What is Prolog?

Prolog is a language which approaches problem-solving in a **declarative** manner. The idea is to define what the problem is, rather than how it should be solved.

In practice, most Prolog programs have a procedural as well as a declarative component — the procedural aspects are often necessary in order to make the programs execute efficiently.

Algorithm = Logic + Control  

Robert A. Kowalski

Prescriptive Languages:

- Describe how to solve problem
- Pascal, C, Ada,...
- Also: Imperative, Procedural

Descriptive Languages:

- Describe what should be done
- Also: Declarative

Kowalski’s equation says that

- Logic – is the specification (what the program should do)
- Control – what we need to do in order to make our logic execute efficiently. This usually includes imposing an execution order on the rules that make up our program.

Objects & Relationships

Prolog programs deal with

- objects, and
- relationships between objects

**English:**

“Christian likes the record”

**Prolog:**

likes(christian, record).
Record Database

is_record(planet_waves).
is_record(desire).
is_record(slow_train).

recorded_by(planet_waves, bob_dylan).
recorded_by(desire, bob_dylan).
recorded_by(slow_train, bob_dylan).

recording_year(planet_waves, 1974).
recording_year(desire, 1975).
recording_year(slow_train, 1979).

Conditional Relationships

Prolog programs deal with conditional relationships between objects.

English:
“C. likes Bob Dylan records recorded before 1979”

Prolog:
likes(christian, X) :-
  is_record(X),
  recorded_by(X, bob_dylan),
  recording_year(X, Year),
  Year < 1979.

Asking Questions

Prolog programs solve problems by asking questions.

English:
“Does Christian like the albums Planet Waves & Slow Train?”

Prolog:
?- likes(christian, planet_waves).
  yes

?- likes(christian, slow_train).
  no

Asking Questions...

English:
“Was Planet Waves recorded by Bob Dylan?”
“Which album was recorded in 1974?”

Prolog:
?- recorded_by(planet_waves, bob_dylan).
  yes

?- recording_year(planet_waves, X).
  X = 1974

?- recording_year(X, 1974).
  X = planet_waves
Asking Questions...

In Prolog

- "", " (a comma), means "and"

**English:**

“Did Bob Dylan record an album in 1974?”

**Prolog:**

?- is_record(X),
    recorded_by(X, bob_dylan),
    recording_year(X, 1974).

yes

Asking Questions...

Sometimes a query has more than one answer:

- Use ";" to get all answers.

**English:**

“What does Christian like?”

**Prolog:**

?- likes(christian, X).

X = planet_waves ;
X = desire ;

no

Asking Questions...

Sometimes answers have more than one part:

**English:**

“List the albums and their artists!”

**Prolog:**

?- is_record(X), recorded_by(X, Y).
X = planet_waves,
Y = bob_dylan ;
X = desire,
Y = bob_dylan ;
X = slow_train,
Y = bob_dylan ;

no

Asking Questions...

Recursive Rules

“People are influenced by the music they listen to. People are influenced by the music listened to by the people they listen to.”

listens_to(bob_dylan, woody_guthrie).
listens_to(arlo_guthrie, woody_guthrie).
listens_to(van_morrison, bob_dylan).
listens_to(dire_straits, bob_dylan).
listens_to(bruce_springsteen, bob_dylan).
listens_to(bjørk, bruce_springsteen).

influenced_by(X, Y) :- listens_to(X, Y).
influenced_by(X, Y) :- listens_to(X,Z),
influenced_by(Z,Y).
Asking Questions...

English:
“Is Björk influenced by Bob Dylan?”
“Is Björk influenced by Woody Guthrie?”
“Is Bob Dylan influenced by Bruce Springsteen?”

Prolog:
?- influenced_by(bjork, bob_dylan).
yes
?- influenced_by(bjork, woody_guthrie).
yes
?- influenced_by(bob_dylan, bruce_s).
no

Answering Questions

(1) scientist(helder).
(2) scientist(ron).
(3) portuguese(helder).
(4) american(ron).
(5) logician(X) :- scientist(X).
(6) ?- logician(X), american(X).

Answering Questions...

?- logician(X), american(X).
logician(X)

(6)
scientist(X)

(1)

scientist(helder)


Answering Questions...

?- logician(X), american(X).

logician(X)

(6)

scientist(X)

(1)

scientist(helder)


Answering Questions...

?- logician(X), american(X).

logician(X)

(6)

scientist(X)

(1)

scientist(helder)


Answering Questions...
Answering Questions...

is_record(planet_waves). is_record(desire). is_record(slow_train).

recorded_by(planet_waves, bob_dylan).
recorded_by(desire, bob_dylan).
recorded_by(slow_train, bob_dylan).

recording_year(planet_waves, 1974).
recording_year(desire, 1975).
recording_year(slow_train, 1979).

likes(christian, X) :-
is_record(X), recorded_by(X, bob_dylan),
recording_year(X, Year), Year < 1979.

Answering Questions...

listens_to(bob_dylan, woody_guthrie).
listens_to(arlo_guthrie, woody_guthrie).
listens_to(van_morrison, bob_dylan).
listens_to(dire_straits, bob_dylan).
listens_to(bruce_springsteen, bob_dylan).
listens_to(bjork, bruce_springsteen).

(1) influenced_by(X, Y) :- listens_to(X, Y).
(2) influenced_by(X, Y) :-
listens_to(X, Z),
influenced_by(Z, Y).

?- influenced_by(bjork, bob_d). 
?- inf_by(bjork, woody_g).
Answering Questions...

Map Coloring

A coloring is OK iff

1. The color of Region 1 ≠ the color of Region 2, and
2. The color of Region 1 ≠ the color of Region 3, ...

\[ \text{color}(R1, R2, R3, R4, R5, R6) : - \\
\text{diff}(R1, R2), \text{diff}(R1, R3), \text{diff}(R1, R5), \text{diff}(R1, R6), \\
\text{diff}(R2, R3), \text{diff}(R2, R4), \text{diff}(R2, R5), \text{diff}(R2, R6), \\
\text{diff}(R3, R4), \text{diff}(R3, R6), \text{diff}(R5, R6). \]

\[ \text{diff}(\text{red}, \text{blue}). \text{diff}(\text{red}, \text{green}). \text{diff}(\text{red}, \text{yellow}). \\
\text{diff}(\text{blue}, \text{red}). \text{diff}(\text{blue}, \text{green}). \text{diff}(\text{blue}, \text{yellow}). \\
\text{diff}(\text{green}, \text{red}). \text{diff}(\text{green}, \text{blue}). \text{diff}(\text{green}, \text{yellow}). \\
\text{diff}(\text{yellow}, \text{red}).\text{diff}(\text{yellow}, \text{blue}). \text{diff}(\text{yellow}, \text{green}). \]

Map Coloring...

?- \text{color}(R1, R2, R3, R4, R5, R6).
R1 = R4 = red, R2 = blue, 
R3 = R5 = green, R6 = yellow ;
R1 = red, R2 = blue, 
R3 = R5 = green, R4 = R6 = yellow

Map Coloring...

“Color a planar map with at most four colors, so that contiguous regions are colored differently.”
Map Coloring – Backtracking

\[
\begin{align*}
\text{color}(R1, R2, R3, R4, R5, R6) \\
\text{diff}(R1, R2) & \quad R1=\text{red} \\
& \quad R2=\text{blue} \\
\text{diff}(R1, R3) & \quad R3=\text{blue} \\
\text{diff}(R1, R4) & \quad R4=\text{blue} \\
\text{diff}(R1, R5) & \quad R5=\text{blue} \\
\text{diff}(R1, R6) & \quad R6=\text{blue} \\
\text{color}(R1, R2, R3, R4, R5, R6) & \quad \text{fail}
\end{align*}
\]

Readings and References

Read Scott, pp. 624–641.

http://dmoz.org/Computers/Programming/Languages/Prolog

- Prolog by Example Coelho & Cotta
- Prolog: Programming for AI Bratko
- Programming in Prolog Clocksin & Mellish
- The Craft of Prolog O'Keefe
- Prolog for Programmers Kluzniak & Szpakowicz
- Prolog Alan G. Hamilton
- The Art of Prolog Sterling & Shapiro
A Prolog program consists of a number of *clauses*:

- **Rules**: Have head + body:
  - head
    - \( \text{likes(chris, X)} :\) :-
      - \( \text{girl(X), black\text{\_}hair(X)} \)
  - body

- Can be recursive

- **Facts**: Head but no body.
  - Always true.

### A question consists of one or more goals:

- ?- \( \text{likes(chris, X), smart(X).} \)
- "," means **and**
- Use ";" to get all answers
- Questions are either
  - Satisfiable (the goal succeeds)
  - Unsatisfiable (the goal fails)

Prolog answers questions (satisfies goals) by:
- instantiating variables
- searching the database sequentially
- backtracking when a goal fails