Introduction to Artificial Intelligence

Week 4 – Introduction to Prolog

Expert Systems – Prolog

Agenda

- Prolog syntax and semantics
- Prolog exercises

Prolog

- **Prolog** is a general-purpose logic programming language associated with artificial intelligence and computational linguistics
- Prolog stands for **Pro**gramming in **Log**ic
- Prolog has its roots in first-order logic, a formal logic, and unlike many other programming languages, Prolog is declarative: the program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a *query* over these relations
- The language was first conceived by a group around Alain Colmerauer in Marseille, France, in the early 1970s and the first Prolog system was developed in 1972 by Colmerauer with Philippe Roussel

Prolog Syntax & Semantics (1)

- Comma (**,**) represents **AND**

  For example: ***tiger(X) :- cat(X), big(X).***

  [X is tiger if X is cat **AND** X is big]

- Semicolon (**;**) represents **OR**

  For example: ***animal(X) :- cat(X); dog(X).***

  [X is animal if X is car **OR** X is dog]

- Dot (**.**) represents the end of the sentence.

Prolog Syntax & Semantics (2)

- Head :- Body.
- Clauses with empty bodies are called facts

  cat(tom). [Tom is cat]

- Clauses with bodies are called rules. Rules describe new rules based on facts and/or other rules

  animal(X) :- cat(X). [X is animal if X is cat]

- Queries are needed to show outputs. It is also called a goal

  ?- cat(X).

  Answer: X = tom

Prolog Syntax & Semantics (3)

- Rules can be more complex and contain **AND (",")** or **OR (";")** operators.

  animal(X) :- cat(X); dog(X). [X is animal if X is cat OR X is dog]

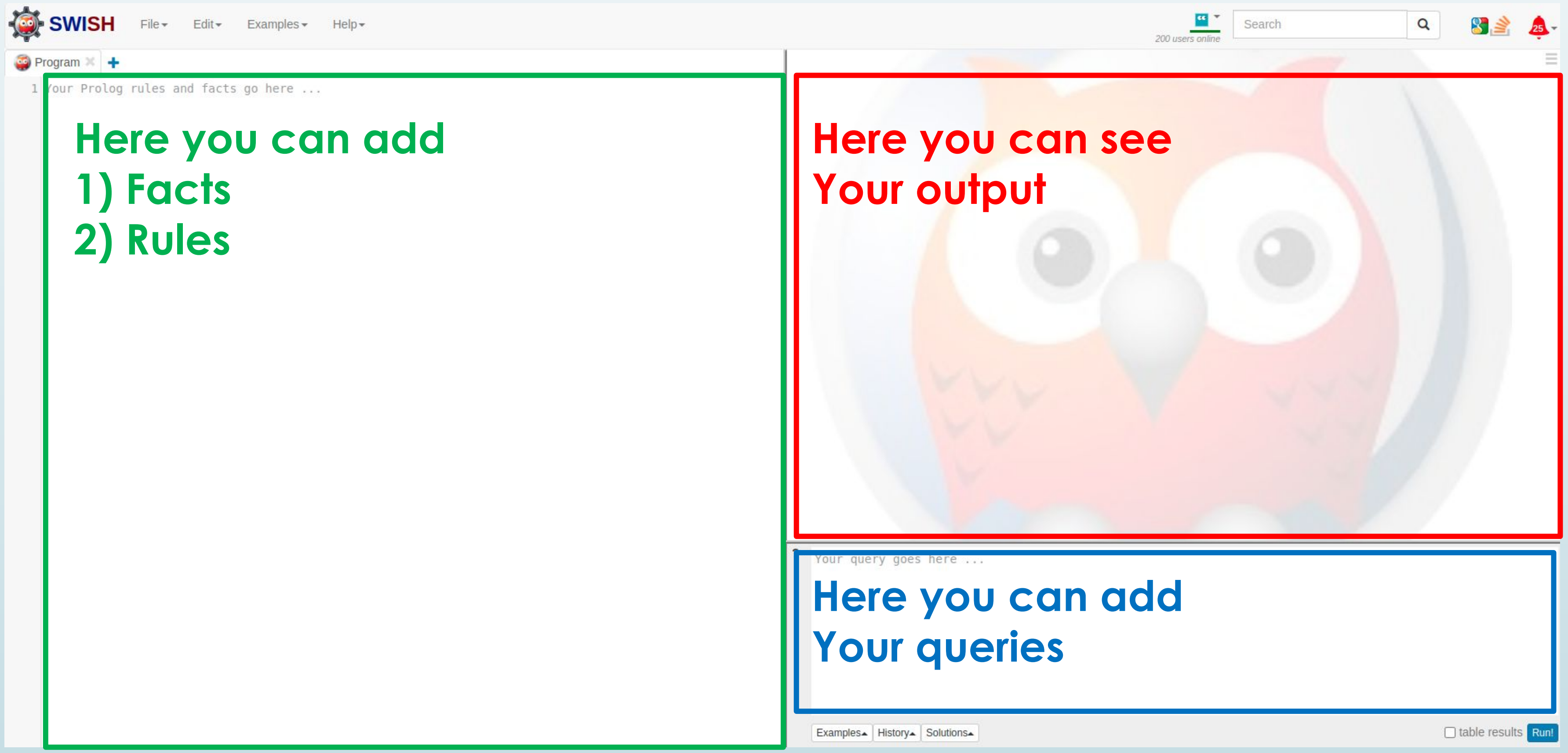
  animal(X) :- cat(X), alive(X). [X is animal if X is cat AND X is alive]

- Rules can also contain **NOT ("not")** operator.

  pet(X) :- cat(X), not(wild(X)). [X is pet if X is cat AND X is not wild]

SWI-Prolog

- Go to https://swish.swi-prolog.org/example/prolog_tutorials.swinb
- It will look as following



Exercise 1: Family Tree - Interactive

1. Based on lecture example draw your own family tree.
2. Write the program with facts and rules for sisters, brothers, mother, father, aunts, uncles, grandfathers and grandmothers. It should be possible to check all these relations.

- Create **facts** with

  male(nick).

- Create **rules** for these people like

  mother(X,Y) :- parent(X,Y), female(X).

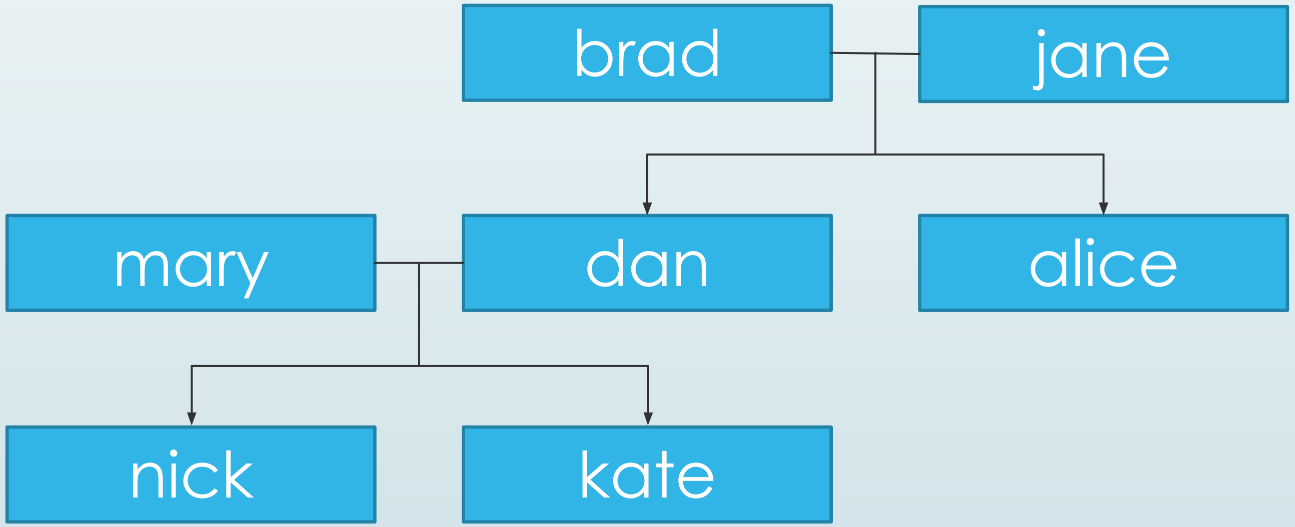
  where *X* is mother, *Y* is a child.

- Check **queries** with

  ?- mother(jane,nick).

- If it is correct, then the output should be ***true.*** Otherwise, ***false***.

Exercise 1: Family Tree



Exercise 2: Ages

Modify previous exercise adding surname, age, height and weight to each person.

- Show all people with your surname
- Show all people with your age
- Show all people with your height
- Show all people with your weight

Exercise 3

- Write the program to count from 1 to 10 using recursive call of rule

Exercise 4: Homework

- **As previous exercise, we want to get all people with same surname without adding rules to your code, only facts are used.**

  Your name is **nick** if you are a boy.

  Your name is **jane** if you are a girl.

- **Hint**:

  you can make a good query that your rules will be embedded in.

References

- https://www.youtube.com/watch?v=0rKD13BIHNQ&t=795s
- http://confreaks.tv/videos/cascadiaruby2012-a-taste-of-prolog