Instant Prolog

• Goal: use first-order logic as a programming language
• Data is set of axioms and rules of inference
• Desired output is theorem to be proved from the axioms
• Program is trace of proof.
• Completely declarative model: no sequencing, no control structures
• Unrealizable in practice: need to define operational semantics to guarantee termination, guide search, etc.
• Prolog is a plausible approximation to logic programming
What it looks like

• father (adam, abel).  % an assertion
• father (adam, cain).  % another one
• brother (X, Y) :- father (Z, X), father (Z,Y).  % a rule
• ?:father (X, abel) .  % query
• X = adam  % response
• ?:father (adam, A).  % bidirectional
• A = abel
• ?:brother (abel, cain).  % nothing to instantiate
• yes
The basics

- A **fact** is an assertion involving constants
- A **rule** is a universally quantified formula consisting of a head and a body.
- The **head** is a term involving variables
- The **body** is a conjunction of terms
- **Terms** consist of uninterpreted function names and their arguments, which are variables, constants, or terms (recursively)
- **Intuitively**: the head is true if there is an instantiation of the variable that makes the body true.
- For all X and Y: X is brother of Y iff there is a Z such that Z is father of X and Z is father of Y
Map coloring in Prolog

- color (X) := member (X, [green, red, blue, yellow]).
- member (X, [X|_]).
- member (X, [_|Y]) :- member (X, Y)
- next (R1, R2) :- color (R1), color (R2), R1 \= R2.

Query:

?-; next (R1,R2), next (R1,R3), next (R1,R4), next (R2,R3), next (R2,R4), next (R3,R4)

Query embodies data
Basic control structure: unification

- Query matches a fact if there is an instantiation of the variables that can unify the query with the fact.
- `?- father (X, abel)`: matches `father(adam, abel)` with map `(X => adam)`
- If query matches the head of clause, must find unifier that satisfies all terms in the body.
- If several matches are possible, system will find one that works, if one exists: backtracking is built-in.
- If no match is found, query is not provable (not necessarily false).
Operational semantics

- A rule is a procedure
- Several rules can have the same head (definition by cases)
  - `fact (N,F) :- fact (N, 1, F).`
  - `fact (N, L, F) :- N > 0, L1 is L*M, N1 is N-1, fact (N1, L1,F)`
  - `fact (0, F, F)`
- Each term in the body is a goal that must be matched
- Unification generalizes parameter passing
- Goals are attempted left-to-right
- The above is **Prolog**, not **logic programming**