



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Τεχνητή Νοημοσύνη Ι

Εργαστηριακή Άσκηση 1

Σγάρμπας Κυριάκος

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

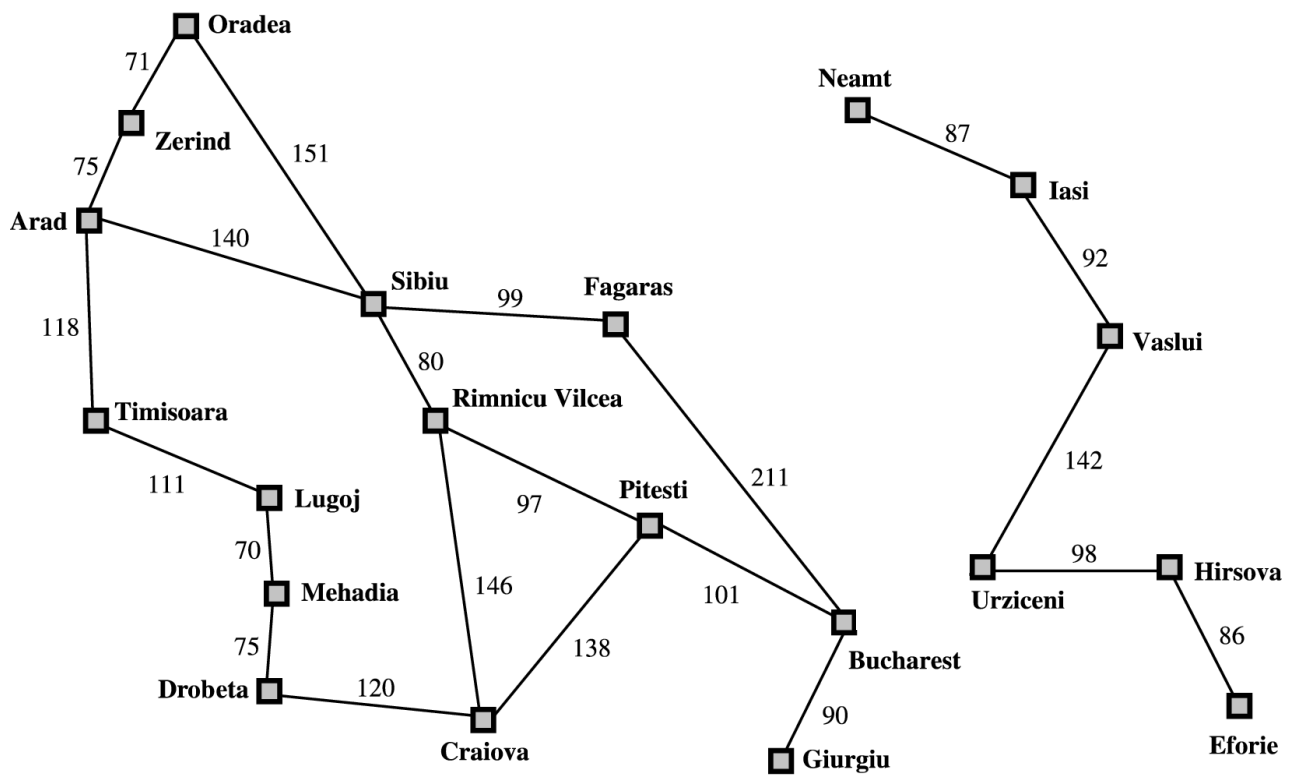
ΑΝΟΙΚΤΑ ακαδημαϊκά **ΠΠ**
μαθήματα



ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ Ι - ΕΡΓΑΣΤΗΡΙΟ

ΕΝΟΤΗΤΑ Α: “Μια Βόλτα στη Ρουμανία” ΑΣΚΗΣΕΙΣ #1, #2, #3

Έστω ο παρακάτω χάρτης με το οδικό δίκτυο της Ρουμανίας.



ΑΣΚΗΣΗ #1

(Απευθείας Ερωτήσεις και Κανόνες)

1. Εκφράστε τις συνδέσεις μεταξύ των πόλεων ως γεγονότα στην Prolog. Πχ.:

`link(d, c, 120).`

`link(s, f, 99).`

`...`

`link(h, e, 86).`

2. Κάντε κατάλληλη ερώτηση στην Prolog ώστε να βρείτε:

(a) όλες τις πόλεις που συνδέονται απευθείας με το Sibiu.

(b) όλες τις πόλεις που συνδέονται απευθείας με το Sibiu και απέχουν λιγότερο από 100 χιλιόμετρα από αυτό.



(c) όλες τις πόλεις που συνδέονται απευθείας:

- i. και με το Arad και με την Oradea (AND).
- ii. με το Arad ή με την Oradea (OR).
- iii. ή με το Arad ή με την Oradea (XOR).

(c) σε ποιές πόλεις μπορείτε να ταξιδέψετε από το Pitesti:

- i. κάνοντας ακριβώς έναν ενδιάμεσο σταθμό,
- ii. ..., χωρίς όμως να καταλήξετε στο Pitesti,
- iii. ..., ούτε σε πόλεις οι οποίες συνδέονται απευθείας με αυτό.

(c) όλες τις πόλεις που συνδέονται απευθείας με τουλάχιστον τρεις άλλες πόλεις. Σημείωση: Παρατηρήστε πόσο περιπλέκονται οι απευθείας ερωτήσεις. Και ότι δεν μπορούμε να βάλουμε συνθήκες μεταξύ διαφορετικών επιστροφών. Τα πράγματα διευκολύνονται αν ορίσουμε βοηθητικές συναρτήσεις (κανόνες).

3. Φτιάξτε μια συνάρτηση Prolog με όνομα **neighbours/2** η οποία θα δέχεται τα ονόματα δυο πόλεων και θα επαληθεύεται μόνο όταν υπάρχει απευθείας σύνδεση μεταξύ τους.

```
?- neighbours(m,l).
```

```
true
```

```
? - neighbours(a,p).
```

```
false
```

Σημείωση: Υπάρχουν 2 τρόποι να την ορίσετε.

4. Φτιάξτε μια συνάρτηση Prolog με όνομα **next/2** η οποία θα δέχεται το όνομα μιας πόλης και θα επιστρέφει όλες τις πόλεις με τις οποίες η πρώτη έχει απευθείας σύνδεση.

```
?- next(a,X).
```

```
X = z ;
```

```
X = s ;
```

```
X = t ;
```

```
false
```

Σχολιάστε την συμμετρία της και την σχέση της με την neighbours/2.

ΑΣΚΗΣΗ #2 (Αναδρομή και !)

5. Ορίστε μια συνάρτηση Prolog **reach(?City1, ?City2, +Link_Distance)** που θα δέχεται δυο πόλεις και ένα μέγεθος απόστασης (μετρημένο σε πλήθος links) και θα ικανοποιείται αν μπορούμε να ταξιδέψουμε από τη μία πόλη στην άλλη με ακριβώς Link_Distance μεταβάσεις (ή Link_Distance-1 ενδιάμεσους σταθμούς).

```
?- reach(a,b,4).
```

```
true
```

```
?- reach(a,b,2).
```

```
false.
```

```
?- reach(a,X,1).
```

```
X = z ;
```

```
X = s ;
```

```
X = t ;
```

```
false
```

6. Ορίστε μια συνάρτηση Prolog **reach2(?City1, ?City2, ?Link_Distance)** που θα λειτουργεί όπως η reach/3, με τη διαφορά ότι θα μπορεί να δίνει και ως έξοδο το Link_Distance.

Υπόδειξη: Θεωρήστε ότι το Link_Distance δεν θα ξεπερνά το 9.

```
?- reach2(n,i,X).
```

```
X = 1 ;
```

```
X = 3 ;
```

```
X = 3 ;
```

```
X = 5 ;
```



```

...
X = 9 ;
false.
?- reach2(n,a,X) .
false.

```

7. Ορίστε μια συνάρτηση Prolog **min_link_dist(?City1, ?City2, ?Link_Distance)** που θα λειτουργεί όπως η **reach2/3**, με τη διαφορά ότι θα επιστρέφει το ελάχιστο **Link_Distance** μεταξύ δύο πόλεων.

```

?- min_link_dist(a,b,X) .
X = 3.

```

Υπόδειξη: Μετρήστε πρώτα την διάμετρο του χώρου καταστάσεων.

8. Έστω η συνάρτηση **find_link_dist/3** που ορίζεται

```

ως εξής: find_link_dist(X,X,0):- !.
find_link_dist(X,Y,1):-
next(X,Y) , !.

```

```

find_link_dist(X,Y,CD):- next(X,Z) , find_link_dist(Z,Y,CDn) ,
CD is CDn+1.

```

Ελέγξτε αν μπορεί να αντικαταστήσει επαρκώς την **min_link_dist/3**. Τί προβλήματα μπορεί να προκαλέσει; Κάτω από ποιές συνθήκες;

9. Έστω η συνάρτηση **find_link_dist2/3** που ορίζεται

```

ως εξής:
find_link_dist2(X,X,0,_):- !.
find_link_dist2(X,Y,1,_):-
next(X,Y) , !.

```

```

find_link_dist2(X,Y,CD,Pass_prev_X):- next(X,Z) , \+(Z==Pass_prev_X) ,
find_link_dist2(Z,Y,CDn,X) ,
CD is CDn+1.

```

ΑΣΚΗΣΗ #3 (Με Χρήση Λιστών)

10. Προσθήκη μνήμης:

- (a) Ορίστε μια συνάρτηση **smart_find_link_dist(+City1, +City2, ?Link_Distance,**

+Memory) που θα λειτουργεί όπως η **reach2/3**, αλλά θα αποφεύγει τους κύκλους με τη χρήση της λίστας **Memory** που θα θυμάται τις πόλεις που έχει συναντήσει.

Υπόδειξη: Κατά την κλήση η **Memory** ξεκινά με αρχική τιμή [].

```

?- smart_find_link_dist(g,s,X, []).
X = 5 ;
X = 9 ;
...
X = 3 ;
false.

```

Τί χρησιμότητα θα είχε μια διαφορετική αρχική τιμή για τη λίστα **Memory**;

- (b) Ορίστε μια συνάρτηση **min_link_dist2/3** που θα λειτουργεί όπως η **min_link_dist/3**, χωρίς όμως να λαμβάνει υπ'όψιν τη διάμετρο του χώρου καταστάσεων.

```

?- min_link_dist2(g,a,X) .
X = 4.

```

Υπόδειξη: Χρησιμοποιήστε την **findall/3**, την **smart_find_link_dist/4** και την **sort/2**.

11. Προσθήκη διαδρομής:

- (a) Ορίστε μια συνάρτηση **smart_find_link_path/5** που θα λειτουργεί όπως η **smart_find_link_dist/4**, αλλά θα επιστρέφει και μια λίστα με το μονοπάτι της διαδρομής.

```

?- smart_find_link_path(a,o,X, [], P) .

```



```

X = 2,
P = [a, z, o] ;
X = 2,
P = [a, s, o] ;
...
X = 9,
P = [a, t, l, m, d, c, p, r, s|...] ;
false.

```

- (b) Ορίστε μια συνάρτηση **min_link_path/4** που θα λειτουργεί όπως η **min_link_dist/3**, αλλά θα επιστρέφει και μια λίστα με το μονοπάτι της διαδρομής. Αν δύο ή περισσότερες διαδρομές έχουν το ίδιο μήκος, να τις επιστρέφει όλες (μη ντετερμινιστικά).

```

?- min_link_path(a,o,X,P) .
X = 2,
P = [a, z, o] ;
X = 2,
P = [a, s, o] ;
false.

```

12. Υπολογισμός πραγματικής απόστασης:

- (a) Ορίστε μια συνάρτηση **smart_find_distance/4** που θα λειτουργεί όπως η **smart_find_link_dist/4** αλλά η απόσταση που θα μετράει δεν θα είναι σε πλήθος links αλλά σε χιλιόμετρα.

```

?- smart_find_distance(a,b,X,[]) .
X = 575 ;
X = 762 ;
X = 607 ;
...
X = 733 ;
false.

```

- (b) Ορίστε μια συνάρτηση **min_distance/3** που θα λειτουργεί όπως η **min_link_dist2/3** αλλά η απόσταση που θα μετράει δεν θα είναι σε πλήθος links αλλά σε χιλιόμετρα.

```

?- min_distance(a,b,X) .
X = 418.

```

- (c) Ορίστε μια συνάρτηση **smart_find_dist_path/5** που θα λειτουργεί όπως η **smart_find_link_path/5**, αλλά θα μετράει την απόσταση σε χιλιόμετρα.

```

?- smart_find_dist_path(a,b,X,[],P) .
X = 575,
P = [a, z, o, s, r, p, b] ;
X = 762,
P = [a, z, o, s, r, c, p, b] ;
...
X = 733,
P = [a, t, l, m, d, c, p, b] ;
false.

```

- (d) Ορίστε μια συνάρτηση **min_dist_path/4** που θα λειτουργεί όπως η **min_link_path/4**, αλλά θα μετράει την απόσταση σε χιλιόμετρα.

```

?- min_dist_path(a,b,X,P) .
X = 418,
P = [a, s, r, p, b] ;
false.

```



Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**

- Έκδοση **1.0** διαθέσιμη [εδώ](#).

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρων, Φακωτάκης Νικόλαος, Σγάρμπας Κυριάκος, Πέππας Παύλος, Μουστάκας Κωνσταντίνος. «Εργαστήριο 1». Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: https://eclass.upatras.gr/modules/course_metadata/opencourses.php?fc=15

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:

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.

