Knowledge Representation and Reasoning

Individual Reflection

Murthy Kanuri

Student ID: 12696139

Table of Contents

1. Introduction

Before starting the Knowledge Representation and Reasoning **(**KRR)module, I had some understanding of how data converts to knowledge or how machines can mimic human reasoning. Throughout all twelve units, I progressed from learning basic definitions to building complex ontologies using industry-grade tools like Protégé (**Stanford Center for Biomedical Informatics Research, 2024**). In this reflective piece, I will summarise my learning process — the challenges I faced, the feedback I received and the skills I acquired, including critical thinking, collaboration, ethical awareness, and digital literacy.

2. Summary of Learning Outcomes

Unit 1 provided basic conceptual clarity on the difference between data, information and knowledge, including the significance of symbolic representation in AI. I learned core definitions, knowledge being a "collection of propositions believed by some putative agent," and reasoning as one's ability to derive new knowledge from existing beliefs (Brachman & Levesque, 2004). Frameworks like Dennett's "intentional stance" helped me relate the machine's behaviour to human terms, making abstract concepts more relatable.

Unit 2 strengthened my foundational skills in set theory and truth tables, which later supported more complex logic. In Unit 3, I learned more about propositional and predicate logic and how logic-based languages like Prolog and Lisp codify symbolic rules. I also briefly experimented with logic-based tools like SWI-Prolog, which helped me understand how these rules can be executed in practice. Applying modus ponens and modus tollens to solve logic puzzles also deepened my analytical thinking. These activities developed my ability to think logically under pressure, a valuable transferable skill.

Unit 4 introduced the practical application of **Protégé**. I became familiar with its declarative style and experimented with knowledge graphs and DL queries. This helped me understand how logical structures can support explainable AI.

Unit 5 covered modelling concepts like Z notation, conceptual ontologies, and probabilistic inference techniques such as Bayesian networks and Monte Carlo simulations (Russell and Norvig, 2016). These tools helped me understand how intelligent systems handle uncertainty and make data-driven decisions.

In Unit 6, I used Protégé to create ontologies and started understanding how to organise concepts like classes, properties, and individuals using Description Logic(DL). I made a few mistakes initially with domain and range declarations, but those errors helped me realise how inference is driven by well-structured ontologies. I began to appreciate how foundational modelling is for system transparency and how minor structural errors can compromise reasoning outcomes in real-world applications.

Unit 7 explored different types of knowledge—declarative, procedural, and meta—and how they can be acquired through methods like interviews and observation. What stood out was the ethical dimension — especially for AI systems that make decisions affecting people (Sowa, 2000). It made me more aware of my responsibility as a future developer.

During Unit 8, I gained more confidence in designing structured ontologies and began to see Protégé as part of a broader process involving data exploration and semantic modelling. I also started identifying like semantic overlap and ambiguity, and how they could impact quality of reasoning (Kalibatiene and Vasilecas, 2011).

Unit 9 expanded on formal logic through DL and OWL, where I learned about TBox, ABox, and RBox, and how they shape ontology-driven systems. I found concepts like entailment, open-world assumption particularly inspiring, especially in how tools like as Protégé apply reasoning to real-world data.

Unit 10 provided insights into how assumptions such as open or closed worlds affect inference and system behaviour. I realised that reasoning in AI is not just technical but careful logical design (Davis et al., 1993).

Unit 11 covered KRR applications in engineering systems, GIS, and software testing. I was particularly interested in real-life projects such as Virtual Factory Framework and SEMA, which use ontologies to manage distributed data and enable cross-domain collaboration (Sowa, 2000). Seeing these examples helped me understand the impact of semantic technologies in professional environments.

Finally, Unit 12 focussed on ontology evaluation. I learned that beyond designing, it must be evaluated against performance, consistency, and real-world usability.

Appendix B includes the exercises, formative activities, and collaborative discussion notes for reference.

3. Feedback from Peers and Tutors

The seminars led by our tutor, Samuel Danso, significantly enhanced my understanding. I enaged with the tutor during the sessions to clarify queries and deepen my learning.

Peer collaboration was vital, particularly when using tools like protégé. We participated in six weeks of collaborative discussions, which broadened my perspective.

In one discussion on the evolution of Knowledge representation, a peer mentioned that KR predated computers, referencing Aristotle. This challenged my earlier assumptions. I reflected on how computing advanced KR while its conceptual roots remain ancient, supported by Davis et al. (1993) and Sowa (2000). Another debate focused on the necessity of reasoning in KR. Peers argued that without inference, KR becomes static. I agreed and cited Brachman and Levesque (2004) to reinforce this idea, particularly in relation to our DL and SPARQL exercises.

Another discussion compared ontology languages like OWL2, RDF, KIF and OWL-Lite. Initially, I favoured RDF for its simplicity. However, after engaging in peer discussions and reading Kalibatiene & Vasilecas (2011), I shifted my preference to OWL2, which offers richer semantics and reasoning capabilities. This shift influenced me to structure my final ontology project , ensuring it was more semantically expressive and compatible with reasoning tools.

I also contributed to the module wiki by sharing useful resources with peers, enhancing collaborative learning. This experience improved my ability to explain technical concepts clearly and reinforced the importance of knowledge sharing in collaborative environments.

4. Professional Skills Matrix and Action Plan (PDP)

Throughout the module, I developed key academic and professional skills. The hands-on tasks and discussions helped improve my digital literacy, logical reasoning, and ethical awareness.

My self-assessed PDP (Appendix A) outlines these skills as well as my current confidence levels, my supporting evidence, and my future development actions.

5. Conclusion

In reflection, this module has helped me build both academic understanding and hands-on skills in knowledge representation and reasoning. I have learnt how to model knowledge, question data critically, and design logical-based systems that are explainable. Most importantly, I have discovered that a good KRR design balances precision with usability, ensuring systems are technically robust, meaningful, and accessible for users. Moving forward, I would like to develop this foundation by exploring linked data, semantic interoperability, and using reasoning to support ethical decision-making in AI systems.

6. References

Brachman, R.J. and Levesque, H.J. (2004) *Knowledge Representation and Reasoning*. Amsterdam: Elsevier.

Davis, R., Shrobe, H. and Szolovits, P. (1993) ‘What is a knowledge representation?’, *AI Magazine*, 14(1), pp. 17–33.

Kalibatiene, D. and Vasilecas, O. (2011) ‘Survey on Ontology Languages’, in Grabis, J. and Kirikova, M. (eds) *Business Informatics Research*. Berlin, Heidelberg: Springer. *Lecture Notes in Business Information Processing*, vol 90.

Russell, S. and Norvig, P. (2016) *Artificial Intelligence: A Modern Approach*. 3rd ed. Harlow: Pearson.

Sowa, J.F. (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks/Cole.

Stanford Center for Biomedical Informatics Research (2024) *Protégé* [software]. Available at: https://protege.stanford.edu/ [Accessed 16 Apr. 2025].

SWI-Prolog downloaded from https://www.swi-prolog.org/download/stable.

Kanuri, M. (2025) *GitHub e-portfolio*. Available at: https://m-kanuri.github.io/ (Accessed: 16 April 2025).

Kanuri, M. (2025) *Other Modules Learning Matrix*. Available at: https://m-kanuri.github.io/artefacts/Other_Modules_Learning_Matrix.pdf (Accessed: 17 April 2025).

Appendix A: Personal Skills matrix and action plan (PDP)

SKILL	LEVEL	EVIDENCE	ACTION PLAN
TIME MANAGEMENT	Expert	Meeting deadlines for all the weekly activities during the course. Using modern tools like Trello, Sticky Notes, Personal calendar reminders. 25 years of professional experience in managing and delivering the projects.	Continue delivering on time with quality and maintaining consistency in results.
CRITICAL THINKING AND ANALYSIS	Developing	Academic reading and applying the principles in practice. Assessing the topics with peers to gain wider knowledge.	Continue comparing academic frameworks and reflect critically in usage.
COMMUNICATION AND LITERACY	Proficient	Discussion with peers during the collaborative discussions. Achieved good percentage in academic assignments. In professional experience did presentations to wider audience across the globe.	Practice explaining KRR concepts. Engaging proactively in academic discussions and forums.
IT AND DIGITAL	Proficient	Learning and using new tools such as Protégé and Prolog. Industry experience of using new tools and technologies.	Applying Protégé in real time professional projects by May 2025. Experimenting with SWRL rule creation, advanced SPARQL queries, and semantic reasoning techniques.

SKILL	LEVEL	EVIDENCE	ACTION PLAN
NUMERACY	Proficient	First order logic (FOL) Statements, Truth Tables, SPARQL and Prolog queries	Applying FOL and Truth table logic for practical problem solving. Writing complex SPARQL queries
RESEARCH	Developing	Using Academic sources during the course where needed. Using the university digital library and online.	Expand research by reading more Journals in AI and critically evaluate sources for bias, credibility and relevance.
INTERPERSONAL	Proficient	Actively involved in all the collaborative discussions. Shared and Received feedback from tutor and from peers.	Leading the future discussion groups and providing summarised feedback.
PROBLEM SOLVING	Proficient	Debugging & Resolving issues when using the new tools. For example, OWLviz graph is not visible unless the plugins not installed, Correcting Syntax errors in SPARQL Queries. Professionally worked in large transformation programmes and solved many technical and non-technical issues.	Handling complex KR scenarios with structured debugging and reasoning. Manage a personal fix log and share it with in discussion threads.
ETHICAL AWARENESS	Aware	During the coursework understood the bias in KR system	I aim to complete professional development courses in AI ethics by August 2025, using platforms like Coursera or EdX to deepen my understanding.

7. Appendix B: e-Portfolio Links

Module Week / Section	Links & Artefacts
e-portfolio main page	**https://m-kanuri.github.io/**
KRR Module	https://m-kanuri.github.io/Module4.html
Unit 1	• **Main Page:** https://m-kanuri.github.io/module4/2025/02/01/KRR-Unit1.html • **Formative Activity 1 & 2:** https://m-kanuri.github.io/artefacts/KRR-Unit1-FormativeActivities.pdf • **Initial Post : https://m-kanuri.github.io/artefacts/KRR-Unit01-InitialPost.pdf** • **Peer Response: https://m-kanuri.github.io/artefacts/KRR-Unit01-Peer_Response.pdf** • **Summary Post: https://m-kanuri.github.io/artefacts/KRR-Unit01-SummaryPost.pdf**
Unit 2	• **Main Page:** https://m-kanuri.github.io/module4/2025/02/08/KRR-Unit2.html • **Exercises:** https://m-kanuri.github.io/artefacts/KRR-Unit2-Excercises.pdf
Unit 3	• **Main Page: https://m-kanuri.github.io/module4/2025/02/15/KRR-Unit3.html** • **Formative Activities:** https://m-kanuri.github.io/artefacts/KRR-Unit3-FormalActivities.pdf
Unit 5	• **Main Page: https://m-kanuri.github.io/module4/2025/03/01/KRR-Unit5.html** • **Activity 1, 2 & 3 : https://m-kanuri.github.io/artefacts/KRR-Unit5-FormalActivities.pdf**
Unit 6	• **Main Page: https://m-kanuri.github.io/module4/2025/03/08/KRR-Unit6.html** • **Formative Activities : https://m-kanuri.github.io/artefacts/KRR-Unit6-FormalActivities.pdf**
Unit 7	• **Main Page:** https://m-kanuri.github.io/module4/2025/03/14/KRR-Unit7.html • **Case Study Review : https://m-kanuri.github.io/artefacts/KRR-Unit7-CaseStudyReview.pdf**
Unit 8	• **Main Page:** https://m-kanuri.github.io/module4/2025/03/21/KRR-Unit8.html • **Formative Activities: https://m-kanuri.github.io/artefacts/KRR-Unit8-FormalActivities.pdf**
Unit 9	• **Main Page: https://m-kanuri.github.io/module4/2025/03/28/KRR-Unit9.html** • **Initial Post : https://m-kanuri.github.io/artefacts/KRR-Unit09-InitialPost.pdf**

Module Week / Section	Links & Artefacts
	• **Peer Response:** https://m-kanuri.github.io/artefacts/KRR-Unit09-Peer_Response.pdf • **Summary Post:** https://m-kanuri.github.io/artefacts/KRR-Unit09-SummaryPost.pdf
Unit 10	• **Main Page:** https://m-kanuri.github.io/module4/2025/04/04/KRR-Unit10.html • **Formative Activities:** https://m-kanuri.github.io/artefacts/KRR-Unit10-FormalActivities.pdf
Unit 11	• **Main Page:** https://m-kanuri.github.io/module4/2025/04/11/KRR-Unit11.html • **Modelling Assignment :** https://m-kanuri.github.io/artefacts/KRR-Unit11-ModellingAssignment.pdf • **GitHub :** https://github.com/m-kanuri/m-kanuri.github.io/blob/main/KRR/Unit11/JobMatching.rdf
Unit 12	• **Main Page:** https://m-kanuri.github.io/module4/2025/04/17/KRR-Unit12.html • **Individual Reflection :**

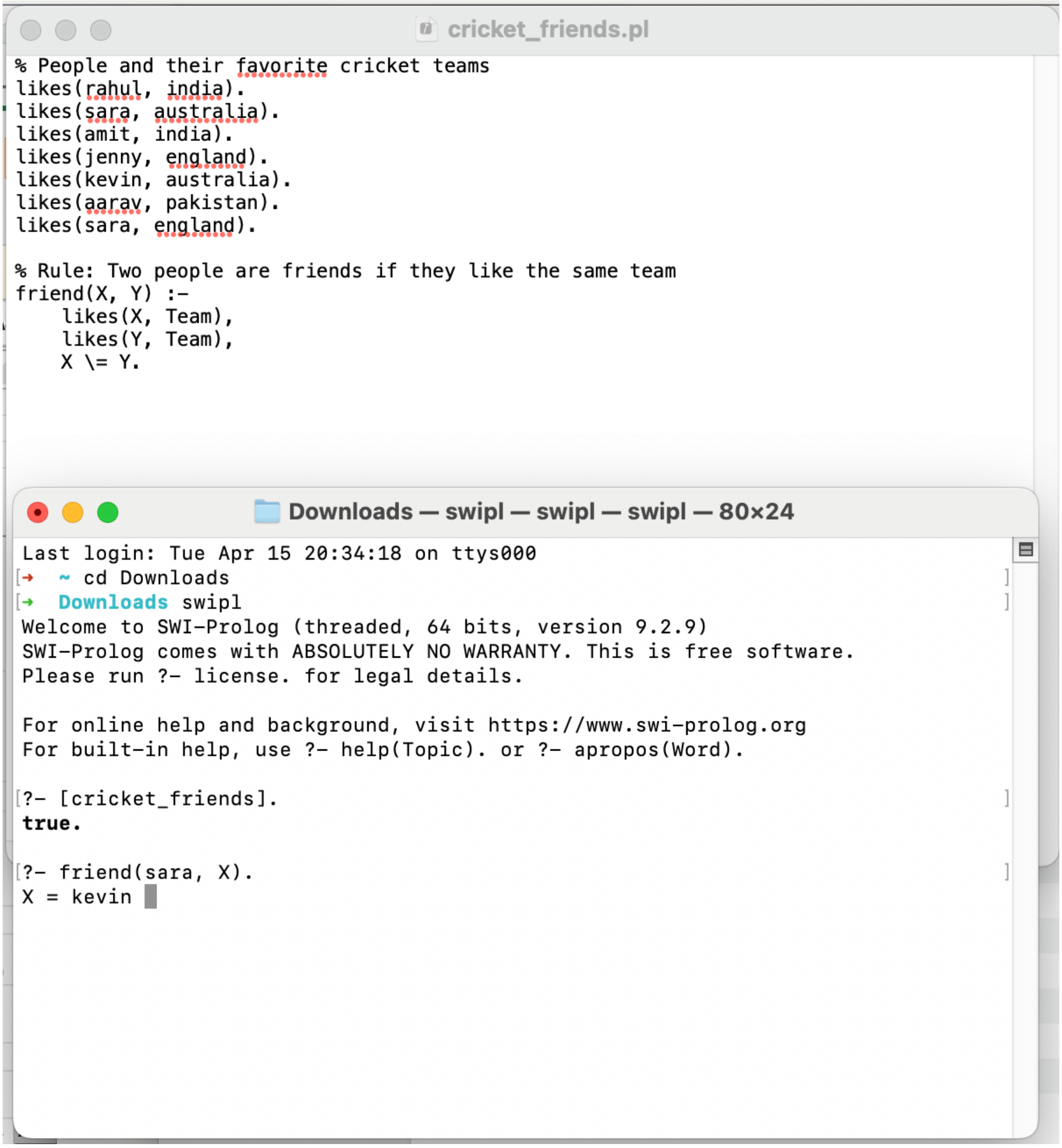
8. Appendix C: Modules Learning Matrix

Please refer to my overall course learning matrix

- https://github.com/m-kanuri/m-kanuri.github.io/blob/main/artefacts/MscAI_LearningMatric_MurthyKanuri.pdf

9. Appendix D: Using Prolog

Prolog program that recommends possible friendships based on favourite cricket teams. If two people are fans of the same team, they are friends. I thought it was interesting that Prolog can figure out which matches to return without the use of any loops or if-statements. Doing this helped me visualize how logic programming functions, where relationships can be expressed by simple rules.


cricket_friends.pl

```
% People and their favorite cricket teams
likes(rahul, india).
likes(sara, australia).
likes(amit, india).
likes(jenny, england).
likes(kevin, australia).
likes(aarav, pakistan).
likes(sara, england).

% Rule: Two people are friends if they like the same team
friend(X, Y) :-
    likes(X, Team),
    likes(Y, Team),
    X \= Y.
```

Downloads — swipl — swipl — swipl — 80×24

```
Last login: Tue Apr 15 20:34:18 on ttys000
➜  ~ cd Downloads
➜  Downloads swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [cricket_friends].
true.

?- friend(sara, X).
X = kevin
```

Prolog program matches persons with hobbies. It's quite simple — if two people like the same thing, and those two people are not the same person, they are considered "hobby buddies." This made Prolog a good fit because you only have to define facts and one rule, and it works out the rest. This taught me that logic programming can be used to find patterns in data without a complicated code.

hobbies.pl

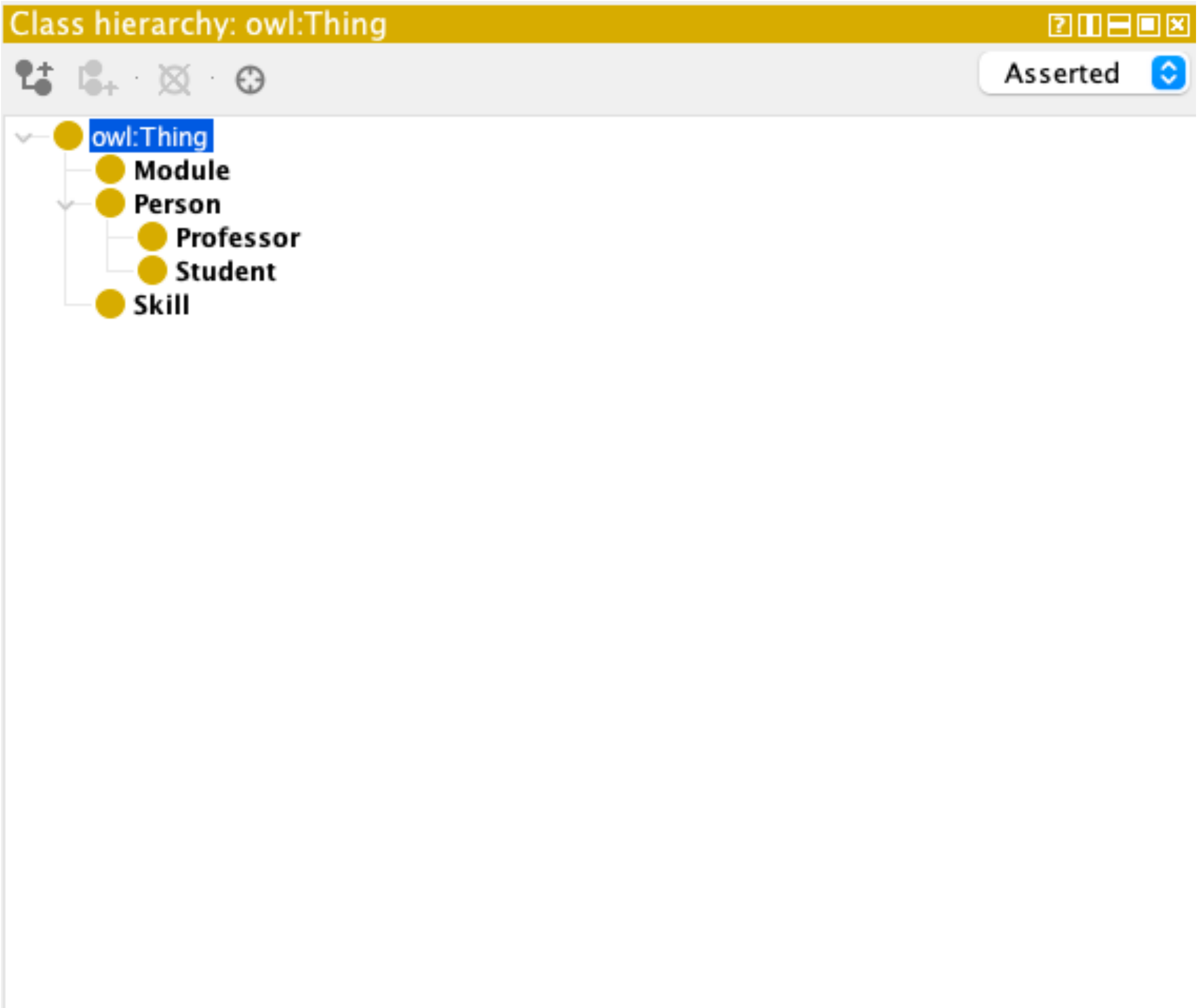
```
% People and their hobbies
likes(ana, painting).
likes(ben, football).
likes(claire, painting).
likes(dan, chess).
likes(emma, football).
likes(frank, chess).

% Rule: People are hobby buddies if they share a hobby
hobby_buddy(X, Y) :-
    likes(X, Hobby),
    likes(Y, Hobby),
    X \= Y.
```

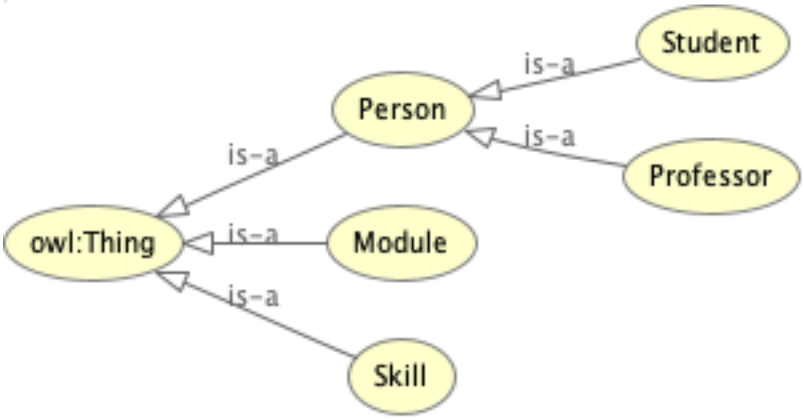
Downloads — swipl — swipl — swipl — 82×25

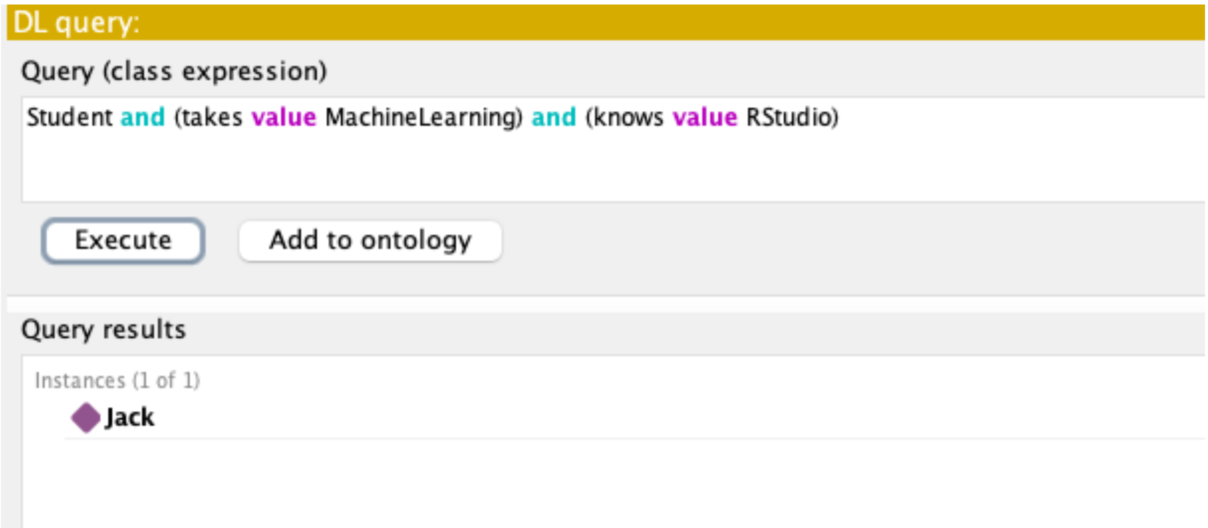
```
Last login: Tue Apr 15 20:39:13 on ttys000
→  ~ cd Downloads
→  Downloads swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [hobbies].
true.

?- hobby_buddy(ana, X).
X = claire.

?- hobby_buddy(ben, X).
X = emma.

?-
```

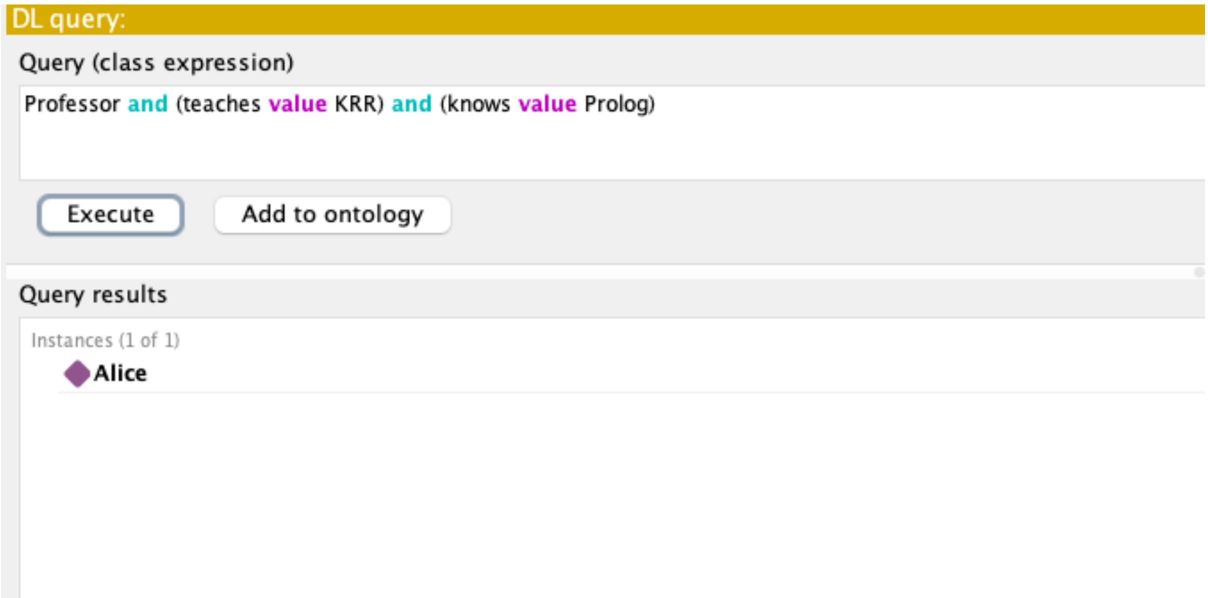
10. Appendix E: Using Protégé

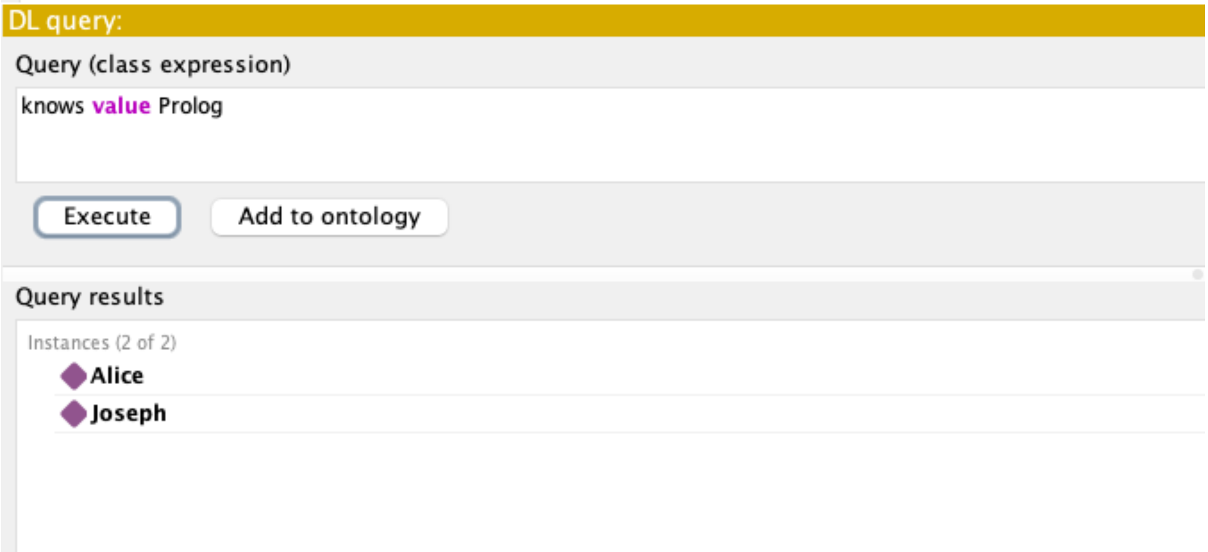
In Protege, you can create a simple ontology according to my real course design. New professors (Tony and Alice), students (Jack and Joseph), and modules (Machine Learning and KRR) were added. I included some skills are Java Python, and Prolog as well. I set up from teaches, takes, hasSkill to everyone. With the DL & SPARQL Query tabs , I can now check who has what skills or who teaches what.






DL query:
Query (class expression)
Student and (takes value MachineLearning) and (knows value RStudio)
Execute
Add to ontology
Query results
Instances (1 of 1)
Jack


DL query:
Query (class expression)
Professor and (teaches value KRR) and (knows value Prolog)
Execute
Add to ontology
Query results
Instances (1 of 1)
Alice


DL query:
Query (class expression)
knows value Prolog
Execute
Add to ontology
Query results
Instances (2 of 2)
Alice
Joseph

SPARQL query:

```
PREFIX : <http://www.semanticweb.org/murthykanuri/ontologies/2025/3/MscAI#>

SELECT ?studentName
WHERE {
 ?student a :Student ;
          :takes :MachineLearning ;
          :knows :RStudio ;
          :hasName ?studentName .
}
```

Jack

SPARQL query:

```
PREFIX : <http://www.semanticweb.org/murthykanuri/ontologies/2025/3/MscAI#>

SELECT ?professorName
WHERE {
 ?professor a :Professor ;
           :teaches :KRR ;
           :knows :Prolog ;
           :hasName ?professorName .
}
```

Alice