

L3 – CS – Semester 6 (2025/2026)

Lab N°2: Symbolic AI with Prolog

A-Installation – Configuration

1) Windows

- Download SWI-Prolog installer (<https://www.swi-prolog.org/download/stable>)
- Run SWI-Prolog installer (e.g: swipl-9.2.9-1.x64.exe)
- Run SWI-Prolog
- Create a "test.pl" file (File/New..)
- Query it (Compile/Compile buffer)

?-mortal(ahmed).

2) Linux (Ubuntu)

- Open your terminal and run:

```
apt update
apt install swi-prolog
```
- Create a Prolog file (test.pl)

- On your terminal: swipl test.pl

- Query it:

?- mortal(ahmed).

3) Online (<https://swish.swi-prolog.org/>)

- Create a program (Type the code on "test.pl")
- Run queries (use the Run! button)

```
%Prolog program
%Facts
human(ahmed).

%Rules
mortal(X):-human(X).
```

test.pl

B-Prolog operators:

- | | | |
|---|---|---|
| <ul style="list-style-type: none">○ Unification :
?- X+1=2+1.
X = 2.

?- 2+X=3+1.
false. | <ul style="list-style-type: none">○ Arithmetic equality:
?- 1+3=:=4.
true.

?- 1+3=:=3+1.
true. | <ul style="list-style-type: none">○ Syntactical difference:
?- dad\=mom.
true.

?- dad\==mom.
true.

?- dad\=dad.
false. |
| <ul style="list-style-type: none">○ Syntactical identity:
?- X==X.
true.

?- dad==mom.
false.

?- "dad"=="dad".
true.

?- dad==dad.
true. | <ul style="list-style-type: none">○ Arithmetic instantiation :
?- X is 3+4.
X = 7.

?- X is +(3,4).
X = 7.

?- X is 3*4+4*5.
X = 32.

?- X is +(*(3,4), *(4,5)).
X = 32. | <ul style="list-style-type: none">○ Arithmetic difference:
?- 3= 6.
true.

?- 3= 3.
false. |

C-Practical examples:

Example 1: (Predicate logic)

- Dogs are animals = $\forall x \text{ dog}(X) \rightarrow \text{animal}(X)$
- Animals will die = $\forall x \text{ animal}(X) \rightarrow \text{mortal}(X)$
- Bobby is a dog = $\text{dog}(\text{bobby})$
- Lulu is a dog = $\text{dog}(\text{lulu})$

Query 1: Will Bobby die? = $\text{mortal}(\text{bobby})$

Query 2: Who will die? = $\text{mortal}(X)$ %use ; for more answers

```
%Prolog program
%Facts
dog(bobby).
dog(lulu).
%Rules
animal(X):-dog(X).
mortal(X):-animal(X).
%Query
?-mortal(bobby).
true.
```

dog.pl

Example 2: (Predicate logic)

- Anyone who can read is educated = $\forall x \text{ read}(X) \rightarrow \text{educated}(X)$
- Dolphins are not educated = $\forall x \text{ dolphin}(X) \rightarrow \text{not educated}(X)$
- Some dolphins are intelligent = $\exists x \text{ dolphin}(X) \text{ and } \text{intelligent}(X)$
- Some intelligent cannot read? = $\exists x \text{ intelligent}(X) \text{ and } \text{not read}(X)$

```
%Facts
dolphin(d).
intelligent(d).
%Rules
educated(X):-read(X).
unread(X):-not(read(X)).
uneducated(X):-not(educated(X)).
uneducated(X):-dolphin(X).
%Query
?- intelligent(d),unread(d).
true.
```

dolphin.pl

Example 3: (Production rule)

IF a person has fever AND cough

THEN the person has flu

```
%Facts
person(karim).
person(khaled).
fever(karim).
cough(karim).
%Rules
flu(X):-person(X),fever(X),cough(X).
%Query
?- flu(karim).
true.
%?- flu(X).
```

flu.pl

Example 4: (Semantic network)



D-Exercises:

Write the prolog programs for the following:

Exercise 1: (Individual)

Define some persons, each with his age (e.g: person(ahmed,50)).

- An individual is a person whatever is the age?
- A minor is a person having less than 18?

Exercise 2: (Family tree)

Define some parental relationships:

father(ahmed,karim). %Ahmed is the father of Karim
 mother(fatma,karim). %Fatma is the mother of Karim

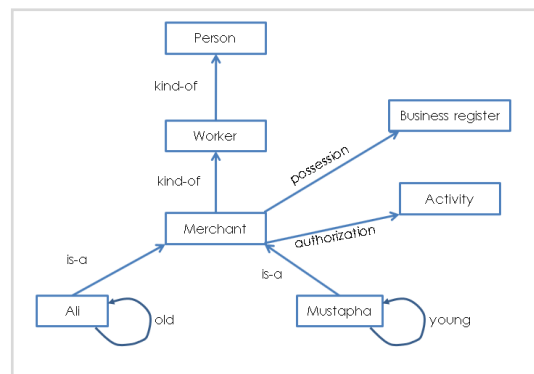
...

Add more facts (as needed) and define the following rules:

- parent(X,Y). %X is the parent of Y
- gradparent(X,Y). %X is the grandparent of Y.
- brother(X,Y). %X is the brother of Y.
- sister(X,Y). %X is the sister of Y.

Exercise 3: (Merchant)

- Write the prolog program for the following semantic net.
- Who possesses a business register?



```
%Facts
is_a(titi,canary).
kind_of(canary, bird).
can_fly(bird).
%Rules
can_fly(X) :- is_a(X, Y),kind_of(Y,Z),can_fly(Z).
%Query
?- can_fly(titi).
true.
%?- can_fly(X).
```

canary.pl