



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Λογικός Προγραμματισμός

Ασκήσεις

Παναγιώτης Σταματόπουλος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Περιεχόμενα

1.	Β' Ομάδα Ασκήσεων "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2007-08.....	3
1.1	Άσκηση 5.....	3
1.2	Άσκηση 6.....	4
1.3	Άσκηση 7.....	6

1. Β' Ομάδα Ασκήσεων "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2007-08

Οι ασκήσεις της ομάδας αυτής πρέπει να αντιμετωπισθούν με τη βοήθεια της τεχνολογίας του λογικού προγραμματισμού με περιορισμούς. Ένα σύστημα λογικού προγραμματισμού που υποστηρίζει την τεχνολογία αυτή είναι η **ECLIPSE**^e. Μπορείτε να χρησιμοποιήσετε είτε την παλιότερη βιβλιοθήκη **fd** είτε την νεώτερη **ic**. Η βιβλιοθήκη **fd** τεκμηριώνεται στο κεφάλαιο 2 του "Obsolete Libraries Manual" και η **ic** στα κεφάλαια 3 και 4 του "Constraint Library Manual". Αν χρησιμοποιήσετε την **ic**, θα σας χρειαστεί και η βιβλιοθήκη **branch_and_bound**, ιδιαιτέρως το κατηγορημα `bb_min/3`, το οποίο τεκμηριώνεται, όπως και όλα τα κατηγορήματα που παρέχει η **ECLIPSE**^e, στο "Alphabetical Predicate Index".

1.1 Άσκηση 5

Γνωρίζετε το πρόβλημα των 8 βασιλισσών, το οποίο συνίσταται στην τοποθέτηση 8 βασιλισσών σε μία σκακιέρα (πλαίσιο 8x8) με τέτοιο τρόπο ώστε να μην απειλούνται μεταξύ τους, με βάση τους κανόνες που ισχύουν στο σκάκι. Αυτό σημαίνει ότι δεν πρέπει να υπάρχουν δύο βασίλισσες στην ίδια γραμμή ή στην ίδια στήλη ή στην ίδια διαγώνιο. Φυσικά, το πρόβλημα αυτό γενικεύεται και για N βασίλισσες που πρέπει να τοποθετηθούν σε ένα πλαίσιο $N \times N$. Μία λύση του προβλήματος των 8 βασιλισσών φαίνεται στο παρακάτω σχήμα.

	1	2	3	4	5	6	7	8
1				⊗				
2		⊗						
3								⊗
4						⊗		
5	⊗							
6			⊗					
7					⊗			
8							⊗	

Το πρόβλημα των N βασιλισσών έχει (για $N \geq 4$) περισσότερες από μία λύση. Αν αριθμήσουμε τις γραμμές και τις στήλες από 1 έως N, έστω ότι σε μία λύση οι βασίλισσες έχουν τοποθετηθεί στις θέσεις $(i_1, j_1), (i_2, j_2), (i_3, j_3), \dots, (i_N, j_N)$, όπου i_k και j_k , είναι η γραμμή και η στήλη, αντίστοιχα, στην οποία έχει τοποθετηθεί η k βασίλισσα. Ορίζουμε το **κόστος** αυτής της λύσης να ισούται με το **μέγιστο** της παράστασης

$$|2 \times i_k - j_k|$$

για κάθε k από 1 έως N.

Γράψτε ένα κατηγορημα `queensopt/3`, το οποίο όταν καλείται σαν `queensopt(N, Solution, Cost)`, για δεδομένη διάσταση πλαισίου και πλήθος βασιλισσών N , να επιστρέφει λύση `Solution` ελαχίστου κόστους `Cost` του προβλήματος των N βασιλισσών. Ας θεωρήσουμε ότι μία λύση του προβλήματος αναπαρίσταται σαν μία λίστα, όπου το k -οστό στοιχείο της ισούται με την στήλη στην οποία έχει τοποθετηθεί βασίλισσα στην k γραμμή. Ουσιαστικά, με την αναπαράσταση αυτή έχουμε $i_k = k$ για κάθε k από 1 έως N και ισχύει ότι το k -οστό στοιχείο της λίστας ισούται με j_k . Κάποιες ενδεικτικές εκτελέσεις είναι οι εξής:

```
?- queensopt(8, Solution, Cost).
```

```
Solution = [4, 2, 8, 6, 1, 3, 5, 7]
```

```
Cost = 9
```

```
?- queensopt(13, Solution, Cost).
```

```
Solution = [2, 4, 6, 8, 10, 12, 1, 3, 5, 7, 9, 11, 13]
```

```
Cost = 13
```

```
?- queensopt(18, Solution, Cost).
```

```
Solution = [7, 4, 6, 11, 18, 14, 12, 2, 9, 1, 3, 5, 8, 10,
            16, 13, 15, 17]
```

```
Cost = 19
```

```
?- queensopt(23, Solution, Cost).
```

```
Solution = [9, 6, 12, 14, 16, 8, 10, 4, 2, 15, 18, 1, 3, 5, 7,
            22, 11, 13, 20, 17, 19, 21, 23]
```

```
Cost = 23
```

Η λύση που δίνεται για $N = 8$ είναι αυτή του σχήματος της προηγούμενης σελίδας. Μέχρι ποια τιμή του N το πρόγραμμά σας βρίσκει βέλτιστη λύση σε λογικά χρονικά πλαίσια;

Παραδοτέο για την άσκηση είναι ένα αρχείο που περιλαμβάνει τον πηγαίο κώδικα `Prolog` που θα γράψετε.

1.2 Άσκηση 6

Μία εταιρεία επιθυμεί να κατασκευάσει έναν αριθμό αποθηκών, από τις οποίες θα μπορεί να τροφοδοτεί τους πελάτες της με τα προϊόντα που παράγει. Από κάποια αρχική μελέτη που έγινε, προέκυψαν διάφορες υποψήφιας θέσεις κατασκευής αποθηκών, έστω πλήθους N . Το κόστος κατασκευής αποθήκης σε κάποια από τις υποψήφιας θέσεις είναι δεδομένο και ισούται με F_i , για κάθε i από 1 μέχρι N . Έστω ότι η εταιρεία προμηθεύει M πελάτες. Το κόστος εξυπηρέτησης του πελάτη j

από την αποθήκη i , εφ' όσον αυτή κατασκευασθεί τελικά, είναι δεδομένο και ισούται με C_{ij} , για κάθε i από 1 μέχρι N και κάθε j από 1 μέχρι M .

Το ζητούμενο του προβλήματος είναι να αποφασισθεί σε ποιες από τις υποψήφιες θέσεις πρέπει η εταιρεία να κατασκευάσει αποθήκες και από ποια αποθήκη θα εξυπηρετείται κάθε πελάτης, έτσι ώστε το συνολικό κόστος κατασκευής των αποθηκών και εξυπηρέτησης των πελατών να είναι το ελάχιστο δυνατό για την εταιρεία.

Δεδομένα για το πρόβλημα, με $N = 20$ υποψήφιες θέσεις για αποθήκες και $M = 40$ πελάτες, βρίσκονται στο http://www.di.uoa.gr/~takis/warehouses_data.pl. Στο αρχείο αυτό, με το κατηγορήμα `fixedcosts/1` δίνονται, στη μορφή λίστας, τα κόστη κατασκευής των αποθηκών στις διάφορες υποψήφιες θέσεις και με το κατηγορήμα `varcosts/1` δίνονται τα κόστη εξυπηρέτησης των πελατών, σαν μία λίστα από λίστες. Κάθε εσωτερική λίστα αντιστοιχεί σε έναν πελάτη, τα στοιχεία της οποίας είναι τα κόστη εξυπηρέτησης του πελάτη από όλες τις υποψήφιες θέσεις αποθηκών.

Γράψτε ένα κατηγορήμα `warehouses/5`, το οποίο να καλείται σαν `warehouses (N1, M1, YesNoLocs, CustSrvs, Cost)`, και λαμβάνοντας υπόψη μόνο τις $N1$ πρώτες υποψήφιες θέσεις και τους $M1$ πρώτους πελάτες από τα δεδομένα του αρχείου που δόθηκε, να βρίσκει λύση ελαχίστου κόστους του προβλήματος, επιστρέφοντας στην μεταβλητή `YesNoLocs` μία λίστα (μήκους $N1$) από 1 και 0, όπου το 1 σημαίνει ότι η αντίστοιχη αποθήκη θα κατασκευασθεί, ενώ 0 ότι δεν θα κατασκευασθεί, στην μεταβλητή `CustSrvs` μία λίστα (μήκους $M1$) από τους αριθμούς αποθηκών από τις οποίες εξυπηρετούνται οι πελάτες και στην μεταβλητή `Cost` το συνολικό κόστος της λύσης. Αν κάποιο από τα $N1$ ή/και $M1$ δοθεί ίσο με 0, τότε να ληφθούν υπόψη όλες οι υποψήφιες θέσεις ή/και όλοι οι πελάτες, αντίστοιχα, που δίνονται στο αρχείο. Κάποιες ενδεικτικές εκτελέσεις είναι οι εξής:

?- `warehouses(5, 10, YesNoLocs, CustSrvs, Cost)`.

`YesNoLocs = [0, 1, 1, 0, 1]`

`CustSrvs = [5, 5, 5, 3, 5, 5, 2, 3, 5, 5]`

`Cost = 105452`

?- `warehouses(15, 20, YesNoLocs, CustSrvs, Cost)`.

`YesNoLocs = [0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]`

`CustSrvs = [8, 7, 5, 6, 8, 5, 2, 6, 8, 8, 4, 5, 6, 5, 7,
8, 4, 9, 4, 7]`

`Cost = 192820`

?- `warehouses(15, 30, YesNoLocs, CustSrvs, Cost)`.

`YesNoLocs = [1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0]`

`CustSrvs = [12, 12, 1, 6, 12, 1, 2, 6, 12, 12, 4, 5, 6, 1, 7,
12, 4, 9, 4, 7, 4, 7, 5, 1, 12, 5, 13, 5, 5, 1]`

Cost = 385634

?- warehouses(0, 0, YesNoLocs, CustServes, Cost).

YesNoLocs = [1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
1, 1, 0, 0]

CustServes = [8, 12, 1, 18, 8, 1, 2, 6, 8, 8, 4, 5, 6, 1, 12,
8, 4, 9, 4, 12, 4, 16, 5, 1, 12, 5, 13, 5, 5,
1, 16, 5, 16, 17, 12, 12, 18, 6, 8, 18]

Cost = 643420

Παραδοτέο για την άσκηση είναι ένα αρχείο που περιλαμβάνει τον πηγαίο κώδικα Prolog που θα γράψετε.

1.3 Άσκηση 7

Θεωρούμε έναν ορθογώνιο αγρό δεδομένων διαστάσεων, έστω $N \times M$. Σε συγκεκριμένες θέσεις (i,j) του αγρού, με το i από 1 έως N και το j από 1 έως M , υπάρχουν K δέντρα. Θέλουμε να τοποθετήσουμε στον αγρό ένα πλήθος από τέντες σε θέσεις που πρέπει να βρεθούν έτσι ώστε:

- Σε τουλάχιστον μία από τις γειτονικές θέσεις κάθε δέντρου, οριζόντια, κάθετα ή διαγώνια, να υπάρχει τέντα.
- Δύο τέντες δεν πρέπει να βρίσκονται σε γειτονικές θέσεις, ούτε οριζόντια, ούτε κάθετα, ούτε διαγώνια.
- Δεν μπορεί να υπάρχει τέντα σε θέση που υπάρχει δέντρο.
- Για κάποιες, όχι απαραίτητα όλες, από τις γραμμές και τις στήλες του αγρού δίνονται μέγιστοι αριθμοί τεντών που μπορεί να υπάρχουν στη γραμμή ή τη στήλη, αντίστοιχα.
- Οι τέντες που θα τοποθετηθούν να είναι οι ελάχιστες δυνατές.

Παρακάτω, στο σχήμα αριστερά, φαίνεται ένας αγρός με $N = 5$ και $M = 5$, στον οποίο βρίσκονται $K = 5$ δέντρα, που σημειώνονται με το σύμβολο Υ , στις θέσεις $(1,2)$, $(2,5)$, $(3,3)$, $(5,1)$ και $(5,5)$. Επίσης, δίνονται οι περιορισμοί ότι στην 1η και στην 4η γραμμή πρέπει να υπάρχουν έως 0 και έως 3 τέντες, αντίστοιχα, και ότι σε καθεμία από την 1η, την 2η και την 5η στήλη πρέπει να υπάρχει έως 1 τέντα. Στο σχήμα δεξιά φαίνεται μία λύση του προβλήματος, με τον ελάχιστο δυνατό αριθμό τεντών, οι θέσεις των οποίων σημειώνονται με το σύμβολο Δ .

	1	2	3	4	5	
1		Υ				0
2					Υ	
3			Υ			
4						3
5	Υ				Υ	

	1	2	3	4	5	
1		Υ				0
2			Δ		Υ	
3			Υ		Δ	
4		Δ				3
5	Υ			Δ	Υ	

1 1 1

1 1 1

Γράψτε ένα κατηγορημα `tents/4`, το οποίο να καλείται σαν `tents(RowTents, ColumnTents, Trees, Tents)`, όπου `RowTents` είναι μία λίστα με τα επιθυμητά συνολικά μέγιστα πλήθη τεντών ανά γραμμή, `ColumnTents` είναι μία λίστα με τα επιθυμητά συνολικά μέγιστα πλήθη τεντών ανά στήλη και `Trees` είναι μία λίστα από συντεταγμένες της μορφής `Row-Column`, στις οποίες βρίσκονται τα δέντρα. Το πλήθος των γραμμών `N` του αγρού ισούται με το μήκος της λίστας `RowTents`, το πλήθος των στηλών `M` ισούται με το μήκος της λίστας `ColumnTents` και το πλήθος των δέντρων `K` ισούται με το μήκος της λίστας `Trees`. Αν δεν θέλουμε να δώσουμε περιορισμό μεγίστου πλήθους για τις τέντες σε κάποια γραμμή ή κάποια στήλη, αρκεί στο αντίστοιχο στοιχείο της λίστας `RowTents` ή `ColumnTents`, αντίστοιχα, να βάλουμε κάποιο αρνητικό αριθμό. Το κατηγορημα που θα γράψετε να επιστρέφει στη μεταβλητή `Tents` μία λίστα με τις συντεταγμένες των θέσεων στις οποίες πρέπει να τοποθετηθούν οι τέντες, ώστε να ισχύουν οι περιορισμοί που έχουν τεθεί. Αν το πρόβλημα έχει περισσότερες από μία λύση με το ελάχιστο δυνατό πλήθος τεντών, να βρίσκονται όλες μέσω οπισθοδρόμησης. Κάποιες ενδεικτικές εκτελέσεις είναι οι εξής:

```
?- tents([0, -1, -1, 3, -1], [1, 1, -1, -1, 1],
        [1 - 2, 2 - 5, 3 - 3, 5 - 1, 5 - 5], Tents).
```

```
Tents = [2 - 3, 3 - 5, 5 - 2, 5 - 4] --> ;
```

```
Tents = ..... --> ;
```

```
Tents = ..... --> ;
```

```
.....
```

```
?- tents([-1, -1, -1, 2, -1, -1, 2, 1],
        [2, 1, -1, 1, 1, -1, 1, -1, -1, 1, 2, -1],
        [1 - 4, 1 - 9, 1 - 12, 2 - 1, 2 - 5, 2 - 8, 3 - 1,
         3 - 6, 3 - 8, 3 - 12, 4 - 5, 4 - 7, 4 - 11, 5 - 3,
         5 - 9, 6 - 1, 6 - 7, 6 - 11, 7 - 5, 8 - 10], Tents).
```

```
Tents = [2 - 4, 2 - 9, 2 - 12, 3 - 2, 4 - 6, 5 - 10,
        6 - 3, 7 - 1, 7 - 6, 8 - 11] --> ;
```

```
Tents = ..... --> ;
```

```
Tents = ..... --> ;
```

```
.....
```

Παραδοτέο για την άσκηση είναι ένα αρχείο που περιλαμβάνει τον πηγαίο κώδικα `Prolog` που θα γράψετε.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Παναγιώτης Σταματόπουλος.
«Λογικός Προγραμματισμός, Η γλώσσα προγραμματισμού Prolog». Έκδοση: 1.0. Αθήνα 2015.
Διαθέσιμο από τη δικτυακή διεύθυνση: <http://opencourses.uoa.gr/courses/DI117/>.

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων

- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει) μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» 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.

