



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Λογικός Προγραμματισμός

Ασκήσεις

Παναγιώτης Σταματόπουλος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Περιεχόμενα

1.	Ασκήσεις "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2000-01.....	3
1.1	Άσκηση 1 (0.3 μονάδες)	3
1.2	Άσκηση 2 (0.3 μονάδες)	3
1.3	Άσκηση 3 (0.4 μονάδες)	4
1.4	Άσκηση 4 (0.5 μονάδες)	5
1.5	Άσκηση 5 (0.5 μονάδες)	5
1.6	Άσκηση 6 (0.5 μονάδες)	7

1. Ασκήσεις "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2000-01

1.1 Άσκηση 1 (0.3 μονάδες)

Ένας πλήρης γράφος με n κορυφές συμβολίζεται σαν K_n . Γράψτε ένα κατηγορημα `kay4/1` το οποίο όταν καλείται σαν `kay4(M)` να σχεδιάζει το γράφο K_4 σαν "τετράγωνο" πλευράς M με κορυφές τα σημεία a, b, c και d . Για παράδειγμα:

```
?- kay4(6).
```

```
a----b
```

```
| \  / |
```

```
|  \ /  |
```

```
| / \  |
```

```
|/  \ |
```

```
c----d
```

```
yes
```

```
?- kay4(7).
```

```
a-----b
```

```
| \    / |
```

```
|  \ /  |
```

```
|   X   |
```

```
| / \  |
```

```
|/    \ |
```

```
c-----d
```

```
yes
```

1.2 Άσκηση 2 (0.3 μονάδες)

Γράψτε ένα κατηγορημα `exchange/2`, το οποίο να παίρνει σαν πρώτο όρισμα μία λίστα θετικών ακεραίων και να επιστρέφει στο δεύτερο όρισμα τη μέγιστη αποτίμηση αυτής της λίστας, όπως θα εξηγηθεί αυτός ο όρος στη συνέχεια. Η βασική αποτίμηση μίας λίστας ακεραίων είναι η αριθμητική τιμή της έκφρασης που προκύπτει αν εναλλάσσουμε αφαιρέσεις και προσθέσεις μεταξύ των στοιχείων της λίστας, με τη σειρά που είναι αυτά δοσμένα. Για παράδειγμα, η λίστα $[9, 8, 2]$ έχει βασική αποτίμηση $9-8+2=3$, ενώ η $[1, 4, 89, 12]$ έχει βασική αποτίμηση $1-4+89-12=74$.

Οποιαδήποτε άλλη αποτίμηση μίας λίστας είναι η βασική αποτίμηση μίας λίστας που παίρνουμε με εναλλαγή στις θέσεις των αριθμών ενός ή περισσότερων ζευγαριών διαδοχικών στοιχείων με ταυτόχρονη αντιστροφή στη σειρά των ψηφίων αυτών των αριθμών. Δηλαδή οι διαδοχικοί αριθμοί $17, 34$ μπορούν να γίνουν $43, 71$ και οι $123, 45$ να γίνουν $54, 321$. Κανείς, όμως, αριθμός δεν μπορεί να μετακινηθεί περισσότερο από μία θέση μακριά από την αρχική του. Το ζητούμενο από το κατηγορημα `exchange/2` είναι να μας επιστρέφει τη μέγιστη δυνατή αποτίμηση μίας λίστας ακεραίων, από τη βασική της ή από αυτές που προκύπτουν με εναλλαγές στοιχείων, όπως περιγράφηκε προηγουμένως. Κάποια παραδείγματα εκτέλεσης είναι τα εξής:

```
?- exchange([1,2],V).
```

```
V = 1
```

```
?- exchange([12,56,34],V).
```

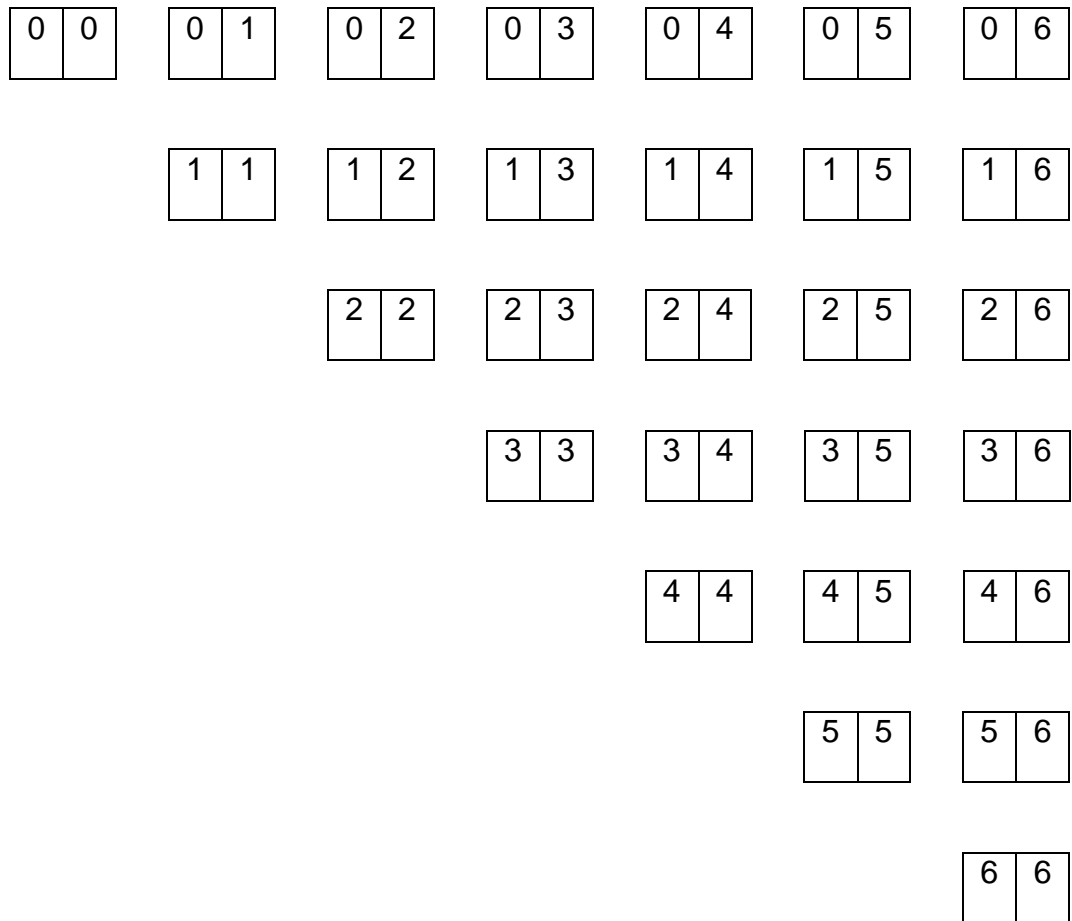
```
V = 78
```

?- exchange ([210, 123, 34, 455, 8, 100], V) .

V = 934

1.3 Άσκηση 3 (0.4 μονάδες)

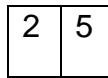
Έστω ότι έχουμε στη διάθεσή μας τις παρακάτω 28 πλάκες "ντόμινο":



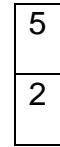
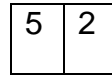
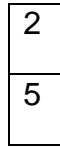
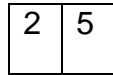
Γράψτε ένα πρόγραμμα Prolog που να μας πληροφορεί πώς πρέπει να τις τοποθετήσουμε σ' ένα πλαίσιο 7x8, έτσι ώστε η τελική διάταξη να είναι η εξής:

3	1	2	6	6	1	2	2
3	4	1	5	3	0	3	6
5	6	6	1	2	4	5	0
5	6	4	1	3	3	0	0
6	1	0	6	3	2	4	0
4	1	5	2	4	3	5	5
4	1	0	2	4	5	2	0

Σημειώστε ότι ένα "ντόμινο" μπορεί να τοποθετηθεί στο πλαίσιο με οποιοδήποτε από τους τέσσερις δυνατούς τρόπους. Δηλαδή το "ντόμινο"



μπορεί να τοποθετηθεί με κάποιον από τους εξής τρόπους:



1.4 Άσκηση 4 (0.5 μονάδες)

Να ορισθεί ένα κατηγορημα `differentiate/2` το οποίο να δέχεται στο πρώτο όρισμά του μία αλγεβρική έκφραση που μπορεί να περιλαμβάνει γινόμενα, αθροίσματα και διαφορές μεταξύ πολυωνύμων μίας μεταβλητής (έστω x) και να βρίσκει την παράγωγο αυτής της έκφρασης, επιστρέφοντας το αποτέλεσμα στο δεύτερο όρισμα σαν το απλούστερο δυνατό πολυώνυμο. Για παράδειγμα:

```
?- differentiate((2*x^2-3*x+2)*(3*x-4)-(4*x^2+18*x-7),P).
P = 18*x^2-42*x
```

```
?- differentiate((2*x-3*x^3)-(3*x-x^2+1)*(-x-1)*2+x,P).
P = -15*x^2+8*x+11
```

1.5 Άσκηση 5 (0.5 μονάδες)

Μία εκδοχή του λεγόμενου job-shop προβλήματος είναι η εξής: Έχουμε να εκτελέσουμε μία σειρά από έργα $j_1, j_2, j_3 \dots$, ανεξάρτητα το ένα από το άλλο, καθένα από τα οποία αποτελείται από έναν αριθμό από εργασίες $t_{11}, t_{12} \dots, t_{21}, t_{22}, \dots, t_{31}, t_{32}, \dots$ οι οποίες πρέπει να εκτελεσθούν με τη δεδομένη σειρά για κάθε έργο. Κάθε εργασία μπορεί να εκτελεσθεί σε συγκεκριμένο τύπο μηχανής και χρειάζεται δεδομένο χρόνο για να έλθει σε πέρας. Επίσης είναι γνωστό το πλήθος των μηχανών που έχουμε διαθέσιμες από κάθε τύπο. Αυτό που θέλουμε να βρούμε είναι πότε και σε ποιες μηχανές πρέπει να εκτελέσουμε τις εργασίες έτσι ώστε όλα τα έργα να έχουν τελειώσει πριν από δεδομένη προθεσμία. Ένα συγκεκριμένο στιγμιότυπο αυτού του προβλήματος δίνεται με τα παρακάτω γεγονότα Prolog:

```
job(j1, [t11, t12]).
job(j2, [t21, t22, t23]).
job(j3, [t31]).
job(j4, [t41, t42]).
```

```
task(t11, m1, 2).
task(t12, m2, 6).
task(t21, m2, 5).
task(t22, m1, 3).
task(t23, m2, 3).
task(t31, m2, 4).
task(t41, m1, 5).
task(t42, m2, 2).
```

```
machine(m1, 1).
machine(m2, 2).
```

```
deadline(14).
```

Τα γεγονότα `job/3` περιγράφουν τα έργα που πρέπει να εκτελέσουμε και τις εργασίες από τις οποίες αποτελούνται, κάθε γεγονός `task/3` δίνει, για μία εργασία, σε τι τύπου μηχανή πρέπει να εκτελεσθεί και πόσες μονάδες χρόνου απαιτούνται για την εκτέλεσή της, τα γεγονότα `machine/2` μας πληροφορούν πόσες μηχανές έχουμε διαθέσιμες από κάθε τύπο και τέλος το κατηγορημα `deadline/1` δίνει την προθεσμία (σε μονάδες χρόνου) μέσα στην οποία πρέπει να έχουν εκτελεσθεί όλα τα έργα. Ορίστε σε Prolog το κατηγορημα `job_shop/1`, το οποίο να επιστρέφει το πρόγραμμα εκτέλεσης των εργασιών στις μηχανές. Παράδειγμα:

```
?- job_shop(S).
S = [execs(m1,[t(t11,0,2),t(t41,2,7),t(t22,7,10)]),
     execs(m2,[t(t12,2,8),t(t42,8,10),t(t23,10,13)]),
     execs(m2,[t(t21,0,5),t(t31,5,9)])] -> ;

S = .....
.....
```

Η παραπάνω απάντηση σημαίνει ότι στη μηχανή τύπου `m1` θα εκτελεστούν από τη χρονική στιγμή 0 έως τη 2 η εργασία `t11`, από τη χρονική στιγμή 2 έως την 7 η εργασία `t41` και από τη χρονική στιγμή 7 έως τη 10 η εργασία `t22`. Μετά ακολουθούν τα προγράμματα εργασίας για τις δύο μηχανές τύπου `m2`.

Μία δεύτερη εκδοχή του `job-shop` προβλήματος είναι αυτή στην οποία για κάθε εργασία έχουμε και ένα δεδομένο πλήθος ατόμων που πρέπει να εργασθούν στη μηχανή που θα τη φέρει σε πέρας. Δηλαδή, τώρα έχουμε το κατηγορημα `task/4` (αντί για `task/3`), όπου το τελευταίο επιπλέον όρισμα δίνει και το πλήθος αυτό των ατόμων. Τα γεγονότα τώρα είναι:

```
task(t11,m1,2,3).
task(t12,m2,6,2).
task(t21,m2,5,2).
task(t22,m1,3,3).
task(t23,m2,3,2).
task(t31,m2,4,2).
task(t41,m1,5,4).
task(t42,m2,2,1).
```

Επίσης έχουμε δεδομένο και το προσωπικό που υπάρχει διαθέσιμο ανά πάσα χρονική στιγμή, το οποίο ορίζεται με το γεγονός `staff/1`, δηλαδή:

```
staff(6).
```

Υποτίθεται ότι κάθε εργαζόμενος μπορεί να δουλέψει σε οποιαδήποτε μηχανή για οποιοδήποτε εργασία.

Ορίστε σε Prolog το κατηγορημα `job_shop_with_manpower/1`, το οποίο να λύνει και αυτήν την εκδοχή του `job-shop` προβλήματος. Παράδειγμα εκτέλεσης:

```
?- job_shop_with_manpower(S).
S = [execs(m1,[t(t11,0,2),t(t41,2,7),t(t22,7,10)]),
     execs(m2,[t(t42,7,9),t(t31,10,14)]),
     execs(m2,[t(t21,0,5),t(t12,5,11),t(t23,11,14)])] -> ;

S = .....
.....
```

1.6 Άσκηση¹ 6 (0.5 μονάδες)

Μία αεροπορική εταιρεία έχει προγραμματίσει να εκτελέσει για μία προκαθορισμένη χρονική περίοδο N πτήσεις, στις οποίες μπορεί να αναφέρεται κανείς με τους κωδικούς $1, 2, 3, \dots, N$. Επιπλέον, μέσω κάποιας μεθόδου που εφάρμοσε, έχει δημιουργήσει M συνδυασμούς πτήσεων P_i ($1 \leq i \leq M$).

Δηλαδή, κάθε P_i περιλαμβάνει κάποιες από τις πτήσεις $1, 2, 3, \dots, N$ και, επίσης, τα P_i δεν είναι κατ' ανάγκη ξένα μεταξύ τους. Οι συνδυασμοί αυτοί είναι έτσι κατασκευασμένοι ώστε να είναι δυνατόν λόγω κανονισμών, συμβάσεων κλπ. να πραγματοποιηθούν ο καθένας, δηλαδή όλες οι πτήσεις που περιλαμβάνει, με έναν από τους διαθέσιμους κυβερνήτες της εταιρείας. Το ζητούμενο είναι να επιλεγούν κάποιοι συνδυασμοί που καλύπτουν ακριβώς τις πτήσεις της εταιρείας, με σκοπό να ανατεθούν σε συγκεκριμένους κυβερνήτες. Δηλαδή, δεν πρέπει ούτε να μείνει πτήση χωρίς κυβερνήτη, ούτε να κάποια πτήση να έχει δύο ή περισσότερους κυβερνήτες. Τέλος, αν ένας συνδυασμός πτήσεων P_i έχει ένα κόστος (έξοδα διανυκτερεύσεων, αποζημιώσεις εκτός έδρας, πληρωμή υπερωριών κλπ.) στην εταιρεία C_i , οι συνδυασμοί που θα επιλεγούν πρέπει να είναι αυτοί που προκαλούν το ελάχιστο συνολικό κόστος. Εφαρμόστε τη μέθοδό σας για εισόδους που προκύπτουν από το `get_flight_data(I, N, P, C)` του προγράμματος `~takis/lp/flight_data.pl` (από το δίκτυο των Suns του τμήματος) δίνοντας στο I έναν αύξοντα αριθμό εισόδου ($1, 2, \dots$) και παίρνοντας στα N, P και C το πλήθος των πτήσεων, μία λίστα συνδυασμών πτήσεων και τη λίστα κοστών των συνδυασμών αυτών, αντίστοιχα.

¹ Να αντιμετωπισθεί με τη βοήθεια της τεχνολογίας του λογικού προγραμματισμού με περιορισμούς, π.χ. στο περιβάλλον της γλώσσας ECLiPS^e.

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.

