

COMP 6591: Introduction to Knowledge-Base Systems

Assignment 2

Summer 2022, sections CC

July 12, 2022

Contents

1	General Information	2
2	Introduction	2
3	Ground rules	2
4	Overview	2
4.1	System Setup	3
4.2	Relational Model	3
4.3	Knowledge Base Items	3
5	Your Assignment	4
5.1	Creating the Knowledge Base in Prolog	4
5.2	Knowledge-Base Queries	4
5.3	The Interactive Client in Java	5
6	What to Submit	5
7	Grading Scheme	6

1 General Information

Date posted: Tuesday, July 12th, 2022.

Date due: Friday, July 22nd, 2022, by 23:59.¹.

Weight: 8% of the overall grade.

2 Introduction

This assignment targets Dynamic Knowledge Representation and Queries in both relational and deductive models. In this assignment you implement a small knowledge-base system that demonstrates dynamic entities and queries.

3 Ground rules

You work on a team of maximum 2 students(including yourself). Each team should designate a leader who will submit the assignment electronically. See Submission Notes for the details. ONLY one copy of the assignment is to be submitted by the team leader. Upon submission, you must book an appointment with the marker team and demo the assignment. All members of the team must be present during the demo to receive the credit. Failure to do so may result in zero credit.

This is an assessment exercise. You may not seek any assistance from others while expecting to receive credit (**You must work strictly within your team**). Failure to do so will result in penalties or no credit.

4 Overview

In the previous assignment, you implemented a Knowledge-base on movie databases. In this assignment, we use the same KB by letting the user add / remove facts from the KB. Additionally, we use an interactive system to execute queries on the KB and report the result to the user.

¹see submission notes

4.1 System Setup

This assignment requires **sqlite**, **swi-prolog**, and **swi-prolog-java** which required **jdk** being installed on the system See the lecture notes for instructions how to install them.

4.2 Relational Model

In this assignment using relational model is not mandatory. You may use `sqlite` or `prolog`'s dynamic facts. You may also revised your KB design so that the queries are easily implemented.

4.3 Knowledge Base Items

The knowledge-base consists of movie information, as follows. Certain elements from the previous assignment have been removed, however you may keep them.

- Movie titles: including the IMDB id, name, description / plot, etc.
- Cast (names)
- Keywords (a few common keywords per movie)
- Genres
- Languages

All above items are dynamic, except for *Genres* and *Languages*. This means a movie is allowed to be listed under certain genres and/or languages, as listed in your knowledge-base. Like the previous assignment, make sure the following data-set requirements are met:

1. include at least 10 titles.
2. include at least two movies with more than one language.
3. include at least 3 cast members per movie.
4. include at least 5 keywords per movie; make sure some movies share one or two keywords.
5. define 3 genres, and 3 languages.

Do not include any TV-series.

5 Your Assignment

Your assignment consist of three parts: (i) Creating the Knowledge Base in Prolog, (ii) Knowledge-Base Queries, (iii) The Interactive Client in Java.

It is strongly recommended that you use command line interface (CLI) for both parts.

5.1 Creating the Knowledge Base in Prolog

Create the KB with regards to the specifications, as listed under section 4.3. Your prolog code must contain all necessary queries that allows a user to *add*, *update*, and *remove* KB items.

Note that this assignment is **not** a database system. Hence, in case of using a relational database, no SQL interface must be exposed by these queries.

Below are examples of a queries that updates the KB:

```
update_movie_title(TT, TITLE) :- ...
    % where TT is the imdb id
    % and title is the new title.

add_movie(TT, TITLE, ...) :-
    % adds the movie to the KB
    % and returns true.
    % fails if duplicate id exists.
```

5.2 Knowledge-Base Queries

Prepare some queries that demonstrates using your KB, created in 5.1. Include some queries that demonstrates violating the constrains: i.e. adding an unknown language, or removing a non-existing movie title. Such queries mush fail.

Submit this list alongside with your assignment, in a separate file.

5.2.1 Categorical Syllogisms

Additionally, prepare some queries and/or rules that demonstrates querying Categorical Syllogisms: \mathbb{A} , \mathbb{E} , \mathbb{I} , and \mathbb{O} . Provide examples for successes and failures (8 queries, in total).

5.3 The Interactive Client in Java

Create a simple client that consults your KB and searches for a movie title. Allow user to add / modify movie titles². Demonstrate the change.

The interactive client strictly uses JPL to communicate with SWI-Prolog. No GUI is required. You may create a simple Java Console Application.

6 What to Submit

The whole assignment is submitted by the due date under the corresponding assignment box. It has to be completed by ALL members of the team in one submission file.

Submission Notes

Clearly include the names and student IDs of all members of the team in the submission. Indicate the team leader.

IMPORTANT: You are allowed to work on a team of 2 students at most (including yourself). Any teams of 3 or more students will result in 0 marks for all team members. If your work on a team, ONLY one copy of the assignment is to be submitted. You must make sure that you upload the assignment to the correct assignment box on Moodle. No email submissions are accepted. Assignments uploaded to the wrong system, wrong folder, or submitted via email will be discarded and no resubmission will be allowed. Make sure you can access Moodle prior to the submission deadline. The deadline will not be extended.

Naming convention for the uploaded file: Create one zip file, containing all needed files for your assignment using the following naming convention. The zip file should be called a#_studids, where # is the number of the assignment, and studids is the list of student ids of all team members, separated by (_). For example, for the first assignment, student 12345678 would submit a zip file named a1_12345678.zip. If you work on a

²You need to demonstrate this on movie titles only, i.e. the title and plot only.

team of two and your IDs are 12345678 and 34567890, you would submit a zip file named `a1_12345678_34567890.zip`.

Submit your assignment electronically on Moodle based on the instruction given by your instructor as indicated above: <https://moodle.concordia.ca>

Please see course outline for submission rules and format, as well as for the required demo of the assignment. A working copy of the code and a sample output should be submitted for the tasks that require them. A text file with answers to the different tasks should be provided. Put it all in a file layout as explained below, archive it with any archiving and compressing utility, such as WinZip, WinRAR, tar, gzip, bzip2, or others. You must keep a record of your submission confirmation. This is your proof of submission, which you may need should a submission problem arises.

7 Grading Scheme

Dynamic Knowledge Base in Prolog (5.1)	25 pts
Update Queries (5.1)	20 pts
KB Queries (5.2)	15 pts
Categorical Syllogisms (5.2.1)	16 pts
Interactive Client (5.3)	24 pts

Total: 100 pts.

References

1. SWI-Prolog: <https://www.swi-prolog.org/Download.html>
2. proSQLite: Prolog File Based Databases via an SQLite Interface:
<https://link.springer.com/content/pdf/10.1007/978-3-642-45284-0.pdf>
3. proSQLite on SWI-Prolog: <https://www.swi-prolog.org/pack/list?p=prosqlite>
4. JPL: <https://jpl7.org/>
5. The Java API: <https://jpl7.org/JavaApiOverview>
6. Categorical Syllogisms: <http://www.philosophypages.com/lg/e08a.htm>