Εισαγωγή

Η παρούσα εργασία εκπονήθηκε στο πλαίσιο του μαθήματος "Τεχνητή Νοημοσύνη" του χειμερινού εξαμήνου 2024-25 στο Οικονομικό Πανεπιστήμιο Αθηνών. Το παρόν έγγραφο αφορά το Β' μέρος της, όπου σκοπός είναι η ανάπτυξη δύο ξεχωριστών προγραμμάτων που αξιοποιούν τον αλγόριθμο forward chaining, για την απόδειξη προτάσεων βάσει γεγονότων και κανόνων που αποθηκεύονται σε μια βάση γνώσης. Την εργασία έχουν επιμεληθεί οι φοιτήτριες Αναΐς Φαρχάτ, Ελένη Αντωνιάδη Τσαραμπουλίδη, Στυλιανή Μουμτζή.

Ερώτημα 1

Ζητούμενο

Στην πρώτη προγραμματιστική εργασία του Β μέρους, καλούμαστε να υλοποιήσουμε ένα πρόγραμμα με forward chaining αλγόριθμο, το οποίο διαβάζει από τη βάση γνώσης μας τους κανόνες και τα γεγονότα που της έχουμε δώσει. Στη συνέχεια, ελέγχει αν η πρόταση που πληκτρολόγησε ο χρήστης είναι αληθής ή ψευδής με βάση αυτά.

Προσέγγιση

Η βασική ιδέα για την υλοποίηση της εργασίας είναι η δημιουργία ενός αλγορίθμου που αναζητά τη βάση γνώσης και ελέγχει αν η πρόταση προς απόδειξη μπορεί να παραχθεί με βάση τους κανόνες και τα γεγονότα.

Παράδειγμα χρήσης:

1. Το πρόγραμμα ζητά από τον χρήστη να πληκτρολογήσει την πρόταση προς απόδειξη.
2. Ξεκινά η διαδικασία του forward chaining.
3. Επιστρέφεται “true” ή “false” αν η πρόταση μπορεί να αποδειχτεί ή όχι αντίστοιχα, και εκτυπώνεται σχετικό μήνυμα.

Υλοποίηση

Rulebook.txt: Το αρχείο Rulebook.txt περιέχει τη βάση γνώσης, η οποία χωρίζεται σε δύο κατηγορίες, γεγονότα (Facts) και κανόνες (Rules). Ο αλγόριθμος είναι σχεδιασμένος έτσι ώστε να επεξεργάζεται και τις δύο κατηγορίες, αναζητώντας τη σχέση μεταξύ τους.

Main.java: Στη main γίνεται η εκκίνηση του προγράμματος. Αρχικά, η βάση γνώσης φορτώνεται και τα δεδομένα μετατρέπονται σε αντικείμενα για επεξεργασία.Έπειτα ο χρήστης καλείται να πληκτρολογήσει την πρόταση προς απόδειξη. Ο αλγόριθμος forward chaining καλείται για να ελέγξει αν η πρόταση αποδεικνύεται. Ανάλογα το αποτέλεσμα, επιστρέφεται “true” ή ”false” από την forward chaining και τέλος εκτυπώνεται μήνυμα διαμορφωμένο στο εκάστοτε αποτέλεσμα.

ForwardChaining.java: Στην ForwardChaining γίνεται η υλοποίηση του forward chaining αλγορίθμου και ξεκινάει με τον έλεγχο όλων των κανόνων και γεγονότων στη βάση γνώσης. Έπειτα,

οι κανόνες εφαρμόζονται και εξετάζονται σειριακά με σκοπό την παραγωγή νέων γεγονότων. Όταν αποδεικνύεται νέο γεγονός, αφού γίνει έλεγχος για να σιγουρευτούμε πως δεν έχει ήδη αποδειχτεί και βρίσκεται στην λίστα μας, προστίθεται σε αυτήν, δηλαδή στα γεγονότα προς εξέταση. Κάθε φορά που εκτελείται ο βρόχος, γίνεται εκ νέου έλεγχος αν αποδεικνύεται το γεγονός με τα μέχρι στιγμής γνωστά γεγονότα. Αν το γεγονός επαληθευτεί επιστρέφεται “true”, διαφορετικά αν εξαντληθούν τα δεδομένα και δεν έχει αποδειχτεί επιστρέφεται «false».

Η κλάση **Literal.java** Αντιπροσωπεύει τα γεγονότα, προσδιορίζοντας και το αν συνοδεύονται από έκφραση άρνησης, η κλάση Rules.java περιγράφει τους κανόνες της βάσεις γνώσης, όπου κάθε κανόνας ορίζεται από μια συλλογή (set) από γεγονότα-προυποθέσεις και ένα γεγονός αποτέλεσμα. Τέλος, η κλάση KnowledgeBase.java αποθηκεύει και οργανώνει τα παραπάνω ώστε να είναι εύκολα και λειτουργικά προσβάσιμα από τον ForwardChaining αλγόριθμο.

Δυνατότητες

Έστω ότι η βάση γνώσης είναι:

#Facts

A

B

#Rules

A,B -> C

A,B -> D

C,D -> E

E -> F

Αν ο χρήστης πληκτρολογήσει την πρόταση F, το πρόγραμμα ξεκινά με τα δεδομένα γεγονότα A και B.Από τον κανόνα A,B -> C προσθέτει το C στα γεγονότα. Παρομοίως συμπεριφέρεται και για το A,B -> D. Έχοντας πλέον ως δεδομένα τα C, D από τον κανόνα C,D -> E προσθέτει το E στα γεγονότα, και αντίστοιχα από τον E->F βρίσκει πλέον το F και στην οθόνη επιστρέφεται η πρόταση “F has been proven”.

Αν ο χρήστης εισάγει μια πρόταση που δεν υπάρχει, όπως G, το πρόγραμμα αναζητά τη βάση γνώσης και εκτελεί τους κανόνες, αδυνατεί να παράγει το G κι έτσι εκτυπώνεται στην οθόνη “G can’t be proved.”.

Ερώτημα 2

Ζητούμενο

Στη δεύτερη προγραμματιστική εργασία του Β μέρους, καλούμαστε να υλοποιήσουμε ένα πρόγραμμα με forward chaining αλγόριθμο, το οποίο διαβάζει από τη βάση γνώσης μας, τα παραδείγματα εκπαίδευσης που της έχουμε δώσει σε πρωτοβάθμια κατηγορηματική λογική. Ύστερα αναζητά αν η πρόταση που πληκτρολόγησε ο χρήστης μας είναι αληθής ή ψευδής, με βάση αυτά.

Προσέγγιση

Η βασική ιδέα για την υλοποίηση της εργασίας είναι να δημιουργηθεί ένας αλγόριθμος ο οποίος αναζητά τη βάση γνώσης και αλλάζει στα παραδείγματα εκπαίδευσης τις μεταβλητές με τα ζητούμενα ονόματα, ώστε να βρούμε αν ο τύπος που αναζητάμε αληθεύει.

Παράδειγμα χρήσης:

1. Το πρόγραμμα ζητάει από τον χρήστη να πληκτρολογήσει τον τύπο προς απόδειξη που επιθυμεί.
2. Το πρόγραμμα ξεκινάει τη διαδικασία του forward chaining.
3. Αν ο τύπος αληθεύει, επιστρέφει true, αλλιώς επιστρέφει false.

Υλοποίηση

KB.txt: Αυτό το αρχείο αναπαριστά τη βάση γνώσης του που θα χρησιμοποιήσουμε στην εξαγωγή δεδομένων μας. Γενικά, τα παραδείγματα γνώσεις χωρίζονται σε δύο κατηγορίες, κανόνες και γεγονότα. Ωστόσο ο αλγόριθμος είναι διαμορφωμένος έτσι ώστε τα γεγονότα να μη χρειάζεται να είναι χωρισμένα σε κατηγορίες μέσα στη βάση γνώσης, για να γίνεται σωστά η εξαγωγή τους.

Main: Στην main, ξεκινάει το πρόγραμμα παίρνοντας το FOL statement που θέλει να αποδείξει ο χρήστης, αν τα δεδομένα που εισάγει είναι σε λάθος μορφή, εμφανίζεται error. Ύστερα ξεκινάει να διαβάζει τη βάση γνώσης και να περνάει τα στοιχεία της σε μια λίστα, αφότου τα μετατρέψει στη κατάλληλη μορφή αντικειμένου, ώστε να είναι έτοιμα για επεξεργασία. Για να το κάνει αυτό καλεί δύο μεθόδους, η getPredicate η οποία δεχόταν τα δεδομένα ως string και τα μετέτρεπε σε αντικείμενα τύπου Predicate και η getClause η οποία τα νέα αντικείμενα Predicate τα έκανε clauses.

Μετά την μετατροπή καλεί την μέθοδο ForwardChaining με είσοδο τη λίστα με τα δεδομένα της βάσης και την forwardChain με όρισμα τον τύπο προς απόδειξη και ξεκινάει η εξαγωγή συμπερασμάτων. Αφότου δοθεί η τιμή αληθείας, εκτυπώνεται και τελείωνει το πρόγραμμα.

ForwardChaining: Εδώ γίνεται η βασική υλοποίηση του προγράμματος. Η κλάση ξεκινάει με αρχικοποίηση διάφορων δομών δεδομένων που θα χρησιμοποιήσουμε παρακάτω, καθώς και τρεις

μεθόδους που αφορούν την λίστα με τα στοιχεία της βάσης γνώσης. Η μέθοδος forwardChain δέχεται ως όρισμα το FOL statement που εισήγαγε ο χρήστης προς απόδειξη.

Αρχικά ελέγχει κάθε clause. Βλέπει με βάσΗ το μέγεθος του αν είναι κανόνας η γεγονός και ελέγχει αν είναι αυτό που ψάχνουμε. Ύστερα ψάχνει όλα τα predicates για κάθε clause παίρνοντας τους όρους τους και κάνοντας Unify με τους όρους του FOL statement προς απόδειξη. Αν το unify ισούται με αληθές, τότε ξεκινάει ξανά επανάληψη για όλα τα clauses και τα predicates της βάσης γνώση εναλλάσοντας τις μεταβλητές με βάση τη δομή substitutions του Unifier. Μετά δημιουργούνται νεα αντικείμενα τύπου Predicate, αν ταυτίζονται με αύτο που ψάχνουμε, τότε η αναζήτηση είναι επιτυχής και επιστρέφεται αληθές. Αν τελειώσουν όλες οι επαναλήψεις και δεν βρεθεί ο τύπος που ψάχνουμε, επιστρέφεται ψευδές.

Clause/ Predicate/ Term: Οι κλάσεις αυτές, έχουν όλα τα απαραίτητα χαρακτηριστικά που αποτελούν ένα αντικείμενο του εκάστοτε τύπου, όπως getters, constructors, toString etc. Αποτελούν κλάσεις από το εργαστήριο του μαθήματος.

Unifier: Στη Unifier, γίνονται οι λειτουργίες που μας βοηθάνε για τις εναλλαγές μεταβλητών και ονομάτων. Αποτελεί κλάση από το εργαστήριο του μαθήματος.

FOLLoader/ Vec2: Κλάσεις που δίνουν μια δομή στην εμφάνιση των αντικειμένων. Αποτελούν κλάσεις από το εργαστήριο του μαθήματος.

Δυνατότητες

Έστω πως θέλουμε να αποδείξουμε τον τύπο Mortal(Socrates) και η βάση γνώσης μας μοιάζει έτσι:

Human(Socrates)

NOT Human(x) OR Mortal(x)

Το λογισμικό μας είναι ικανό να ψάξει τη βάση μας, να αντικαταστήσει με Socrates όπου x, και τέλος να εξάγει το αποτέλεσμα μας , όπου είναι αληθές. Αν του δίναμε τον τύπο IsDog(Milos) θα επέστρεφε ψευδές καθώς στη βάση μας δεν υπάρχει αντίστοιχο παράδειγμα εκπαίδευσης.