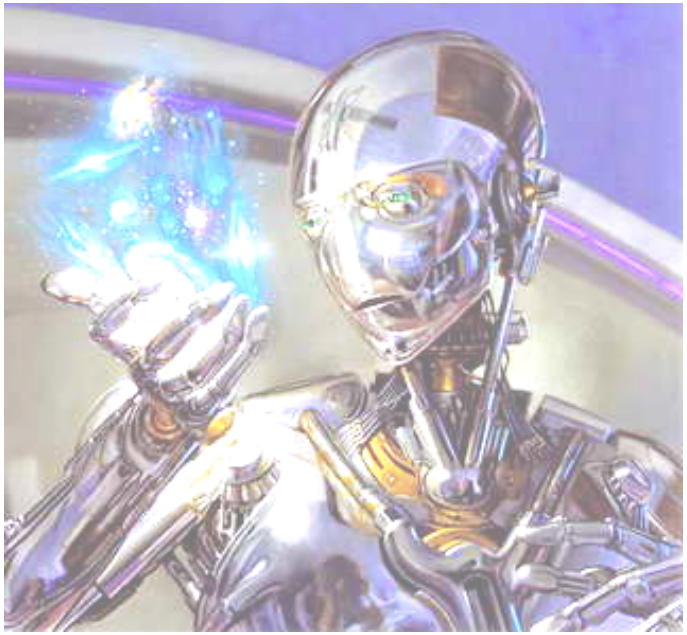


An Overview of Prolog

Chapter One: Introduction to Prolog

Outlines:

1. What's Prolog
2. Basic Idea of Prolog
3. Defining relations by facts
4. Defining relations by rules
5. Exercises

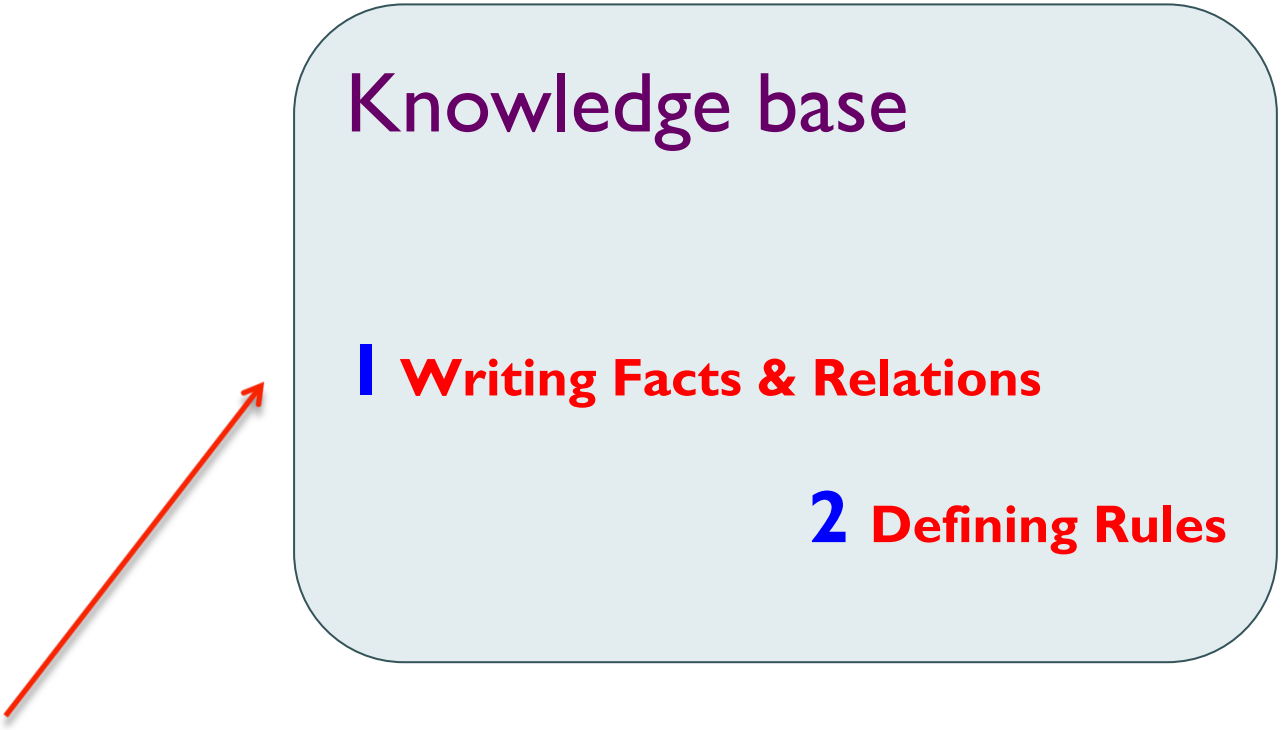
What is Artificial Intelligence ?

- It is the science and engineering of making intelligent machines, especially intelligent computer programs.
- AI is the simulation of intelligent human processes
- It is related to the similar task of using computers to understand human intelligence

What is Prolog

- Prolog is a powerful programming language for AI
- Developed by Alain Colmerauer & at the University of Marseille, France
- Become very popular in Europe and Japan
- Stands for programming in logic
- Declarative (declare facts and rules)
- Very different from other (procedural) programming languages (non-algorithmic)
- Well suited for problems that involves structured objects and relations between them

Basic Idea of Prolog (Three basic steps)



Basic Idea of Prolog

1. Describe the situation of interest by specifying *relationships* among objects and properties of objects.
2. Ask a question.
3. Prolog logically deduces new facts about the situation we described.
4. Prolog gives us its deductions back as answers.

Basic Idea of Prolog

- The program logic is expressed in terms of relations, represented as *facts* and *rules*.

Ex: When we say, "John owns the book", we are declaring the ownership relationship between two objects: John and the book.

- A computation is initiated by running a *query* over these relations.

Ex: When we ask, "Does John own the book?" we are trying to find out about a relationship.

Basic Idea of Prolog

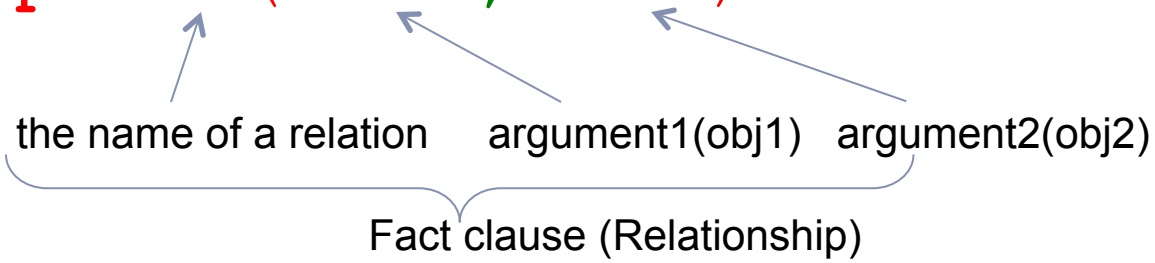
- **Facts:**
  - describes explicit relationships between objects and properties objects might have (e.g. Mary likes pizza, grass has color green)
- **Rules:**
  - defines implicit relationships between objects
  - defining implicit object properties (e.g. Sara passes the exam *if* Sara studies well for the exam).

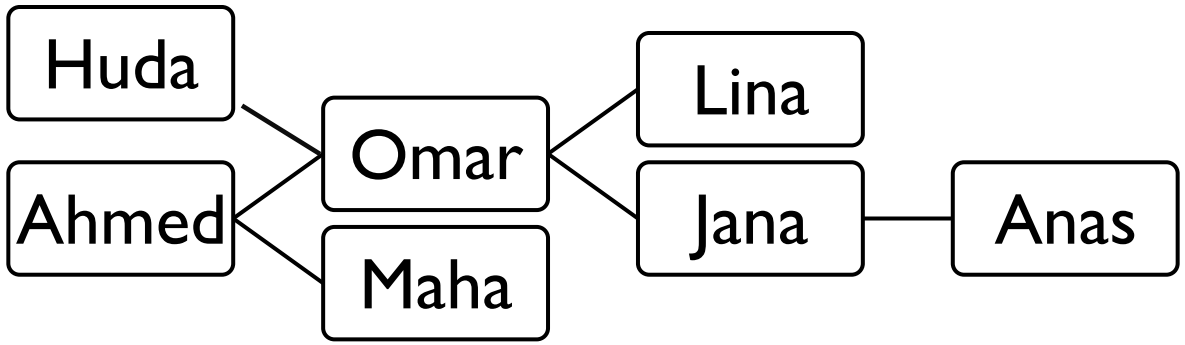
Defining relations by facts

Example

The fact that Ahmed is a parent of Omar

```
parent(ahemd, omar).
```





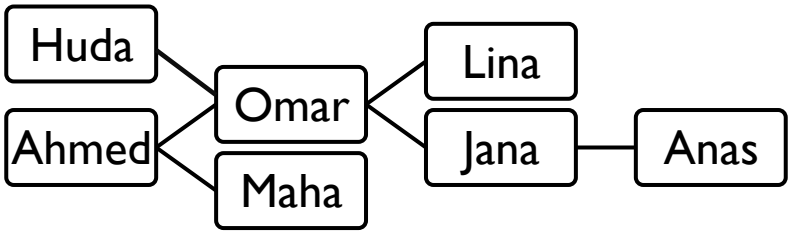
Prolog Operators

- Operators
  - Implication **:- << (if)**
  - Conjunction **, << (and)**
  - Disjunction **; << (or)**

Defining relations by facts

- The family tree is defined by the following Prolog program:

```
parent(huda, omar).
parent(ahmed, omar).
parent(ahmed, maha).
parent(omar, lina).
parent(omar, jana).
parent(jana, anas).
```



- This program consists of **six clauses**.
- Each clause declares **one fact** about the **parent relation**
- A **relation** is the set of all its **instances**
- Each clause is a particular **instance** of the **parent relation**
- An Instance is also called relationship

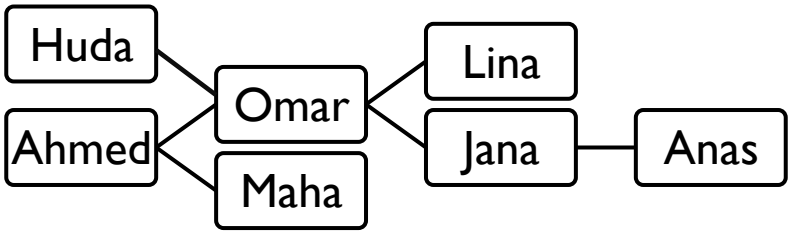
Defining relations by facts

When the program communicated to Prolog system, Prolog can be posed some **questions** about the parent relation:

Is Omar a parent of Jana?

```
?- parent(omar, jana).
```

Prolog answer: `yes`




```
?- parent(maha, jana).
```

Prolog answer: `no`

```
?- parent(ahmed, rami).
```

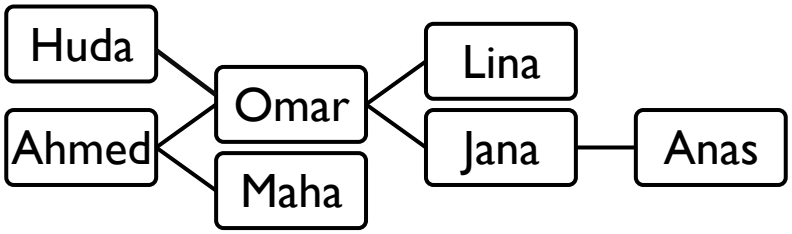
Prolog answer: `no`

Defining relations by facts

Who is Maha's parent?

```
?- parent(X, maha).
```

Prolog answer: `X = ahmed`



Who are Omar's children?

```
?- parent(omar, X)
```

Prolog answer: `X = lina`

We can request anther solution by typing ;

Prolog answer: `X = jana`

If we request more solution, Prolog will answer: `no`

Defining relations by facts

Find X and Y such that X is a parent of Y.

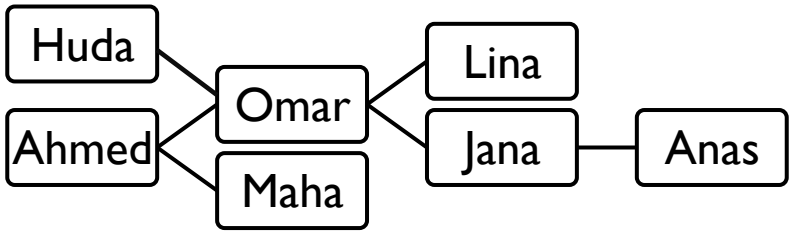
```
?- parent (X, Y)
```

Prolog answers:

```
X = huda
Y = omar;

X = ahmed
Y = omar;

X = ahmed
Y = maha;
```

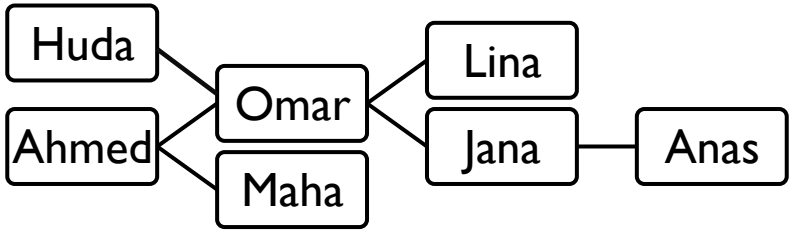


Defining relations by facts

Who is a grandparent of Anas?

Who is the parent of Anas? Assume that this is Y

Who is the parent of Y? Assume that this is X




```
?- parent(Y, anas), parent (X, Y)
```

Prolog answer: `X = omar`

`Y = jana`

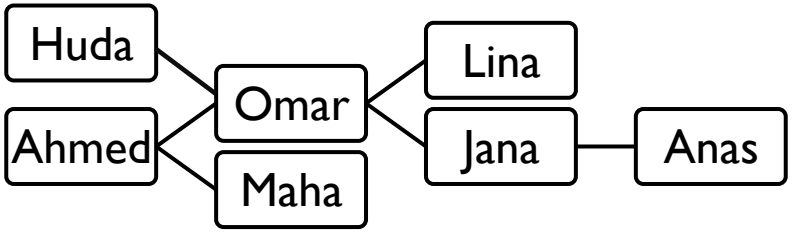
* Changing the order of the requirements will NOT affect the result.

Defining relations by facts

Do Lina and Jana have a common parent?

Who is a parent X of Lina?

Is X a parent of Jana?



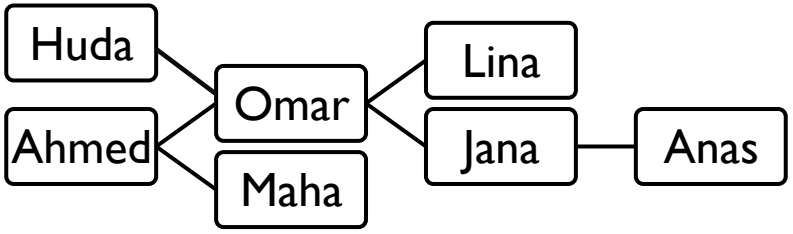

```
?- parent (X, lina), parent(X, jana)
```

Prolog answer: `X = omar`

Defining relations by facts

- We can add the information on the gender of the people that occur in the parent relation:

```
female(huda).
male(ahmed).
male(omar).
female(maha).
female(jana).
female(lina).
male(anas).
```



=

```
gender(huda, feminine).
gender(ahmed, masculine).
gender(omar, masculine).
```

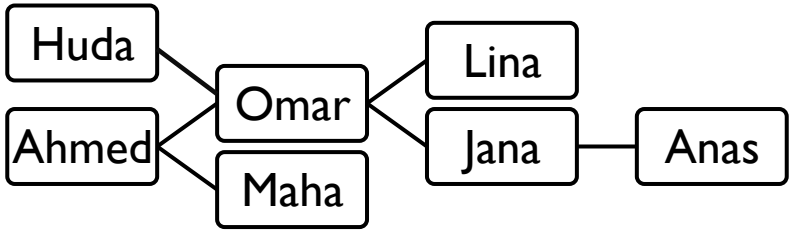
male and **female** are **unary relations**, whereas **parent** is a **binary relation**.

Defining relations by rules

Let us introduce the **offspring** relation (inverse of **parent**)

```
offspring(maha, ahmed).
offspring(omar, ahmed).
...
```

Fact caluses



The logical statement is:

For all X and Y,

Y is an offspring of X if

X is a parent of Y

```
Offspring(Y, X) :- parent(X, Y).
```

Rule clause

Defining relations by rules

There is an important difference between **facts** and **rules**:

- **Facts** is something that is always, unconditionally, true

```
parent(ahmed, maha).
```

- **Rules** specify things that are true if some condition is satisfied. Rules have:
  - Condition part (right-hand side) = body
  - Conclusion part (left-hand side) = head

```
Offspring(Y, X) :- parent(X, Y).
```

head body

Defining relations by rules

Questions

- Is Maha an offspring of Ahmed?

```
?- offspring(maha, ahmed).
```

How Prolog answer this question using the rule:

```
Offspring(Y, X) :- parent(X, Y).
```

Prolog answer: `yes`

Exercises:

- How to express:
  - Mother relation
  - Grandparent relation
  - Sister relation

in Prolog?

Solution

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

2- grandparent(X, Z) :- parent(X, Y) , parent(Y, Z)

3- sister(X, Z) :- parent(Y, X) , parent(Y, Z) , female(X)