LAB 2 REPORT

Introduction to Artificial Intelligence

I. Group Information

Name	Student ID
Nguyễn Lê Hùng	22127135
Trần Văn Quyết	21127680

II. Accomplished requirements

Requirements overview

Propositional Logic

Stage	Requirement	Proress
1	Đọc dữ liệu đầu vào và lưu trồng cấu trúc dữ liệu phù hợp	100%
2	Cài đặt giải thuật hợp giải trên logic mệnh đề	100%
3	Các bước suy diễn phát sinh đủ mệnh đề và kết luận đúng	100%
4	Tuân thủ mô tả định dạng của đề bài	100%
5	Báo cáo test case và đánh giá	100%

Technical specifications

Tools	Role
Python version >= 3.1	Program structure, backbone bootstrapper for other modules.

III. Propositional Logic

A,The Resolution Algorithm

The Resolution Algorithm is a method to see if a knowledge base (KB) supports a statement (α). It does this by assuming the opposite (not α) is true and trying to prove that this leads to a contradiction.

The algorithm works best with statements in a specific format (CNF), so luckily we don't need to worry about converting the input data (KB and not α) because it's already guaranteed to be in that format.

Resolution works by comparing pairs of statements in the knowledge base. If a pair has opposite ideas (complementary literals), it combines them into a new statement. This new statement is added to the knowledge base as long as it's not already there.

The process continues until one of two things happens:

- We end up with a statement that's always false (empty clause). This means there's a contradiction in the knowledge base, or the knowledge base actually supports the original statement (α).
- No new statements can be created by comparing pairs. This means the knowledge base doesn't support the original statement (α).

Below is the pseudocode code of the resolution algorithm from the book Artificial Intelligence: A Modern Approach (AIMA) Third Edition, Chapter 7, Figure

```
function PL-RESOLUTION(KB, α) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
          α, the query, a sentence in propositional logic

  clauses ← the set of clauses in the CNF representation of KB ∧ ¬α
  new ← { }
  loop do
    for each pair of clauses C_i, C_j in clauses do
      resolvents ← PL-RESOLVE(C_i, C_j)
      if resolvents contains the empty clause then return true
      new ← new ∪ resolvents
    if new ⊆ clauses then return false
    clauses ← clauses ∪ new
```

Figure 7.12 A simple resolution algorithm for propositional logic. The function PL-RESOLVE returns the set of all possible clauses obtained by resolving its two inputs.

B,Test cases

Number	Input	Output
1	D 4 -A OR -B OR C -B OR -C OR D B C	5 -B OR -C -A OR C -C OR D -A OR -B OR D -B OR D

	-	5 -A OR -B -A OR D -B D -C 2 {} -A YES
2	S OR T 6 -P OR Q OR U R OR -S P OR T -Q OR -R -T OR U Q	6 -P OR -R OR U P OR U -R Q OR T OR U -Q OR -S P 5 -R OR T OR U -P OR -S OR U -R OR U -S OR T OR U Q OR U 1 -S OR U 0 NO
3	E 9 A -A OR B -B OR -C OR -D OR E -A OR F -G OR H -F OR G -G OR E C E	10 -B OR -C OR -D -F OR H B {} -B OR -D OR E -A OR -C OR -D OR E F -G -A OR G E OR -F YES
4	H 12 -A OR C -B OR E -A OR D -B OR D	16 -A OR F -A OR -B OR -D OR F D OR -E -A OR -D OR -E OR H -F OR H

	-A OR -E OR -D OR F -E OR B -E OR C -A OR F OR -G -A OR F OR H -E OR -F OR H E F	B -A OR -E OR F -B OR C -A OR -D OR F -E OR -F -A OR -E OR -G OR H -A OR -B OR -E OR F -E OR H -A OR -E OR H -B OR -F OR H C 20 -A OR -G OR H -B OR -F -A OR -E OR -G -F -A OR -D OR -E -A OR -E -A OR -D OR H D -A OR -B OR F -A OR H -A OR -B OR -E -A OR -B OR -G OR H -A OR -B OR H -B OR H -E H -A OR -B OR -E OR H -A OR -B OR -D OR -E -A OR -B OR -D OR H -A OR -B OR -D OR -E OR H 8 -A OR -G -A OR -B -A OR -B OR -D -A {} -B -A OR -D -A OR -B OR -G YES
5	-P OR Q 8 S OR T OR U	11 R OR S OR U -Q OR U

	Q OR -R -P OR T -B OR U R OR -T -S B OR -Q OR U Q OR S	T B OR S OR U S -P OR R T OR U B OR -R OR U -R Q OR -T Q 15 B OR -P OR U S OR U B OR U {} -P OR Q R Q OR U -P -T OR U B OR -T OR U -R OR U -T Q OR S OR U U R OR U YES

C.Evaluate

- The algorithm efficiently operates on data that has been standardized according to conventions.
- The number of propositions generated is relatively large, increasing rapidly when the number of different literals in the initial propositions of the knowledge base is high.

IV. Prolog Logic

Chủ đề: "**Hệ thống giáo dục"**

1. Đối tượng:
    - Học sinh
    - Giáo viên
    - Chương trình học
    - Bộ môn học
    - Chương trình học
2. Quan hệ:
    - Tham gia
    - Dạy
    - Học
3. Cơ sở tri thức (fact):

```
% Students
student(john).
student(emma).
student(liam).
student(olivia).
student(charlie).
student(sophia).
student(noah).
student(isabella).
student(logan).
student(ava).

% Teachers
teacher(mr_smith).
teacher(mrs_jones).
teacher(mr_davis).
teacher(miss_wilson).
teacher(mr_brown).
```

```
% Students and their genders
male(john).
female(emma).
male(liam).
female(olivia).
male(charlie).
female(sophia).
male(noah).
female(isabella).
male(logan).
female(ava).

% Teachers and their genders
male(mr_smith).
female(mrs_jones).
male(mr_davis).
female(miss_wilson).
male(mr_brown).


% Classes
class(ClassName, SubjectName).
class(math_class, math).
class(science_class, science).
class(english_class, english).


% Subjects
subject(math).
subject(science).
subject(english).

% Grades
grade(john, math, 85).
grade(john, science, 92).
grade(emma, math, 55).
grade(emma, science, 33).
grade(emma, english, 51).
grade(liam, math, 78).
grade(liam, science, 80).
```

```
grade(liam, english, 54).
grade(olivia, science, 92).
grade(olivia, english, 90).
grade(charlie, math, 82).
grade(charlie, science, 85).
grade(charlie, english, 78).
grade(sophia, math, 45).
grade(sophia, science, 57).
grade(sophia, english, 67).
grade(noah, math, 88).
grade(noah, english, 90).
grade(isabella, math, 62).
grade(isabella, english, 31).
grade(logan, math, 78).
grade(logan, science, 85).
grade(logan, english, 48).
grade(ava, math, 95).
grade(ava, science, 92).

enrolled(john, math_class).
enrolled(john, science_class).
enrolled(emma, math_class).
enrolled(emma, science_class).
enrolled(emma, english_class).
enrolled(liam, math_class).
enrolled(liam, science_class).
enrolled(liam, english_class).
enrolled(olivia, science_class).
enrolled(olivia, english_class).
enrolled(charlie, math_class).
enrolled(charlie, science_class).
enrolled(charlie, english_class).
enrolled(sophia, math_class).
enrolled(sophia, science_class).
enrolled(sophia, english_class).
enrolled(noah, math_class).
enrolled(noah, english_class).
enrolled(isabella, math_class).
enrolled(isabella, english_class).
enrolled(logan, math_class).
```

enrolled(logan, science_class).
enrolled(logan, english_class).
enrolled(ava, math_class).
enrolled(ava, science_class).

teaches(mr_smith, math).
teaches(mrs_jones, math).
teaches(mrs_jones, science).
teaches(mr_davis, english).
teaches(miss_wilson, math).
teaches(miss_wilson, english).
teaches(mr_brown, science).
teaches(mr_brown, english).

activity(football).
activity(basketball).
activity(chess_club).
activity(debate_team).
activity(drama_club).

participates_in(john, football).
participates_in(emma, chess_club).
participates_in(liam, basketball).
participates_in(olivia, debate_team).
participates_in(john, debate_team).
participates_in(charlie, drama_club).
participates_in(charlie, football).
participates_in(charlie, basketball).
participates_in(sophia, football).
participates_in(noah, chess_club).
participates_in(isabella, basketball).
participates_in(emma, basketball).
participates_in(logan, debate_team).
participates_in(ava, drama_club).

4. Vị từ:
    - male_student(Student): nam sinh
    - female_student(Student): nữ sinh
    - female_teacher(Teacher): cô giáo
    - male_teacher(Teacher): thầy giáo

- enrolled_in_subject(Student, Subject): Quan hệ này xác định xem một học sinh có được đăng ký vào một môn học cụ thể hay không.
- passed(Student, Subject): Quan hệ này xác định xem một học sinh đã vượt qua một môn học cụ thể hay không, dựa trên điểm số (điểm số lớn hơn hoặc bằng 50).
- high_grade(Student, Subject): Quan hệ này xác định xem một học sinh có đạt điểm cao (điểm số lớn hơn hoặc bằng 90) trong một môn học cụ thể hay không.
- needs_tutoring(Student, Subject): Quan hệ này xác định xem một học sinh có cần hướng dẫn (nếu điểm số của học sinh dưới 65) trong một môn học cụ thể hay không.
- fails(Student, Subject): Quan hệ này xác định xem một học sinh đã rớt một môn học cụ thể hay không, dựa trên điểm số (điểm số dưới 50).
- teaches_multiple_subjects(Teacher) : Quan hệ này xác định xem một giáo viên có giảng dạy nhiều môn học hay không.
- assessment(Student, Subject, AssessmentType, Score): Quan hệ này xác định điểm số của một học sinh trong một bài kiểm tra (test) trong một môn học cụ thể.
- class_roster(Class, Roster) : Quan hệ này trả về danh sách các học sinh đã đăng ký vào một lớp cụ thể.
- teaches_class(Teacher, Class) : Quan hệ này xác định xem một giáo viên có dạy một lớp cụ thể hay không.
- class_information(Class, Teacher) : Quan hệ này cung cấp thông tin về giáo viên và môn học được giảng dạy trong một lớp cụ thể.
- class_size(Class, Size) : Quan hệ này xác định kích thước của một lớp cụ thể (số lượng học sinh).
- classes_taught_by_teacher(Teacher, Classes) : Quan hệ này trả về danh sách các lớp mà một giáo viên dạy.
- class_average_grade(Class, Subject, AverageGrade) : Quan hệ này tính toán điểm trung bình của một lớp trong một môn học cụ thể.
- students_participating_in_activity(Activity, Students) : Quan hệ này trả về danh sách các học sinh tham gia một hoạt động ngoại khóa cụ thể.
- students in_class_with_grade_above(Class, Subject, Grade, Students) : Quan hệ này trả về danh sách các học sinh trong một lớp có điểm số cao hơn một ngưỡng nhất định trong một môn học cụ thể.
- teachers_teaching_subject(Subject, Teachers) : Quan hệ này trả về danh sách các giáo viên dạy một môn học cụ thể.
- student_activities(Student, Activities) : Quan hệ này trả về danh sách các hoạt động ngoại khóa mà một học sinh tham gia.

- student_grade_above_average(Student, Subject) : Quan hệ này xác định xem một học sinh có điểm số cao hơn điểm trung bình của lớp trong một môn học cụ thể hay không.
- students_taking_subject(Subject, Students) : Quan hệ này trả về danh sách các học sinh đang học một môn học cụ thể.
- passed_subjects(Student, PassedSubjects) : Quan hệ này trả về danh sách các môn học mà một học sinh đã vượt qua.
- students_passing_class(Class, PassedStudents) : Quan hệ này trả về danh sách các học sinh đã vượt qua một lớp cụ thể.
- teacher_average_grade(Teacher, Subject, AverageGrade) : Quan hệ này tính toán điểm trung bình của một giáo viên dạy trong một môn học cụ thể.
- studentactivitiescount(Student, Count) : Quan hệ này đếm số lượng hoạt động ngoại khóa mà một học sinh tham gia.
- student_subjects_count(Student, Count) : Quan hệ này đếm số lượng môn học mà một học sinh đang học.
- student_grades(Student, Grades) : Quan hệ này trả về danh sách các điểm số của một học sinh trong các môn học.
- student_average_grade(Student, AverageGrade) : Quan hệ này tính toán điểm trung bình của một học sinh trong tất cả các môn học.

V. Reference

Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig (Chapter 3.3.2) ([URLaima3e stanford ON Stanford University people.cs.stanford.edu])