ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΙΑ 5ου ΕΞΑΜΗΝΟΥ 2024

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

ΜΕΛΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Αθανάσιος-Ζώης Δημητρακόπουλος (Α.Μ. 3220039)

Ανδρέας Λάμπος (Α.Μ. 3220105)

ΔΙΔΑΣΚΩΝ

Ίων Ανδρουτσόπουλος

1η Εργασία - Μέρος Β’

1ο Ερώτημα

Θα αναλύσουμε την υλοποίηση για την **Εξαγωγή συμπερασμάτων προς τα εμπρός (forward chaining) για προτάσεις Horn προτασιακής λογικής.**

---

1. Εισαγωγή

Το πρόγραμμα αφορά την εφαρμογή του **αλγορίθμου εξαγωγής συμπερασμάτων προς τα εμπρός (forward chaining)**, χρησιμοποιώντας μια βάση γνώσεων και κανόνες. Ο στόχος είναι να αποφασιστεί αν μια δεδομένη πρόταση (query) είναι αληθής ή ψευδής, και να παρουσιάσει το αντίστοιχο δέντρο απόδειξης.

---

2. Λεπτομέρειες Αρχείων και Μεθόδων

Αρχείο knowledge_base.txt

Το αρχείο περιέχει τη βάση γνώσεων με:

1. Θετικά γεγονότα (π.χ. P, R).
2. Κανόνες που συνδέουν γεγονότα με αποτελέσματα (π.χ. P -> Q).

Παραδείγματα:

- Θετικά γεγονότα: P, R, U, A, B.
- Κανόνες:
  - P -> Q: Αν το P είναι αληθές, τότε το Q είναι επίσης αληθές.
  - A, B -> C: Αν ισχύουν και τα A και B, τότε ισχύει το C.

Αρχείο forward_chaining.py

Το αρχείο υλοποιεί τον αλγόριθμο εξαγωγής συμπερασμάτων προς τα εμπρός (forward chaining), και περιλαμβάνει βοηθητικές μεθόδους για την ανάγνωση της βάσης γνώσεων, τη δημιουργία του δέντρου απόδειξης, και την παρουσίαση των αποτελεσμάτων.

Κύριες Μέθοδοι

1. Μέθοδος forward_chaining_with_proof()

Υλοποιεί την εμπρόσθια αλυσιδωτή συλλογιστική για την απόδειξη του query.

Είσοδοι:

- **facts**: Το σύνολο των γνωστών θετικών γεγονότων.
- **rules**: Μια λίστα κανόνων από τη βάση γνώσεων.
- **query**: Το ερώτημα που πρέπει να αποδειχθεί.

Λειτουργία:

1. Ξεκινά με τα αρχικά γεγονότα (known_facts).
2. Εξετάζει κάθε κανόνα:
    - Αν όλες οι προϋποθέσεις του κανόνα ισχύουν και το συμπέρασμα δεν έχει ήδη παραχθεί, το συμπέρασμα προστίθεται στα γνωστά γεγονότα, δηλαδή στη βάση γνώσης μας.
3. Συνεχίζει τη διαδικασία μέχρι να μη βρεθούν νέα γεγονότα.
4. Ελέγχει αν το query είναι στα θετικά ή αρνητικά γεγονότα.
5. Επιστρέφει:
    - True και το δέντρο απόδειξης αν το query είναι αποδείξιμο.
    - False αν δεν είναι αποδείξιμο ή αν ο τύπος προς απόδειξη δεν υπάρχει στη βάση γνώσης.

---

2. Μέθοδος build_proof_tree(fact, tree)

Δημιουργεί το δέντρο απόδειξης για το query.

Είσοδος:

- **Fact:** Το γεγονός που ελέγχεται για την απόδειξη.
- **Tree:** Ένα λεξικό που περιγράφει τους κανόνες. Κάθε κλειδί είναι ένα γεγονός και η τιμή του είναι η λίστα των προϋποθέσεων για την απόδειξή του.

Λειτουργία:

- Αν το γεγονός είναι αρχικό (είναι γνωστό), επιστρέφεται ως έχει.
- Αν έχει εξαρτήσεις, αναδρομικά κατασκευάζονται τα δέντρα για κάθε προϋπόθεση.

Έξοδος:

- Ένα δέντρο απόδειξης (λεξικό) που δείχνει τη λογική δομή των εξαρτήσεων.

Παράδειγμα:
Για το query C και τους κανόνες A, B -> C, θα επιστραφεί:

- {'C': ['A', 'B']}

---

3. Μέθοδος print_proof_tree()

Εκτυπώνει το δέντρο απόδειξης σε μορφή κατανοητή για τον χρήστη.

Χρησιμοποιεί αναδρομή για να εμφανίσει κάθε επίπεδο του δέντρου με κατάλληλη εσοχή.

Για το παραπάνω **παράδειγμα**, θα επιστραφεί C ↓
A
B

---

4. Μέθοδος KBread()

Η βάση γνώσεων αναλύεται από το αρχείο knowledge_base.txt:

- Παίρνει ως μεταβλητή το όνομα του αρχείου της βάσεις γνώσεις.
- Γεγονότα και κανόνες αποθηκεύονται σε σύνολα και λίστες.

Ανακεφαλαίωση

Το project περιλαμβάνει:

- Μια βάση γνώσεων με θετικά/αρνητικά γεγονότα και κανόνες.
- Έναν αλγόριθμο εξαγωγής συμπερασμάτων προς τα εμπρός (forward chaining) για την εξαγωγή συμπερασμάτων.
- Έξοδο με δέντρο απόδειξης για πλήρη τεκμηρίωση της διαδικασίας.

Παραδείγματα Εκτέλεσης

1.

```
Enter the path to the input file: knowledge_base.txt
Enter the query you want to check (e.g., F): L

The query 'L' is derivable (True).

Proof tree:
L ↓
  T ↓
    P
    R
  C ↓
    A
    B

Enter the query you want to check (e.g., F) or enter '0' to exit: 0

Process finished with exit code 0
```

2.

```
Enter the path to the input file: knowledge_base.txt
Enter the query you want to check (e.g., F): T

The query 'T' is derivable (True).

Proof tree:
T ↓
  P
  R

Enter the query you want to check (e.g., F) or enter '0' to exit: 0

Process finished with exit code 0
```

1η Εργασία – Μέρος Β'

2ο Ερώτημα

Εξαγωγή συμπερασμάτων προς τα εμπρός (forward chaining) για προτάσεις Horn πρωτοβάθμιας κατηγορηματικής λογικής.

Σε αυτήν την εργασία θα αναλυθεί η υλοποίηση για την Εξαγωγή Συμπερασμάτων προς τα Εμπρός (forward chaining) για οριστικές προτάσεις Horn πρωτοβάθμιας κατηγορηματικής λογικής.

1. Εισαγωγή

Το πρόγραμμα αφορά την εφαρμογή του αλγορίθμου εξαγωγής συμπερασμάτων προς τα εμπρός, χρησιμοποιώντας μια βάση γνώσεων και κανόνες. Ο στόχος είναι να αποφασιστεί αν μια δεδομένη πρόταση (query) είναι αληθής ή ψευδής και να παρουσιαστεί το αντίστοιχο δέντρο απόδειξης.

---

2. Λεπτομέρειες Αρχείων και Μεθόδων

Αρχείο knowledge_base_2.txt

Το αρχείο περιέχει τη βάση γνώσεων με:

1. Θετικά γεγονότα (π.χ. Loves(John, Mary), Friend(John, Paul)).
2. Κανόνες που συνδέουν γεγονότα με αποτελέσματα
   (π.χ. Loves(John, x) -> Likes(x,IceCream)).

Παραδείγματα:

- Θετικά γεγονότα: Loves(John, Mary), Friend(John, Paul), Respects(Anna, John).
- Κανόνες:
  - Loves(John, x) -> Likes(x, IceCream) : Αν ο John αγαπά κάποιον, τότε αυτός ο κάποιος αγαπάει το παγωτό.
  - Likes(x, IceCream) -> Happy(x) : Αν κάποιος αγαπάει το παγωτό, τότε είναι χαρούμενος.

Αρχείο forward_chaining_2.py

Το αρχείο υλοποιεί τον αλγόριθμο εξαγωγής συμπερασμάτων προς τα εμπρός (forward chaining) και περιλαμβάνει μεθόδους για την ανάγνωση της βάσης γνώσεων, τη δημιουργία του δέντρου απόδειξης και την παρουσίαση των αποτελεσμάτων.

Κύριες Μέθοδοι

Μέθοδος KBread()

Υλοποιεί την ανάγνωση και ανάλυση της βάσης γνώσεων από το αρχείο knowledge_base_2.txt.

Λειτουργία:

1. Αποδέχεται τη διαδρομή ενός αρχείου από τον χρήστη.
2. Ελέγχει για σφάλματα πρόσβασης στο αρχείο.
3. Αναλύει το περιεχόμενο του αρχείου, ξεχωρίζοντας κανόνες από γεγονότα.

Έξοδος:

Μια λίστα μορφής: [{'type': 'rule', 'if': parse('A'), 'then': parse('B')}, {'type': 'fact', 'fact': parse('C')}], για ανάλυση των δεδομένων από την forward_chaining().

Μέθοδος forward_chaining(predicates, query)

Υλοποιεί την εμπρόσθια αλυσιδωτή διαδικιασία για την απόδειξη του ερωτήματος.

Είσοδοι:

- predicates: Η λίστα των κανόνων και γεγονότων από τη βάση γνώσεων.
- query: Το ερώτημα προς απόδειξη.

Λειτουργία:

1. Ξεκινά με τα αρχικά γνωστά γεγονότα.
2. Εξετάζει αν κάποιοι κανόνες μπορούν να ενεργοποιηθούν από τα τρέχοντα γεγονότα.
3. Ενημερώνει τη βάση γνώσης με νέα γεγονότα και επαναλαμβάνει.

4. Ελέγχει αν το ερώτημα περιλαμβάνεται στα θετικά γεγονότα.

Έξοδος:

- True: Αν το ερώτημα αποδεικνύεται.
- False: Αν το ερώτημα δεν αποδεικνύεται.

Μέθοδος unify(predicate, fact)

Εκτελεί την ενότητα (unification) μεταξύ ενός κανόνα και ενός γεγονότος.

Μέθοδος apply_substitution(predicate, substitution)

Εφαρμόζει την αντικατάσταση μεταβλητών σε έναν κανόνα.

---

3. Παραδείγματα Εκτέλεσης

Βάση Γνώσεων:

Loves(John, Mary)
Loves(John, x) -> Likes(x, IceCream)
Likes(x, IceCream) -> Happy(x)

Ερώτημα: Happy(Mary)

Ανάλυση:

1. Ο κανόνας Loves(John, x) -> Likes(x, IceCream) ενεργοποιείται για x = Mary.
2. Το γεγονός Likes(Mary, IceCream) προστίθεται στη βάση.
3. Ο κανόνας Likes(x, IceCream) -> Happy(x) ενεργοποιείται για x = Mary.
4. Το γεγονός Happy(Mary) αποδεικνύεται.

Αποτέλεσμα: Το ερώτημα Happy(Mary) είναι True.

```
Enter the path to the input file: knowledge_base_2.txt
Enter the query predicate: Happy(Mary)

Initial Facts:
- Loves(John, Mary)
- Friend(John, Paul)
- Friend(Mary, Paul)
- Respects(Anna, John)
- Respects(Anna, Paul)
```

2.

```
Enter the path to the input file: knowledge_base_2.txt
Enter the query predicate: Likes(John, IceCream)

Initial Facts:
- Loves(John, Mary)
- Friend(John, Paul)
- Friend(Mary, Paul)
- Respects(Anna, John)
- Respects(Anna, Paul)

The query 'Likes(John, IceCream)' is not derivable (False).
Enter the query predicate or enter '0' to exit:
```