



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Λογικός Προγραμματισμός

Ασκήσεις

Παναγιώτης Σταματόπουλος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Περιεχόμενα

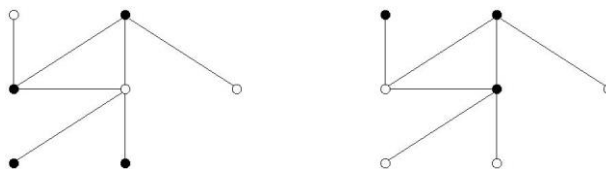
1.	Β' Ομάδα Ασκήσεων "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2008-09.....	3
1.1	Άσκηση 5.....	3
1.2	Άσκηση 6.....	4
1.3	Άσκηση 7.....	5

1. Β' Ομάδα Ασκήσεων "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2008-09

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 ECLiPS^e. Μπορείτε να χρησιμοποιήσετε είτε την παλιότερη βιβλιοθήκη `fd` είτε την νεότερη `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 κατηγορήμα `bb_min/3`, το οποίο τεκμηριώνεται, όπως και όλα τα κατηγορήματα που παρέχει η ECLiPS^e, στο "Alphabetical Predicate Index".

1.1 Άσκηση 5

Στη Θεωρητική Πληροφορική, το πρόβλημα της βέλτιστης κάλυψης κορυφών (optimal vertex cover) ενός γράφου συνίσταται στην εύρεση ενός συνόλου κορυφών, ελαχίστου πλήθους, τέτοιου ώστε για κάθε ακμή του γράφου, τουλάχιστον μία από τις κορυφές-άκρα της να ανήκει στο σύνολο αυτό. Στο σχήμα αριστερά φαίνεται μία κάλυψη κορυφών του δεδομένου γράφου (μαύροι κόμβοι), η οποία όμως δεν είναι βέλτιστη. Περιλαμβάνει τέσσερις κόμβους, ενώ στο δεξιό σχήμα η κάλυψη κορυφών είναι βέλτιστη (με τρεις κόμβους), αφού δεν είναι δυνατόν να υπάρξει κάποια με δύο κόμβους.



Για την αντιμετώπιση του προβλήματος αυτού, πρέπει να χρησιμοποιήσετε γράφους που κατασκευάζονται μέσω του κατηγορήματος `create_graph(N, D, G)`, όπως αυτό ορίζεται στο αρχείο <http://www.di.uoa.gr/~takis/graph.pl>. Κατά την κλήση του κατηγορήματος δίνεται το πλήθος N των κόμβων του γράφου και η πυκνότητά του D (σαν ποσοστό των ακμών που υπάρχουν στον γράφο σε σχέση με όλες τις δυνατές ακμές) και επιστρέφεται ο γράφος G σαν μία λίστα από ακμές της μορφής $N1-N2$, όπου $N1$ και $N2$ είναι οι δύο κόμβοι της ακμής (οι κόμβοι του γράφου παριστάνονται σαν ακέραιοι από το 1 έως το N). Ένα παράδειγμα εκτέλεσης του κατηγορήματος αυτού είναι το εξής:

```
?- seed(1), create_graph(9, 30, G).
```

```
G = [1 - 5, 2 - 4, 2 - 6, 3 - 4, 3 - 6, 3 - 9, 4 - 7, 5 - 7,  
     6 - 7, 6 - 8, 6 - 9]
```

Για την άσκηση αυτή θα πρέπει να ορίσετε ένα κατηγορήμα `vertexcover/3`, το οποίο όταν καλείται σαν `vertexcover(N, D, C)`, αφού δημιουργήσει ένα γράφο N κόμβων και πυκνότητας D , καλώντας το κατηγορήμα `create_graph/3`, να επιστρέφει στο C μία λίστα από κόμβους που

αποτελούν μία βέλτιστη κάλυψη κορυφών του γράφου. Κάποια παραδείγματα εκτέλεσης είναι τα εξής¹:

```
?- seed(2009), vertexcover(9, 30, C).
```

```
C = [2, 4, 6, 8, 9]
```

```
?- seed(1000), vertexcover(17, 50, C).
```

```
C = [1, 2, 3, 4, 5, 6, 8, 9, 11, 15, 16, 17]
```

```
?- seed(12345), vertexcover(33, 65, C).
```

```
C = [1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18,  
19, 20, 21, 23, 26, 27, 28, 29, 30, 31, 32, 33]
```

```
?- seed(1), vertexcover(100, 70, C), length(C, L).
```

```
C = [2, 3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, .....
```

```
L = 94
```

```
?- seed(100), vertexcover(100, 35, C), length(C, L).
```

```
C = [1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  
19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, .....
```

```
L = 87
```

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

1.2 Άσκηση 6

Το πρόβλημα της πρόβας ορχήστρας συνίσταται στην εύρεση της σειράς με την οποία πρέπει να γίνει η πρόβα των διαφόρων μερών ενός κονσέρτου που πρόκειται να δώσει μία ορχήστρα, έτσι ώστε να ελαχιστοποιηθεί ο συνολικός χρόνος αναμονής των μουσικών της ορχήστρας. Κάθε μουσικός

¹ Σε μηχάνημα Linux του εργαστηρίου του Τμήματος.

συμμετέχει σε συγκεκριμένα μέρη του κονσέρτου και αφού αποφασισθεί η σειρά με την οποία θα γίνει η πρόβα των μερών, ο μουσικός οφείλει να είναι παρών όταν αρχίζει η πρόβα του πρώτου μέρους στο οποίο συμμετέχει και αποδεσμεύεται όταν τελειώσει η πρόβα του τελευταίου μέρους στο οποίο συμμετέχει. Οι μουσικοί αμείβονται ανάλογα με τον χρόνο που τους ζητείται να είναι παρόντες στην πρόβα, πράγμα που σημαίνει ότι ενδιαφερόμαστε να ελαχιστοποιήσουμε, όσο είναι δυνατόν, τα χρονικά διαστήματα που οι μουσικοί είναι παρόντες, αλλά γίνονται πρόβες μερών στα οποία δεν συμμετέχουν. Επίσης, για κάθε μέρος του κονσέρτου, είναι γνωστή η διάρκειά του. Δείτε ένα παράδειγμα στον παρακάτω πίνακα:

Μέρος	1	2	3	4	5	6	7	8	9
Μουσικός 1	1	1	0	1	0	1	1	0	1
Μουσικός 2	1	1	0	1	1	1	0	1	0
Μουσικός 3	1	1	0	0	0	0	1	1	0
Μουσικός 4	1	0	0	0	1	1	0	0	1
Μουσικός 5	0	0	1	0	1	1	1	1	0
Διάρκεια	2	4	1	3	3	2	5	7	6

Στο παράδειγμα αυτό, το κονσέρτο αποτελείται από 9 μέρη και συμμετέχουν σ' αυτό 5 μουσικοί. Η διάρκεια κάθε μέρους φαίνεται στην τελευταία γραμμή του πίνακα. Τα 1 και 0 στον πίνακα δείχνουν αν ο κάθε μουσικός συμμετέχει ή όχι στο αντίστοιχο μέρος του κονσέρτου. Αν οι πρόβες των μερών γινότουσαν με τη σειρά του πίνακα, τότε οι χρόνοι αναμονής των μουσικών θα ήταν 11 ($=1+3+7$), 6 ($=1+5$), 9 ($=1+3+3+2$), 20 ($=4+1+3+5+7$) και 3, αντίστοιχα, δίνοντας ένα συνολικό χρόνο αναμονής για τους μουσικούς ίσο με 49 ($=11+6+9+20+3$). Υπάρχουν όμως και καλύτερες λύσεις απ' αυτήν για τη σειρά των μερών στην πρόβα. Ορίστε ένα κατηγορημα `rehearsal/2`, το οποίο όταν καλείται σαν `rehearsal(Sequence, WaitTime)` να επιστρέφει στο `Sequence` τη βέλτιστη σειρά πρόβας των μερών ενός κονσέρτου και στο `WaitTime` τον συνολικό χρόνο αναμονής των μουσικών γι' αυτή τη βέλτιστη σειρά. Τα δεδομένα του προβλήματος δίνονται, όπως φαίνεται στο http://www.di.uoa.gr/~takis/rehearsal_data.pl, με τα γεγονότα `players/1` και `durations/1`. Μία ενδεικτική εκτέλεση του ζητούμενου κατηγορημάτος είναι η εξής:

```
?- rehearsal(Sequence, WaitTime).
Sequence = [3, 8, 2, 7, 1, 6, 5, 4, 9]
WaitTime = 17
```

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

1.3 Άσκηση 7

Έστω ότι έχουμε N άντρες και N γυναίκες και το ζητούμενο είναι να τους παντρέψουμε με τέτοιο τρόπο ώστε να προκύψουν N σταθεροί γάμοι. Κάθε άντρας προτιμά σε διαφορετικό βαθμό τις υποψήφιες συζύγους του και κάθε γυναίκα επίσης έχει διαβαθμίσει τους υποψήφιους συζύγους της. Ένας γάμος A - Γ είναι σταθερός όταν για κάθε γυναίκα Γ_1 που ο άντρας A προτιμά περισσότερο από τη σύζυγό του Γ αυτή προτιμά το σύζυγό της A_1 περισσότερο από τον A και για κάθε άντρα A_2 που η γυναίκα Γ προτιμά περισσότερο από το σύζυγό της A αυτός προτιμά τη σύζυγό του Γ_2 περισσότερο από τη Γ . Αν δουλέψετε με τη βιβλιοθήκη `fd` της `ECLiPSe`, δεδομένα θα βρείτε στο http://www.di.uoa.gr/~takis/stable_data.pl (για την `ic`, στο

http://www.di.uoa.gr/~takis/stableic_data.pl). Ορίστε ένα κατηγορημα `stable/1`, το οποίο να επιστρέφει μία λύση του προβλήματος, και μέσω οπισθοδρόμησης όλες τις λύσεις τελικά, σαν μία λίστα από γάμους Α-Γ. Έστω ότι τα δεδομένα (από το http://www.di.uoa.gr/~takis/stable_data.pl) για το πρόβλημα είναι τα εξής:

```
men([rick,jack,john,hugh,greg,nick,bill,andy,alan,dick]).
```

```
women([helen,tracy,linda,sally,wanda,maria,diana,patty,jenny,lilly]).
```

```
prefers(rick,[tracy,linda,jenny,maria,wanda,sally,diana,patty,helen,lilly]).
```

```
prefers(jack,[tracy,lilly,patty,sally,diana,linda,jenny,helen,maria,wanda]).
```

```
prefers(john,[diana,jenny,wanda,linda,patty,tracy,sally,helen,lilly,maria]).
```

```
prefers(hugh,[helen,wanda,diana,maria,sally,patty,linda,lilly,jenny,tracy]).
```

```
prefers(greg,[jenny,maria,lilly,patty,sally,linda,tracy,diana,helen,wanda]).
```

```
prefers(nick,[sally,linda,diana,maria,jenny,lilly,wanda,helen,patty,tracy]).
```

```
prefers(bill,[linda,tracy,diana,patty,lilly,sally,jenny,helen,wanda,maria]).
```

```
prefers(andy,[helen,jenny,lilly,maria,sally,patty,wanda,tracy,diana,linda]).
```

```
prefers(alan,[patty,sally,maria,jenny,linda,diana,helen,lilly,tracy,wanda]).
```

```
prefers(dick,[sally,wanda,helen,maria,lilly,diana,jenny,tracy,patty,linda]).
```

```
prefers(helen,[alan,rick,nick,bill,jack,hugh,dick,john,andy,greg]).
```

```
prefers(tracy,[andy,john,alan,hugh,bill,rick,greg,dick,jack,nick]).
```

```
prefers(linda,[dick,hugh,john,nick,andy,greg,alan,jack,bill,rick]).
```

```
prefers(sally,[nick,dick,bill,rick,greg,andy,jack,hugh,alan,john]).
```

```
prefers(wanda,[greg,bill,dick,jack,john,nick,alan,andy,rick,hugh]).
```

```
prefers(maria,[john,alan,nick,rick,greg,hugh,jack,bill,andy,dick]).
```

```
prefers(diana,[greg,hugh,andy,nick,john,bill,alan,dick,jack,rick]).
```

```
prefers (patty, [alan, bill, jack, hugh, greg, dick, rick, nick, andy, john]) .
prefers (jenny, [greg, dick, jack, bill, alan, john, andy, hugh, rick, nick]) .
prefers (lilly, [hugh, jack, dick, nick, andy, alan, greg, rick, bill, john]) .
```

Τότε, μία ενδεικτική εκτέλεση του ζητούμενου κατηγορήματος είναι:

```
?- stable(M) .
```

```
M = [rick-helen, jack-lilly, john-maria, hugh-diana, greg-jenny,
      nick-sally, bill-wanda, andy-tracy, alan-patty, dick-linda]
More --> ;
```

```
M = [rick-maria, jack-lilly, john-diana, hugh-helen, greg-jenny,
      nick-sally, bill-linda, andy-tracy, alan-patty, dick-wanda]
More --> ;
```

```
M = [rick-maria, jack-lilly, john-linda, hugh-diana, greg-jenny,
      nick-sally, bill-helen, andy-tracy, alan-patty, dick-wanda]
More --> ;
```

```
M = [rick-tracy, jack-lilly, john-diana, hugh-helen, greg-jenny,
      nick-sally, bill-linda, andy-maria, alan-patty, dick-wanda]
```

Βρείτε μέχρι ποιο μέγεθος δεδομένων εισόδου το πρόγραμμά σας βρίσκει όλες τις λύσεις για το πρόβλημα σε εύλογο χρόνο. Χρησιμοποιήστε το C πρόγραμμα που θα βρείτε στο <http://www.di.uoa.gr/~takis/randstabledata.c>, το οποίο παράγει με τυχαίο τρόπο δεδομένα για το πρόβλημα (τα δεδομένα αυτά είναι κατάλληλα για την βιβλιοθήκη `fd`, αλλά πολύ εύκολα μπορείτε να επεκτείνετε το πρόγραμμα για να καλυφθεί και η `ic`).

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.

