



Al-Mustaqbal University
College of Science



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قسم علوم الذكاء الاصطناعي

المحاضرة الثانية

Facts, Rules, Questions, and Operators



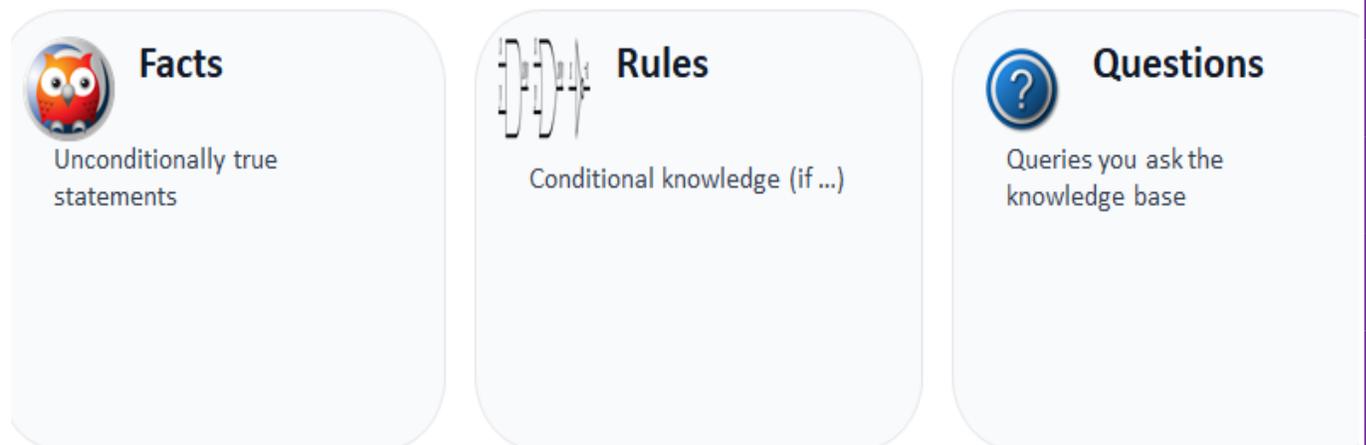
المادة: Prolog languages
المرحلة: الاولى
اسم الاستاذ: م.م اية محمد حسين محمد علي



In this lecture we will cover:

- The 3 basic constructs in Prolog: Facts, Rules, and Questions
- How to write facts and build a small knowledge base
- How rules express conditional knowledge ("if")
- How to ask questions (queries) and interpret answers

Three basic constructs:



A collection of facts + rules is called a knowledge base (database).



What is a Fact?

- A fact represents knowledge that is true in the program.
- Facts are written as predicates ending with a period (.).
- Names of relationships and objects begin with a lowercase letter.
- Examples represent relationships like father/2, mother/2, friend/2.

Example Syntax

father(john, jim).

mother(jane, jim).

friend(jack, john).

Facts store these relations in the knowledge base.





Key points for writing facts

Rules for identifiers

- Predicate and atom names start with lowercase letters.
- Use underscores _ to separate words when needed.
- A fact ends with a period (.).

Arity (number of arguments)

Prolog predicates often written as name/arity:

father/2

mother/2

friend/2

Example: father(john, jim) has arity 2.



From natural language to Prolog facts

Statements

- John is the father of Jim.
- Jane is the mother of Jim.
- Jack is the friend of John.

Prolog facts

father(john, jim).

mother(jane, jim).

friend(jack, john).

Store these in the Clauses section of your program.

Rules express conditional knowledge

- Use rules when a fact depends on other facts.
- A rule has a head (conclusion) and a body (conditions).
- ":-" can be read as "if" or "is implied by".



General form

Head :- Condition1, Condition2, ..., Condition N.

Questions ask about facts and rules

- A question looks like a fact (a predicate call).
- Facts and rules go in the Clauses section.
- Questions go in the Goal section.
- Prolog answers by matching and unifying terms.



Example query

?- parent(john, jim).



Three common query styles

1) Constant

- Returns Yes/No by matching facts/rules.

```
?- age(a, 10).  
Yes.
```

2) Constant + Variable

- Find values for variables that make the predicate true.

```
?- age(a, X).  
X = 10.
```

3) Variable(s)

- Find all solutions (can return multiple answers).

```
?- age(X, Y).  
X=a, Y=10 ;  
X=b, Y=20 ;  
...
```

Example of Questions

Sample knowledge base

```
age(a,10).  
age(b,20).  
age(c,30).
```

Queries and results

```
?- age(a, X).    % X = 10  
?- age(X, 20).  % X = b  
?- age(X, Y).   % multiple solutions
```



Type of questing in the goal

There are three type of question in the goal summarized as follow:

- 1. Asking with constant: prolog matching and return Yes/No answer.
- 2. Asking with constant and variable: prolog matching and produce result for the Variable.
- 3. Asking with variable: prolog produce result

How to read the output?

- Prolog binds variables (instantiation).
- Each solution must satisfy all goals.
- Use “;” to ask for the next solution (in many Prolog systems).



Tip: Use the keyboard to request more answers (;) and stop (.).



Arithmetic & comparison operators are built-in

Common arithmetic

- + addition
- - subtraction
- * multiplication
- / division
- mod remainder

Example

```
?- X is 7 + 3*2.  
X = 13.
```

Comparison

- == numeric equality
- =\= numeric not equal
- < less than
- =< less or equal
- > greater than
- >= greater or equal

Example

```
?- 10 > 3.  
Yes.
```



Combine facts, rules, and a query

% Facts

father(john, jim).

mother(jane, jim).

% Rules

parent(P1, P2) :- father(P1, P2).

parent(P1, P2) :- mother(P1, P2).

% Query

?- parent(jane, jim).

Expected output

Yes.