CS 313 Lecture 27

Functional programming in Python
Intro to Prolog
Functional Programming in Python

• lambda, e.g.
  
  lambda x: x*x
  lambda x, y: x + y

• map, filter, reduce
  [in Python 3, to get reduce: from functools import reduce ]

• list comprehensions, e.g.
  [x*x for x in [1, 2, 3, 4] if x > 1]
Part 4: Logic Programming

Prolog (Sethi Ch 11, 15.7)

• developed in 1972 in France
  • Alain Colmerauer
  • Philippe Roussel
  • Robert Kowalski
Logic Programming

• program =
  set of logic statements

• program execution =
  controlled deduction (search) of logic statements

• “algorithm = logic + control”
Logic Programming

• “static” view of programming
  • just list the facts
  • program tries to solve the problem

+ short programs
+ clean – like mathematical program descriptions
- easy to overlook how problem is solved
  • can be inefficient
- sometimes easier to think “procedurally”
Applications of Logic Programming

Mostly in AI domain

• Expert systems
• Relational databases
• Natural language understanding
• Theorem proving
• Planning, decision making
Prolog: Programming in Logic

Prolog programming:
• declaring facts
• defining rules
• asking questions

Examples:
• Facts:
  • Jen and Laura are sisters
  • Elliott likes ice cream

• Rules:
  • two people are sisters if they are both female and have the same parents

• Questions:
  • what does Elliott like?
  • are Laura and Mary sisters?
in Prolog:

Facts:

likes(elliott, icecream).
valuable(gold).
parents(ashley, jill, joe).  /* child, mother, father */
kingOf(john, france).

likes, gold = atoms (need to start with lowercase letter)
a(b, c) = compound term
in Prolog:

Rules:

bird(X) :- animal(X), has_feathers(X).
sisterOf(X, Y) :-
    female(X),
    parents(X, Mother, Father),
    parents(Y, Mother, Father),
    X \= Y.

X, Mother = variables (need to start with uppercase letter or _
Prolog rules

A Prolog rule

\[ P : - Q_1, Q_2, \ldots Q_n. \]

means

\[ P \text{ if } Q_1 \text{ and } Q_2 \ldots \text{ and } Q_n \]

Control:

to deduce \( P \),
    deduce \( Q_1 \),
    deduce \( Q_2 \),
    ...
    deduce \( Q_n \)
Prolog queries

• Facts and rules go into a prolog file (.pl)
• Queries are asked at the prompt

?- valuable(gold).
true.
?- valuable(silver).
false. /* not in database of facts, i.e., can’t prove */
?- valuable(X).
X = gold.
Example

- family.pl