ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Λογικός Προγραμματισμός

Ασκήσεις

Παναγιώτης Σταματόπουλος

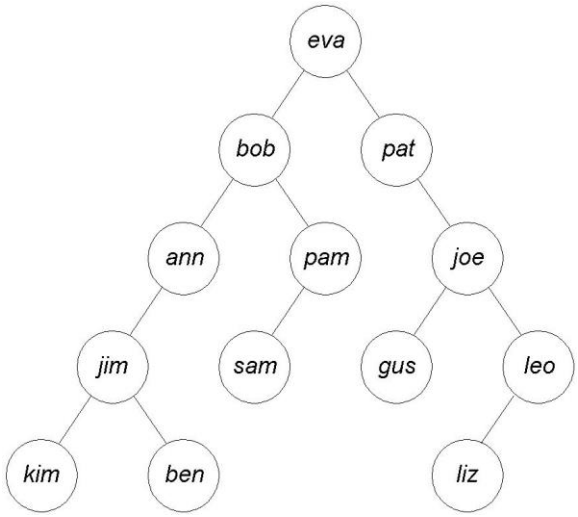
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Περιεχόμενα

1. Α' Ομάδα Ασκήσεων "Λογικού Προγραμματισμού" Ακαδημαϊκού Έτους 2009-10

1.1 Άσκηση 1

Διατυπώστε σε Prolog, με τη βοήθεια ενός κατηγορήματος `parent/2`, το γενεαλογικό δέντρο που φαίνεται στο διπλανό σχήμα. Επίσης, μέσω των κατηγορημάτων `male/1` και `female/1`, διατυπώστε και τη γνώση ότι οι `bob`, `joe`, `jim`, `sam`, `gus`, `leo` και `ben` είναι άνδρες και ότι οι `eva`, `pat`, `ann`, `pam`, `kim` και `liz` είναι γυναίκες. Υλοποιήστε, επίσης, σε Prolog και τα κατηγορήματα `uncle/2` και `common_female_ancestor/3` έτσι ώστε:



- Το `uncle(X,Y)` να αληθεύει όταν ο `X` είναι θείος του/της `Y`.
- Το `common_female_ancestor(X,Y,Z)` να αληθεύει όταν η `Z` είναι γυναίκα κοινή πρόγονος των `X` και `Y`.

Σχεδιάστε το δέντρο ανάλυσης για την ερώτηση "*Ποιος/ποια έχει θείο τον bob;*", έχοντας φροντίσει να διατυπώσετε τους κανόνες σας με τέτοιο τρόπο ώστε το δέντρο ανάλυσης να μην είναι ιδιαίτερα μεγάλο. Σχεδιάστε επίσης το δέντρο ανάλυσης, τουλάχιστον μέχρι την εύρεση της πρώτης λύσης, και για την ερώτηση "*Ποιες είναι οι γυναίκες κοινές πρόγονοι του gus και του joe;*" και, αν επιθυμείτε, και για την εύρεση των υπόλοιπων λύσεων, αν υπάρχουν. Και για την ερώτηση αυτή, φροντίστε οι κανόνες σας να είναι έτσι διατυπωμένοι, ώστε να μην οδηγηθείτε σε τεράστιο δέντρο ανάλυσης.

Πιστεύετε ότι ανάλογα με τη μορφή κάποιας ερώτησης που υποβάλλουμε για το κατηγόρημα `uncle/2` (και τα δύο ορίσματα σταθερές, το πρώτο σταθερά και το δεύτερο μεταβλητή ή αντίστροφα, ή και τα δύο μεταβλητές) ενδέχεται κάποια υλοποίηση του κατηγορήματος να είναι προτιμότερη, όσον αφορά την αποδοτικότητα στον υπολογισμό της απάντησης, από άλλες πιθανές υλοποιήσεις; Ισχύει κάτι παρόμοιο και για το κατηγόρημα `common_female_ancestor/3`;

Παραδοτέο για την άσκηση είναι ένα αρχείο zip, στο οποίο θα περιλαμβάνεται το πηγαίο αρχείο του προγράμματος Prolog που θα γράψετε, αρχεία εικόνας με τα δέντρα ανάλυσης και κάποιο αρχείο κειμένου με τις απαντήσεις σας στις τελευταίες ερωτήσεις κρίσεως.

1.2 Άσκηση 2

Ο Γιάννης αποφάσισε μία ημέρα να επισκεφθεί τα καταστήματα για την αγορά ενός υπολογιστικού συστήματος αποτελούμενου από την κεντρική υπολογιστική μονάδα, μία οθόνη, έναν εκτυπωτή και ένα σαρωτή. Με σκοπό να βρει τις πιο συμφέρουσες τιμές, επισκέφθηκε τέσσερα γνωστά καταστήματα, το Πλαίσιο, το Multirama, το Fnac και το E-shop, όχι κατ' ανάγκη με αυτή τη σειρά. Συμπωματικά, αγόρασε τελικά από καθένα απ' αυτά τα καταστήματα ένα τμήμα του υπολογιστικού του συστήματος. Όταν γύρισε σπίτι του, ο πατέρας του τον ρώτησε τι έκανε τελικά με τις αγορές του, αλλά ο Γιάννης του απάντησε με γρίφους. Συγκεκριμένα, του είπε ότι τον σαρωτή τον αγόρασε από το Multirama, ότι αμέσως μετά την αγορά της κεντρικής μονάδας δεν επισκέφθηκε το E-shop, ότι το Πλαίσιο ήταν ο δεύτερος σταθμός του και ότι στη μεθεπόμενη επίσκεψη μετά το Fnac αγόρασε την

οθόνη. Γράψτε ένα πρόγραμμα Prolog για να βοηθήσετε τον πατέρα του Γιάννη να μάθει τι αγόρασε ο γιος του από κάθε κατάστημα και με ποια σειρά τα επισκέφθηκε. Συγκεκριμένα, ορίστε ένα κατηγόρημα `buycomp/1`, έτσι ώστε όταν καλείται σαν `buycomp(S)` να επιστρέφει στο S μία αναπαράσταση της ζητούμενης λύσης. Για παράδειγμα:

```
?- buycomp(S).

S = [buy(fnac,computer),buy(plaisio,printer),

     buy(e_shop,display),buy(multirama,scanner)]
```

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο Prolog.

1.3 Άσκηση 3

Έστω ότι έχετε στη διάθεσή σας έναν επεξεργαστή που διαθέτει N καταχωρητές R1, R2, ..., RN οι οποίοι έχουν δακτυλιοειδή διευθέτηση. Αυτό σημαίνει ότι μπορεί κάποιος να μεταφέρει τα περιεχόμενα του καταχωρητή Ri στον καταχωρητή Ri+1, για $1 \leq i < N$, και του RN στον R1 με τις εντολές `move(i)`, για $1 \leq i < N$, και `move(N)`, αντίστοιχα. Επίσης, μπορεί να αντιμεταθέσει τα περιεχόμενα των καταχωρητών Ri και Rj με την εντολή `swap(i,j)`, όπου $i < j$. Έστω, τώρα, ότι σας δίνονται τα αρχικά περιεχόμενα των N καταχωρητών, καθώς και τα επιθυμητά τελικά περιεχόμενα. Το ζητούμενο είναι να βρεθεί η μικρότερη αλληλουχία από τις εντολές `move` και `swap` που πρέπει να εκτελεσθούν για να επιτευχθεί ο ζητούμενος μετασχηματισμός. Συγκεκριμένα, ορίστε το κατηγόρημα `codegen/3`, έτσι ώστε όταν αυτό καλείται με πρώτο όρισμα τη λίστα των αρχικών περιεχομένων των καταχωρητών και με δεύτερο όρισμα τη λίστα των τελικών περιεχομένων, να επιστρέφει στο τρίτο όρισμα τη λίστα των απαραίτητων (ελάχιστων) εντολών που απαιτούνται για το μετασχηματισμό. Σημειώστε ότι είναι δυνατόν στην αναπαράσταση των περιεχομένων των καταχωρητών να έχουμε, τόσο στην αρχική όσο και στην τελική κατάσταση, το σύμβολο `*`, που σημαίνει, για μεν την αρχική κατάσταση "δεν ξέρω τι περιέχεται στον καταχωρητή", για δε την τελική κατάσταση "δεν με ενδιαφέρει τι περιέχεται στον καταχωρητή". Κάποια παραδείγματα εκτέλεσης είναι τα εξής:

```
?- codegen([a,b,c,d],[a,d,a,b],L).

L = [move(2),move(1),swap(3,4),swap(2,3)]
```

```
?- codegen([a,*,c],[c,a,*],L).

L = [move(1),move(3)]
```

```
?- codegen([a,b,c],[a,a,*],L).

L = [move(1)]
```

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο Prolog.

1.4 Άσκηση 4

Μία εκδοχή του λεγόμενου job-shop προβλήματος είναι η εξής: Έχουμε να εκτελέσουμε μία σειρά από έργα `j1, j2, j3 …`, ανεξάρτητα το ένα από το άλλο, καθένα από τα οποία αποτελείται από έναν αριθμό από εργασίες `t11, t12 …, t21, t22, …, t31, t32, …` 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/2` 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 κατηγόρημα `deadline/1` δίνει την προθεσμία (σε μονάδες χρόνου) μέσα στην οποία πρέπει να έχουν εκτελεσθεί όλα τα έργα. Ορίστε σε Prolog το κατηγόρημα `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 εκτελεσθούν 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 κατηγόρημα `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 κατηγόρημα `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(t21,0,5),t(t12,5,11),t(t23,11,14)]),
```

```
     execs(m2,[t(t42,7,9),t(t31,10,14)])]                -> ;


S = ..................................................

....................................................................
```

Τα δεδομένα του προβλήματος, στη μορφή των γεγονότων Prolog που δίνονται στην εκφώνηση, βρίσκονται στο http://www.di.uoa.gr/~takis/jobshop_data.pl.

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο Prolog.

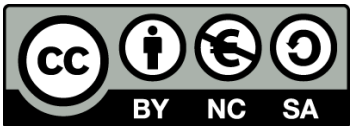
Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης



[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.