PROLOG

Artificial Intelligence
Dr. Hossein Karshenas

University Of Isfahan
Winter 2025

PRESENTED BY
Pouria Sameti

University Of Isfahan

Contents

Introduction

- Prolog is programming language.
- Declarative Paradigm.
  - facts, rules, and queries.
- Rule-based Reasoning.
  - rolog uses rules to derive conclusions.

Fact

- A fact is a basic statement that represents a piece of knowledge about the problem domain.

- predicate
  - It represents the name of the relationship or property.

- Argument
  - they represent entities or values involved in the relationship.

Example

Facts

```
% Facts

person(jack).
person(josh).
person(mary).
person(alice).

age(jack, 20).
age(alice, 40).
age(mary, 30).
age(josh, 70).
```

Rules

head :- body.

- Rules define logical relationships between facts and enable reasoning by specifying conditions under which something is true.
- Head
  - Represents the conclusion or the statement that becomes true if the body is satisfied.
- Body
  - Represents the conditions (facts or other rules) that must be true for the head to be true.
- Head :- Body.
  - The :- symbol means “if“ or “is true if.“

Example

Rules

```
14 % Rules
15 is_adult(X) :- age(X, A), A >= 18.
16
17 is_senior(X) :- age(X, A), A >= 60.
18
19 is_younger(X, Y) :- age(X, A1), age(Y, A2), A1 < A2.
20
21 is_older(X, Y) :- age(X, A1), age(Y, A2), A1 > A2.
```

Query

?- goal.

- queries are used to ask questions about the knowledge base (facts and rules) to infer information or validate a condition.

- Goal
  - Represents a fact, rule, or condition you want to evaluate.
  - If the query is true, Prolog returns true or variable bindings.
  - If it’s false, Prolog returns false.

Example

Query

```
?- is_adult(mary).
true.

?- is_adult(jack).
true.

?- is_adult(josh).
true.

?- is_senior(jack).
false.

?- is_senior(mary).
false.
```

```
?- person(X)
X = jack ;
X = josh ;
X = mary ;
X = alice.
```

```
?- age(X, Y)
X = jack,
Y = 20 ;
X = alice,
Y = 40 ;
X = mary,
Y = 30 ;
X = josh,
Y = 70.
```

Arithmetic Functions

Addition	**+**
Subtraction	-
Multiplication	*
Division (returns float)	/
Integer Division	//
Modulus (remainder)	mod
Exponentiation	^

Relational Operators

Operator	Description
Equality	=
Inequality	\=
Term Equality (non-evaluated)	==
Term Inequality	\==
Less than	<
Greater than	>

Logical Operators

Operator	Description
AND	,
OR	;
NOT	\+

Example

Operators

```
?- X is 2+5.
X = 7.
```

```
?- Y = 10, X is Y*3.
Y = 10,
X = 30.
```

```
is_even(X) :- 0 is X mod 2.
```

```
weekend(Day) :- Day = saturday; Day = sunday.
```

Functions

function/number

- **predicate arity**
  - function: Refers to the name of the predicate (or function) in Prolog.
  - number: Refers to the number of arguments that the predicate takes. Single Argument (Arity 1), Two Arguments (Arity 2)

write/1

Arity 1

Purpose: Outputs text or terms to the console.

```
?- write(hello).
hello
true.
```

length/2

Purpose: Determines the length of a list or creates a list of a specific length.

- If the first argument is a list, it calculates its length.
- If the first argument is a variable, it generates a list of the specified length.

```
?- length([1, 2, 3], L).
L = 3.
```

```
?- length(X, 4).
X = [_, _, _, _].
```

member/2

Purpose: Checks if an element belongs to a list or enumerates list elements.

```
?- member(1, [1, 2, 3, 4, 5]).
true .
```

```
?- member(X, [1, 2, 3, 4, 5]).
X = 1 ;
X = 2 ;
X = 3 ;
X = 4 ;
X = 5.
```

append/3

This predicate is used to concatenate two lists to form a third list.

```
?- append([1], [2, 3, 4], L).
L = [1, 2, 3, 4].
```

is/2

Purpose: Evaluates arithmetic expressions.

```
?- X is 5 + 3.
X = 8.
```

```
?- Y = 10, Z is Y * 2.
Y = 10,
Z = 20.
```

University Of Isfahan

End of Chapter

Good luck ;)