

CSL 303 : Artificial Intelligence

TUTORIAL ASSIGNMENT 1

Introduction to Prolog

Date Assigned: 13th August, 2021

Date Submitted: 16th August, 2021

Submitted By:

Name: S. Bhavyesh

Roll Number: 191210045

Branch: Computer Science and Engineering

Semester : 5th

Submitted to: Dr. Chandra Prakash

Department of Computer Science and Engineering



NATIONAL INSTITUTE OF TECHNOLOGY DELHI

A-7, Institutional Area, near Satyawadi Raja Harish Chandra Hospital, New Delhi,
Delhi-110040

1. What is Prolog and how it is connected to Artificial Intelligence.

Introduction

PROLOG is a Logic Programming Language that was created in early 1970s to program artificial intelligent systems. Its applications include proving theorems, building expert systems, and for Natural Language Processing. It belongs to the class of declarative programming languages, where the focus is on achieving a particular result, no matter what algorithm is followed (‘what’ you want, not ‘how’ you want). This is different from imperative programming, where we explicitly tell the computer what steps are to be taken to perform some task.

A general PROLOG program can be thought of as a knowledge base – a repository of facts (axioms) and rules(conditional statements). This knowledge base is ‘compiled’ and is accessed at execution time by a set of queries. The program then looks for relationships between facts, based on the rules(if given) and then arrives at some answer(s). The diagram below summarises the operation of a PROLOG program.

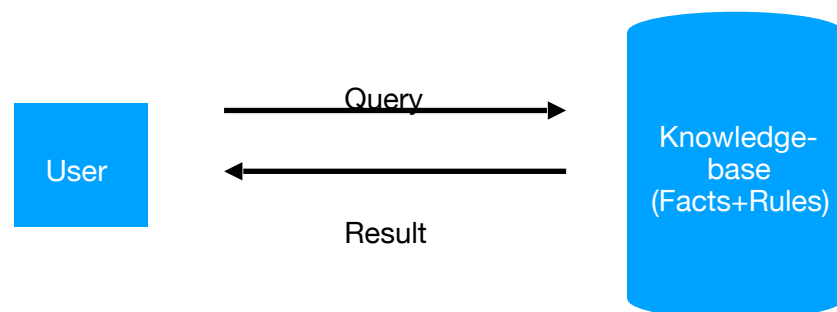


Figure 1: Diagram showing the working of PROLOG

Connection with Artificial Intelligence

The mathematical principle behind the working of PROLOG is first-order-predicate-logic (FOPL). FOPL is ‘stronger’ than propositional logic, in the sense that predicate logic is better suited for representing naturally-spoken languages, and can be used to represent relationships between objects in a more general sense.

For example, we have the following statement:

p : It is hot today,

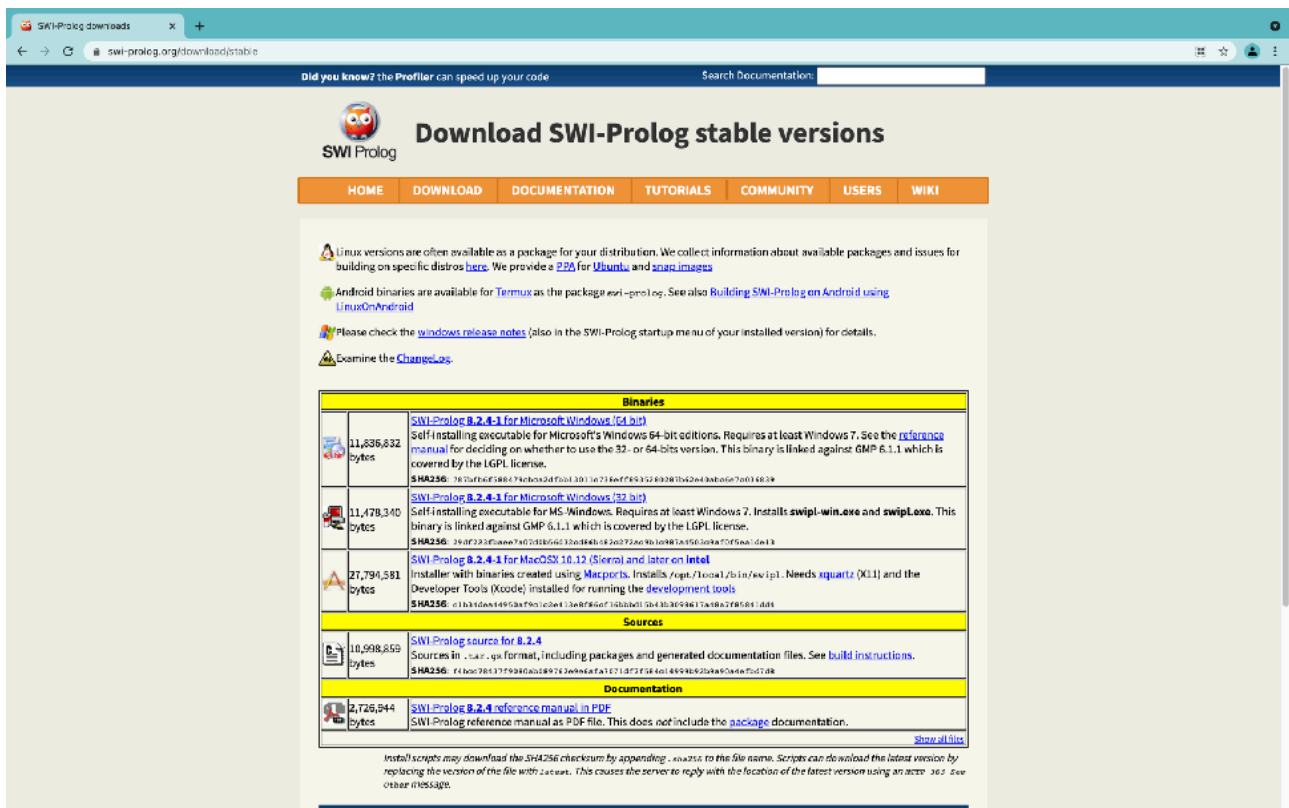
and I ask the machine if it is hot today. The answer will be true since that fact is present in the knowledge base. But we won’t really call this intelligence, as the machine simply searches for this fact. However, if I add the following rule :

$p \rightarrow q$: If it is hot, it will rain,

where p, q are the respective statements, separated by ‘if’. Then if I ask if it will rain today, the answer will be true. But we didn’t explicitly mention in the knowledge base that it will rain today. This means that the machine used the existing axiom and the given rule to arrive at the conclusion. This can be related to the reasoning techniques we use in our day-to-day life. And artificial intelligence, to grossly simplify, is the simulation of human reasoning, by machines. And PROLOG is a medium to ‘train’ a machine in this way.

2. Download the SWI-Prolog from following link as per your operating system.

Go to the mentioned website and download SWI-PROLOG(from <https://www.swi-prolog.org/download/stable>) as per your operating system (in my case it is macOS):



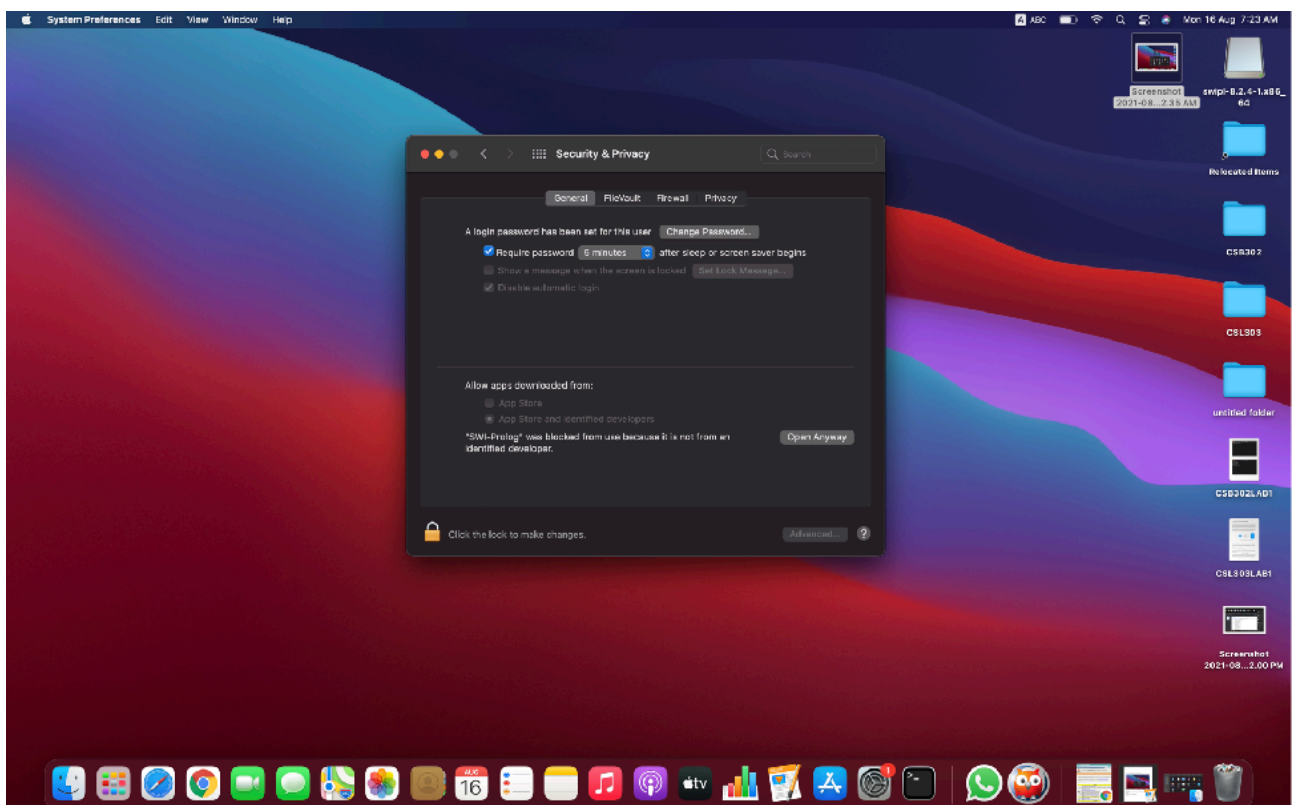
Then, open the .dmg file and drag the SWI-PROLOG icon from the there to Applications folder:



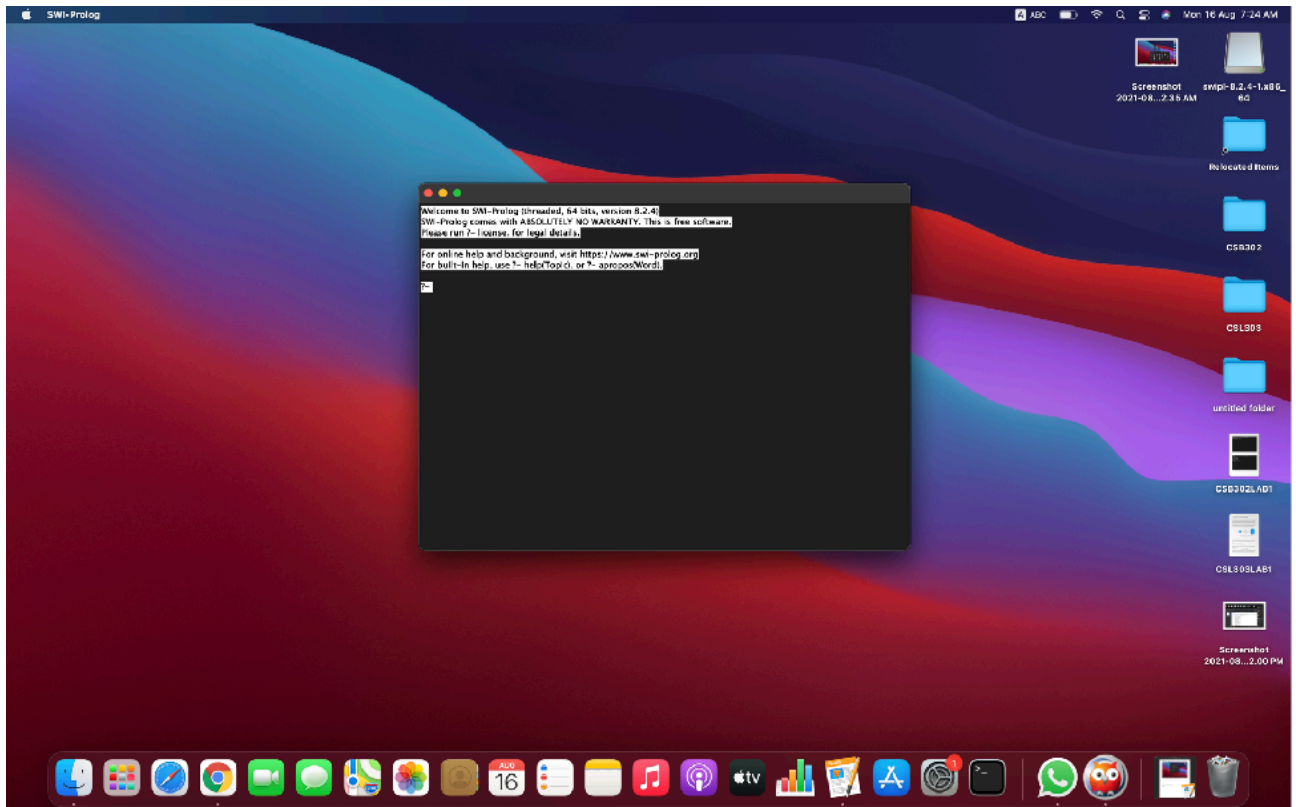
Try opening the SWI-PROLOG application. It will disallow at first.



To overcome this, go to “apple” > System Preferences > Security and Privacy > click on “Open Anyway” at the bottom.



Now we can access the SWI-PROLOG terminal.



3. Convert following statements [facts/rules/queries] into PROLOG.

Examples (Facts):

“A dog is a mammal”

“A sparrow is a bird”

Examples (Rules):

“Something is an animal, if it is a mammal or a bird”

Examples (Queries):

“is a sparrow an animal?”

“is a table an animal?”

“what is a dog?”

Since SWI-PROLOG does not come with a native text editor when installed on macOS, one can use Text-Edit or any other text-editor. I have used VSCode. The code is given below:



```
TA_1_Prob_1.pl
Users > scsr > Desktop > CSL303 > TA_1_Prob_1.pl
1  group(dog,mammal).           % a dog is a mammal
2  group(sparrow,bird).        % a sparrow is a bird
3  animal(X):-group(X,mammal);group(X,bird). % something is an animal, if it is a mammal or a bird
```

To run the program, open the PROLOG terminal and the first line to be entered is
[‘absolute path of program file’].

After that, regular queries can follow:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
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).
```

```
?- ['Users/scsr/Desktop/CSL303/TA_1_Prob_1.pl'].
true.
```

```
?- animal(sparrow).
true.
```

```
?- animal(table).
false.
```

```
?- group(dog,X).
X = mammal.
```

```
?-
```

OBSERVATION/COMMENTS:

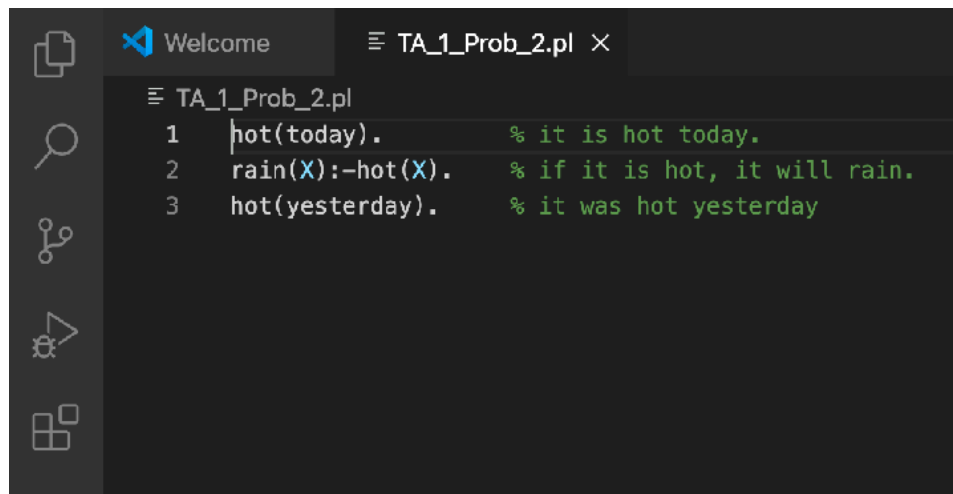
1. `group(dog,mammal)` is an axiom, signifying that dog belongs to the group of mammals.
2. `animal(X):-group(X,mammal) ; group(X,bird)` is a conditional(rule), meaning that if X belongs to the group of mammals or birds, then X is an animal (*result ← cause*).
3. Note that there is a semicolon that separates the two conditions. This is the OR operator in PROLOG. The AND operator is denoted by a comma.
4. Also, variable names start with a capital letter. This distinguishes them from ‘tokens’ or fixed entities such as dog, sparrow etc.
5. All statements in PROLOG have to end with a dot. It is analogous to the semicolon in C.
6. Clearly, a sparrow is an animal, a table is not. These have been stated in the terminal correctly.
7. `group(dog,X)` tells the program to find the category of dog. This shows that PROLOG is not only capable of displaying truth values of statements, but it also helps in finding values that fit a particular criterion.

4. Find the answer of the query based on the following statement / knowledge base.

It is hot today
If it is hot, it will rain.
It was hot yesterday.

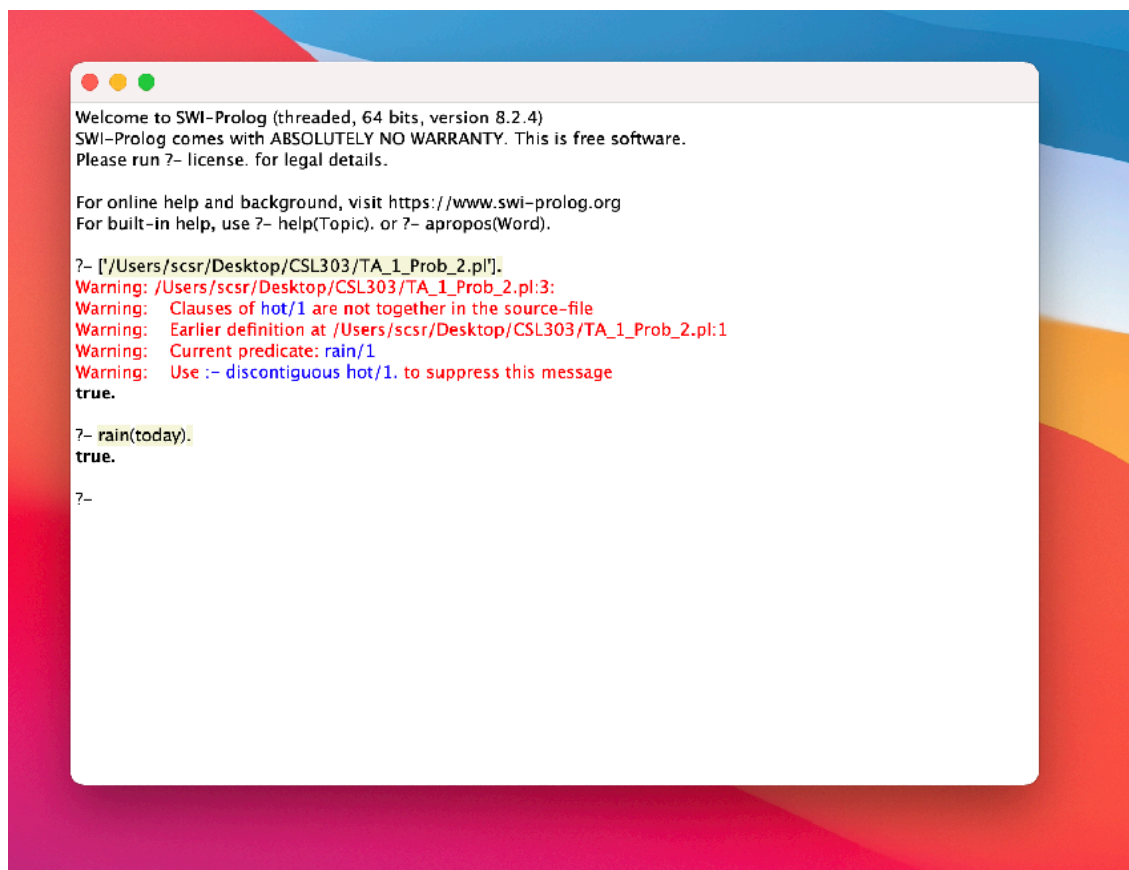
Query : Will it rain today ?

A new file for the program is created with the following code:



```
TA_1_Prob_2.pl
1 hot(today). % it is hot today.
2 rain(X):-hot(X). % if it is hot, it will rain.
3 hot(yesterday). % it was hot yesterday
```

Open the PROLOG terminal and type the absolute path of the program as shown below, and post the query:



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
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).

?- ['Users/scsr/Desktop/CSL303/TA_1_Prob_2.pl'].
Warning: /Users/scsr/Desktop/CSL303/TA_1_Prob_2.pl:3:
Warning: Clauses of hot/1 are not together in the source-file
Warning: Earlier definition at /Users/scsr/Desktop/CSL303/TA_1_Prob_2.pl:1
Warning: Current predicate: rain/1
Warning: Use :- discontiguous hot/1. to suppress this message
true.

?- rain(today).
true.

?-
```


OBSERVATION/COMMENTS:

1. As we can see, a warning is thrown but it does not really affect the program. The warning is possibly due to the fact that similar clauses are not together in the code.
2. Comments can be added in PROLOG programs by a % sign. Anything to the right of it is a comment.
3. The terminal correctly states that it will rain today. This can be verified by the *modus ponens* rule.