

Μεταπτυχιακό Πρόγραμμα Εφαρμοσμένης Πληροφορικής

Προχωρημένη Τεχνητή Νοημοσύνη

Εργασία 2

Prolog-Λογικός Προγραμματισμός με Περιορισμούς
Χειμερινό Εξάμηνο 2016

Άσκηση 1 - Μαθήματα

Σε μια πανεπιστημιακό τμήμα, πρέπει να γίνουν οι ακόλουθες εξετάσεις:

- εξέταση μαθήματος Α, διάρκειας 150' και φοιτητές 120,
- εξέταση μαθήματος Β, διάρκειας 180' και φοιτητές 200,
- εξέταση μαθήματος Γ, διάρκειας 230' και φοιτητές 190,
- εξέταση μαθήματος Δ, διάρκειας 130' και φοιτητές 250,
- εξέταση μαθήματος Ε, διάρκειας 210' και φοιτητές 70,

Υποθέστε ότι:

- υπάρχει ένας μεγάλος χώρος εξετάσεων χωρητικότητας 400 ατόμων και ότι
- ένα ή περισσότερα μαθήματα μπορούν να εξεταστούν ταυτόχρονα σε αυτό τον χώρο,
- οι χρόνοι έναρξης και τέλους κάθε εξέτασης είναι ανεξάρτητοι για κάθε μάθημα, ακόμη και αν γίνονται στην ίδια αίθουσα.
- Το μάθημα Α δεν μπορεί να συμπίπτει με το μάθημα Γ.

Ζητούμενα είναι (α) ο ελάχιστος χρόνος που θα πρέπει να χρησιμοποιηθεί ο χώρος εξετάσεων για να ολοκληρωθούν όλες οι εξετάσεις και (β) η χρονική στιγμή που θα πρέπει να ξεκινήσει η κάθε εξέταση (από την χρονική στιγμή 0, όπου και ξεκινούν όλες οι εξετάσεις). Να γράψετε ένα κατηγορημα **exams/2** σε CLP το οποίο να επιστρέφει τον χρόνο έναρξης της κάθε εξέτασης και το συνολικό χρόνο που θα χρησιμοποιηθεί η αίθουσα. Για παράδειγμα:

```
?- exams(L, Duration).  
L = [0, 0, 150, 380, 180]  
Duration = 510
```

Άσκηση 2 - Ακολουθίες

Υπάρχουν τρεις λίστες ακεραίων, όπως φαίνονται παρακάτω.

```
list(1, [2, 3, 4, 10, 22, 11, 17]).  
list(2, [1, 2, 3, 4, 10, 22, 11, 11, 10, 24]).  
list(3, [2, 3, 4, 5, 10, 23, 10, 22, 11, 17]).
```

Στόχος είναι να βρεθεί η **κοινή τριάδα διαδοχικών ακεραίων** στις παραπάνω λίστες, η οποία έχει μέγιστο άθροισμα. Προφανώς σε κάθε λίστα η τριάδα μπορεί να βρίσκεται σε διαφορετική θέση. Στο παράδειγμά μας η τριάδα που αναζητείται φαίνεται παρακάτω.

```
[2, 3, 4, 10, 22, 11, 17]).  
      | (θέση 4)  
[1, 2, 3, 4, 10, 22, 11, 11, 10, 24]).  
      | (θέση 5)  
[2, 3, 4, 5, 10, 23, 10, 22, 11, 17]).  
      | (θέση 7)
```

Να υλοποιήσετε ένα κατηγορημα **align/3** το οποίο στο πρώτο όρισμα επιστρέφει τις θέσεις από όπου ξεκινούν οι τριάδες που αναζητούμε σε κάθε λίστα, στο δεύτερο όρισμα την τριάδα, και στο τρίτο το άθροισμά της.

```
?- align(Pos, Triple, Sum).  
Pos = [4, 5, 7]  
Triple = [10, 22, 11]  
Sum = 43  
Yes (0.00s cpu)
```

Άσκηση 3 - Η Κατασκευαστική Εταιρεία

Μια κατασκευαστική εταιρεία πρέπει να κατασκευάσει ένα πολυόροφο κτίριο. Οι εργασίες που πρέπει να γίνουν σε κάθε όροφο είναι οι ακόλουθες:

- κατασκευή του ορόφου (μπετά),
- ηλεκτρολογική εγκατάσταση,
- υδραυλική εγκατάσταση, και οι
- λοιπές εργασίες.

Για κάθε όροφο η εταιρεία έχει υπολογίσει τις μέρες και τον αριθμό των εργαζομένων που απαιτούνται. Οι ανάγκες καταγράφονται σε γεγονότα όπως θα εξηγηθεί στα ακόλουθα.

Κατασκευή του ορόφου

Η κατασκευή του κάθε ορόφου απαιτεί συγκεκριμένο/αμετάβλητο αριθμό εργαζομένων, διαφέρει όμως στον αριθμό των ημερών που απαιτούνται. Η πληροφορία καταγράφεται σε γεγονότα της μορφής:

```
builders(<Όροφος>, days(<N>)) .
```

όπου το πρώτο όρισμα είναι ο όροφος και το δεύτερο ο αριθμός των ημερών για την ολοκλήρωση του σε ημέρες. Για παράδειγμα το γεγονός:

```
builders(a1, days(12)) .
```

δηλώνει ότι ο πρώτος όροφος (a1) απαιτεί 12 μέρες για την ολοκλήρωσή του.

Ηλεκτρολογική εγκατάσταση

Οι ηλεκτρολογικές εγκαταστάσεις για κάθε όροφο απαιτούν διαφορετικό αριθμό εργαζομένων. Η πληροφορία καταγράφεται σε γεγονότα της μορφής:

```
electricians(<Όροφος>, <Αριθμός Εργαζομένων>, days(<N>))
```

όπου το πρώτο όρισμα είναι ο όροφος, το δεύτερο ο αριθμός των εργαζομένων και το τρίτο ο αριθμός των ημερών για την ολοκλήρωση του. Για παράδειγμα το γεγονός:

```
electricians(a2, 5, days(30)) .
```

δηλώνει ότι ο δεύτερος όροφος (a2) απαιτεί 5 εργαζόμενους ηλεκτρολόγους και 30 μέρες για την ολοκλήρωσή των ηλεκτρολογικών εγκαταστάσεων του.

Υδραυλική εγκατάσταση

Με παρόμοιο τρόπο ορίζονται και τα γεγονότα που αφορούν τις υδραυλικές εργασίες.

```
plumbers(<Όροφος>, <Αριθμός Εργαζομένων>, days(<N>))
```

Για παράδειγμά το ακόλουθο γεγονός:

```
plumbers(a1, 4, days(8)) .
```

δηλώνει ότι ο πρώτος όροφος (a1) απαιτεί 4 εργαζόμενους υδραυλικούς και 8 μέρες για την ολοκλήρωσή των υδραυλικών εγκαταστάσεων του.

Λοιπές εργασίες

Τέλος οι λοιπές εργασίες (finishing) ορίζονται με παρόμοια γεγονότα:

```
finishers(<Όροφος>, <Αριθμός Εργαζομένων>, days(<N>))
```

Για παράδειγμα το γεγονός:

```
finishers(a1, 3, days(10)) .
```

δηλώνει ότι ο πρώτος όροφος (a1) απαιτεί 3 εργαζόμενους και 10 μέρες για την ολοκλήρωσή των λοιπών εργασιών.

Ισχύουν οι ακόλουθοι περιορισμοί:

- Πρέπει πρώτα να γίνει η κατασκευή του ορόφου (μπετά) πριν από κάθε άλλη εργασία στον όροφο.
- Οι όροφοι πρέπει να κατασκευαστούν με την σειρά (!), δηλ. δεν μπορεί να ξεκινήσει η κατασκευή

- (μπετά) του a2 (δεύτερος όροφος) πριν την ολοκλήρωση του a1 (πρώτος όροφος).
- Οι ηλεκτρολογικές και υδραυλικές εργασίες μπορούν να γίνουν με οποιαδήποτε σειρά, αλλά όχι ταυτόχρονα σε ένα όροφο.
- Οι λοιπές εργασίες ξεκινούν μετά το πέρας των ηλεκτρολογικών και υδραυλικών εργασιών.
- Τα συνεργεία μπορούν να εργαστούν σε οποιοδήποτε όροφο ο οποίος έχει κατασκευαστεί (μπετά).
- Το έργο (όλοι οι όροφοι) πρέπει να ολοκληρωθεί *ακριβώς στις 140 μέρες (#=)*.
- Ο μέγιστος αριθμός εργατών από κάθε κατηγορία που μπορεί η εταιρεία να προσλάβει είναι είκοσι (20), δηλαδή μπορεί να προσλάβει το πολύ 20 ηλεκτρολόγους, 20 υδραυλικούς και 20 λοιπούς εργαζόμενους. Σημειώνεται ότι ο αριθμός 20 είναι το μέγιστο, δηλαδή η εταιρεία μπορεί να προσλάβει και λιγότερους εργαζόμενους.

Εφόσον το έργο έχει συγκεκριμένη ημερομηνία παράδοσης, αυτό που ενδιαφέρει την εταιρεία είναι να καθυστερήσει την έναρξη του έργου όσο το δυνατό περισσότερο και παράλληλα να γνωρίζει πόσους εργαζόμενους θα χρειαστεί για να το πετύχει.

Να μοντελοποιήσετε το πρόβλημα σε CLP δίνοντας την υλοποίηση ενός κατηγορήματος **construction_plan(Start, Duration, Electricians, Plumbers, Finishers)**, το οποίο πετυχαίνει όταν το έργο μπορεί να ξεκινήσει την ημέρα Start, να ολοκληρωθεί σε Duration μέρες, με τον αριθμό των εργαζομένων να εμφανίζεται στα αντίστοιχα ορίσματα. Για παράδειγμα:

```
?-construction_plan(Start,Duration,Electricians,Plumbers,Finishers) .
Start = 40
Duration = 100
Electricians = 18
Plumbers = 11
Finishers = 18
```

Η εκτέλεση του παραπάνω κατηγορήματος, σημαίνει ότι το αργότερο που μπορεί να ξεκινήσει η κατασκευή είναι η χρονική στιγμή 40, Η συνολική του διάρκεια θα είναι 100 στιγμές και θα απαιτηθούν 18 ηλεκτρολόγοι, 11 υδραυλικοί και 18 λοιποί εργαζόμενοι για να επιτευχθεί αυτό.

Η υλοποίησή σας θα πρέπει να μπορεί να απαντήσει και σε ερωτήματα της ακόλουθης μορφής:

```
?- construction_plan(Start, Duration, 10, 10, 10) .
Start = 30
Duration = 110
Yes (319.96s cpu)
```

Το παραπάνω ερώτημα καθορίζει το αριθμό των εργαζομένων για κάθε κατηγορία και αναζητά την αργότερη ημερομηνία που μπορεί να ξεκινήσει η κατασκευή του έργου, καθώς και την συνολική του διάρκεια.

Σημειώσεις

- Τα γεγονότα που αποθηκεύουν τα δεδομένα του προβλήματος θα τα βρείτε στο `compus` στο αρχείο **exec13.ecl**.
- Το πρόβλημα έχει αρκετές λύσεις και η εύρεση της βέλτιστης λύσης μπορεί να πάρει σημαντικό χρόνο. Καλό είναι να εκτελέσετε την αναζήτηση με τον αλγόριθμο `branch_and_bound` δίνοντας ένα χρονικό όριο. Για παράδειγμα το ακόλουθο κάνει αναζήτηση τιμών για τις μεταβλητές `Vars`, ελαχιστοποιώντας το κόστος `Cost`, με χρονικό περιθώριο τα 120 δευτερόλεπτα:

```
bb_min(labeling(Vars),Cost,bb_options{timeout:120}) .
```

Μπορείτε να ορίσετε το χρόνο αναζήτησης, όπως εσείς επιθυμείτε και ανάλογα με την ισχύ του H/Y σας.

ΠΑΡΑΔΟΣΗ

Θα παραδώσετε εντός της ημερομηνίας που αναφέρεται στο `compus` τα ακόλουθα:

- Ένα αρχείο με το όνομα **exec13.ecl** το οποίο θα περιέχει τις λύσεις (κατηγορήματα) **όλων των ασκήσεων**.
- Ένα αρχείο **report.pdf** (σε μορφή pdf) το οποίο θα περιέχει:
 - Στην πρώτη σελίδα το Όνομά σας, τον Αριθμό μητρώου σας και το email σας.
 - Για κάθε μια από τις ασκήσεις:
 - τον **κώδικα** σας (ασχέτως αν βρίσκεται και στο αρχείο `exec1.ecl`, `airplanes.ecl`) και σχολιασμό σχετικά με αυτόν.
 - Παραδείγματα εκτέλεσης (1-2 για κάθε κατηγορία)
 - Bugs και προβλήματα που έχει ο κώδικάς σας.

ΠΡΟΣΟΧΉ: ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΑΥΣΤΗΡΑ ΤΑ ΟΝΟΜΑΤΑ ΚΑΙ ΤΗ ΣΕΙΡΑ ΤΩΝ ΟΡΙΣΜΑΤΩΝ ΠΟΥ ΔΙΝΕΤΑΙ ΠΑΡΑΠΑΝΩ (ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ ΚΩΔΙΚΑ)

Καλή επιτυχία (και *have fun with Prolog!*)