

Τεχνητή Νοημοσύνη

2^ο Θέμα

Ονόματα: Δραγώνας Σωτήρης 03114187, Ζαρόγιαννης Γιάννης 03114186

Πληροφορίες για την υλοποίηση:

Διαβάσαμε τα διάφορα .csv μέσω της Java και δημιουργήσαμε αρχεία γεγονότων Prolog. Όπου δεν είχαμε πληροφορία για κάποιο δεδομένο χρησιμοποιήσαμε το keyword «unknown». Δημιουργήσαμε τα γεγονότα client και taxis με όλα τα διαθέσιμα δεδομένα. Έπειτα δημιουργήσαμε ένα αρχείο γεγονότων με τα στοιχεία των γραμμών και με βάση αυτό και τον κανόνα «unavailable» του «rules.pl» προσθέσαμε στα γεγονότα μόνο τους κόμβους που ανήκουν σε διαθέσιμη γραμμή. Στα γεγονότα των κόμβων προσθέσαμε και ένα index ώστε να κρατήσουμε τη σειρά με την οποία διαβάστηκαν. Φτιάξαμε και αρχείο γεγονότων με τα στοιχεία του traffic.

Διαβάζοντας το nodes.csv μέσω της Java και χρησιμοποιώντας τον κανόνα «next» για κάθε τριάδα διαδοχικών κόμβων δημιουργήσαμε σημεία με πίνακες γειτνίασης και τα προσθέσαμε σε ένα hash map ίδιο με της πρώτης άσκησης.

Ο A* είναι ίδιος με την πρώτη άσκηση με την προσθήκη του priority το οποίο είναι ένα βάρος με το οποίο πολλαπλασιάζουμε την ευκλείδεια απόσταση του επόμενου κόμβου από τον προορισμό. Το priority εξαρτάται από το δρόμο που ανήκει ο κόμβος και από την ώρα του πελάτη και υλοποιείται με τον κανόνα «priority» και ο σκοπός του είναι χρησιμοποιώντας τα γεγονότα της prolog να γίνεται λιγότερο ή περισσότερο πιθανό να επιλεγούν κάποιοι δρόμοι ανάλογα με τα χαρακτηριστικά τους. Αυτό συμβαίνει διότι ανάλογα με τα χαρακτηριστικά του δρόμου και χρησιμοποιώντας ως τιμή βάσης την ευκλείδεια απόσταση την πολλαπλασιάζουμε με αυτό το βάρος ώστε να φαίνεται σαν ο προορισμός να είναι είτε πιο μακριά είτε πιο κοντά.

----- (Λεπτομέρειες στα σχόλια του rules.pl) -----

Για όσα δεδομένα δεν είχαμε πληροφορία θέσαμε το ουδέτερο priority: 1.0 ώστε να μην επηρεάζουν το αποτέλεσμα. Αν το «Oneway» δεν είχε πληροφορία, θεωρήσαμε τον δρόμο διπλό.

Ανάλογα με τα χαρακτηριστικά των ταξί π.χ.: γλώσσα, θέσεις κλπ. κάποια ταξί απορρίπτονται και δεν υπολογίζεται γι' αυτά διαδρομή. Για τα υπόλοιπα εκτελείται ο A* και δημιουργούνται 2 κατατάξεις. Η 1^η εξαρτάται από το μήκος της διαδρομής του ταξί μέχρι τον πελάτη και η 2^η από το rating κάθε ταξί.

```
Reading input files and creating the corresponding prolog data files (enjoy)...
-Learning the rules...
-I feel so much smarter now...
-Asking the client...
-Had a very interesting conversation with the client...
-Let's find out which taxis are around...
-So many good taxi choices...
-Let me walk every single road of the map...
-My legs are in pain with so much walking...
-Marking every point...
-I lost count with so many points...
-I can't stand the traffic, please get me out...
-Done with the honking...
-Stalking the neighbours...
-Getting you a taxi...
-Lets find out how to arrive to your destination...
Available taxis sorted by distance to client
1 --- Taxi id=110
2 --- Taxi id=140
Available taxis sorted by rating to client
1 --- Taxi id=140, rating=9.1
2 --- Taxi id=110, rating=8.0
Type which taxi you prefer by giving its ID and pressing <Enter>:110

That's a very good choice
KML file will be created...
It was a pleasure to calculate stuff for you, bye bye :)

Process finished with exit code 0
|
```

Αφού δοθούν στον πελάτη οι 2 κατατάξεις του ζητείται να γράψει ποιο ταξί επιλέγει δίνοντας το ID του και τέλος δημιουργείται το αρχείο kml με τις διαδρομές όλων των ταξί μέχρι τον πελάτη σε κόκκινο εκτός της διαδρομής του επιλεγμένου από τον πελάτη ταξί που είναι με πράσινο. Επίσης με καφέ, φαίνεται η διαδρομή του πελάτη από την αφετηρία στον προορισμό του.

Για τον A* δοκιμάσαμε 3 μεγέθη μετώπου αναζήτησης: 10, 100, 1000.

Ακολουθούν οι κατατάξεις για τη 1^η επιλογή σημείων:

- Queue size = 10

Available taxis sorted by distance to client

1 --- Taxi id=140

2 --- Taxi id=160

3 --- Taxi id=110

4 --- Taxi id=190

Available taxis sorted by rating to client

1 --- Taxi id=140, rating=9.1

2 --- Taxi id=110, rating=8.0

3 --- Taxi id=190, rating=7.2

4 --- Taxi id=160, rating=7.1

- Queue size = 100

Available taxis sorted by distance to client

1 --- Taxi id=140

2 --- Taxi id=160

3 --- Taxi id=110

4 --- Taxi id=190

Available taxis sorted by rating to client

1 --- Taxi id=140, rating=9.1

2 --- Taxi id=110, rating=8.0

3 --- Taxi id=190, rating=7.2

4 --- Taxi id=160, rating=7.1

- Queue size = 1000

Available taxis sorted by distance to client

1 --- Taxi id=140

2 --- Taxi id=160

3 --- Taxi id=110

4 --- Taxi id=190

Available taxis sorted by rating to client

1 --- Taxi id=140, rating=9.1

2 --- Taxi id=110, rating=8.0

3 --- Taxi id=190, rating=7.2

4 --- Taxi id=160, rating=7.1

Ακολουθούν οι κατατάξεις για τη 2^η επιλογή σημείων:

- Queue size = 10

Available taxis sorted by distance to client

1 --- Taxi id=110

2 --- Taxi id=140

Available taxis sorted by rating to client

1 --- Taxi id=140, rating=9.1

2 --- Taxi id=110, rating=8.0

- Queue size = 100

Available taxis sorted by distance to client

1 --- Taxi id=110

2 --- Taxi id=140

Available taxis sorted by rating to client

1 --- Taxi id=140, rating=9.1

2 --- Taxi id=110, rating=8.0

- Queue size = 1000

Available taxis sorted by distance to client

1 --- Taxi id=110

2 --- Taxi id=140

Available taxis sorted by rating to client

1 --- Taxi id=140, rating=9.1

2 --- Taxi id=110, rating=8.0