Introduction to Prolog

By David Fishman, Kenneth Skaggs, Alex Packard, Jared Vanderford

Logic Programming

- Based on mathematical logic
- Facts and Rules

Prolog Basics

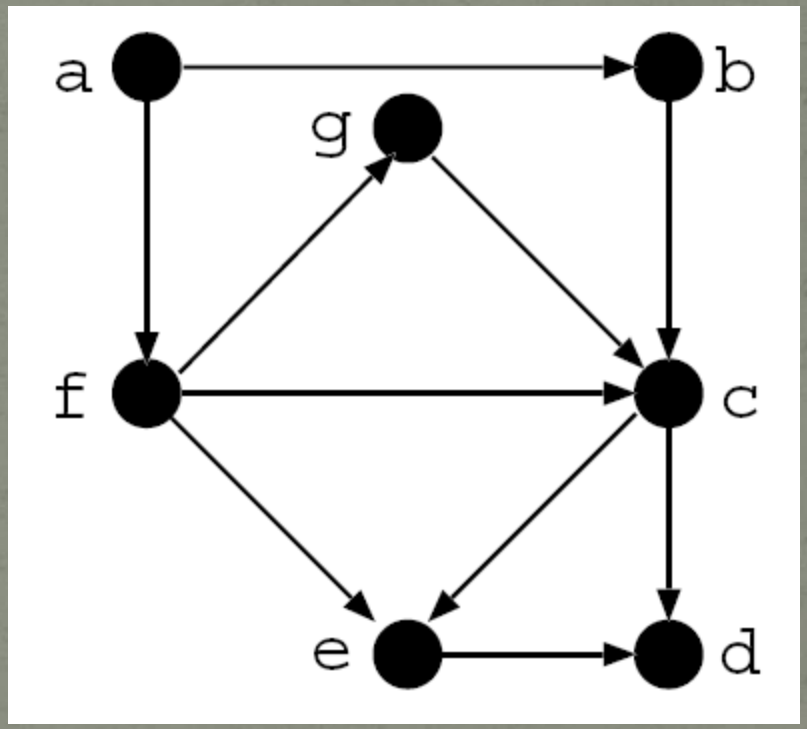
- No types
- Identifiers starting with lower-case denote data values
- All others denote variables
- Includes mechanisms for tuples and lists

Facts and Rules

- A *fact* is a Prolog statement consisting of an identifier followed by an n-tuple of constants.
- A relation identifier (or fact) is referred to as a *predicate*
- When a tuple of values is in a relation it *satisfies* the predicate
- A *rule* is a prolog statement which gives conditions under which tuples satisfy a predicate.

Directed Graph Example

- edge(a,b).
- edge(a,f).
- edge(b,c).
- edge(f,g).
- edge(f,e).
- edge(f,c).
- edge(g,c).
- edge(c,d).
- edge(c,e).
- edge(e,d).



Count Edges

- "," means "and"
- "." ends a statement
- ":-" defines a rule
- Above reads: "The tuple (Node1, Node2) is true if the tuple (Node1,SomeNode) exists and the tuple (SomeNode, Node2) exists."

```
twoEdges(Node1, Node2):-
                    edge(Node1,SomeNode),
                    edge(SomeNode, Node2).
```

Querying Prolog

```
23 ?- twoEdges(a,s).
false.

24 ?- twoEdges(a,b).
false.

25 ?- twoEdges(a,X).
X = c ;
X = g ;
X = e ;
X = c.

26 ?- twoEdges(a,c).
true
```

Exercise

- Write the rule threeEdges(Node1,Node2) which will return true if there are three edges between the specified nodes and false otherwise.