) $\wedge$ $\neg$On(e,f1) $\wedge$ On(x,f2) $\wedge$ On(e,f2))

  A) Γράψτε τον ορισμό για τον τελεστή STRIPS Call(x,e,f) ο οποίος περιγράφει την ενέργεια κλήσης ενός ανελκυστήρα e από ένα άτομο x, που βρίσκεται στον όροφο f, έτσι ώστε ο ανελκυστήρας να φτάνει τελικά στον όροφο f.

  B) Ο John είναι στον όροφο 2 και θέλει να πάει στον όροφο 3, αλλά ο μόνος ανελκυστήρας που λειτουργεί είναι ο ανελκυστήρας e που βρίσκεται στον όροφο 7. Δείξτε διαγραμματικά πλάνο που επιλύει το πρόβλημα. Δείξτε ξεκάθαρα τις προϋποθέσεις και αποτελέσματα των βημάτων, καθώς και τις αιτιολογικές συνδέσεις.

  C) Υπάρχουν πάνω από ένας τρόποι για να λυθεί το πρόβλημα; Εξηγήστε γιατί ναι ή γιατί όχι.

- Υποθέστε ότι έχετε τους παρακάτω τελεστές STRIPS (αγνοήστε το γεγονός ότι η λίστα παραμέτρων είναι κενή):

Operator 1	Operator 2	Operator 3	Operator 4
PRECOND: A	PRECOND: A $\wedge$ D	PRECOND: B	PRECOND: A $\wedge$ C
EFFECTS: Z $\wedge$ $\neg$B	EFFECTS: E $\wedge$ $\neg$A	EFFECTS: D $\wedge$ $\neg$C	EFFECTS: B $\wedge$ $\neg$E

  Υποθέστε ότι ο στόχος είναι να κατασκευαστεί ένα πλάνο που μετατρέπει την αρχική κατάσταση A $\wedge$ C στην τελική κατάσταση D $\wedge$ E $\wedge$ Z. Σχεδιάστε ένα πλάνο κατασκευασμένο με χρήση σχεδιασμού μερικής διάταξης (POP) το οποίο επιλύει το πρόβλημα. <u>Δείξτε ξεκάθαρα τις αιτιολογικές συνδέσεις και τους περιορισμούς διάταξης στο πλάνο</u>.

- Υποθέστε ότι έχετε τους παρακάτω τελεστές STRIPS (αγνοήστε το γεγονός ότι η λίστα παραμέτρων είναι κενή):

Operator 1	Operator 2	Operator 3
PRECOND: A	PRECOND: A $\wedge$ C	PRECOND: A $\wedge$ D
EFFECTS: D $\wedge$ $\neg$B	EFFECTS: E $\wedge$ $\neg$D	EFFECTS: Z $\wedge$ $\neg$A

Ο στόχος είναι να κατασκευαστεί πλάνο που μετατρέπει την αρχική κατάσταση A $\wedge$ B $\wedge$ C σε κατάσταση που περιλαμβάνει το D $\wedge$ E $\wedge$ Z.

A) Δείξτε τα δέντρα αναζήτησης των αλγορίθμων αναζήτησης κατα βάθος (DFS) και κατα πλάτος (BFS) που προκύπτουν όταν οι αλγόριθμοι έχουν επεκτείνει τους 4 πρώτους κόμβους. **B)** Πόσους κόμβους επεκτείνει ο κάθε αλγόριθμος μέχρι να βρει λύση? Για τα A και B υποερωτήματα υποθέστε ορθή διάχιση του χώρου καταστάσεων με αποφυγή επέκτασης επαναλαμβανόμενων καταστάσεων. **Γ)** Δείξτε το δέντρο αναζήτησης που προκύπτει από την εφαρμογή ενός εκ των DFS ή BFS (δική σας επιλογή) μέχρι την εύρεση λύσης, υποθέτοντας ανάστροφη διάσχιση του χώρου καταστάσεων με αποφυγή επέκτασης επαναλαμβανόμενων καταστάσεων. <u>Σε όλα τα υποερωτήματα υποθέστε ότι για κάθε κόμβο η σειρά δημιουργίας των κόμβων-παιδιών του από αριστερά προς τα δεξιά ακολουθεί τους δείκτες των τελεστών</u> (δηλ. πρώτα Operator1 κ.ο.κ.).