ΥΣ02 Τεχνητή Νοημοσύνη – Χειμερινό Εξάμηνο 2026-2027

Εργασία Τρίτη
(2 μονάδες του συνολικού βαθμού στο μάθημα)

Ημερομηνία Ανακοίνωσης: 25/11/2026
Ημερομηνία Παράδοσης: 15/12/2026 ώρα 23:59

Αντιγραφή: Σε περίπτωση που προκύψουν φαινόμενα αντιγραφής, οι εμπλεκόμενοι θα βαθμολογηθούν **στην άσκηση με βαθμό μηδέν.**

Χρήση μοντέλων Παραγωγικής Τεχνητής Νοημοσύνης: Θα ήταν ιδανικό να κάνετε την εργασία χωρίς την χρήση εργαλείων όπως το ChatGPT. Αν όμως δυσκολευτείτε σε κάποιο ερώτημα και χρησιμοποιήσετε κάποιο τέτοιο εργαλείο, η απάντηση σας στο ερώτημα θα πρέπει να συνοδεύεται από όλο τον διάλογο σας με το εργαλείο. **Δεν θα αφαιρεθούν βαθμοί σε αυτή την περίπτωση.** Προσέξτε όμως η απάντηση σας θα πρέπει να είναι με βάση τις διαλέξεις και τα φροντιστήρια του μαθήματος, και όχι απλό copy-paste της απάντησης του εργαλείου. **Σε αντίθετη περίπτωση, θα μηδενιστεί το αντίστοιχο ερώτημα.**

Πρόβλημα 1:
Θεωρήστε ένα πρόβλημα ικανοποίησης περιορισμών με n μεταβλητές, c περιορισμούς, όπου κάθε περιορισμός περιλαμβάνει το πολύ k μεταβλητές με $k>2$, και το πεδίο ορισμού κάθε μεταβλητής έχει πληθικό αριθμό το πολύ d. Ένα τέτοιο πρόβλημα είναι το παζλ κρυπταριθμητικής που δίνεται στη σελίδα 14 των διαφανειών της Ενότητας 7 των διαλέξεων ή αυτό που δίνεται στο Πρόβλημα (2) παρακάτω. Να δώσετε ένα αλγόριθμο ο οποίος παίρνει σαν είσοδο ένα τέτοιο πρόβλημα ικανοποίησης περιορισμών και το μετατρέπει σε συνεπές ως προς τις ακμές. Ποια είναι η πολυπλοκότητα χειρίστης περίπτωσης του αλγόριθμου αυτού σαν συνάρτηση των παραμέτρων n, c, k και d;
(5+5=10 μονάδες)

Πρόβλημα 2:
Στο πρόβλημα αυτό θα χρησιμοποιήσουμε τους αλγόριθμους επίλυσης προβλημάτων ικανοποίησης περιορισμών που βρίσκονται στην ιστοσελίδα του βιβλίου AIMA (https://github.com/aimacode/aima-python/blob/master/aima/csp.py). Πιθανά να χρειαστεί να επεκτείνετε αυτούς τους αλγόριθμους ώστε να χρησιμοποιηθούν στο πρόβλημα που έχετε να λύσετε.

Το πρόβλημα ικανοποίησης περιορισμών που θα μας απασχολήσει είναι η επίλυση παζλ κρυπταριθμητικής. Για παράδειγμα, θεωρήστε το παρακάτω παζλ:

```
    SEND
+   MORE
-------------
   MONEY
```

Αυτό που θέλουμε είναι να βρούμε μια ανάθεση ψηφίων (0, 1, 2, …, 9) στα γράμματα του παζλ ώστε να γίνεται η πρόσθεση που δίνεται.

Έχετε να απαντήσετε τα ακόλουθα ερωτήματα:

1. Να ορίσετε το παραπάνω παζλ σαν πρόβλημα ικανοποίησης περιορισμών.
2. Να μελετήσετε την εργασία

   Fahiem Bacchus, Xinguang Chen, Peter van Beek, and Toby Walsh. Binary vs. non-binary constraints. Artificial Intelligence, 140:1-37, 2002.

   Να χρησιμοποιήσετε την τεχνική των κρυμμένων μεταβλητών (hidden variables) που παρουσιάζεται στην εργασία αυτή για να μετατρέψετε το παραπάνω παζλ σε δυαδικό πρόβλημα.
3. Να λύσετε το δυαδικό πρόβλημα του ερωτήματος (2) χρησιμοποιώντας τους αλγόριθμους BT, BT+MRV, FC+MRV, MAC και min-conflicts. Να συγκρίνετε τους αλγόριθμους αυτούς με βάση τους κόμβους που δημιουργούν στο δένδρο αναζήτησης, τους ελέγχους συνέπειας που πραγματοποιούν, τον αριθμό των οπισθοδρομήσεων (backtracks) που κάνουν, τον χρόνο διάδοσης των περιορισμών και τον συνολικό χρόνο εκτέλεσης. Να παρουσιάσετε τα αποτελέσματα σας χρησιμοποιώντας κατάλληλους πίνακες όπως αυτούς στις διαφάνειες του μαθήματος.

(5+5+40=50 μονάδες)

Πρόβλημα 3:

Θεωρήστε το σταυρόλεξο του παρακάτω σχήματος.

1	2	3
4		
5		

Να βρεθούν έξι λέξεις με τρία γράμματα (τρεις λέξεις οριζόντια και τρεις λέξεις κάθετα) που επιλύουν το σταυρόλεξο. Κάθε λέξη πρέπει να διαλεχθεί από την παρακάτω λίστα δεκαοχτώ λέξεων:

add, age, aid, aim, air, are, arm, art, bad, bat, bee, boa, dim, ear, eel, eft, lee, oaf

Προτείνεται η αρχική επίλυση να γίνει διαισθητικά, στη συνέχεια χρησιμοποιώντας συνέπεια πεδίου (1-consistency) και έπειτα χρησιμοποιώντας συνέπεια ακμής (2-consistency).

Υπάρχουν τουλάχιστον δύο τρόποι για να περιγραφεί το παραπάνω σταυρόλεξο σαν πρόβλημα ικανοποίησης περιορισμών.

Ο πρώτος είναι να αναπαρασταθούν οι θέσεις των λέξεων (1-οριζόντια, 4-οριζόντια κλπ.) ως μεταβλητές με πεδίο το σύνολο των διαθέσιμων λέξεων. Ο περιορισμός είναι ότι οι λέξεις πρέπει να έχουν το ίδιο γράμμα στο σημείο που τέμνονται.

Ο δεύτερος είναι να αναπαρασταθούν τα εννέα τετράγωνα ως μεταβλητές. Το πεδίο κάθε μεταβλητής είναι το σύνολο των γραμμάτων του αλφάβητου {a, b, c, …, z}. Οι περιορισμοί αφορούν το γεγονός ότι τα γράμματα πρέπει να σχηματίζουν μια λέξη που ανήκει στο σύνολο των διαθέσιμων λέξεων. Για παράδειγμα, το πάνω αριστερά τετράγωνο και το πάνω κεντρικό τετράγωνο δεν μπορούν να έχουν και τα δύο την τιμή a, επειδή δεν υπάρχει λέξη, στο σύνολο διαθέσιμων λέξεων, που να ξεκινάει με aa.

Έχετε να απαντήσετε τα εξής ερωτήματα:

1. Να δοθεί ένα παράδειγμα απαλοιφής τιμής που οφείλεται σε συνέπεια πεδίου χρησιμοποιώντας την πρώτη αναπαράσταση (αν υπάρχει).

2. Να δοθεί ένα παράδειγμα απαλοιφής τιμής που οφείλεται σε συνέπεια ακμής χρησιμοποιώντας την πρώτη αναπαράσταση (αν υπάρχει).
3. Είναι η συνέπεια πεδίου σε συνδυασμό με τη συνέπεια ακμής αρκετές για να λυθεί το δοσμένο πρόβλημα, χρησιμοποιώντας την πρώτη αναπαράσταση; Γιατί;
4. Να δοθεί ένα παράδειγμα απαλοιφής τιμής που οφείλεται σε συνέπεια πεδίου χρησιμοποιώντας τη δεύτερη αναπαράσταση (αν υπάρχει).
5. Να δοθεί ένα παράδειγμα απαλοιφής τιμής που οφείλεται σε συνέπεια ακμής χρησιμοποιώντας τη δεύτερη αναπαράσταση (αν υπάρχει).
6. Είναι η συνέπεια πεδίου σε συνδυασμό με τη συνέπεια ακμής αρκετές για να λυθεί το δοσμένο πρόβλημα, χρησιμοποιώντας την δεύτερη αναπαράσταση; Γιατί;
7. Ποια αναπαράσταση οδηγεί στην πιο αποδοτική λύση με τη χρήση τεχνικών που βασίζονται στη συνέπεια; Να δοθούν τα επιχειρήματα πάνω στα οποία βασίζεται η απάντηση σας.

(7*3=21 μονάδες)

Πρόβλημα 4:
Ο αλγόριθμος AC-3 (σελίδα 54 των διαφανειών της Ενότητας 7 των διαλέξεων) επαναφέρει στην ουρά κάθε ακμή (X_k, X_i) όποτε διαγράφεται κάποια τιμή από το πεδίο ορισμού της X_i, ακόμη και αν κάθε τιμή της X_k είναι συνεπής με αρκετές από τις εναπομείνασες τιμές της X_i. Υποθέστε ότι, για κάθε ακμή (X_k, X_i), διατηρούμε τον αριθμό των εναπομεινασών τιμών της X_i που είναι συνεπείς με κάθε τιμή της X_k. Εξηγήστε πώς μπορούν να ενημερώνονται αυτοί οι αριθμοί με αποδοτικό τρόπο και, κατά συνέπεια, δείξτε ότι η συνέπεια ακμών (arc consistency) μπορεί να επιτευχθεί σε συνολικό χρόνο $O(n^2d^2)$. Υλοποιήστε το νέο αλγόριθμο σε Python και δοκιμάστε τον στο Πρόβλημα 2. Πόσο βελτιώθηκε ο MAC με το νέο αλγόριθμο;
(5+5=10 μονάδες)

Πρόβλημα 5:
Στο πρόβλημα αυτό θα μελετήσουμε την άλγεβρα χρονικών διαστημάτων (Interval Algebra) που ορίστηκε από τον James Allen (1983). Δείτε την ιστοσελίδα https://ics.uci.edu/~alspaugh/cls/shr/allen.html και το άρθρο του Allen https://cse.unl.edu/~choueiry/Documents/Allen-CACM1983.pdf. Αυτό που μας ενδιαφέρει στο πρόβλημα αυτό είναι να δούμε αν το Gemini (Έκδοση που παρέχεται δωρεάν για φοιτητές: https://gemini.google/gr/students/?hl=el. Μην χρησιμοποιήσετε άλλη έκδοση ή άλλο chatbot γιατί δεν θα μπορούμε να ελέγξουμε τα αποτελέσματα σας.) μπορεί να χρησιμοποιηθεί για τον υπολογισμό του πίνακα σύνθεσης (composition table) της άλγεβρας (Πίνακας 4α στην παραπάνω ιστοσελίδα ή Σχήμα 4 στο άρθρο του Allen). Να σχεδιάσετε κατάλληλα prompts που θα σας επιτρέψουν να κάνετε τους σχετικούς ελέγχους. Δεν είναι απαραίτητο να ελέγξετε όλα τα 13*13 κελιά του πίνακα σύνθεσης. Επικεντρωθείτε στα πιο ενδιαφέροντα (τα πιο δύσκολα, τα πιο εύκολα, άλλα;). Τι συμπεραίνετε για τις ικανότητες του Gemini για χρονικό λογισμό;
(15 μονάδες)

Σύνολο μονάδων εργασίας: 10+50+21+10+15=106 μονάδες (=άριστα)