COMP 348: Principles of Programming Languages
Assignment no.1

Duc Nguyen - Antonio Verdicchio - Zahra Nikbakht

Gina Cody School of Computer Science and Software Engineering
Concordia University, Montreal, QC, Canada

Summer 2020

Contents

1 Example to demonstrate LaTeX

Here is an example for the latex file

```
all_sections(CNAM, CNUM, L) :-
findall(L, takes_course(_, CNAM, CNUM, L), X),
sort(X, L).
```

2 Question 8

2.1 ? magic(Hermione)

In this non ground query, we will find all atoms such that they satisfy magic(Hermione). First, magic(Hermione) is unified with the head of the rule magic(X):- house_elf(X). So X is instantiated to Hermione. Through resolution, we go to the body of the rule and house_elf(Hermione) is now our goal to satisfy and it is treated as a new non ground query. We have house_elf(dobby) in our knowledge base so house_elf(Hermione) unifies with house_elf(dobby) and Hermione is instantiated to dobby so dobby satisfies our goal and thus it is one answer to the initial query.

There are no more answers for house_elf(Hermione), so using backtracking, we reach the next rule for magic(X). magic(Hermione) is unified with the head of the rule magic(X):- wizard(X). So X is instantiated to Hermione. Then in the resolution step, wizard(Hermione) is now our goal to satisfy. We have wizard(dobby) in our knowledge base so wizard_elf(Hermione) unifies with wizard(dobby) and Hermione is instantiated to dobby so dobby is another answer of the query.

There are no more answers for wizard(Hermione), so using backtracking, we reach the next rule for magic(X). magic(Hermione) is unified with the head of the rule magic(X):- witch(X). X is instantiated to Hermione. Then in the resolution step, we make a transition to the body of the rule and witch(Hermione) is now our goal to satisfy. We have witch(hermione), witch(mcGonagall) and witch(rita_skeeter) in our knowledge base so Hermione is first instantiated hermione and when we continue the search, there are other answers mcGonagall and rita_skeeter to the non groud query witch(Hermione). Through backtracking, we can see that there is no other answer for our initial query.
The final answer to the query is:

```
Hermione = dobby;
Hermione = dobby;
Hermione = hermione;
Hermione = mcGonagall;
Hermione = rita_skeeter;
```

2.2 ? magic(hermione)

This is a ground query, so we expect either true or false as an answer. First, Prolog searches the database from top to bottom. It matches the query with head of the rule magic(X):- house_elf(X). So X is instantiated to hermione. Now our goal is house_elf(hermione) (through resolution) and we treat it as a new ground query. We can see that house_elf(hermione) cannot match any clause in our database so we go to the next rule through backtracking: magic(X):- wizard(X). Now X is instantiated to hermione. In the resolution step, we make a transition to the body of the rule and our goal is to satisfy (find) wizard(hermione). We can clearly see in the database that hermione is a wizard so the result of our goal query (?- wizard(hermione).) is true. Therefore we can tell that the initial query also returns true.