

Inteligență Artificială

Tema 2: *Backward Reasoner*

29 noiembrie 2019

1 Obiectivul temei

Tema extinde noțiunile discutate la curs și la laborator pentru a crea un motor de raționament logic. Programul (în Python) va putea prelucra un fișier text care conține afirmații și interogări, pentru a oferi răspunsuri la interogările date.

- Afirmațiile date vor fi fie atomi, fie reguli având ca premise unul sau mai mulți atomi și un singur atom drept concluzie.
- Interogările vor fi atomi.

Raționamentul se va face folosind **backward chaining**: pornind de la elementele concluziei și demonstrându-le prin găsirea unor reguli cu care să poată fi demonstrate, trecând la demonstrarea premiselor acelor reguli, etc.

2 Cerințe

- **Cerința 1 (2p)**: Dezvoltați o reprezentare eficientă pentru atomi, și citiți fișierul de intrare într-o listă de fapte, reguli, și interogări. Afișați această listă la consolă la rularea programului.
- **Cerința 2 (6p)**: Implementați procesul de backward chaining pentru a putea răspunde interogărilor date.
 - Afișați într-un format lizibil etapele procesului – ce scopuri se încearcă să se demonstreze la un moment dat, și ce reguli sunt încercate pentru a demonstra fiecare scop. Dacă interogarea
 - Dacă interogarea conține variabile, și demonstrația cere ca acestea să fie legate la anumite valori pentru ca scopul să fie demonstrat, se vor afișa aceste legări;
 - Altfel, se va afișa **True** dacă scopul poate fi demonstrat sau **False** dacă scopul nu poate fi demonstrat.
- **Cerința 3 (2p)**: Implementați suport pentru funcții, inclusiv evaluarea lor la constante. Implementați funcțiile conform cu comentariile din teste. La evaluarea funcțiilor în timpul demonstrării afișați rezultatul evaluării în output.
- **Cerința 4 (2p)**: Implementați suport pentru coeficienți de încredere, conform celor studiate la curs. Coeficienții pentru fapte și reguli se vor găsi ca numere după un simbol % ce urmează faptului sau regulii. Dacă o afirmație nu are un coeficient de încredere dat explicit, acesta este considerat 1. La demonstrare afișați și coeficientul obținut pentru fiecare scop și pentru scopul final de demonstrat.

Notă: Cerințele 3 și 4 pot fi implementate independent.

3 Specificații

Tema se va realiza ca un program Python care primește ca unic argument calea către un fișier text care conține afirmații și interogări. Fișierul va respecta următoarea gramatică:

Linie ::= (Afirmație | Interogare) Comentariu | Comentariu

Afirmație ::= Atom | Atom ":" Condiție
Condiție ::= Atom | Atom "," Condiție
Atom ::= Nume "(" Termeni ")" | Nume "()"
Termeni ::= Termen | Termen "," Termeni
Termen ::= Constantă | Variabilă | Funcție
Funcție ::= Nume "(" Termeni ")" | Nume "()"
Variabilă ::= "?" Nume
Nume ::= [A-Za-z_]+
Constantă ::= Număr | [A-Za-z] [A-Za-z0-9]*
Număr ::= "0" | [1-9] [0-9]*

Interogare ::= "?" Atom

Comentariu ::= ("#" | "%" | ":") [A-Za-z0-9 -_]

Este posibil ca fișierul să conțină și linii goale, care vor fi ignorate. De asemenea, poate exista spațiu liber oriunde în afară de: între numele predicatului / funcției și paranteza deschisă; între semnul întrebării și numele variabilei; sau în interiorul numelor de variabile, funcții, predicate, sau în conținutul constantelor.

Interogările din fișier pot fi urmate de rezultatul așteptat al interogării, în comentariu, pentru ușurința testării.

4 Trimiterea temei

Arhiva temei va conține un fișier cu numele `NumePrenumeZZ.py` , formate din numele complet (doar litere mari și mici) și ziua nașterii. Orice alte fișiere necesare vor avea numele începând tot cu `NumePrenumeZZ.py` .