

## Prolog — **PRO**gramming in **LOGIC**

1. A Prolog program is a collection of procedures.
2. Data in Prolog are (rather simple) procedures.
3. Prolog supports recursion.
4. Prolog has no assignment statement.
5. Prolog has no distinction between actual & formal parameters, input & output parameters.
6. Prolog is (essentially) typeless.
7. Prolog has no run-time errors.
8. In Prolog, we tell the computer *WHAT* result we want computed, not *HOW* to compute it.

# Prolog Syntax

- **variables** – 1<sup>st</sup> char. is upper case
- **constants** – 1<sup>st</sup> char. is lower case
- **delimiters** – parentheses and commas
- **procedures or rules:**

Rules are of the form (in BNF-like notation)

$\langle \text{head} \rangle \text{ :- } \langle \text{body} \rangle .$

where  $\langle \text{head} \rangle$  is a goal, and  $\langle \text{body} \rangle$  is a (possibly empty) list of goals separated by commas, followed by a period.

The rule:  $B \text{ :- } A_1, A_2, \dots, A_n.$

can be interpreted *logically* as

“B is true **if** each of the  $A_i$ ’s is true.”

It can also be interpreted *operationally* as

“To solve B,

it suffices to solve each of the  $A_i$ ’s.”

A Prolog program is a collection of rules.

# A simple example: Chemical Reactions

We use constants to represent chemicals:

mg	magnesium
o2	oxygen
mgo	magnesium oxide
h2	hydrogen
h2o	dihydrogen oxide (water)
c	carbon
co2	carbon dioxide
h2co3	carbonic acid

Rules represent chemical reactions:

mg :- mgo, h2.	(you have mg <i>if</i> you have mgo and h2)
h2o :- mgo, h2.	(you have h2o <i>if</i> you have mgo and h2)
co2 :- c, o2.	(you have co2 <i>if</i> you have c and o2)
h2co3 :- h2o, co2.	(you have h2co3 <i>if</i> you have h2o and co2)

mg magnesium  
o2 oxygen  
mgo magnesium oxide  
h2 hydrogen  
h2o dihydrogen oxide (water)  
c carbon  
co2 carbon dioxide  
h2co3 carbonic acid

Assertions represent given information:

mgo :- . (there is magnesium oxide) mgo.  
h2 :- . (there is hydrogen) h2.  
o2 :- . (there is oxygen) o2.  
c :- . (there is carbon) c.

Submitting a goal list at Prolog's prompt:

?- h2co3.

will produce the response "Yes".

mg :- mgo, h2.	(you have mg <i>if</i> you have mgo and h2)
h2o :- mgo, h2.	(you have h2o <i>if</i> you have mgo and h2)
co2 :- c, o2.	(you have co2 <i>if</i> you have c and o2)
h2co3 :- h2o, co2.	(you have h2co3 <i>if</i> you have h2o and co2)
mgo.	(there is magnesium oxide)
h2.	(there is hydrogen)
o2.	(there is oxygen)
c.	(there is carbon)

## How did Prolog arrive at “Yes”?

?- h2co3.	The initial goal list
?- h2o, co2.	was replaced by this (4th rule);
?- mgo, h2, co2.	h2o was replaced by mgo & h2 (2nd rule)
?- h2, co2.	mgo was removed (5th rule)
?- co2.	h2 was removed (6th rule)
?- c, o2.	co2 was replaced by c & o2 (3rd rule)
?- o2.	c was removed (8th rule)
?- <empty>	o2 was removed (7th rule)