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).

What is Prolog?

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
Conditional Relationships

Prolog programs deal with

- conditional relationships between objects.

“C. likes Bob Dylan records recorded before 1979”

Prolog: 

\[ \text{likes(christian, X)} \leftarrow \text{is_record(X)}, \text{recorded_by}(X, \text{bob_dylan}), \text{recording_year}(X, \text{Year}), \text{Year} < 1979. \]

Asking Questions

Prolog programs

- solve problems by asking questions.

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

Prolog: 

\[ \text{likes(christian, X)} \leftarrow \text{is_record(X)}, \text{recorded_by}(X, \text{bob_dylan}), \text{recording_year}(X, \text{Year}), \text{Year} < 1979. \]

\[ \text{?- likes(christian, planet_waves)}. \text{yes} \]

\[ \text{?- likes(christian, slow_train)}. \text{no} \]

Asking Questions...

In Prolog

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

“Did Bob Dylan record an album in 1974?”

Prolog: 

\[ \text{?- recording_year(X, 1974)}. \text{X} = \text{planet_waves} \]

\[ \text{?- recorded_by(planet_waves, bob_d)}. \text{yes} \]

…”Was Planet Waves recorded by Bob Dylan?”

“When was Planet Waves recorded?”

“Which album was recorded in 1974?”
**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(björk, bruce_springsteen).

inf_by(X, Y) :- listens_to(X, Y).
inf_by(X, Y) :- listens_to(X, Z), inf_by(Z, Y).

**Asking Questions...**

Sometimes a query has more than one answer:

- Use ";" to get all answers.

<table>
<thead>
<tr>
<th>English:</th>
<th>Prolog:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“What does Christian like?”</td>
<td>:- likes(christian, X).</td>
</tr>
<tr>
<td>X = planet_waves ;</td>
<td>X = desire ;</td>
</tr>
<tr>
<td>X = desire ;</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>English:</th>
<th>Prolog:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Is Björk influenced by Bob Dylan?”</td>
<td>:- inf_by(bjork, bob_dylan).</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>“Is Bob Dylan influenced by Bruce Springsteen?”</td>
<td>:- inf_by(bob_dylan, bruce_s).</td>
</tr>
<tr>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>English:</th>
<th>Prolog:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“List the albums and their artists!”</td>
<td>:- is_record(X), recorded_by(X, Y).</td>
</tr>
<tr>
<td>X = planet_waves,</td>
<td>X = desire,</td>
</tr>
<tr>
<td>Y = bob_dylan ;</td>
<td>Y = bob_dylan ;</td>
</tr>
<tr>
<td>X = slow_train,</td>
<td>no</td>
</tr>
</tbody>
</table>
Answering Questions...

\[ \text{logician}(X), \text{american}(X). \]

\[ \text{logician}(X) \]
\[ \text{american}(X) \]
\[ X = \text{ron} \]
\[ \text{scientist}(X), \text{american}(\text{helder}), \text{fail} \]
\[ \text{scientist}(\text{helder}), \text{scientist}(\text{ron}) \]

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

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

Answering Questions...

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

logician(X) american(X)

american(held)

X = ron

scientist(ron)

scientist(leader)
**Answering Questions...**

?- inf_by(bjork, bob_d).

(1) l_to(bjork, bob_d) fail
(2) l_to(bjork, Z) inf_by(Z, bob_d) Z=bruce_s

l_to(bjork, bob_d) fail

(1) l_to(bjork, woody_g) fail
(2) l_to(bjork, Z) inf_by(Z, woody_g) Z=bruce_s

l_to(bjork, woody_g) fail

(1) inf_by(X, Y) :- l_to(X, Y).
(2) inf_by(X, Y) :-
   l_to(X, Z),
   inf_by(Z, Y).

?- inf_by(bjork, bob_d).

succeed

Yes, I understand that the predicate \( \text{inf}_\text{by} \) is defined with two clauses:

1. \( \text{inf}_\text{by}(X, Y) \) if \( \text{l}_\text{to}(X, Y) \) fails;
2. \( \text{inf}_\text{by}(X, Y) \) if \( \text{l}_\text{to}(X, Z), \text{inf}_\text{by}(Z, Y) \) succeeds.

The example queries use these definitions to answer questions about the artist and year of a recording.

**Answering Questions...**

?- likes(christian, X).

is_record(X) artist(X, bob_d) recording_year(X, Y) \( Y<1979 \)

\( X = \text{planet_waves} \) \( Y=1979 \) succeed
\( X = \text{desire} \) \( Y=1975 \) succeed
\( X = \text{slow_train} \) \( Y=1974 \) fail

?- inf_by(bjork, woody_g).

listens_to(bob_dylan, woody_g).
listens_to(arlo_guthrie, woody_g).
listens_to(van_morrison, bob_d).
listens_to(dire_straits, bob_d).
listens_to(bruce_springsteen, bob_d).
listens_to(björk, bruce_s).

?– inf_by(bjork, bob_d).

succeed

?– inf_by(bjork, woody_g).
Color a planar map with at most four colors, so that contiguous regions are colored differently.

?- 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

Map Coloring V – Backtracking

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, ...

color(R1, R2, R3, R4, R5, R6) :-
   diff(R1, R2), diff(R1, R3), diff(R1, R5), diff(R1, R6),
   diff(R2, R3), diff(R2, R4), diff(R2, R5), diff(R2, R6),
   diff(R3, R4), diff(R3, R6), diff(R5, R6).

diff(red,blue). diff(red,green). diff(red,yellow).
diff(blue,red). diff(blue,green). diff(blue,yellow).
diff(green,red). diff(green,blue). diff(green,yellow).
diff(yellow, red).diff(yellow,blue). diff(yellow,green).
Readings and References...

<table>
<thead>
<tr>
<th>Computing with Logic</th>
<th>Maier &amp; Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Systems Through Prolog</td>
<td>Steven H. Kim</td>
</tr>
<tr>
<td>Natural Language Processing in Prolog</td>
<td>Gazdar &amp; Mellish</td>
</tr>
<tr>
<td>Language as a Cognitive Process</td>
<td>Winograd</td>
</tr>
<tr>
<td>Prolog and Natural Language Analysis</td>
<td>Pereira and Shieber</td>
</tr>
<tr>
<td>Computers and Human Language</td>
<td>George W. Smith</td>
</tr>
<tr>
<td>Introduction to Logic</td>
<td>Irving M. Copi</td>
</tr>
<tr>
<td>Beginning Logic</td>
<td>E.J. Lemmon</td>
</tr>
</tbody>
</table>

Map Coloring VI – Backtracking

A Prolog program consists of a number of clauses:

**Rules** – Have head + body:

- head
- body
- Can be recursive

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

Readings and References

- Read Scott, pp. 624–641.

<table>
<thead>
<tr>
<th>Prolog by Example</th>
<th>Coelho &amp; Cotta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prolog: Programming for AI</td>
<td>Bratko</td>
</tr>
<tr>
<td>Programming in Prolog</td>
<td>Clocksin &amp; Mellish</td>
</tr>
<tr>
<td>The Craft of Prolog</td>
<td>O'Keefe</td>
</tr>
<tr>
<td>Prolog for Programmers</td>
<td>Kluzniak &amp; Szpakowicz</td>
</tr>
<tr>
<td>Prolog</td>
<td>Alan G. Hamilton</td>
</tr>
<tr>
<td>The Art of Prolog</td>
<td>Sterling &amp; Shapiro</td>
</tr>
</tbody>
</table>
Prolog So Far...

- A clause consists of
  - atoms Start with lower-case letter.
  - variables Start with upper-case letter.
- Prolog programs have a
  - Declarative meaning
    * The relations defined by the program
  - Procedural meaning
    * The order in which goals are tried

Prolog So Far...

- A question consists of one or more goals:
  - `?- 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