University of Asia Pacific

Report No: 01

Report Name: University Course Prerequisite & Scheduling System

Submitted to

Noor Mairukh Khan Arnob

Lecturer

Department of Computer Science & Engineering

Submitted by

Md. Arifur Rahman Akash

Roll-21201091, Section – B2, 52nd Batch,

Department of Computer Science & Engineering

Course Title: Artificial Intelligence and Expert Systems Lab

Course Code: CSE 404

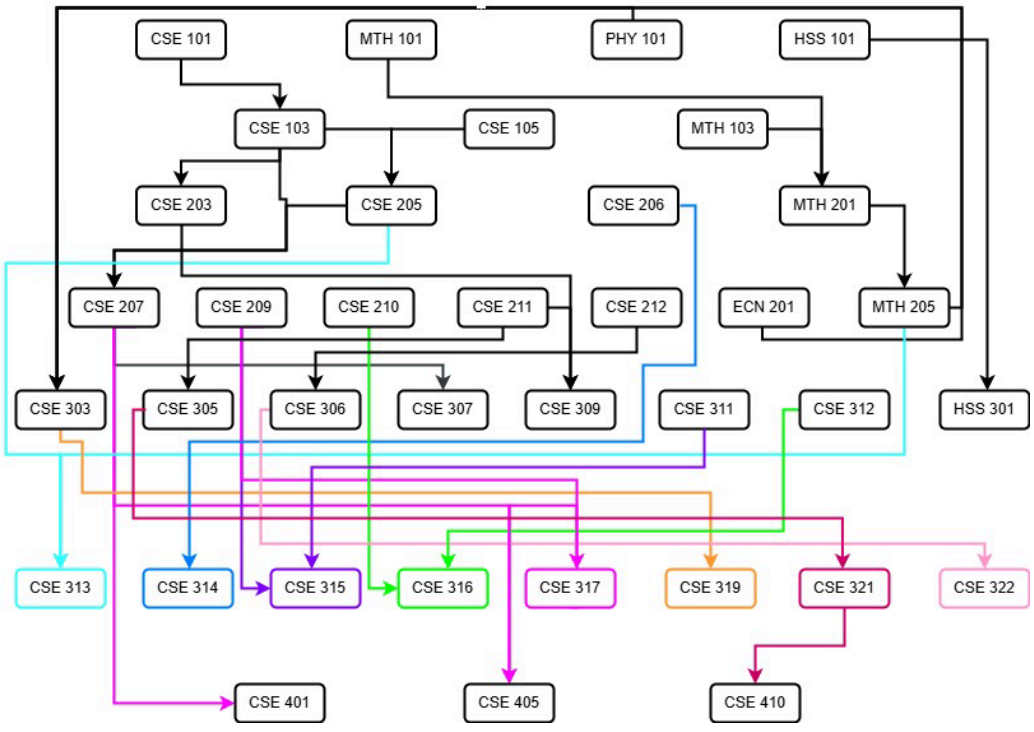
Problem Title: University Course Prerequisite & Scheduling System

Problem Description: Managing university course enrollments efficiently requires a system that ensures students meet all prerequisites before registering for advanced courses. This system automates prerequisite checking, course availability determination, and course recommendations. Using Prolog, we can model course dependencies, check eligibility, and provide an automated course planner.

Tools and Languages:

- **Programming Language**: Prolog
- **Development Environment**: SWI-Prolog
- **Data Representation**: Facts and Rules
- **Logic Processing**: Recursive queries for course eligibility checking

Dependency Diagram:



Sample Queries & Output

1. List courses with no prerequisites

```
?- no_prereq_course(Course).
Course = cse101 ;
Course = phy101 ;
Course = mth101 ;
Course = cse102 ;
Course = hss101 ;
Course = hss111a ;
Course = hss111b ;
Course = phy102 ;
Course = cse104 .
```

2. Check if Arif has completed CSE101

```
?- has_completed(arif, cse101).
true
```

3. Determine if Sami is eligible for CSE205

```
?- eligible(sami, cse205).
true
```

4. Retrieve all (direct and indirect) prerequisites for CSE205

```
?- prereq_chain(cse205, Prereq).
Prereq = cse101 ;
Prereq = cse103 ;
Prereq = cse105 .
```

5. List all courses Arafat can potentially take next

```
?- next_possible_course(arafat, Course).
Course = cse203 ;
Course = cse207 .
```

6. Calculate the total credits Sami has completed

```
?- credits_completed(sami, TotalCredits).
TotalCredits = 6.0.
```

7. Calculate the total credits of the entire program

```
?- program_total_credits(Total).
Total = 155.25.
```

8. Determine if Arafat is eligible for CSE205

```
?- eligible(arafat, cse205).
true .
```

9. Retrieve all courses with credit value less than 2

```
?- course(Course, Credit), Credit < 2.
Course = cse102,
Credit = 1.5 ;
Course = phy102,
Credit = 1.5 ;
Course = cse104,
Credit = 1.5 ;
Course = eee122,
Credit = 1.5 ;
Course = chem112,
Credit = 1.5 ;
Course = cse204,
Credit = 1.5 ;
Course = cse206,
Credit = 1.5 ;
Course = eee222,
Credit = 1.5 ;
Course = cse208,
Credit = 1.5 ;
Course = cse210,
Credit = 1.5 ;
Course = cse212,
Credit = 1.5 ;
Course = cse304,
Credit = 0.75 ;
Course = cse306,
Credit = 0.75 ;
Course = cse310,
Credit = 1.5 ;
Course = cse312,
Credit = 1.5 ;
Course = cse320,
Credit = 1.5 ;
Course = cse322,
Credit = 0.75 ;
Course = cse330,
Credit = 1.5 ;
Course = cse314,
Credit = 0.75 ;
Course = cse316,
Credit = 1.5 ;
Course = cse404,
Credit = 1.5 ;
Course = cse406,
Credit = 1.5 ;
Course = cse410,
Credit = 1.5 ;
Course = cse426,
Credit = 1.5 ;
Course = cse430,
Credit = 1.5 ;
Course = bus402,
Credit = 0.75.
```

10. Retrieve all courses with exact value of 3 credit

```
?- course(Course, 3.0).
Course = cse101 ;
Course = phy101 ;
Course = mth101 ;
Course = hss101 ;
Course = cse103 ;
Course = cse105 ;
Course = eee121 ;
Course = mth103 ;
Course = chem111 ;
Course = cse203 ;
Course = cse205 ;
Course = mth201 ;
Course = mth203 ;
Course = cse207 ;
Course = cse211 ;
Course = mth205 ;
Course = cse303 ;
Course = cse305 ;
Course = cse307 ;
Course = cse309 ;
Course = cse311 ;
Course = cse317 ;
Course = cse319 ;
Course = cse321 ;
Course = cse313 ;
Course = cse315 ;
Course = cse401 ;
Course = cse403 ;
Course = cse405 ;
Course = cse427 ;
Course = cse400 ;
Course = cse425 ;
Course = cse429 ;
Course = bus401.
```

Conclusion

The Prolog course scheduling system effectively models courses, prerequisites, and student progress. It supports direct and recursive queries, enabling dynamic checks on eligibility and credit calculations.

Challenges

- **Recursion:** Ensuring termination without infinite loops.
- **Data Completeness:** Incomplete facts can yield unexpected results.
- **Scalability:** Performance may degrade as data size increases.