
Αναπαράσταση Γνώσης με Λογική

Εργασία 2

Έλεγχος λογικής κάλυψης μέσω ανάλυσης και
τοπικής αναζήτησης

Εργασία 2 – Προτασιακή Λογική

- Ο στόχος της εργασίας είναι η υλοποίηση ενός απλού πράκτορα που βασίζεται στη γνώση. Ο πράκτορας θα διαθέτει:
 - μια βάση γνώσης (ΒΓ) υλοποιημένη σε ένα αρχείο κειμένου
 - έναν μηχανισμό εξαγωγής συμπερασμάτων, ο οποίος θα χρησιμοποιεί τον κανόνα της ανάλυσης (resolution) και τον αλγόριθμο τοπικής αναζήτησης GSAT
- Ο πράκτορας θα πρέπει να μπορεί να συμπεράνει εάν μια ατομική πρόταση που εισάγει ο χρήστης γίνεται entailed από την ΒΓ

Βάση Γνώσης

- Η ΒΓ του πράκτορα θα είναι αποθηκευμένη σε ένα απλό αρχείο κειμένου (text file) και θα κατασκευάζεται με τυχαίο τρόπο
- Η ΒΓ θα είναι σε CNF. Δηλ. κάθε πρόταση θα είναι μια διάζευξη λεκτικών (λεκτικό – literal) είναι μια προτασιακή μεταβλητή ή η άρνηση της).
 - π.χ. $\neg a \vee b \vee \neg c$
 - η κάθε πρόταση θα είναι γραμμένη σε ξεχωριστή γραμμή του αρχείου
 - Στην πρώτη γραμμή του αρχείου θα υπάρχουν πληροφορίες σχετικά με την ΒΓ. Συγκεκριμένα, η πρώτη γραμμή θα έχει, π.χ. την εξής μορφή
10 100 4
Όπου 10 – το πλήθος των προτασιακών μεταβλητών
100 – το πλήθος των προτάσεων
4 – το μέγιστο μήκος (πλήθος λεκτικών) της κάθε πρότασης
 - Οι υπόλοιπες γραμμές του αρχείου θα περιέχουν τις προτάσεις

Μηχανισμός Εξαγωγής Συμπερασμάτων

- Ο μηχανισμός εξαγωγής συμπερασμάτων (ΜΕΣ) θα είναι μια διαδικασία δύο φάσεων.
 1. τοπική αναζήτηση με GSAT
 2. υλοποίηση του κανόνα της ανάλυσης (resolution)

Ο ΜΕΣ θα χρησιμοποιείται για να μπορεί ο πράκτορας να αποφανθεί εάν μια ατομική πρόταση (δηλ. ένα λεκτικό), που θα εισάγει ο χρήστης, καλύπτεται λογικά (δηλ. γίνεται entailed) από την ΒΓ.

Μηχανισμός Εξαγωγής Συμπερασμάτων

- Αν ο χρήστης εισάγει π.χ. το λεκτικό $\neg f$, ο ΜΕΣ θα πρέπει να προσθέσει (προσωρινά) την άρνηση του λεκτικού (δηλ. το f) στην ΒΓ
- Μετά θα προσπαθήσει να βρει ένα μοντέλο της πρότασης $B\Gamma \wedge f$ χρησιμοποιώντας τον αλγόριθμο GSAT
 - αν βρεθεί μοντέλο τότε η πρόταση $B\Gamma \wedge f$ είναι ικανοποιήσιμη και το $\neg f$ δεν γίνεται entailed από την ΒΓ
- Αν ο GSAT δεν βρει λύση εντός ενός ορίου επαναλήψεων/επανεκκινήσεων τότε ο ΜΕΣ τερματίζει τον GSAT και εκκινεί την διαδικασία της ανάλυσης
 - εφαρμόζει τον κανόνα της ανάλυσης επαναληπτικά. ξεκινώντας από την πρόταση $B\Gamma \wedge f$. μέχρι είτε να καταλήξει σε άτοπο είτε να μην μπορεί να εφαρμοστεί περαιτέρω.
 - Αν καταλήξει σε άτοπο τότε το $\neg f$ γίνεται entailed από την ΒΓ. Αλλιώς, δεν γίνεται

Εργασία 2 – Προτασιακή Λογική

- Το πρόγραμμα που θα κατασκευάσετε πρέπει αρχικά να ζητάει από τον χρήστη:
 - το πλήθος των προτάσεων C που θα περιλαμβάνονται στην ΒΓ
 - το μέγιστο μήκος (πλήθος λεκτικών) L της κάθε πρότασης
 - το πλήθος P των προτασιακών μεταβλητών. Οι μεταβλητές αυτές αναπαρίστανται από τα μικρά γράμματα του αγγλικού αλφαβήτου (και συνεπώς το μέγιστο πλήθος τους είναι 26)
- Αφού ο χρήστης εισάγει τιμές για τα C, L, P θα ακολουθεί η κατασκευή της αρχικής ΒΓ.

Εργασία 2 – Κατασκευή της ΒΓ

- Η αρχική ΒΓ θα κατασκευάζεται ως εξής:
- Για κάθε μια από τις C προτάσεις:
 - θα δημιουργείται ένας τυχαίος αριθμός ανάμεσα στο 1 και στο L . Αυτός ο αριθμός (έστω x) θα προσδιορίζει το πλήθος των λεκτικών της πρότασης.
 - Το κάθε ένα από τα x λεκτικά θα δημιουργείται επιλέγοντας πρώτα, τυχαία, ένα προτασιακό σύμβολο ανάμεσα στα P διαθέσιμα. Μετά, το επιλεγθέν σύμβολο θα αντιστρέφεται με πιθανότητα 50%.
 - Π.χ. αν στην τύχη επιλεγθεί το k , τότε, στην τύχη, με πιθανότητα 50% θα εμφανίζεται στην πρόταση ως k και με 50% ως $\neg k$
- Για κάθε πρόταση που δημιουργείται θα πρέπει να γίνεται έλεγχος ότι δεν υπάρχει ήδη στην ΒΔ. Αν υπάρχει τότε θα δημιουργείται μια άλλη με τον ίδιο τρόπο.
- Η διαδικασία δημιουργίας της ΒΓ πρέπει να γίνεται στην κεντρική μνήμη (δηλ. οι προτάσεις πρέπει να αποθηκεύονται προσωρινά σε κατάλληλη δομή). Μόλις ολοκληρωθεί η διαδικασία, η ΒΓ θα γράφεται σε ένα αρχείο κειμένου.

Εργασία 2 – Κατασκευή της ΒΓ

- Κατά την κατασκευή της ΒΓ ο κώδικας σας πρέπει να εξασφαλίζει τα εξής:
 - ο ίδιος όρος δεν εμφανίζεται πάνω από μια φορά
 - $(P \vee \neg Q \vee R) \wedge (P \vee \neg Q \vee R)$ απαγορεύεται!
 - σε κανέναν όρο δεν εμφανίζεται το ίδιο literal πάνω από μια φορά
 - $(P \vee \neg Q \vee P)$ απαγορεύεται
 - σε κανέναν όρο δεν εμφανίζεται ένα ατομικό σύμβολο και η άρνηση του
 - $(P \vee \neg Q \vee R) \wedge (Q \vee R \vee \neg Q)$ απαγορεύεται!
 - Γιατί;

Εργασία 2 – Προτασιακή Λογική

- Αφού έχει ολοκληρωθεί η κατασκευή της ΒΓ, το πρόγραμμα θα πρέπει να ζητάει από την χρήστη την εισαγωγή ενός λεκτικού.
- Μόλις ο χρήστης εισάγει ένα λεκτικό, π.χ. h , οι προτάσεις της ΒΓ συν την ατομική πρόταση $\neg h$ πρέπει να φορτώνονται σε κατάλληλη δομή στην κεντρική μνήμη και να ξεκινάει η λειτουργία του ΜΕΣ
 - Το πρώτο βήμα αφορά την κλήση του αλγορίθμου GSAT με συγκεκριμένα όρια επανεκκινήσεων και επαναλήψεων. Αν βρεθεί λύση από τον GSAT τότε το λεκτικό δεν γίνεται entailed. Αν ο GSAT φτάσει στο όριο των επιτρεπόμενων επανεκκινήσεων τότε τερματίζεται.
 - Το δεύτερο βήμα αφορά την εφαρμογή της ανάλυσης. Αν η ανάλυση καταλήξει σε άτοπο (κενή πρόταση) τότε πρέπει να εμφανίζεται κατάλληλο μήνυμα, και το λεκτικό που εισήγαγε ο χρήστης, καθώς και όλα τα ενδιάμεσα αποτελέσματα της ανάλυσης, πρέπει να προστίθενται στο αρχείο της ΒΓ. Αλλιώς, απλά εμφανίζεται ένα κατάλληλο μήνυμα.
- Αυτό πρέπει να γίνεται επαναληπτικά για όσα λεκτικά επιθυμεί ο χρήστης

Φάση 1 - GSAT

- Παράμετροι του αλγόριθμου (οι δύο πρώτες θα εισάγονται από τον χρήστη):
 - **το όριο των επαναλήψεων / αλλαγών τιμών σε μεταβλητές (max flips)**
 - όταν ο αλγόριθμος φτάσει αυτό το όριο χωρίς να έχει βρει λύση επαναλαμβάνει τη διαδικασία με διαφορετική αρχική ανάθεση
 - **το όριο των επανεκκινήσεων / προσπαθειών (max tries) (δηλ. πόσες φορές θα επαναληφθεί η διαδικασία)**
 - όταν ο αλγόριθμος φτάσει αυτό το όριο χωρίς να έχει βρει λύση, τερματίζει ανεπιτυχώς
 - **το σύνολο α των προτάσεων (όρων) προτασιακής λογικής σε CNF**

Ο αλγόριθμος GSAT

```
procedure GSAT( $a$ ,  $maxTries$ ,  $maxFlips$ )  
  for  $i := 1$  to  $maxTries$  do  
     $A :=$  randomly chosen assignment of the variables in  $a$   
    for  $j := 1$  to  $maxFlips$  do  
      if  $A$  satisfies  $a$  then return ( $A$ )  
      else  
         $x :=$  randomly chosen variable of  $a$  whose flip satisfies the  
          maximum number of clauses under the current assignment  $A$   
        if by flipping  $x$  you get a cost  $\leq$  current cost then  
          flip value of  $x$  in  $A$   
        endif  
      endfor  
    endfor  
  return (“No model found”)
```

Θέματα Υλοποίησης GSAT

- Τι δομές δεδομένων χρειάζονται?
 - μια δομή για να κρατάει την τρέχουσα ανάθεση τιμών A στις μεταβλητές
 - Μονοδιάστατος πίνακας αρκεί
 - μια δομή για να κρατάει το κόστος κάθε πιθανού flip (αλλαγής τιμής σε μεταβλητή)
 - πίνακας
 - μια δομή που κρατάει τη CNF φόρμουλα
 - Ένας δισδιάστατος πίνακας αρκεί. Λεπτομέρειες σε λίγο!
- Ο GSAT επιτρέπει τις πλάγιες κινήσεις!
 - όταν δεν υπάρχουν γειτονικές καταστάσεις με καλύτερο κόστος από την τρέχουσα αλλά υπάρχουν κάποιες με ίδιο κόστος, διαλέγει μια από αυτές στην τύχη

GSAT: Παράδειγμα

1) $A \vee \neg B \vee C$

2) $A \vee C \vee D$

3) $B \vee D \vee \neg E$

4) $C \vee D \vee \neg E$

5) $\neg B \vee C \vee \neg E$

μεταβλητές

όροι

A	B	C	D	E	1	2	3	4	5	κόστος
0	1	0	0	1	F	F	T	F	F	4

Φάση 2 – Ανάλυση

- Η υλοποίηση της ανάλυσης μπορεί να γίνει απλά ελέγχοντας για κάθε πρόταση (ξεκινώντας από την $\neg h$) αν μπορεί να αναλυθεί με κάποια από τις υπόλοιπες. Αν μπορεί, τότε το αποτέλεσμα της ανάλυσης πρέπει να προστίθεται στην δομή που αποθηκεύει τις προτάσεις, εκτός αν το αποτέλεσμα είναι η κενή πρόταση, οπότε η διαδικασία ολοκληρώνεται
 - Αν η ανάλυση καταλήξει σε άτοπο (κενή πρόταση) τότε πρέπει να εμφανίζεται κατάλληλο μήνυμα, και το λεκτικό που εισήγαγε ο χρήστης, καθώς και όλα τα ενδιάμεσα αποτελέσματα της ανάλυσης, πρέπει να προστίθενται στο αρχείο της ΒΓ. Αλλιώς, απλά εμφανίζεται ένα κατάλληλο μήνυμα.

Θέματα Υλοποίησης

- **οι όροι πρέπει να αποθηκεύονται σε κάποια δομή δεδομένων**
 - απλή λύση -> δισδιάστατος πίνακας $C \times L$
 - Αν ο πίνακας ονομάζεται `clauses`, τότε `clauses[i][j]` είναι το literal που βρίσκεται στη θέση j στον i -th όρο
 - Ένα ειδικό σύμβολο να συμβολίζει το κενό (χρειάζεται όταν ένας όρος έχει λιγότερα από L λεκτικά)
- **πως θα ξεχωρίσετε ένα ατομικό σύμβολο από την άρνηση του?**
 - απλή λύση -> αν έχετε 10 ατομικά σύμβολα (μεταβλητές) τα αριθμείτε από 1 ως 10. Οι αρνήσεις τους αριθμούνται από -1 ως -10.
 - `clauses[3][0] = -6` σημαίνει ότι το πρώτο literal στον τέταρτο όρο είναι το $\neg x_6$

Εργασία 2

- Μετά την ολοκλήρωση της εκτέλεσης του προγράμματος θα τυπώνεται το αποτέλεσμα (δηλ. για κάθε λεκτικό που εισήγαγε ο χρήστης αν γίνεται entailed ή όχι)
- Επίσης να τυπώνεται ο τρόπος με τον οποίο έγινε η απόδειξη (GSAT ή ανάλυση).
- Η υλοποίηση μπορεί να γίνει σε μια γλώσσα όπως Java, C, C++, Python, ή σε Matlab, σε ομάδες ως και τεσσάρων ατόμων.
- Προθεσμία παράδοσης άσκησης: **24/01/2021**.
- Τρόπος παράδοσης: μέσω eclass
 - Θα πρέπει να αναρτηθεί ο κώδικας + μια τεχνική αναφορά που θα περιγράφει τις βασικές λειτουργίες του προγράμματος και όποια προβλήματα έχουν προκύψει