ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Λύση Ασκήσεων

2007-2008

Ιωάννης Χατζηλυγερούδης
Αικατερίνη Μπαγουλή
Όθων Μιχαήλ

Πάτρα 2008

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
2007-2008
1η Σειρά Ασκήσεων
ΑΠΑΝΤΗΣΕΙΣ

1. Μετατρέψτε τις παρακάτω προτάσεις φυσικής γλώσσας (είτε τις Ελληνικές είτε τις Αγγλικές) σε προτάσεις ΚΛΠΤ.

α. "Ο Πλούτο αγαπά το αφεντικό του"
"Pluto loves its master"

Αν θεωρήσουμε πως η πρόταση αναφέρεται σε ένα συγκεκριμένο αφεντικό του Πλούτο που είναι μοναδικό, έχουμε:

αγαπά(Πλούτο, αφεντικό_του(Πλούτο))
loves(Pluto, master_of(Pluto))

Θεωρώντας πως η πρόταση λέει ότι ο Πλούτο αγαπά οποιοδήποτε αφεντικό του, έχουμε:

$(\forall x)$ (αφεντικό(x,Πλούτο) $\Rightarrow$ αγαπά(Πλούτο,x))
$(\forall x)$ (master(x,Pluto) $\Rightarrow$loves(Pluto,x))

Τέλος, θεωρώντας πως η πρόταση λέει ότι ο Πλούτο ίσως έχει πολλά αφεντικά και αγαπάει τουλάχιστον ένα, έχουμε:

$(\exists x)$ (αφεντικό(x,Πλούτο) $\wedge$ αγαπά(Πλούτο,x))
$(\exists x)$ (master(x,Pluto) $\wedge$loves(Pluto,x))

Πιο σωστή απάντηση είναι η πρώτη. Όταν δεν είμαστε σίγουροι για την ακριβή σημασία της πρότασης που μας δίνεται στην εκφώνηση, μαζί με την απάντησή μας γράφουμε και την θεώρηση στην οποία βασιστήκαμε.

β. "Κάθε σκύλος έχει κάποιο αφεντικό"
"Every dog has a master"

$(\forall x)(\exists y)$ (σκύλος(x) $\Rightarrow$ αφεντικό(y,x))
$(\forall x)(\exists y)$ (dog(x) $\Rightarrow$ master(y,x))
ή
$(\forall x)$ (σκύλος(x) $\Rightarrow$ $(\exists y)$αφεντικό(y,x))
$(\forall x)$ (dog(x) $\Rightarrow$ $(\exists y)$master(y,x))

γ. "Ο Γιάννης ή μισεί τον Γιώργο ή είναι φιλόδοξος"
"John either hates George or is ambitious"

μισεί(Γιάννης, Γιώργος) $\vee$ φιλόδοξος(Γιάννης)
hates(John, George) $\vee$ ambitious(John)

δ. "Τα μήλα είναι ένα είδος τροφής"
"Apples are a kind of food"

Θεωρώντας την κατηγορία «μήλα» ως μια οντότητα, δηλ. σταθερά, έχουμε:

τροφή(μήλα)
food(apples)

Θεωρώντας το κάθε μήλο ως μια οντότητα και ότι τα «μήλα» αντιπροσωπεύουν ένα σύνολο τέτοιων οντοτήτων, έχουμε :

$(\forall x)$ (μήλο(x) $\Rightarrow$τροφή(x))
$(\forall x)$ (apple(x) $\Rightarrow$ food(x))

ε. "Κάθε σαρκοβόρο ζώο τρώει όλα τα ζώα τα μικρότερα από αυτό"
"Every carnivorous animal eats all animals smaller than itself"

$(\forall x)(\forall y)$((σαρκοβόρο_ζώο(x) $\wedge$ ζώο(y) $\wedge$μικρότερο(y,x)) $\Rightarrow$ τρώει(x,y))
$(\forall x)(\forall y)$((carnivorous_animal(x) $\wedge$ animal(y) $\wedge$ smaller(y,x)) $\Rightarrow$ eats(x,y))
ή
$(\forall x)$ (σαρκοβόρο_ζώο(x) $\Rightarrow$ (($\forall y$) (ζώο(y) $\wedge$μικρότερο(y,x)) $\Rightarrow$ τρώει(x,y)))
$(\forall x)$ (carnivorous_animal(x) $\Rightarrow$(($\forall y$) (animal(y) $\wedge$ smaller(y,x)) $\Rightarrow$ eats(x,y)))
ή
$\neg$(($\exists x$)($\exists y$) (σαρκοβόρο_ζώο(x) $\wedge$ ζώο(y) $\wedge$μικρότερο(y,x) $\wedge$ $\neg$τρώει(x,y))
$\neg$(($\exists x$)($\exists y$) (carnivorous_animal(x) $\wedge$ animal(y) $\wedge$smaller(y,x) $\wedge$ $\neg$eats(x,y))

στ. "Κανείς άνδρας δεν συμπαθεί μια γυναίκα που είναι χορτοφάγος"
" No man likes a woman who is vegeterian"

$(\forall x)(\forall y)$((άνδρας(x) $\wedge$ γυναίκα(y) $\wedge$ χορτοφάγος(y)) $\Rightarrow$ $\neg$συμπαθεί(x,y))
$(\forall x)(\forall y)$((man(x) $\wedge$ woman(y) $\wedge$ vegeterian(y)) $\Rightarrow$ $\neg$likes(x,y))
ή
$(\forall x)$ (άνδρας(x) $\Rightarrow$ ($\forall y$)((γυναίκα(y) $\wedge$ χορτοφάγος(y)) $\Rightarrow$ $\neg$συμπαθεί(x,y)))
$(\forall x)$ (man(x) $\Rightarrow$ ($\forall y$)((woman(y) $\wedge$ vegeterian(y)) $\Rightarrow$ $\neg$likes(x,y)))
ή
$(\forall y)$ ((γυναίκα(y) $\wedge$ χορτοφάγος(y)) $\Rightarrow$ $\neg$($\exists x$)(άνδρας(x) $\wedge$συμπαθεί(x,y)))
$(\forall y)$((woman(y) $\wedge$ vegeterian(y)) $\Rightarrow$ $\neg$($\exists x$)(man(x) $\wedge$ likes(x,y)))
ή
$\neg$(($\exists x$)($\exists y$)(άνδρας(x) $\wedge$ γυναίκα(y) $\wedge$ χορτοφάγος(y) $\wedge$συμπαθεί(x,y)))
$\neg$(($\exists x$)($\exists y$)(man(x) $\wedge$ woman(y) $\wedge$ vegeterian(y) $\wedge$likes(x,y)))

Κάθε πρόταση ΚΛΠΤ εκφράζεται σε Προτασιακή Μορφή <u>με μοναδικό τρόπο</u>. Οπότε, για να ελέγξουμε αν διαφορετικές προτάσεις ΚΛΠΤ είναι ισοδύναμες αρκεί να τις εκφράσουμε σε Προτασιακή Μορφή και να δούμε αν τα σύνολα προτάσεων που προκύπτουν γι αυτές ταυτίζονται μεταξύ τους.

2. Να μετατραπούν σε προτασιακή μορφή οι παρακάτω προτάσεις ΚΛΠΤ.

α. $(\forall x)\ (\forall y)\ (\forall z)\ ((pet(x) \wedge master(x, y) \wedge lives(y, z)) \Rightarrow lives(x, z))$

1. <u>Απαλοιφή συνεπαγωγών:</u>
$(\forall x)\ (\forall y)\ (\forall z)\ (\neg(pet(x) \wedge master(x,y) \wedge lives(y,z)) \vee lives(x,z))$

2. <u>Περιορισμός εμβέλειας του ~:</u>
$(\forall x)\ (\forall y)\ (\forall z)\ ((\neg pet(x) \vee \neg master(x,y) \vee \neg lives(y,z)) \vee lives(x,z))$

3. <u>Μετονομασία μεταβλητών:</u>
Μη εφαρμόσιμο

4. <u>Μετατροπή σε ΚΜΡ</u>
Μη εφαρμόσιμο

5. <u>Απαλοιφή υπαρξιακών ποσοσδεικτών</u>
Μη εφαρμόσιμο

6. <u>Απαλοιφή καθολικών ποσοσδεικτών</u>
((¬pet(x) ∨ ¬master(x,y) (¬lives(y,z)) (lives(x,z))

7. Μετατροπή σε ΚΣΜ
Μη εφαρμόσιμο

8. Απαλοιφή διασυνδεκτικών και καταγραφή προτάσεων
$\varphi \equiv (\neg pet(x), \neg master(x,y), \neg lives(y,z), lives(x,z))$

9. Μετονομασία μεταβλητών
Μη εφαρμόσιμο (μία μόνο πρόταση)

β. (((x) (((y) a(y) (b(x, y))) ((((x) c(x))

1. <u>Απαλοιφή συνεπαγωγών:</u>
$((\forall x)\ ((\exists y)\ \neg a(y) \vee b(x,y))) \vee ((\forall x)\ c(x))$

2. <u>Περιορισμός εμβέλειας:</u>
Μη εφαρμόσιμο

3. <u>Μετονομασία μεταβλητών:</u>
$((\forall x)\ ((\exists y)\ \neg a(y) \vee b(x,y))) \vee ((\forall z)\ c(z))$

4. <u>Μετατροπή σε ΚΜΡ</u>
$(\forall x)(\exists y)(\forall z)\ (\neg a(y) \vee b(x, y) \vee c(z))$

5. <u>Απαλοιφή υπαρξιακών ποσοσδεικτών {y = f(x)}</u>
$(\forall x)(\forall z)\ (\neg a(f(x)) \vee b(x,f(x)) \vee c(z))$

6. <u>Απαλοιφή καθολικών ποσοδεικτών</u>
   $(\neg a(f(x)) \vee b(x,f(x))$ ($c(z))$

7. <u>Μετατροπή σε ΚΣΜ</u>
   Μη εφαρμόσιμο

8. <u>Απαλοιφή διασυνδεκτικών και καταγραφή προτάσεων</u>
   $\varphi \equiv (\neg a(f(x))\ ,\ b(x, f(x))\ ,\ c(z))$

9. <u>Μετονομασία μεταβλητών</u>
   Μη εφαρμόσιμο (μία μόνο πρόταση)

3. Προσπαθείστε να ενοποιήσετε τα παρακάτω ζεύγη στοιχείων. Αν ενοποιούνται, βρείτε την γενικότερη ενοποιήτρια. Αν όχι, εξηγείστε γιατί.

α. <u>p(x, y) , p(a, z)</u>

Ενοποιούνται με γενικότερη ενοποιήτρια την σ = {a/x, y/z}

β. <u>p(x, x) , p(a, b)</u>

Δεν ενοποιούνται διότι δεν υπάρχει γ.ε. Αν υπήρχε, θα ήταν η «σ = {a/x, b/x}», η οποία όμως δεν είναι έγκυρη, διότι η μεταβλητή «x» έχει σαν προσδέσεις δύο διαφοερετικές σταθερές, επομένως δημιουργείται σύγκρουση.

γ. <u>απόγονος(x, πατέρας-του(x)) , απόγονος(γιάννης, βασίλης)</u>

Δεν ενοποιούνται., διότι ενώ η «x» ενοποιείται με την σταθερά «γιάννης», η συνάρτηση «πατέρας-του(x)» δεν μπορεί να ενοποιηθεί με μια σταθερά (την «βασίλης»).

δ. <u>απόγονος(x, y) , απόγονος(βασίλης, πατέρας-του(βασίλης))</u>

Ενοποιούνται με γ.ε. την σ = {βασίλης/x, πατέρας-του(βασίλης)/y}

ε. <u>q(x, a, y) , q(z, z, b)</u>

Και αυτά ενοποιούνται με γ.ε. την σ = {z/x, a/z, b/y}, η οποία καταλήγει στην
σ = {a/x, a/z, b/y}

στ. <u>q(x) , ¬q(a)</u>

Δεν ενοποιούνται, επειδή έχουν αντίθετη πολικότητα.

4. Δίνονται οι παρακάτω προτάσεις ΚΛΠΤ

(1) works-in(george, patras)
(2) works-in(paul, rio)
(3) master(george, pluto)
(4) master(paul, boby)
(5) $(\forall x)(\forall y)$(works-in(x, y) $\Rightarrow$ lives-in(x, y))
(6) $(\forall x)(\forall y)(\forall z)$((master(x, y) $\wedge$ lives-in(x, z)) $\Rightarrow$ lives-in(y, z))
όπου x, y, z είναι μεταβλητές.

(α) Χρησιμοποιώντας τη μέθοδο της αντίφασης της επίλυσης, αποδείξτε ότι "Pluto lives in Patra"

Πρώτα πρέπει να τις μετατρέψουμε σε προτασιακή μορφή:

(1) works-in(george, patras)
(2) works-in(paul, rio)
(3) master(george, pluto)
(4) master(paul, boby)
(5) ($\neg$works-in(x1, y1) , lives-in(x1, y1))
(6) ($\neg$master(x2, y2) , $\neg$lives-in(x2, z), lives-in(y2, z))

Η πρόταση που θέλουμε να αποδείξουμε είναι η φ = lives-in(pluto, patras) και το σύνολο των αξιωμάτων S με βάση τα οποία θα την αποδείξουμε είναι οι προτάσεις του (α). Παίρνουμε την άρνηση της φ, $\neg$φ = $\neg$lives-in(pluto, patras) και, αφού βρίσκεται ήδη σε προτασιακή μορφή, την προσθέτουμε στο S. Στη συνέχεια, εφαρμόζοντας διαρκώς επίλυση μεταξύ προτάσεων του S προσπαθούμε να καταλήξουμε στην κενή πρόταση (). Κάθε επιλύουσα που δεν είναι η κενή πρόταση προστίθεται στο S και η διαδικασία συνεχίζεται. Αν προκύψει επιλύουσα () σταματάμε την διαδικασία των επιλύσεων αφού η εξαγωγή της κενής πρότασης από το S$\cup\neg$φ σημαίνει πως η φ συνεπάγεται λογικά από το S.

Έχουμε:

(1) (works-in(george, patras))
(2) (works-in(paul, rio))
(3) (master(george, pluto))
(4) (master(paul, boby))
(5) ($\neg$works-in(x1, y1) , lives-in(x1, y1))
(6) ($\neg$master(x2, y2) , $\neg$lives-in(x2, z), lives-in(y2, z))
(7) ($\neg$lives-in(pluto, patras))

---

(8)	($\neg$lives-in(george, z), lives-in(pluto, z))	(3, 6) με σ={george/x2 , pluto/y2}
(9)	(lives-in(george, patras))	(1, 5) με σ={george/x1 , patras/y1}
(10)	(lives-in(pluto, patras))	(8, 9) με σ={patras/z}
(11)	( )	(7, 10)

Οπότε αποδεικνύεται η πρόταση φ.

Αφού εδώ δεν μας περιορίζει κάποια στρατηγική στα βήματα που θα κάνουμε, οποιαδήποτε σειρά επιλύσεων διαλέξει κανείς είναι σωστή, αρκεί να οδηγεί στην () με σωστές επιλύσεις.

(β) Χρησιμοποιώντας τη μέθοδο της αντίφασης της επίλυσης, απαντήστε στο ερώτημα "Who lives in Rio?" (βρείτε δηλ. τις τιμές της μεταβλητής x για τις οποίες αληθεύει η πρόταση (∃x) lives-in(x, rio)).

Ακολουθούμε την προηγούμενη διαδικασία με την διαφορά πως εδώ παίζουν μεγαλύτερο ρόλο από πριν οι αντικαταστάσεις που έχουν γίνει όταν φτάνουμε στην κενή
πρόταση. Από αυτές βρίσκουμε τις ζητούμενες τιμές του x. Μια ακόμα διαφορά ειναι πως τώρα δεν θα σταματήσουμε την πρώτη φορά που θα καταλήξουμε στην (). Θα επαναλάβουμε την διαδικασία για να δούμε με πόσες διαφορετικές τιμές του x οδηγούμαστε σε απόδειξη της πρότασης.

Η πρόταση που θέλουμε να αποδείξουμε είναι η φ = (∃x) lives-in(x, rio).
Παίρνουμε την άρνηση της φ, ¬φ = ¬(∃x) lives-in(x, rio) = (∀x) ¬lives-in(x, rio), την μετατρέπουμε σε προτασιακή μορφή : ¬lives-in(x, rio) και την προσθέτουμε στο S.

Έχουμε:

(1) (works-in(george, patras))
(2) (works-in(paul, rio))
(3) (master(george, pluto))
(4) (master(paul, boby))
(5) (¬works-in(x1, y1) , lives-in(x1, y1))
(6) (¬master(x2, y2) , ¬lives-in(x2, z), lives-in(y2, z))
(7) (¬lives-in(x, rio))

---

(8) (lives-in(paul, rio)) (2, 5) με σ={paul/x1 , rio/y1}
(9) ( ) (7, 8) με σ={paul/x}
Η διαδικασία επαναλαμβάνεται για την επόμενη τιμή του x:

(8) (¬lives-in(paul, z), lives-in(boby, z)) (4, 6) με σ={paul/x2 , boby/y2}
(9) (¬works-in(paul, z), lives-in(boby, z)) (5, 8) με σ={paul/x1 , z/y1}
(10) (lives-in(boby, rio)) (2, 9) με σ={rio/z}
(11) ( ) (7, 8) με σ={boby/x}

Εδώ μπορούμε να σταματήσουμε αφού συνεχίζοντας θα βρούμε ξανά τις ίδιες τιμές:

(8) (¬works-in(x, rio)) (5, 7) με σ={x/x1 , rio/y1}
(9) ( ) (2, 8) με σ={paul/x}

και

(8) (¬master(x2, x), ¬lives-in(x2, rio)) (6, 7) με σ={x/y2 , rio/z}
(9) (¬works-in(x2, rio), ¬master(x2, x)) (5, 8) με σ={x2/x1 , rio/y1}
(10) (¬master(paul, x)) (2, 9) με σ={paul/x2}
(11) ( ) (4, 10) με σ={boby/x}

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
2007-2008
2η Σειρά Ασκήσεων
ΑΠΑΝΤΗΣΕΙΣ

1. Δίνονται οι παρακάτω προτάσεις, σε προτασιακή μορφή:
(1) {p(x1, y1), ¬s(x1), ¬t(x1)}
(2) {p(x2, y2), ¬q(x2, f(x2)), ¬r(x2, f(y2))}
(3) {q(b, y3)}
(4) {q(a, y4), ¬r(a, f(a))}
(5) {r(a, f(a))}
(6) {r(b, f(b))}
(7) {s(a)}
(8) {s(b)}
(9) {t(a)}.

(α) Αν η προς αποδειξη πρόταση (στόχος) είναι η p(x, y), σχεδιάστε το SLD-δέντρο.

SLD Δέντρο:

$$(\neg p(x,y))$$

$(1)\ \sigma_1=\{x1/x,\ y1/y\}$ $\qquad$ $(2)\sigma_2=\{x2/x,\ y2/y\}$

$(\neg s(x1), \neg t(x1))$ $\qquad$ $(\neg q(x2, f(x2)), \neg r(x2, f(y2)))$

$(7)\sigma_7=\{a/x1\}$ $\quad$ $(8)\ \sigma_8=\{b/x1\}$ $\qquad$ $(3)\sigma_3=\{b/x2,\ f(b)/y3\}$ $\quad$ $(4)\sigma_4=\{a/x2,\ f(a)/y4\}$

$(\neg t(a))$ $\quad$ $(\neg t(b))$ $\qquad$ $(\neg r(b, f(y2)))$ $\qquad$ $(\neg r(a, f(y2)), \neg r(a, f(a)))$

$(9)\downarrow$ $\quad$ $\downarrow$ $\qquad$ $\downarrow (6)\sigma_6=\{b/y2\}$ $\qquad$ $\downarrow (5)\sigma_7=\{a/y2\}$

$(\ )$ $\quad$ failure $\qquad$ $(\ )$ $\qquad$ $(\neg r(a, f(a)))$

$\sigma=\sigma_1\circ\sigma_7=\{a/x, a/x1, y1/y\}$ $\qquad$ $\sigma=\sigma_2\circ\sigma_3\circ\sigma_6=\{b/x, b/y, b/x2, b/y2, f(b)/y3\}$ $\qquad$ $(5)\downarrow$

$(\ )$

$\sigma=\sigma_2\circ\sigma_4\circ\sigma_5=\{a/x, a/y, a/x2, a/y2, f(a)/y4\}$

Στο δέντρο SLD φαίνονται όλα τα δυνατά μονοπάτια απόδειξης της πρότασης p(x,y). Τρεις διαδρομές οδηγούν στην κενή πρόταση και κατά συνέπεια αποδεικνύουν πως αληθεύει η πρόταση. Παίρνουμε τις αντικαταστάσεις κατά μήκος καθεμιάς από αυτές τις διαδρομές και βρίσκουμε πως οι τιμές των x και y που επαληθεύουν την p είναι: {a/x, y1/y}, {b/x, b/y} και {a/x, a/y}.

Συνεπώς, η πρόταση p(x,y) αληθεύει αν το x έχει την τιμή a και το y έχει οποιαδήποτε τιμή, ή αν x = y = b.

(β) Με ποιά στρατηγική από τις depth-first with backtracking και breadth-first βρίσκουμε γρηγορότερα (δηλ. σε λιγότερα βήματα) τη λύση; Εξηγείστε.

Χρησιμοποιώντας depth-first with backtracing βρίσκουμε την πρώτη λύση σε 3 βήματα - επιλύσεις, τις (1) - (7) - (9).
Χρησιμοποιώντας breadth-first βρίσκουμε την ίδια λύση σε 7 βήματα – επιλύσεις, τις (1) - (2) - (7) - (8) - (3) - (4) - (9) .
Συνεπώς, με τη πρώτη στρατηγική βρίσκουμε γρηγορότερα τη λύση.

(γ) Τι μπορούμε να κάνουμε για μειώσουμε τα βήματα και στις δύο περιπτώσεις;

Συνήθως, για να μειώσουμε τα βήματα εύρεσης της λύσης το μόνο που μπορούμε να κάνουμε είναι να αλλάξουμε την σειρά κάποιων προτάσεων ή την σειρά κάποιων όρων στις προτάσεις. Στην συγκεκριμένη περίπτωση, όσες εναλλαγές και να κάνουμε θα παρατηρήσουμε πως δεν μειώνονται τα βήματα καμίας από τις παραπάνω στρατηγικές.

2. Δίνεται το παρακάτω σύνολο προτάσεων (σε προτασιακή μορφή):
{R, ¬Q, ¬T}, {R, ¬Q, ¬P}, {¬S, T, P}, {Q}, {S}, {¬T}

(α) Ποια από τις στρατηγικές P1–Επίλυση και Επίλυση Εισόδου είναι ασφαλέστερη για αυτό το σύνολο προτάσεων και γιατί;

Η P1-Επίλυση είναι πάντα πλήρης. Αντίθετα, η Επίλυση Εισόδου είναι πλήρης μόνο για προτάσεις τύπου Horn. Όμως, για παράδειγμα, η πρόταση **{¬S, T, P}** δεν είναι τύπου Horn αφού έχει περισσότερα από ένα θετικά στοιχεία, επομένως η Επίλυση Εισόδου δεν είναι πλήρης για το συγκεκριμένο σύνολο προτάσεων. Άρα ασφαλέστερη στρατηγική είναι η P1-Επίλυση που είναι πλήρης.

(β) Χρησιμοποιείστε τη διαδικασία της αντίφασης της επίλυσης και την ασφαλέστερη στρατηγική που επιλέξατε στο (α) για να αποδείξετε ότι η R είναι θεώρημα.

Σύμφωνα με την διαδικασία της αντίφασης της επίλυσης, τοποθετούμε την αντίφαση της προς απόδειξη πρότασης στο αρχικό σύνολο προτάσεων και επιλύοντας σύμφωνα με την επιλεγείσα στρατηγική προσπαθούμε να καταλήξουμε στην κενή πρόταση.

Να θυμηθούμε στο σημείο αυτό ότι στην P1-Επίλυση πρέπει σε κάθε βήμα της επίλυσης τουλάχιστον ένας απ’ τους γονείς να είναι θετική πρόταση, δηλαδή να είναι μια πρόταση με όλα της τα στοιχεία θετικά.

Να επισημάνουμε επίσης ότι πρέπει να επιλύουμε οτιδήποτε μπορεί να επιλυθεί ακόμα και αν μας οδηγεί σε πρόταση με περισσότερα στοιχεία απ’ τους γονείς της. Αν δεν το κάνουμε αυτό μπορεί να χάσουμε λύσεις ή ίσως στην χειρότερη περίπτωση να μην ακολουθήσουμε ποτέ την μοναδική διαδρομή που θα μας οδηγούσε σε λύση.

1. **(R, ¬Q, ¬T)**
2. **(R, ¬Q, ¬P)**
3. **(¬S, T, P)**
4. **(Q)**
5. **(S)**
6. **(¬T)**
7. **(¬R) ---> Η αντίφαση της προς απόδειξη πρότασης**

---

8. **(R, ¬T)** **(4, 1) Επιλύσαμε δηλ. την 4 με την 1**
9. **(R, ¬P)** **(4, 2)**
10. **(T, P)** **(5, 3) Θετική πρόταση**

---

11. **(R, ¬Q, P)** **(10, 1)**
12. **(R, ¬Q, T)** **(10, 2)**
13. **(P)** **(10, 6) Θετική πρόταση**
14. **(R, P)** **(10, 8) Θετική πρόταση**
15. **(R, T)** **(10, 9) Θετική πρόταση**

Για να μην γράφουμε πολύ, σκεφτόμαστε ως εξής: Ποιά απ’ αυτές τις προτάσεις θα μπορούσε ίσως να μας οδηγήσει στην κενή πρόταση; Παρατηρούμε ότι αν επιλύσουμε την 15 με την 7 θα προκύψει η πρόταση (T). Επομένως στο επόμενο βήμα θα μπορούμε να επιλύσουμε την (T) με την 6 (¬T), επίλυση η οποία θα μας δώσει την κενή πρόταση.

Επομένως η R είναι θεώρημα.

3. Δίνονται τα εξής γεγονότα: «Ο Παύλος είναι πατέρας του Γιάννη και της Γεωργίας» και «Η Ελένη είναι μητέρα της Μαρίας και του Πέτρου». Επίσης, μας δίνεται και η εξής γνώση τύπου κανόνα, που αφορά το πότε δύο άνθρωποι είναι αδέλφια: «Δύο άνθρωπο είναι αδέλφια αν είτε έχουν τον ίδιο πατέρα είτε έχουν την ίδια μητέρα».

(α) Ζητείται να αναπαρασταθεί η παραπάνω γνώση σε Prolog, ώστε να δημιουργηθεί αντίστοιχο πρόγραμμα.

Το πρόγραμμα θα έχει ως εξής:

```
is_father(paul, john).
is_father(paul, georgia).
is_mother(helen, mary).
is_mother(helen, peter).
siblings(X, Y) :- is_father(Z, X), is_father(Z, Y).
siblings(X, Y) :- is_mother(Z, X), is_mother(Z, Y).
```

(β) Τι θα απαντήσει η Prolog στο ερώτημα «Ποια είναι αδέλφια του Γιάννη;» (Υλοποιείστε το ερώτημα σε Prolog).

Αν θέσουμε στην Prolog το ερώτημα:

```
?- siblings(georgia, john).
```

τότε θα εκτυπωθεί :

```
Yes
```

Το ζητούμενο ερώτημα σε Prolog είναι: siblings(X, john).Αν θέσουμε αυτό το ερώτημα:

```
?- siblings(X, john).
```

τότε θα εκτυπωθεί :

```
X = john ;
X = georgia ;
No
```

Εδώ το No σημαίνει ότι δεν υπάρχουν άλλες απαντήσεις-λύσεις.

Όπως είναι φανερό, το πρόγραμμα εξάγει το συμπέρασμα ότι ο Γιάννης είναι αδελφός του εαυτού του, το οποίο δεν θα θέλαμε να εξάγεται.

(γ) Αν οι απαντήσεις στο (β) περιλαμβάνουν και τον Γιάννη (ως αδελφό του εαυτού του), τότε τροποποιείστε το πρόγραμμα που κάνατε στο (α) για να εξαληφθεί αυτή η περίπτωση.

Για να αποτρέψουμε την προηγούμενη κατάσταση, κάνουμε τις εξής προσθήκες:

```
is_father(paul, john).
is_father(paul, georgia).
is_mother(helen, maria).
is_mother(helen, peter).
equal(john, john).
equal(georgia, georgia).
equal(maria, maria).
equal(peter, peter).
siblings(X, Y) :- is_father(Z, X), is_father(Z, Y),not(equal(X, Y)).
siblings(X, Y) :- is_mother(Z, X), is_mother(Z, Y),not(equal(X, Y)).
```

Εναλλακτικά, κρατάμε το πρόγραμμα του (α) όπως έχει αλλάζοντας μόνο τους δύο κανόνες ως εξής:

```
siblings(X, Y) :- is_father(Z, X), is_father(Z, Y), \+ X==Y.
siblings(X, Y) :- is_mother(Z, X), is_mother(Z, Y), \+ X==Y.
```

Στη θέση του <u>\+ X==Y</u> μπορεί να μπει και το <u>not(X=Y).</u> Μπορεί ακόμα να χρησιμοποιηθεί το <u>X \= Y.</u>

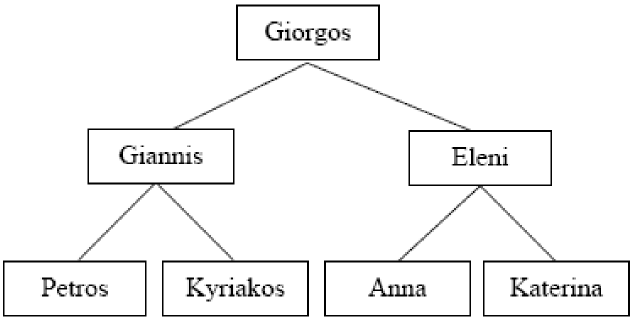
Τώρα, αν θέσουμε το ερώτημα:

```
?- siblings(X, john).
```

θα εκτυπωθεί

```
X = georgia ;
No
```

4. Δίνεται το παρακάτω δυαδικό δέντρο:



(α) Να γράψετε πρόγραμμα Prolog που να αναπαριστά το παραπάνω δέντρο με μια σειρά από γεγονότα, χρησιμοποιώντας το κατηγόρημα tree/3.

Για να αναπαραστήσουμε την λειτουργία του παραπάνω δέντρου στην Prolog ορίζουμε γεγονότα της μορφής tree(X, Y, Z) που δηλώνουν πως ο κόμβος X του δέντρου είναι γονέας των Y και Z, με τον Y να είναι το αριστερό παιδί του X και τον Z να είναι το δεξί παιδί του X.

Εχουμε:

```
tree(giorgos, giannis, eleni).
tree(giannis, petros, kyriakos).
tree(eleni, anna, katerina).
tree(petros,nil,nil).
tree(kyriakos, nil, nil).
tree(anna,nil,nil).
tree(katerina,nil,nil).
```

(β) Να συμπληρώσετε το πρόγραμμα με κανόνα(ες) έτσι ώστε να διαπερνά το δέντρο κατά βάθος με ταυτόχρονη εκτύπωση των ονομάτων των κόμβων (δηλ. Giorgos-Giannis-Petros-Kyriakos-Eleni-Anna-Katerina).

Αυτό που χρειάζεται είναι ο παρακάτω αναδρομικός κανόνας:

```
print_tree(X):-tree(X,Y,Z), write(X), nl, print_tree(Y), print_tree(Z).
print_tree(X):-tree(X,nil,nil).
```

Προκειμένου η εκκίνηση του προγράμματος να γίνεται με το κατηγόρημα go (χωρίς ορίσματα), προσθέτουμε τον κανόνα:

```
go:- write('Give Root: '), read(T), nl, print_tree(T).
```

Το πρόγραμμα αυτό μπορεί να εφαρμοστεί σε δέντρα οποιουδήποτε βάθους.

(γ) Το ίδιο με το (β) για διαπέραση κατά πλάτος (δηλ. Giorgos-Giannis-Eleni-Petros-Kyriakos-Anna-Katerina).

Χρησιμοποιούμε τους παρακάτω κανόνες:

```
print_br(X):-write(X), nl, print_tree(X).

print_tree(X):-tree(X,nil,nil).
print_tree(X):-tree(X,Y,Z), write(Y), nl, write(Z), nl, print_tree(Y), print_tree(Z).
```

Προκειμένου η εκκίνηση του προγράμματος να γίνεται με το κατηγόρημα go (χωρίς ορίσματα), προσθέτουμε τον κανόνα:

```
go:- write('Give Root: '), read(T), nl, print_br(T).
```

Το πρόγραμμα αυτό μπορεί να διαπεράσει κατα πλάτος το συγκεκριμένο δέντρο και οποιοδήποτε υποδέντρο του αλλά δεν δουλεύει σωστά για δέντρα οποιουδήποτε βάθους.

Για να επιτύχουμε διαπέραση κατα πλάτος σε δέντρα οποιουδήποτε βάθους, χρειάζεται να χρησιμοποιήσουμε λίστες.

Η λύση με λίστα θα ήταν:

```
print_br(X,L):-write(X), nl, print_tree(X,L).

print_tree(X,L):-tree(X,nil,nil).
print_tree(X,L):-tree(X,Y,Z),write(Y),nl,write(Z),nl,
                 L=[X|Rest],
                 append(Rest,[Y,Z],L2),
                 L2=[Next|Tail],
                 print_tree(Next,L2).

go:-  write('Give Root: '), read(T), nl, append([],[T],L), print_br(T,L).
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
2007-2008
3η Σειρά Ασκήσεων
ΑΠΑΝΤΗΣΕΙΣ

1. Δίνεται το γνωστό πρόβλημα των δύο δοχείων: «Υπάρχουν δύο δοχεία χωρητικότητας 4 και 3 λίτρων αντίστοιχα και μια βρύση. Κατ' αρχήν, τα δοχεία είναι άδεια. Θέλουμε να απομονώσουμε σ' ένα από τα δοχεία ποσότητα 2 λίτρων. Οι δυνατές ενέργειες είναι: γέμισμα των δοχείων από τη βρύση, άδειασμα των δοχείων στο έδαφος, άδειασμα του ενός δοχείου στο άλλο, μερικώς ή ολικώς.»

Ζητούνται:

α) Να ορίσετε (α1) την αρχική κατάσταση, (α2) την/τις τελική/ές κατάσταση/εις και (α3) τους τελεστές μετάβασης, με βάση την αναπαράσταση μιας κατάστασης, όπως δόθηκε στις παραδόσεις.

Ορίζουμε σαν αναπαράσταση μιας τυχαίας κατάστασης την: (x, y)
όπου x η ποσότητα νερού στο δοχείο των 4 λίτρων (με δυνατές τιμές 0, 1, 2, 3, 4) και y η ποσότητα νερού στο δοχείο των 3 λίτρων (με δυνατές τιμές 0, 1, 2, 3).

α1) Αρχική κατάσταση: (0,0)
α2) Τελικές καταστάσεις : (2,0), (2,1), (2,2), (2,3), (0,2), (1,2), (3,2), (4,2)
Με άλλα λόγια, το κριτήριο τερματισμού είναι: x=2 ή y=2.
α3) Τελεστές μετάβασης:

ΤΕΛΕΣΤΗΣ:ΠΕΡΙΓΡΑΦΗ	ΠΡΟΫΠΟΘΕΣΕΙΣ	ΑΠΟΤΕΛΕΣΜΑ
T1: Γέμισε το Α	$x<4$	(4,y)
T2: Γέμισε το Β	$y<3$	(x,3)
T3:Άδειασε το Α	$x>0$	(0,y)
T4: Άδειασε το Β	$y>0$	(x,0)
T5: Άδειασε το Α στο Β	$x>0$, $y<3$	Αν $x\geq3-y$ τότε (x-(3-y),3), αλλιώς (0,y+x)
T6: Άδειασε το Β στο Α	$x<4$, $y>0$	Αν $y\geq4-x$ τότε (4, y-(4-x)), αλλιώς (y+x,0)

β) Προσδιορίστε τον χώρο καταστάσεων του προβλήματος. Υπάρχουν ανέφικτες καταστάσεις; Αν ναι, μπορούν να προσδιοριστούν;

Χώρος καταστάσεων: (x.y): $x \in \{0,1,2,3,4\}$, $y \in \{0,1,2,3\}$

Αναλυτικά:

(0,0), (0,1), (0,2), (0,3)
(1,0), **(1,1)**, **(1,2)**, (1,3)
(2,0), **(2,1)**, **(2,2)**, (2,3)
(3,0), **(3,1)**, **(3,2)**, (3,3)
(4,0), (4,1), (4,2), (4,3)

Ανέφικτες καταστάσεις είναι αυτές στις οποίες, με δεδομένη την αρχική κατάσταση και τους τελεστές μετάβασης, δεν θα φτάσουμε ποτέ. Ξεκινάμε, λοιπόν, εφαρμόζοντας στην αρχική κατάσταση τους τελεστές που μπορούν να εφαρμοστούν. Με την εφαρμογή αυτών των τελεστών προκύπτουν κάποιες νέες καταστάσεις. Σε καθεμία από τις νέες καταστάσεις εφαρμόζουμε τους τελεστές που μπορούν να εφαρμοστούν σε αυτήν κοκ. Οταν προκύπτουν καταστάσεις που έχουν ήδη αναπτυχθεί δεν εφαρμόζουμε ξανά τελεστές σ'αυτές (θα προκύψουν καταστάσεις που έχουν ήδη προκύψει). Σε κάποια στιγμή φτάνουμε σ'ένα σημείο από το οποίο και μετά οι καταστάσεις πλέον διαρκώς θα ανακυκλώνονται. Όταν δεν μπορούμε να συνεχίσουμε άλλο παρατηρούμε πως κάποιες από τις καταστάσεις του Χώρου Καταστάσεων δεν εμφανίστηκαν καθόλου στον γράφο και ούτε πρόκειται να εμφανιστούν όσο κι αν συνεχίσουμε. Έτσι βρίσκουμε ποιές είναι οι ανέφικτες καταστάσεις. Ανέφικτες καταστάσεις εδώ είναι όσες από τις παραπάνω έχουν σημειωθεί πιο έντονα, δηλαδή οι καταστάσεις στις οποίες κανένα από τα δύο δοχεία δεν είναι άδειο ούτε γεμάτο. Επομένως, εφικτές είναι μόνο οι καταστάσεις όπου ένα από τα δύο δοχεία είναι άδειο ή γεμάτο.

γ) Ορίστε μια συνάρτηση κόστους g(n) και μια ευρετική συνάρτηση h(n).

Συνάρτηση κόστους:

Η συνάρτηση κόστους σχετίζεται με το **κόστος μετάβασης**. Το κόστος μετάβασης χαρακτηρίζει μια μετάβαση από μια κατάσταση n'= (x', y') σε μια άλλη κατάσταση n = (x, y). Στην περίπτωσή μας θεωρούμε ότι το κόστος αυτό σχετίζεται με τον όγκο του νερού που διακινείται σε μια μετάβαση και το εκφράζουμε ως εξής:

$$g(n', n) \begin{cases} |x'-x|+|y'-y| & \text{αν } (x=x' \text{ ή } y=y') \\ (|x'-x|+|y'-y|)/2 & \text{αν } (x\neq x' \text{ και } y\neq y') \end{cases}$$

Επομένως, το συνολικό κόστος μετάβασης (δηλ. το κόστος μετάβασης από την αρχή) στην κατάσταση n είναι: g(n) = g(n') + g(n', n)

Ευρετική συνάρτηση:

Σαν ευρετικό, σε μια κατάσταση (x, y), θεωρούμε το πόσο κοντά στα 2 λίτρα βρίσκεται η ποσότητα νερού σε κάθε δοχείο. Επίσης, θέλουμε να είναι h(n)=0 στις καταστάσεις-στόχους. Γι’ αυτό, εκφράζουμε την ευρετική συνάρτηση ως εξής:

$$h(n) \begin{cases} (|2-x|+|2-y|)/2 & \text{αν } x, y \neq 2 \\ 0 & \text{αν } x=2 \text{ ή } y=2 \end{cases}$$

δ) Εφαρμόστε τα τρια πρώτα βήματα των αλγορίθμων (δ1) αναζήτηση δέσμης και (δ2) διακλάδωση και δέσμευση, σχεδιάζοντας τα αντίστοιχα επίπεδα στο δέντρο καταστάσεων.

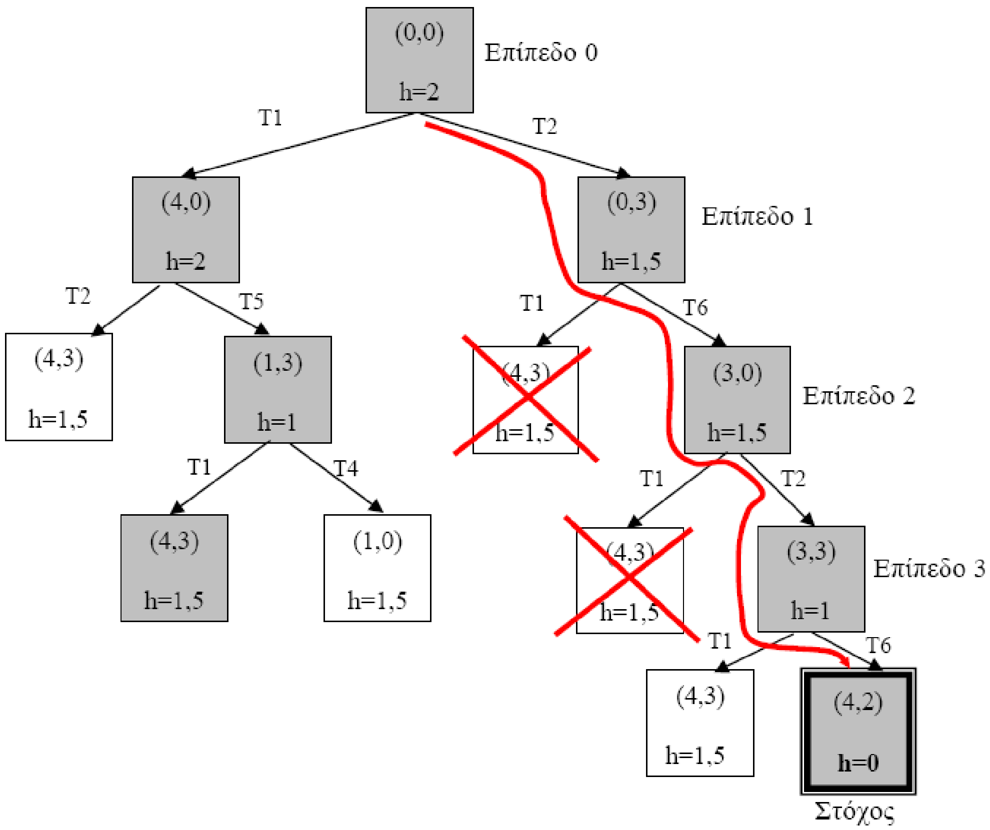
Αναζήτηση Δέσμης

Ο αλγόριθμος αναζήτησης δέσμης επιλέγει τα m καλυτερα παιδιά (με βάση το h(n)) από κάθε επίπεδο για περαιτέρω ανάπτυξη. Τα υπόλοιπα τα διαγράφει. Επίσης καθορίζουμε τα εξής για αποδοτικότερη λειτουργία του αλγορίθμου:

(α) Διαγράφονται όσες καταστάσεις έχουν ήδη αναπτυχθεί σε προηγούμενο επίπεδο είτε είναι στο ίδιο μονοπάτι είτε όχι (αφού μας ενδιαφέρει να βρούμε μια λύση).
(β) Διαγράφεται κάθε ίδια κατάσταση που παράγεται στο ίδιο επίπεδο, δηλ. κρατάμε μόνο την πρώτη που παράγεται.
(γ) Στην περίπτωση δύο ή περισσότερων καταστάσεων που έχουν την ίδια ευρετική τιμή προτιμούμε τέτοια διάταξη, ώστε μια κατάσταση να ανήκει σε διαφορετικό γονέα από την προηγούμενη και την επόμενή της.

(δ) Μεταξύ δύο καταστάσεων του ίδιου γονέα με την ίδια ευρετική συνάρτηση επιλέγουμε την πρώτη αριστερά.

Στη συνέχεια φαίνεται η ανάπτυξη (όλου) του δέντρου για m=2, όπου δεν αναγράφονται καθόλου τα κλαδιά που οδηγούν σε καταστάσεις που έχουν ήδη αναπτυχθεί (δηλ. όσες εμπίπτουν στην περίπτωση (α)), με κόκκινο Χ παριστάνεται η διαγραφή καταστάσεων της περίπτωσης (β) και με γκρι εμφανίζονται οι καταστάσεις που επιλέγονται σε κάθε επίπεδο.



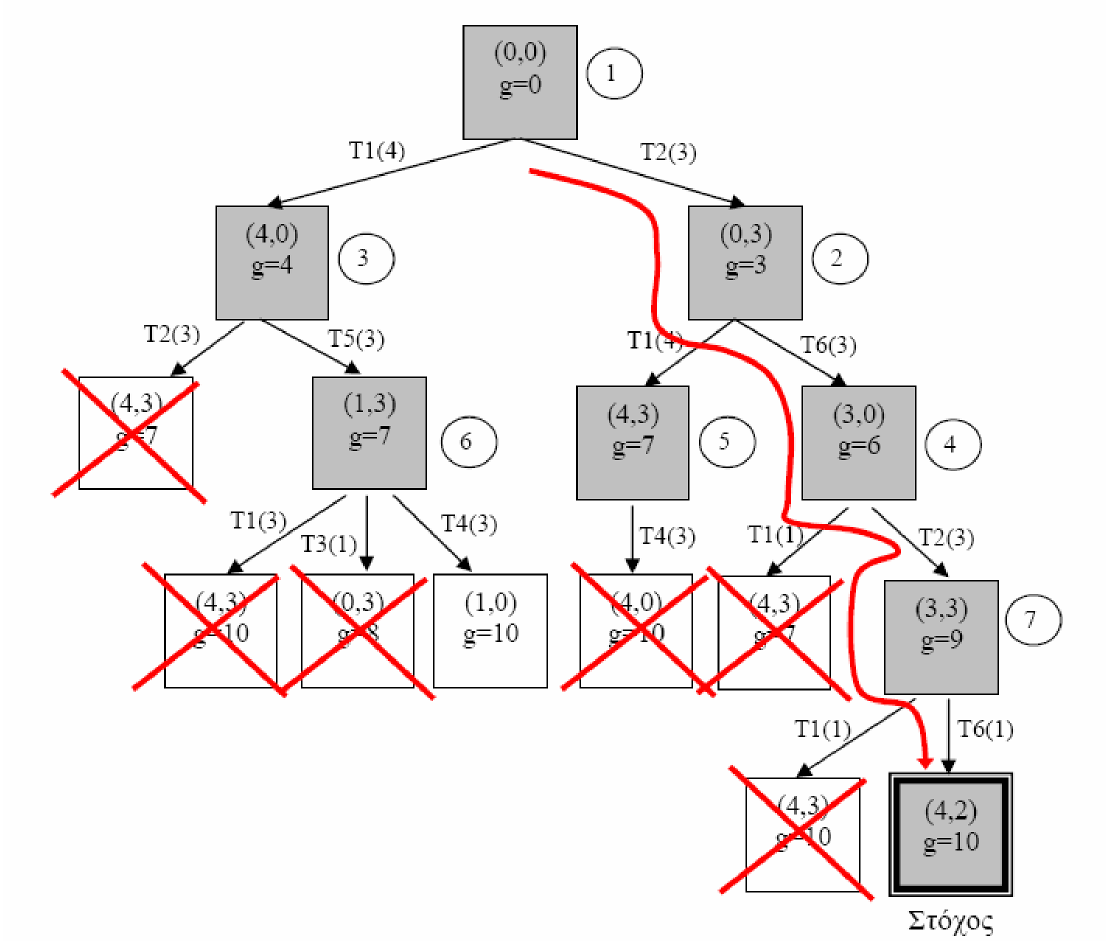
<u>Σημείωση :</u> Στην παραπάνω εικόνα ο κόμβος που έχει διαγραφεί στο επίπεδο 3 είναι λάθος. Αντιστοιχεί στην κατάσταση (4,0) με h=2 που έχει αναπτυχθεί στο επίπεδο 1.

Η διαδρομή λύσης που βρέθηκε (και είναι μία από τις δυνατές) φαίνεται με κόκκινη γραμμή. Άρα μια λύση του προβλήματος είναι: T2-T6-T2-T6, δηλαδή:

Γέμισε το δοχείο Β (Τ2)
Άδειασε το δοχείο Β στο Α (Τ6)
Γέμισε το δοχείο Β (Τ2)
Άδειασε το δοχείο Β στο Α (Τ6)

<u>Διακλάδωση και δεύσμευση (ή επέκταση και οριοθέτηση)</u>

Ο αλγόριθμος διακλάδωσης και δέσμευσης επιλέγει κάθε φορά από τις τρέχουσες ανοικτές καταστάσεις (που δεν έχουν αναπτυχθεί) την κατάσταση που «απέχει» λιγότερο από την αρχή, με βάση τη συνάρτηση κόστους g(n). Ως γνωστόν, βρίσκει τη βέλτιστη λύση (δηλ. τη λύση με το μικρότερο κόστος), αν η συνάρτηση κόστους είναι σωστή. Θεωρούμε παρόμοια με τον προηγούμενο αλγόριθμο:
(α) Διαγράφονται όσες καταστάσεις έχουν ήδη αναπτυχθεί σε προηγούμενο επίπεδο.
(β) Διαγράφεται κάθε ίδια ανοικτή κατάσταση με μεγαλύτερο κόστος.
(γ) Στην περίπτωση δύο ή περισσότερων ίδιων καταστάσεων που έχουν το ίδιο κόστος διαγράφουμε αυτή που παρήχθη πιο πρόσφατα.
(δ) Στην περίπτωση δύο ή περισσότερων καταστάσεων που έχουν το ίδιο κόστος προτιμούμε αυτή που παρήχθη παλαιότερα.



<u>Σημείωση :</u> Στην παραπάνω εικόνα ο κόμβος-παιδί του (3,0)- που έχει διαγραφεί στο επίπεδο 3 είναι λάθος. Αντιστοιχεί στην κατάσταση (4,0) που έχει αναπτυχθεί στο επίπεδο 1.

Η διαδρομή λύσης που βρέθηκε (και είναι η βέλτιστη) φαίνεται με κόκκινη γραμμή. Παρατηρείστε τώρα ότι και ο αλγόριθμος αναζήτησης δέσμης βρήκε την ίδια λύση, που κατά σύμπτωση είναι η βέλτιστη. Άρα η βέλτιστη λύση του προβλήματος είναι:

T2-T6-T2-T6.

2. Δίνεται το γνωστό πρόβλημα των Ιεραποστόλων και Κανιβάλων:

«Υπάρχουν τρείς ιεραπόστολοι, τρεις κανίβαλοι και μια βάρκα στη μια όχθη ενός ποταμού. Θέλουμε όλοι να μεταφερθούν στην απέναντι όχθη με τη βάρκα. Όμως, η βάρκα για να κινηθεί χρειάζεται τουλάχιστον ένα άτομο, χωρά μόνο μέχρι δύο άτομα και σε καμμιά περίπτωση δεν πρέπει ο αριθμός των κανιβάλων να είναι μεγαλύτερος από αυτόν των ιεραποστόλων σε κάποια όχθη.»

Ζητούνται:

α) Να περιγραφεί σαν πρόβλημα αναζήτησης, δηλ. να οριστούν η αρχική κατάσταση, η τελική κατάσταση, οι τελεστές μετάβασης, μια συνάρτηση κόστους g(n) και μια ευρετική συνάρτηση h(n).

Η αναπαράσταση μιας τυχαίας κατάστασης του προβλήματος μπορεί να είναι η εξής:

(K1, I1, K2, I2, O)

όπου:

K1, K2 είναι ο αριθμός των καννιβάλλων στην όχθη 1, 2 αντίστοιχα,
I1, I2 είναι ο αριθμός των ιεραποστόλων στην όχθη 1, 2 αντίστοιχα,
O είναι 1 ή 2 ανάλογα με το σε ποιά όχθη βρίσκεται η βάρκα.

Με βάση τα παραπάνω έχουμε:

Αρχική κατάσταση: (3, 3, 0, 0, 1)

Τελική κατάσταση: (0, 0, 3, 3, 2)

Πέντε τελεστές μετάβασης είναι οι παρακάτω. Οι υπόλοιποι 5 (για να καλύψουμε και τις περιπτώσεις της αντίστροφης μεταφοράς) είναι παρόμοιοι:

Α/Α	ΠΕΡΙΓΡΑΦΗ	ΠΡΟΫΠΟΘΕΣΕΙΣ	ΑΠΟΤΕΛΕΣΜΑ
T1	Μεταφορά 1 ιεραποστόλου από την Οχθη1 στην Οχθη2	I1>0, (I1>=K1+1 ή I1=1), O=1	(K1, I1-1, K2, I2+1, 2)
T2	Μεταφορά 1 καννιβάλου από την Οχθη1 στην Οχθη2	K1>0, (I2>K2 ή K2=0 ή I2=0), O=1	(K1-1, I1, K2+1, I2, 2)
T3	Μεταφορά 2 ιεραποστόλων από την Οχθη1 στην Οχθη2	I1>1, (I1>=K1+2 ή I1=2), O=1	(K1, I1-2, K2, I2+2, 2)
T4	Μεταφορά 2 καννιβάλων από την Οχθη1 στην Οχθη2	K1>1, (I2>K2+1 ή I2=0),O=1	(K1-2, I1, K2+1, I2, 2)
T5	Μεταφορά 1 ιεραπ. και 1 καν. από την Οχθη1 στην Οχθη2	K1,I1>0, (K2=0 ή I2>0), O=1	(K1-1, I1-1, K2+1, I2+1, 2)

Μια άλλη αναπαράσταση μιας τυχαίας κατάστασης του προβλήματος, συνοπτικότερη και επομένως αποδοτικότερη, μπορεί να είναι η εξής:
(K, I, O)
όπου K είναι ο αριθμός των καννιβάλλων στην όχθη 1
I είναι ο αριθμός των ιεραποστόλων στην όχθη 1
O είναι 1 ή 2 ανάλογα με το σε ποιά όχθη βρίσκεται η βάρκα.
Οπότε, αρχική κατάσταση: (3, 3, 1), τελική κατάσταση: (0, 0, 2)

Πέντε τελεστές μετάβασης είναι οι παρακάτω. Οι υπόλοιποι 5 (για να καλύψουμε και τις περιπτώσεις της αντίστροφης μεταφοράς) είναι παρόμοιοι:

Α/Α	ΠΕΡΙΓΡΑΦΗ	ΠΡΟΫΠΟΘΕΣΕΙΣ	ΑΠΟΤΕΛΕΣΜΑ
Τ1	Μεταφορά 1 ιεραποστόλου από την Οχθη1 στην Οχθη2	I>0, (I>=K+1 ή I=1), O=1	(K, I-1, 2)
Τ2	Μεταφορά 1 καννιβάλου από την Οχθη1 στην Οχθη2	K>0, (I=3 ή I=0),O=1	(K-1, I, 2)
Τ3	Μεταφορά 2 ιεραποστόλων από την Οχθη1 στην Οχθη2	I>1, (I>=K+2 ή I=2), O=1	(K, I-2, 2)
Τ4	Μεταφορά 2 καννιβάλων από την Οχθη1 στην Οχθη2	K1>1, (K=3 ή I=0),O=1	(K-2, I, 2)
Τ5	Μεταφορά 1 ιεραπ. και 1 καν. από την Οχθη1 στην Οχθη2	K,I>0, (K=3 ή I<3), O=1	(K-1, I-1, 2)

<u>Συνάρτηση Κόστους</u>

Δεν υπάρχει κάτι που να διαφοροποιεί το «κόστος» (π.χ. τον κόπο ή το έργο που απαιτείται) στις διάφορες μεταβάσεις, οπότε θεωρούμε ότι το κόστος μετάβασης είναι πάντα 1, δηλ. g(n) = 1 + g(n-1) για κάθε κατάσταση n (όπου n-1 είναι η κατάσταση-γονέας της n).

<u>Ευρετική Συνάρτηση</u>

Σαν ευρετική συνάρτηση, δηλ. μια συνάρτηση που μετρά την απόσταση κάθε κατάστασης από τον στόχο (τελική κατάσταση), ορίζουμε την εξής:

h(n) = K1+I1 ή h(n) = K+I
(ανάλογα με την αναπαράσταση)

Η συνάρτηση αυτή επιστρέφει τον αριθμό των καννιβάλων και ιεραποστόλων που απομένουν ακόμη για μετακίνηση.

β) Ποιός αλγόριθμος πρέπει να χρησιμοποιηθεί ώστε να βρούμε β1) μια οποιαδήποτε λύση, β2) όλες τις λύσεις, β3) τη συντομότερη, β4) τη βέλτιστη λύση. Δικαιολογείστε τις απαντήσεις σας.

(β1) μία λύση:
- <u>κατά βάθος</u> (depth-first).
  (Είναι συνήθως υπολογιστικά πιο «φθηνός» και πιο γρήγορος στο να βρίσκει μια λύση)

(β2) όλες τις λύσεις:
- <u>κατά πλάτος</u> (breadth-first)
  (Βρίσκει όλες τις λύσεις με ασφάλεια και χωρίς χρήση ευρετικών)
- <u>επαναληπτική εκβάθυνση</u> (iterative-deepening)
  (Βρίσκει όλες τις λύσεις με ασφάλεια, σε περισσότερο χρόνο από τον κατά πλάτος, αλλά με λιγότερη χρήση μνήμης)

(β3) συντομότερη λύση = λιγότερα βήματα:
- <u>κατά πλάτος</u>
  (Βρίσκει πάντα τη συντομότερη λύση, δηλ. αυτή με τα λιγότερα βήματα, που δε σημαίνει όμως ότι είναι και η βέλτιστη. Στην περίπτωσή μας όμως, λόγω σταθερού κόστους βρίσκει και την βέλτιστη.)

(β4) βέλτιστη λύση = μικρότερο κόστος :

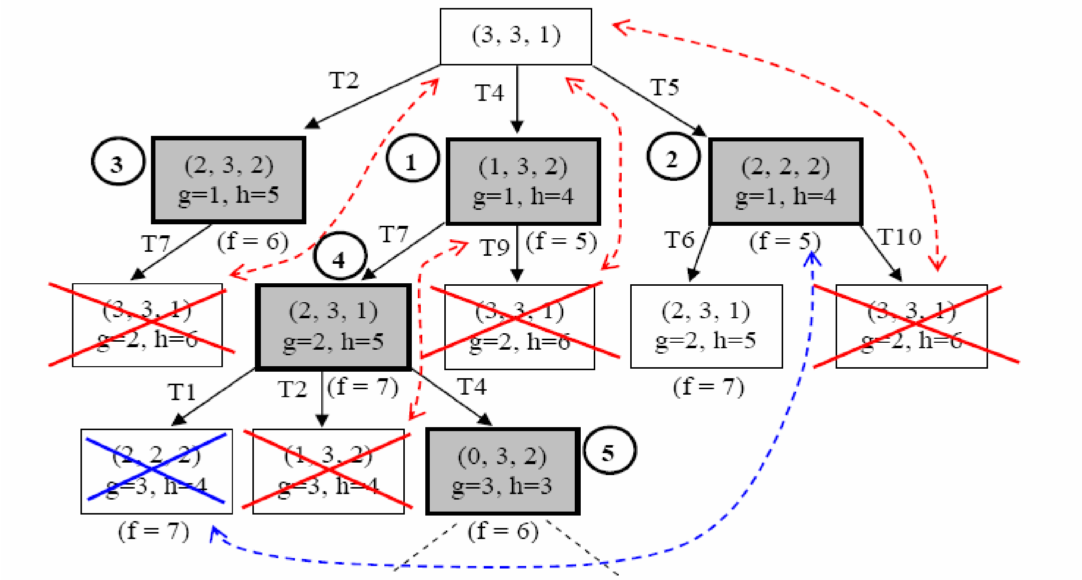
- A*

(Εγγυάται εύρεση της βέλτιστης λύσης. Δεδομένου όμως ότι το κόστος είναι σταθερό, ουσιαστικά μετατρέπεται σε αλγόριθμο βέλτιστου κόμβου (best-first). Οπότε, θα μπορούσε να προταθεί και ο βέλτιστου κόμβου, ως υπολογιστικά «ελαφρύτερος».)

Στο β4 θα μπορούσε να προταθεί και ο B&B, με την επιφύλαξη της συνάρτησης κόστους, αλλά στην περίπτωσή μας μεταπίπτει στον κατα πλάτος λόγω σταθερού κόστους. Επομένως, θα μπορούσε να προταθεί και ο κατά πλάτος, ο οποίος όμως είναι συνήθως πιο αργός από τον A*.

γ) Εφαρμόστε τα τρια πρώτα βήματα του αλγορίθμου που προτείνατε για το β4, σχεδιάζοντας τα αντίστοιχα επίπεδα στο δέντρο καταστάσεων.

A* : (Εφαρμόζουμε τη δεύτερη μέθοδο αναπαράστασης καταστάσεων)



Στο παραπάνω σχήμα έχουν αναπτυχθεί περισσότερα από 3 βήματα για εκπαιδευτικούς λόγους. Οι σκιασμένοι κόμβοι είναι αυτοί που προτιμήθηκαν για ανάπτυξη λόγω καλύτερης εκτίμησης f και οι αριθμοί δίπλα τους μέσα σε κύκλο δείχνουν τη σειρά ανάπτυξης.

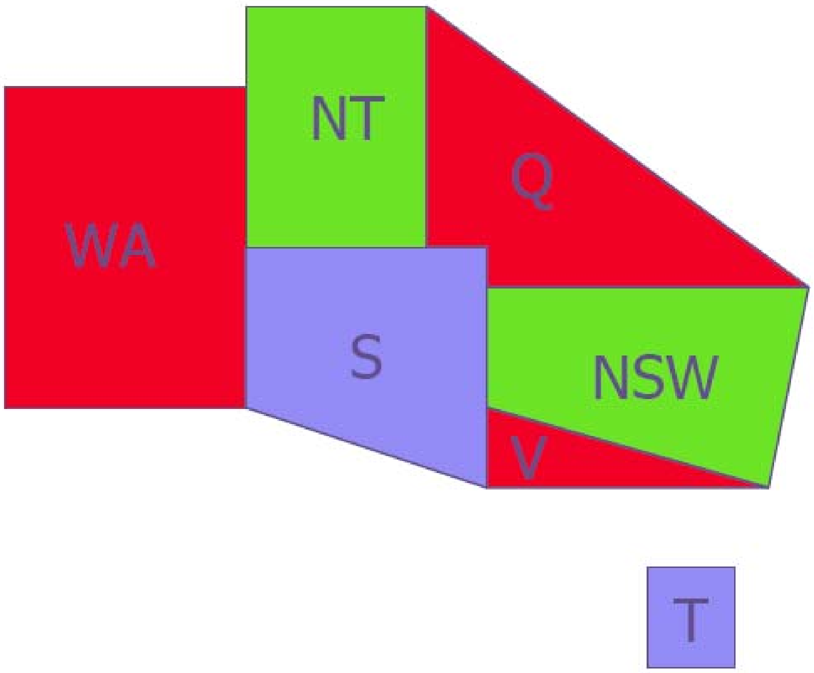
Στην ανάπτυξη του δέντρου ακολουθήθηκαν οι παρακάτω κανόνες:

1. Μια κατάσταση που είναι ίδια με μία πρόγονο κατάσταση, δεν αναπτύσσεται (βλ. καταστάσεις με κόκκινη διαγράμμιση στο σχήμα).
2. Αν δύο καταστάσεις έχουν την ίδια εκτίμηση f και βρίσκονται στο ίδιο επίπεδο, τότε αναπτύσσεται πρώτα η αριστερότερα ευρισκόμενη.
3. Αν έχουμε δύο ίδιες καταστάσεις και δεν βρίσκονται στο ίδιο επίπεδο, τότε διαγράφεται αυτή που απέχει περισσότερο από τον στόχο (π.χ. βλ. κατάσταση με μπλε διαγράμμιση στο σχήμα).

Γενικά, οι κανόνες αυτοί ακολουθούνται σε τέτοιους αλγορίθμους.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
2007-2008
4η Σειρά Ασκήσεων
ΑΠΑΝΤΗΣΕΙΣ

1. Το πρόβλημα χρωματισμού χαρτών (map coloring problem) συνίσταται στο να χρωματιστούν διαφορετικές περιοχές ενός χάρτη έχοντας έναν ορισμένο αριθμό χρωμάτων, έτσι ώστε γειτονικές περιοχές να μην έχουν το ίδιο χρώμα.



Μία λύση στο πρόβλημα χρωματισμού του συγκεκριμένου χάρτη με τρία χρώματα

α. Για τον παραπάνω χάρτη περιγράψτε το πρόβλημα ως πρόβλημα ικανοποίησης
περιορισμών και δώστε τα πρώτα βήματα εκτέλεσής του.

Στο συγκεκριμένο πρόβλημα μπορούμε να ορίσουμε 7 μεταβλητές, τις:
{WA, NT, S, Q, NSW, V, T}

Και οι 7 έχουν κοινό πεδίο τιμών, το:
{Κ, Π, Μ}, όπου Κ: Κόκκινο, Π: Πράσινο και Μ: Μπλε

Ο μόνος περιορισμός που μας δίνεται απ' το πρόβλημα είναι ότι δύο γειτονικές μεταβλητές (περιοχές στον χάρτη) δεν μπορούν να έχουν ταυτόχρονα την ίδια τιμή (χρώμα). Επομένως θα πρέπει να ικανοποιούνται οι ακόλουθοι περιορισμοί:

WA ≠ NT, WA ≠ S, NT ≠ S, NT ≠ Q, S ≠ Q, S ≠ NSW, S ≠ V, Q ≠ NSW, NSW ≠ V

Στον παρακάτω πίνακα φαίνονται τα πρώτα βήματα εκτέλεσης:

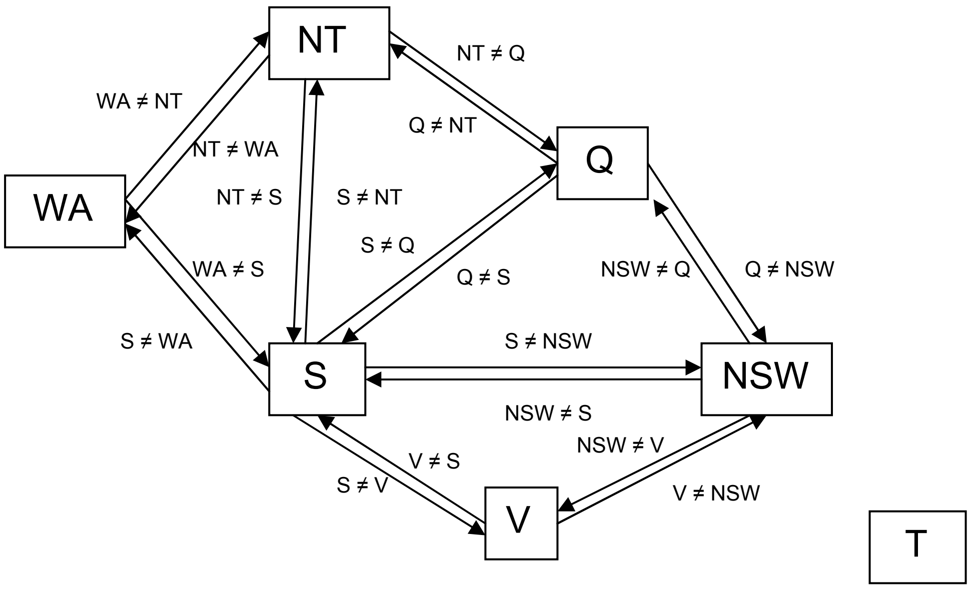
WA	NT	Q	NSW	V	S	T
Κ,Π,Μ	Κ,Π,Μ	Κ,Π,Μ	Κ,Π,Μ	Κ,Π,Μ	Κ,Π,Μ	Κ,Π,Μ
Κ	Π,Μ	Κ,Π,Μ	Κ,Π,Μ	Κ,Π,Μ	Π,Μ	Κ,Π,Μ
Κ	**Π**	**Κ**	**Π**	**Κ**	**Μ**	**Μ**

Έτσι, για WA = K, NT = Π, Q = K, NSW = Π, V = K, S = M, T = M, έχουμε μία λύση του προβλήματος.

Η Τ δεν συνδέεται με κανέναν κόμβο οπότε μπορούμε να της δώσουμε όποια τιμή θέλουμε. Της δώσαμε τελικά την τιμή 'Μπλε', για να καταλήξουμε στον χρωματισμό του σχήματος.

β. Σχεδιάστε ένα γράφο δυαδικών περιορισμών για το παραπάνω πρόβλημα και εξηγήστε τι εκφράζουν οι κόμβοι και τα τόξα στον γράφο αυτό. Στον γράφο θα πρέπει να εμφανίζονται και οι αντίστοιχοι περιορισμοί.

Ο γράφος δυαδικών περιορισμών για το παραπάνω πρόβλημα είναι ο ακόλουθος:



Οι κόμβοι εκφράζουν τις μεταβλητές του προβλήματος.

Κάθε τόξο από ένα κόμβο u σε ένα κόμβο w, δηλώνει την ύπαρξη ενός δυαδικού περιορισμού μεταξύ των μεταβλητών των κόμβων u και w. Ο περιορισμός πρέπει να εμφανίζεται και προς τις δύο κατευθύνσεις, δηλ. και από τον u στον w και από τον w στον u.

γ. Αν επιλύαμε το πρόβλημα χρωματισμού χαρτών με κάποιον κλασσικό αλγόριθμο αναζήτησης, θα μπορούσαμε να επιταχύνουμε την διαδικασία εύρεσης λύσης χρησιμοποιώντας μια ευρετική συνάρτηση που να βασίζεται στην αρχή της συντομότερης αποτυχίας (first fail principle). Σύμφωνα με την αρχή αυτή, επιλέγεται σε κάθε εφαρμογή του τελεστή ανάθεσης τιμής η μη-δεσμευμένη μεταβλητή η οποία είναι πιθανότερο να οδηγήσει συντομότερα σε αποτυχημένους κόμβους στην αναζήτηση. Για ποιο λόγο πιστεύετε ότι μια τέτοια προσέγγιση μπορεί να επιταχύνει την εύρεση λύσης; Θεωρείτε ότι οι κλασσικές μέθοδοι

αναζήτησης είναι επαρκείς για την επίλυση προβλημάτων ικανοποίησης/επίλυσης περιορισμών και αν όχι γιατί;

First Fail Principle:

Σύμφωνα με την αρχή αυτή, επιλέγεται σε κάθε εφαρμογή του τελεστή ανάθεσης τιμής η μη-δεσμευμένη μεταβλητή η οποία είναι πιθανότερο να οδηγήσει συντομότερα σε αποτυχημένους κόμβους στην αναζήτηση. Έτσι, επιλέγεται εκείνη με το μικρότερο πεδίο τιμών, καθώς λιγότερες διαθέσιμες τιμές σημαίνει και μεγαλύτερη πιθανότητα αποτυχίας. Σε περίπτωση που περισσότερες μεταβλητές έχουν τον ίδιο αριθμό τιμών στα πεδία τους, τότε επιλέγεται εκείνη η μεταβλητή που συμμετέχει σε περισσότερους περιορισμούς (most constrained principle). Η λογική στην οποία βασίζεται η αρχή είναι ότι όσο συντομότερα ανακαλυφθούν οι αποτυχημένες αναθέσεις τιμών σε ένα πρόβλημα, τόσο μικρότερο μέρος του δέντρου αναζήτησης θα εξεταστεί για την εύρεση λύσεων, άρα τόσο γρηγορότερα θα οδηγηθούμε σε λύση.

Η αναζήτηση (κλασικές μέθοδοι) δεν είναι επαρκής για την επίλυση προβλημάτων CSP.

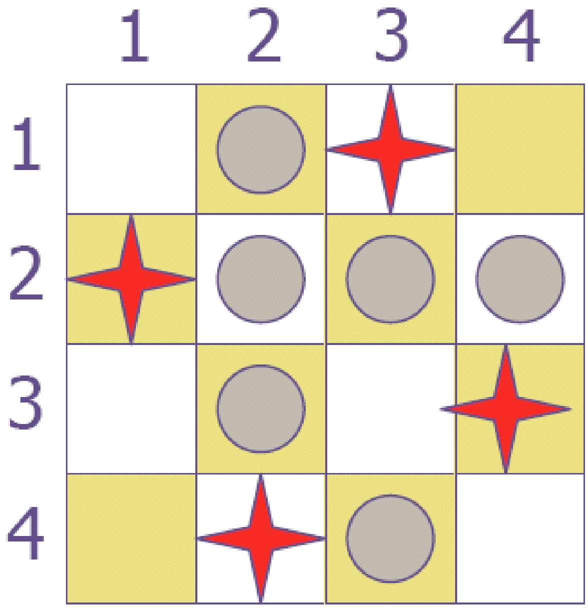
Αν και οι ευριστικοί αλγόριθμοι βελτιώνουν σε μεγάλο βαθμό την απόδοση των τυφλών αλγορίθμων αναζήτησης, εντούτης δεν επιλύουν σε ικανοποιητικό χρόνο προβλήματα μεγάλου μεγέθους. Το μεγάλο μειονέκτημα των κλασικών αυτών μεθόδων είναι ότι δεν εκμεταλλεύονται σε μεγάλο βαθμό τους περιορισμούς που υπάρχουν σε τέτοιου είδους προβλήματα. Ο λόγος είναι ότι ο έλεγχος για το αν η λύση ικανοποιεί τους περιορισμούς γίνεται μετά την παραγωγή της (a posteriori), δηλαδή πρώτα δημιουργείται η λύση (έστω και μερική) και μετά ελέγχεται για το αν είναι ικανοποιητική. Το γεγονός αυτό σε συνδυασμό με το φαινόμενο της συνδυαστικής έκρηξης, κάνει την επίλυση προβλημάτων με τις μεθόδους αυτές εξαιρετικά χρονοβόρα.

Συνδυαστική Έκρηξη (Combinatorial Explosion):

Τα προβλήματα ικανοποίησης περιορισμών χαρακτηρίζονται από μεγάλο αριθμό μεταβλητών και πιθανών τιμών (πεδίων), γεγονός που οδηγεί στο φαινόμενο της συνδυαστικής έκρηξης, δηλαδή στην πολύ μεγάλη αύξηση του χώρου αναζήτησης, πράγμα που κάνει την επίλυσή τους εξαιρετικά χρονοβόρα.

Στα μαθηματικά μία συνδυαστική έκρηξη περιγράφει το αποτέλεσμα συναρτήσεων που μεγαλώνουν ταχύτατα λόγω πολλαπλών συνδυασμών.

2. **Το πρόβλημα των *N* βασιλισσών αφορά στην τοποθέτηση σε μια σκακιέρα *N*x*N* των βασιλισσών, χωρίς να απειλούν η μια την άλλη. Μια λύση του προβλήματος για *N* = 4, φαίνεται στο παρακάτω σχήμα.**



Μια λύση του προβλήματος των τεσσάρων βασιλισσών (οι κύκλοι δείχνουν τις θέσεις που απειλεί η βασίλισσα που βρίσκεται στην (2,1))

α. Να αναπαρασταθεί το πρόβλημα των τεσσάρων βασιλισσών ως πρόβλημα ικανοποίησης περιορισμών.

Μπορούμε να ορίσουμε **4 μεταβλητές**, τις:
{Q1, Q2, Q3, Q4}
Κάθε μία αντιπροσωπεύει μία στήλη της σκακιέρας.

Και οι τέσσερις έχουν **κοινό πεδίο τιμών** και συγκεκριμένα:
D = {1, 2, 3, 4}
Κάθε τιμή του D αντιπροσωπεύει μία γραμμή.

Ικανοποιούμε εξ' αρχής τον πρώτο περιορισμό που λέει ότι δύο βασίλισσες δεν μπορούν να βρίσκονται στην ίδια στήλη αναθέτοντας σε κάθε στήλη j την μεταβλητή Q_j. Επομένως:

Q_j : Η βασίλισσα της στήλης j.

Ας δούμε τώρα τι γίνεται με τις γραμμές. Αφού το πεδίο τιμών της μεταβλητής Q_j είναι οι τέσσερις γραμμές της σκακιέρας, τότε μπορούμε να ορίσουμε πλήρως την θέση μίας βασίλισσας, έστω αυτής της στήλης j, ως εξής:

$\mathbf{Q_j = i}$: Αυτό σημαίνει απλά ότι η βασίλισσα της στήλης j είναι τοποθετημένη στην γραμμή i, άρα με τον τρόπο αυτό έχουμε ορίσει και τις δύο συντεταγμένες της βασίλισσας άρα και την ακριβή θέση της πάνω στο επίπεδο-σκακιέρα.

Περιορισμοί:

Ας δούμε τώρα ποιοί ακριβώς είναι οι περιορισμοί που ανακύπτουν από ένα τέτοιο πρόβλημα λαμβάνοντας υπ' όψιν μας τα εξής:

1. Κάθε βασίλισσα απειλεί οριζόντια
2. Κάθε βασίλισσα απειλεί διαγώνια
3. Το ότι απειλεί και κάθετα δεν μας απασχολεί γιατί με τον τρόπο που ορίσαμε τις μεταβλητές δεν θα μπορέσουμε ποτέ να τοποθετήσουμε δύο βασίλισσες στην ίδια στήλη
4. Δεν μας απασχολεί τι γίνεται προς τα αριστερά της βασίλισσας που τοποθετούμε σε κάθε βήμα διότι η τοποθέτηση θα ξεκινάει πάντα απ' την πρώτη στήλη. Το μόνο που θα ελέγχουμε για κάθε βασίλισσα είναι το σε ποιές γραμμές της τρέχουσας στήλης δικαιούμαστε να την τοποθετήσουμε με βάση τους προς τα δεξιά περιορισμούς των βασιλισσών που έχουν ήδη τοποθετηθεί. Εν συνεχεία θα ανανεώνουμε τους προς τα δεξιά περιορισμούς με βάση τους νέους περιορισμούς που προκύπτουν λόγω της βασίλισσας που μόλις τοποθετήσαμε.

$$\forall i, j : Q_j \neq Q_i$$

$Q_j \neq Q_{j+n} + n$, για $n > 1$ και $j + n \leq 4$

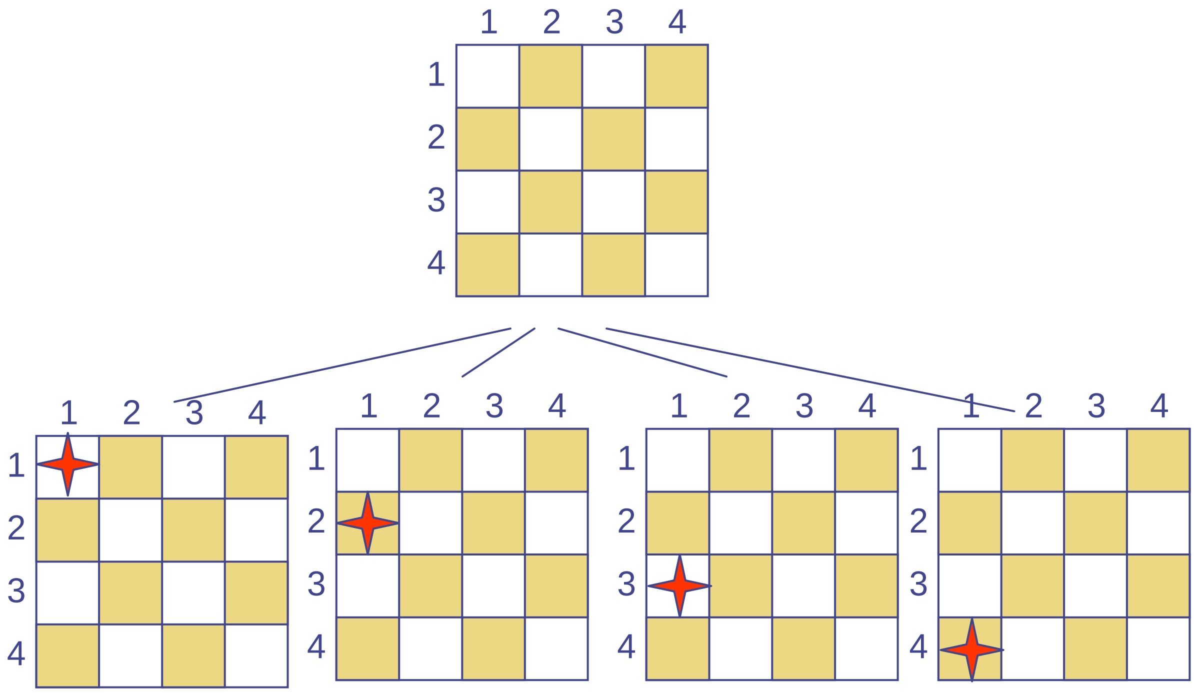
$Q_j \neq Q_{j+n} - n$, για $n > 1$ και $j + n \leq 4$

Ας μεταφράσουμε λίγο τα παραπάνω μαθηματικά:

Το $\forall i, j : Q_j \neq Q_i$ δηλώνει ότι οι περιορισμοί αυτοί ισχύουν για βασίλισσες διαφορετικής στήλης όπως ακριβώς θέλαμε να συμβαίνει.
Ο περιορισμός $Q_j \neq Q_{j+n} + n$ εξασφαλίζει ότι δεν θα τοποθετηθεί βασίλισσα διαγώνια και προς τα πάνω απ' την βασίλισσα της στήλης j. Λέει με άλλα λόγια ότι η γραμμή της βασίλισσας που απέχει *n* στήλες απ' την βασίλισσα της στήλης j (δηλ. η Q_{n+j}) απαγορεύεται να είναι τέτοια, που αν της προσθέσουμε *n* γραμμές, δηλαδή αν την κατεβάσουμε κατά *n* γραμμές, να έρχεται στην ίδια γραμμή με τη βασίλισσα της στήλης j. Σκεφτείτε για να το καταλάβετε δύο βασίλισσες τοποθετημένες ως εξής: Η μία βρίσκεται στην τέταρτη γραμμή της πρώτης στήλης και η δεύτερη στην πρώτη γραμμή της τέταρτης στήλης. Είναι προφανές ότι απειλούνται διαγώνια. Κατεβάστε τώρα κατά τέσσερις γραμμές, όσες δηλαδή είναι και οι στήλες που τις χωρίζουν, την δεύτερη βασίλισσα. Είναι προφανές ότι έρχεται στην ίδια ευθεία με την πρώτη βασίλισσα και αυτό είναι που μας λέει ο παραπάνω περιορισμός ότι δεν θα πρέπει να συμβαίνει.

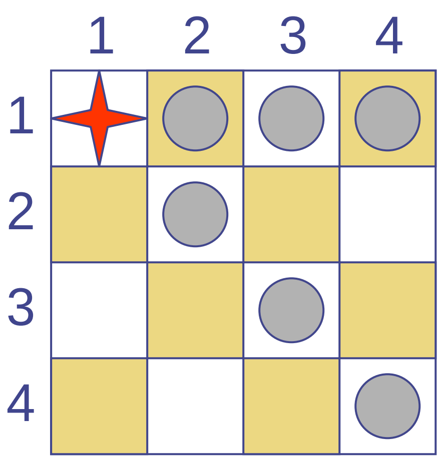
Ο $Q_j \neq Q_{j+n} - n$ λέει το ίδιο με τον προηγούμενο αλλά για την διαγώνια και προς τα κάτω απειλή.

β. Με βάση την αναπαράσταση που επιλέξατε στο *α* να βρείτε μία λύση συνδυάζοντας αναζήτηση και διάδοση περιορισμών. Συνιστάται να χρησιμοποιήσετε κατά πλάτος αναζήτηση. Ξεκινώντας, δηλαδή, από μία αρχική κατάσταση, σε κάθε κόμβο του δέντρου αναζήτησης θα πραγματοποιείται διάδοση περιορισμών (με όποιον αλγόριθμο θέλετε εσείς π.χ. AC-3) και μόνο όταν ο αλγόριθμος επιστρέψει θα συνεχίζεται η αναζήτηση. Με λίγα λόγια, η αναζήτηση θα ψάχνει για λύση ενώ η διάδοση περιορισμών θα την βοηθάει να μην εξετάζει κάποιες περιπτώσεις που δεν πρόκειται να την οδηγήσουν σε λύση.



Οι δυνατές επιλογές για το πρώτο βήμα της αναζήτησης

Έστω ότι στο πρώτο βήμα της αναζήτησης επιλέγουμε Q_1 = 1, δηλαδή η βασίλισσα της πρώτης στήλης να τοποθετηθεί στην γραμμή 1. Τότε προκύπτουν οι ακόλουθοι περιορισμοί:



Επομένως αν εφαρμόσουμε διάδοση περιορισμών για το τρέχον βήμα της αναζήτησης έχουμε τα εξής:

Q_1 = {1} Q_2 = {3, 4} Q_3 = {2, 4} Q_4 = {2, 3}

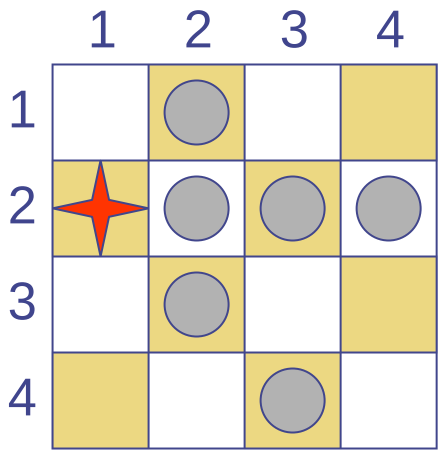
Αν για την Q_2 επιλέξουμε την τιμή 3, τότε παρατηρούμε ότι απειλούνται οι θέσεις (2, 3) και (4, 3) οπότε δεν μας μένει κανένα ελεύθερο τετράγωνο για να τοποθετήσουμε την βασίλισσα Q_3.

Αν για την Q_2 επιλέξουμε την τιμή 4, τότε:

Q_1 = {1} Q_2 = {4} Q_3 = {2} Q_4 = {}

οπότε τώρα δεν μας έμεινε καμία θέση να τοποθετήσουμε την βασίλισσα Q_4.
Άρα το συγκεκριμένο βήμα της αναζήτησης έχει αποτύχει.

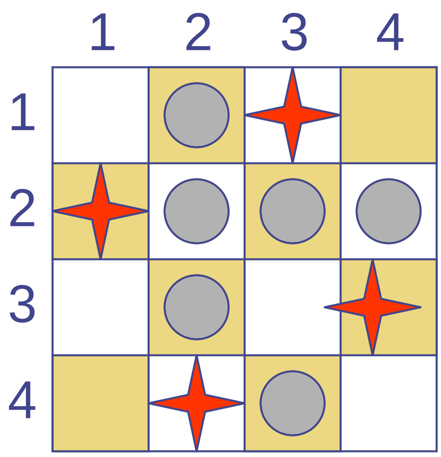
Δοκιμάζουμε το επόμενο βήμα της αναζήτησης που είναι το ακόλουθο:



Για την διάδοση περιορισμών ισχύουν τα εξής:

$Q_1 = \{2\}$ $Q_2 = \{4\}$ $Q_3 = \{1, 3\}$ $Q_4 = \{1, 3, 4\}$
$Q_1 = \{2\}$ $Q_2 = \{4\}$ $Q_3 = \{1\}$ $Q_4 = \{1, 3\}$ Τοποθέτηση της βασίλισσας Q_2 στην γραμμή 4
$Q_1 = \{2\}$ $Q_2 = \{4\}$ $Q_3 = \{1\}$ $Q_4 = \{3\}$ Τοποθέτηση της βασίλισσας Q_3 στην γραμμή 1

Επομένως, βρήκαμε μία λύση του προβλήματος. Σχηματικά:



Φυσικά θα μπορούσαμε να συνεχίσουμε την αναζήτηση, εφαρμόζοντας διάδοση περιορισμών σε κάθε βήμα της, για να βρούμε όλες τις δυνατές λύσεις.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
2007-2008
5η Σειρά Ασκήσεων
ΑΠΑΝΤΗΣΕΙΣ

1. Δίνεται η παρακάτω βάση κανόνων:

R1: if A and B then C
R2: if C and D then E
R3: if A and I then ¬H
R4: if A and ¬D then E
R5: if C and ¬D then I
R6: if E and I then ¬H
R7: if E and H then ¬G
R8: if E and ¬H then G

(α) Ζητείται να εξαχθεί το K, αν
- **το αρχικό περιεχόμενο της μνήμης εργασίας είναι WM = {A, B, ¬D, E}**
- **χρησιμοποιηθεί αλυσίδωση προς τα εμπρός (forward chaining)**
- **ο πρώτος υποψήφιος κανόνας που συναντάται πυροδοτείται (σειρά αναγραφής)**
- **ο ίδιος κανόνας μόνο μια φορά πυροδοτείται**
(Περιγράψτε τα βήματα της εξαγωγής)

Αρχικά: WM = {A, B, ~D, E} , ΣΣ = Σύνολο Σύγκρουσης

ΣΣ = {R1, R4}
R1→ WM = {A, B, ~D, E, **C**} - παραγωγή του C

ΣΣ = {R4, R5}
Ο R1 έχει ήδη πυροδοτηθεί μια φορά οπότε δεν ξαναπυροδοτείται
R4→ WM = {A, B, ~D, E, **C**} - παραγωγή του E (υπάρχει ήδη)

ΣΣ = {R5}
Οι R1, R4 έχουν ήδη πυροδοτηθεί μια φορά οπότε δεν ξαναπυροδοτούνται
R5→ WM = {A, B, ~D, E, **C**, **I**} - παραγωγή του I

ΣΣ = {R3, R6}
Οι R1, R2, R5 έχουν ήδη πυροδοτηθεί μια φορά οπότε δεν ξαναπυροδοτούνται
R3→ WM = {A, B, ~D, E, **C**, **I**, **~H**} - παραγωγή του ~H

ΣΣ = {R6, R8}
Οι R1, R2, R5, R3 έχουν ήδη πυροδοτηθεί μια φορά οπότε δεν ξαναπυροδοτούνται
R6→ WM = {A, B, ~D, E, **C**, **I**, **~H**} - παραγωγή του ~H

ΣΣ = {R8}
Οι R1, R2, R5, R3, R6 έχουν ήδη πυροδοτηθεί μια φορά οπότε δεν ξαναπυροδοτούνται
R8→ WM = {A, B, ~D, E, **C**, **I**, **~H**, **G**} – παραγωγή του G → **Τέλος**

(β) Αν χρησιμοποιηθεί ως στρατηγική ελέγχου η προσφατότητα και δευτερευόντως η σειρά αναγραφής, θα υπάρξει διαφορά; Εξηγείστε

Όταν λέμε Προσφατότητα εννοούμε την Επικαιρότητα. Αυτό σημαίνει ότι προτιμούνται οι κανόνες που ενεργοποιούνται από πιο πρόσφατα δεδομένα. Επειδή με το μάτι δεν μπορούμε αν αποφανθούμε για το αν τελικά θα υπάρξει βελτίωση, θα πρέπει να εφαρμόσουμε και πάλι εξαγωγή συμπεράσματος με τις στρατηγικές ελέγχου που μας ζητάει το ερώτημα αυτό. Θα φέρνουμε μπροστά τους πιο πρόσφατους κανόνες (κανόνες που παρήχθησαν στο τρέχον βήμα) και αν είναι περισσότεροι από ένας θα πρέπει να τους ταξινομούμε με βάση την σειρά αναγραφής ($2^η$ στρατηγική που αναφέρεται στην εκφώνηση) τους στην βάση κανόνων.

Αρχικά: WM = {A, B, ~D, E} , ΣΣ = Σύνολο Σύγκρουσης

ΣΣ = {R1, R4}
R1→ WM = {A, B, ~D, E, **C**} - παραγωγή του C

ΣΣ = {R5, R4}
R5→ WM = {A, B, ~D, E, **C**, **I** } - παραγωγή του I

ΣΣ = {R3, R6, R4}
R3→ WM = {A, B, ~D, E, **C**, **I**, **~H**} - παραγωγή του ~H

ΣΣ = {R8, R6, R4}
R8→ WM = {A, B, ~D, E, **C**, **I**, **~H**, **G**} - παραγωγή του G → **Τέλος**

Παρατηρούμε ότι θα υπάρξει διαφορά και πιο συγκεκριμένα θα υπάρξει βελτίωση κατά 2 κανόνες αφού με την στρατηγική του προηγούμενου ερωτήματος το G παρήχθη σε 6 βήματα (πυροδότηση 6 κανόνων συνολικά), ενώ με την στρατηγική που μόλις εφαρμόσαμε σε 4.

2. α) Υποθέστε ότι είναι γνωστά τα εξής:

- **Η πιθανότητα να δούμε κροκόδειλο στη λίμνη X είναι 0,65**
- **Η πιθανότητα να δούμε πάπιες στη λίμνη X, ενώ βλέπουμε ένα κροκόδειλο στη λίμνη είναι 0,04**
- **Η πιθανότητα να δούμε πάπιες στη λίμνη X, ενώ δεν βλέπουμε κροκόδειλο στη λίμνη είναι 0,3**

Να υπολογιστούν με βάση τον κανόνα του Bayes οι πιθανότητες:

α1) Να δούμε κροκόδειλο στη λίμνη, ενώ βλέπουμε πάπιες

Τα δεδομένα είναι:
P(K) = 0.65 , η πιθανότητα να δούμε κροκόδειλο,
P(Π/K) = 0.04 , η πιθανότητα να δούμε πάπιες ενώ βλέπουμε κροκόδειλο,
P(Π/~K) = 0.3 , η πιθανότητα να δούμε πάπιες ενώ δεν βλέπουμε κροκόδειλο
P(~K) = 1- P(K) = 0.35

Η πιθανότητα να δούμε κροκόδειλο στη λίμνη, ενώ βλέπουμε πάπιες είναι:

$$P(K/\Pi)=\frac{P(K)P(\Pi/K)}{P(K)P(\Pi/K)+P(\sim K)P(\Pi/\sim K)}=\frac{0.65*0.04}{0.65*0.04+0.35*0.3}=\frac{0.026}{0.131}=0.198$$

α2) Να δούμε κροκόδειλο στη λίμνη, ενώ δεν βλέπουμε πάπιες

Η πιθανότητα να δούμε κροκόδειλο στη λίμνη, ενώ δεν βλέπουμε πάπιες είναι:

$$P(K/\sim\Pi)=\frac{P(K)P(\sim\Pi/K)}{P(\sim\Pi)}$$

όπου P(~Π) = 1-P(Π) = 1-P(K)P(Π/K)-P(~K)P(Π/~K) = 1-0.131 = 0.869
Επίσης P(~Π/K) = 1-P(Π/K) = 1-0.04 = 0.96

Άρα $P(K/\sim\Pi)=\frac{P(K)P(\sim\Pi/K)}{P(\sim\Pi)}=\frac{0.65*0.96}{0.869}=0.718$

(β) Δίνονται τα παρακάτω στοιχεία (στον πίνακα) που παριστάνουν πιθανότητες σχετικές με τρεις αμοιβαία αποκλειόμενες και εξαντλητικές υποθέσεις (ασθένειες) (*h*1: κρυολόγημα, *h*2: αλλεργία, *h*3: ευπάθεια στο φως) και δύο υπό συνθήκη ανεξάρτητα στοιχεία (συμπτώματα) (*e*1: φτάρνισμα, *e*2: πυρετός) για ένα πρόσωποΥ. Χρησιμοποιώντας συλλογισμό κατά Bayes, να απαντήσετε στο ερώτημα «τι ασθένεια έχει ο Υ;», δεδομένου ότι παρατηρούνται και τα δύο συμπτώματα στον Υ.

	h1	*h2*	*h3*
p(hi)	0,65	0,3	0,05
p(e1/hi)	0,2	0,7	0,25
p(e2/hi)	0,7	0,8	0,0

→ Η πιθανότητα ο Υ να έχει την ασθένεια h1 δεδομένου ότι παρατηρούνται και τα δυο συμπτώματα είναι:

$$P(h1/e1e2)=\frac{P(e1/h1)P(e2/h1)P(h1)}{P(e1/h1)P(e2/h1)P(h1)+P(e1/h2)P(e2/h2)P(h2)+P(e1/h3)P(e2/h3)P(h3)}$$

$$=\frac{0.2*0.7*0.65}{0.2*0.7*0.65+0.7*0.8*0.3+0.25*0*0.05}=\frac{0.091}{0.259}=0.351$$

→ Η πιθανότητα ο Υ να έχει την ασθένεια h2 δεδομένου ότι παρατηρούνται και τα δυο συμπτώματα είναι:

$$P(h2/e1e2)=\frac{P(e1/h2)P(e2/h2)P(h2)}{P(e1/h1)P(e2/h1)P(h1)+P(e1/h2)P(e2/h2)P(h2)+P(e1/h3)P(e2/h3)P(h3)}$$

$$=\frac{0.7*0.8*0.3}{0.2*0.7*0.65+0.7*0.8*0.3+0.25*0*0.05}=\frac{0.168}{0.259}=0.649$$

→ Η πιθανότητα ο Y να έχει την ασθένεια h3 δεδομένου ότι παρατηρούνται και τα δυο συμπτώματα είναι:

$$P(h3/e1e2)=\frac{P(e1/h3)P(e2/h3)P(h3)}{P(e1/h1)P(e2/h1)P(h1)+P(e1/h2)P(e2/h2)P(h2)+P(e1/h3)P(e2/h3)P(h3)}$$

$$=\frac{0.25*0*0.05}{0.2*0.7*0.65+0.7*0.8*0.3+0.25*0*0.05}=\frac{0}{0.259}=0$$

Άρα το πιο πιθανό είναι ο Y να έχει την ασθένεια h2 (αλλεργία).

3. Δίνονται οι παρακάτω πέντε (5) κανόνες που χρησιμοποιούν συντελεστές βεβαιότητας και αποτελούν τη βάση κανόνων ενός συστήματος

R1
if shape is long
then fruit is banana (0.5)

R2
if shape is long
and color is yellow
then fruit is banana (0.8)

R3
if shape is long
and color is green
then fruit is pear (0.6)

R4
if shape is long
and connection is bunch
then fruit is banana (0.9)

R5
if shape is long
and color is green
and width is uniform
then fruit is pear (0.6)

Δίνονται από τον χρήστη τα παρακάτω δεδομένα διαδοχικά (μέσα σε παρένθεση οι συντελεστές βεβαιότητας που δίνει ο χρήστης): “shape is long (0.7)”, “color is green (0.85)”, “connection is bunch” και “width is uniform (0.6)”. Ποιο είναι το τελικό συμπέρασμα; Καταγράψτε τα βήματα εξαγωγής του συμπεράσματος.

Μετά την πρώτη στιχομυθία (shape is long (0.7), έχουμε:
R1→ cfR1 = 0.7 x 0.5 = 0.35
Και στην μνήμη έχουμε την πρόταση: **{fruit is banana (0.35)}**

Μετά τη δεύτερη στιχομυθία (color is green (0.85)), έχουμε:
R3→ cf1 = 0.7, cf2 = 0.85, cfh = 0.6
cfe = min{0.7, 0.85} = 0.7
cfR3 = cfe x cfh = 0.7 x 0.6 = 0.42
Και στην μνήμη έχουμε τις προτάσεις: **{fruit is banana (0.35)}**
{fruit is pear (0.42)}

Μετά την τρίτη στιχομυθία (connection is bunch (1.0)), έχουμε:
R4→ cf1 = 0.7, cf2 = 1.0, cfh = 0.9

$cfe = \min\{0.7, 1.0\} = 0.7$

$cf_{new} = cfe \times cfh = 0.7 \times 0.9 = 0.63$

Αλλά επειδή υπάρχει ήδη πρόταση “fruit is banana” στην μνήμη έχουμε:

$cfR4 = cf_{old} + cf_{new}(1 - cf_{old}) = 0.35 + 0.63(1\text{-}0.35) = 0.7595$

Και στην μνήμη έχουμε τις προτάσεις: **{fruit is banana (0.7595)}**

{fruit is pear (0.42)}

Μετά την τέταρτη στιχομυθία (width is uniform (0.6)), έχουμε:

R5→ $cf1 = 0.7$, $cf2 = 0.85$, $cf3 = 0.6$, $cfh = 0.6$

$cfe = \min\{0.7, 0.85, 0.6\} = 0.6$

$cf_{new} = cfe \times cfh = 0.6 \times 0.6 = 0.36$

Αλλά επειδή υπάρχει ήδη πρόταση “ fruit is pear” στην μνήμη έχουμε:

$cfR5 = cf_{old} + cf_{new}(1 - cf_{old}) = 0.42 + 0.36(1\text{-}0.42) = 0.6288$

Και στην μνήμη έχουμε τις προτάσεις: **{fruit is banana (0.7595)}**

{ fruit is pear (0.6288)}

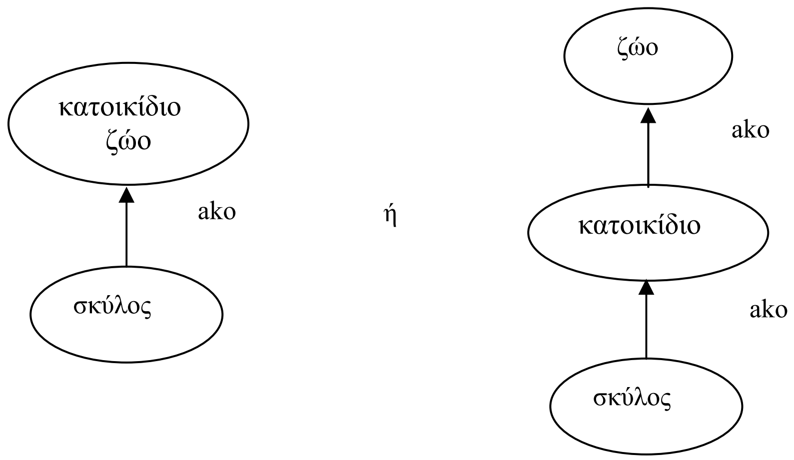
Άρα το τελικό συμπέρασμα είναι : “**fruit is banana**” με βεβαιότητα **0.7595**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
2007-2008
6η Σειρά Ασκήσεων
ΑΠΑΝΤΗΣΕΙΣ

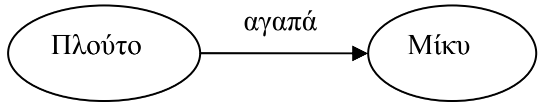
1. Για κάθε μια από τις παρακάτω προτάσεις σχεδιάστε (τμήμα από) το σημαντικό δίκτυο που την αναπαριστά:
(α) «Οι σκύλοι είναι ένα είδος κατοικίδιων ζώων.»
(β) «Ο Πλούτο αγαπά τον Μίκυ»
(γ) «Ο Μίκυ είναι αφεντικό του Πλούτο»
(δ) «Ο Γιάννης έχει βάρος 65 κιλά»

(α)

Σαν τμήμα ιεραρχικού δικτύου μπορεί να παρασταθεί ως εξής:

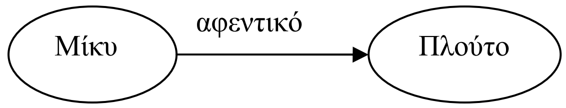


(β)
Σαν τμήμα γενικού δικτύου ή ιεραρχικού δικτύου μπορεί να παρασταθεί ως εξής (στην περίπτωση του ιεραρχικού δικτύου το «αγαπά» αποτελεί σύνδεσμο ιδιότητας):

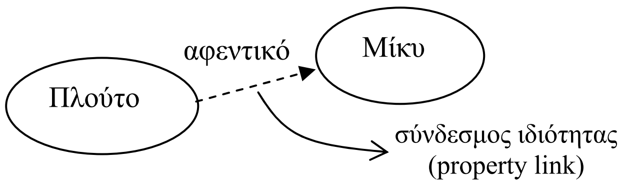


ενώ σαν τμήμα προτασιακού δικτύου ως

(γ)
Σαν τμήμα γενικού δικτύου μπορεί να παρασταθεί ως

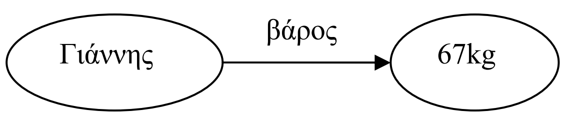


Σαν τμήμα ιεραρχικού δικτύου μπορεί να παρασταθεί ως



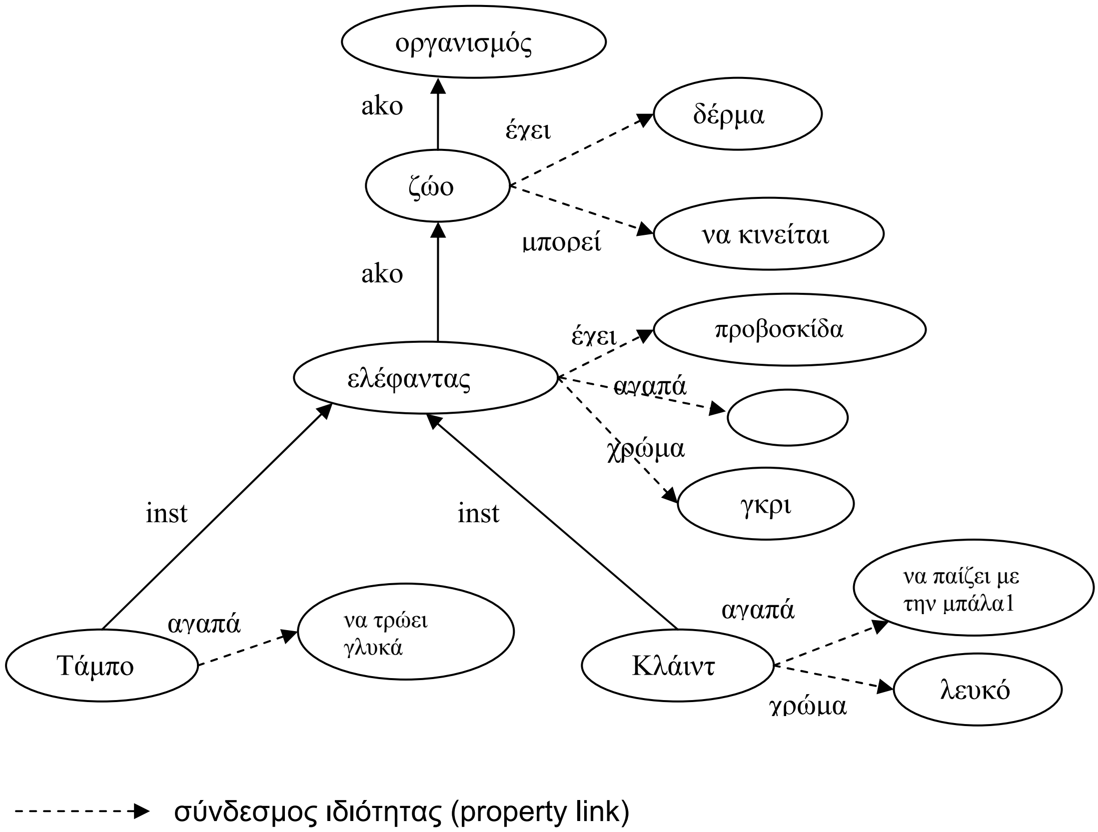
Σαν τμήμα προτασιακού δικτύου μπορεί να παρασταθεί ως

(δ)
Σαν τμήμα γενικού ή ιεραρχικού δικτύου μπορεί να παρασταθεί ως εξής (στην περίπτωση του ιεραρχικού δικτύου το «βάρος» αποτελεί σύνδεσμο ιδιότητας):



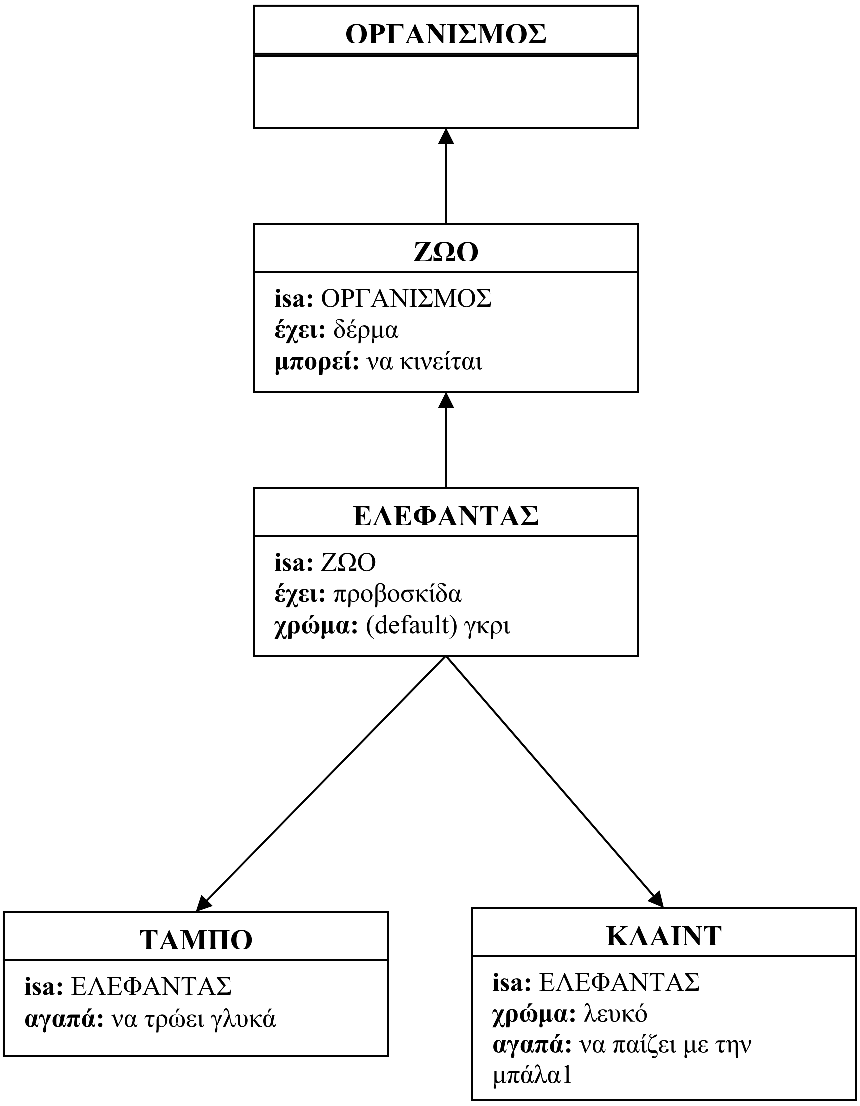
2. Δίνεται η εξής περιγραφή: «Τα ζώα είναι οργανισμοί που έχουν δέρμα και μπορούν να κινούνται. Οι ελέφαντες είναι ζώα που διαθέτουν προβοσκίδα και είναι συνήθως χρώματος γκρι. Ο Τάμπο και ο Κλάιντ είναι ελέφαντες. Όμως, ο Κλάιντ είναι λευκός ελέφαντας. Ο Τάμπο αγαπά να τρώει γλυκά, ενώ ο Κλάιντ να παίζει με μια συγκεκριμένη μπάλλα.»

(α) Να αναπαρασταθεί η παραπάνω γνώση με ένα σημαντικό δίκτυο.



Ο σύνδεσμος ιδιότητας «αγαπά» εισάγεται στον κόμβο-κλάση «ελέφαντας» με κάποια άγνωστη τιμή για να είναι δυνατή η ύπαρξή της ιδιότητας «αγαπά» στους κόμβους –στιγμιότυπα «Τάμπο» και «Κλάιντ».

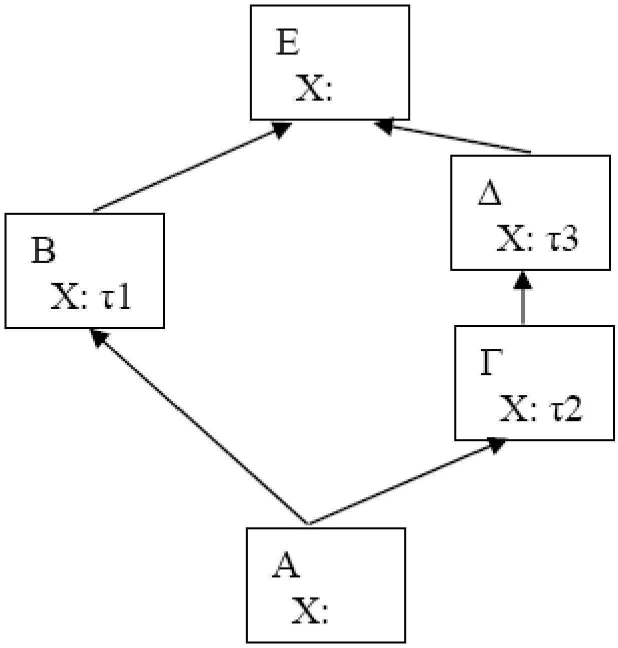
(β) Να αναπαρασταθεί η παραπάνω γνώση με πλαίσια.



(γ) Εξηγείστε πώς η αναπαράσταση εξασφαλίζει την ιδιαιτερότητα του χρώματος του Κλάιντ.

Όταν γίνει ερώτημα για το «χρώμα» στον κόμβο (ή πλαίσιο) «Κλάιντ» τότε επειδή συναντάται τοπικά τιμή για την ιδιότητα αυτή, επιστρέφει «λευκό». Αντίθετα, για τον «Τάμπο», επειδή δεν υπάρχει τοπικά τιμή, κληρονομεί την τιμή για την ιδιότητα «χρώμα» από την κλάση «ελέφαντας», οπότε επιστρέφει «γκρι».

3. Στην παρακάτω ιεραρχία πλαισίων να εξηγήσετε τι απάντηση θα δώσει ο Βασισμένος στην Απόσταση Συλλογισμού αλγόριθμος στην ερώτηση για την τιμή του χαρακτηριστικού Χ του πλαισίου Α.



Ζητούμενο: Τιμή του χαρακτηριστικού Χ στο πλαίσιο Α.

Αλγόριθμος Βασισμένος στη Συλλογιστική Απόσταση

1. Εφάρμοσε αναζήτηση κατά πλάτος (ή βάθος) ακολουθώντας όλες τις δυνατές διαδρομές από το Α προς τα πάνω και αποθήκευσε στη λίστα VALUES όλες τις τιμές που θα βρεις για το Χ.
   Άρα τελικά: **VALUES = {τ2, τ3, τ1}**

2. Για κάθε τιμή στη VALUES εξέτασε αν υπάρχει άλλη τιμή που προέρχεται από πλαίσιο που βρίσκεται σε μικρότερη συλλογιστική απόσταση από το Α. Αν υπάρχει, διάγραψε την τιμή.

   Να θυμήσουμε στο σημείο αυτό ότι η απόσταση ενός πλαισίου F1 από ένα πλαίσιο F2 είναι μικρότερη από την απόστασή του από το πλαίσιο F3 αν και μόνο αν υπάρχει διαδρομή συλλογισμού από το F1 στο F3 δια μέσου του F2.

   Μπορούμε δηλαδή να συγκρίνουμε την συλλογιστική απόσταση δύο πλαισίων από ένα τρίτο πλαίσιο μόνο όταν πρόκειται για πλαίσια που βρίσκονται στο ίδιο μονοπάτι.

   Για το συγκεκριμένο παράδειγμα: Η τιμή της τ3 διαγράφεται απ' την λίστα VALUES διότι η τ2 προέρχεται από πλαίσιο που βρίσκεται σε μικρότερη συλλογιστική απόσταση από το Α.

3. Αν απομείνουν 0 τιμές, τότε δεν υπάρχει απάντηση. Αν απομείνει μια (1) τιμή είναι η απάντηση. Αν απομείνουν περισσότερες από μία, τότε υπάρχει αντίφαση.

   Εδώ τελικά: **Χ = {τ2, τ1}** → **Αντίφαση**