LING 388: Language and Computers

Sandiway Fong
Lecture 5
Administrivia

• **Reminder**
  – Homework 2: due this Wednesday by midnight
    • email: Ben Martin bamartin@email.arizona.edu

• Today’s Topic:
  – Recursion (Chapter 3 of learnprolognow.org)
Prolog online resources

- Some background in logic or programming?
- Useful Online Tutorials

   - Learn Prolog Now!
     - Patrick Blackburn, Johan Bos & Kristina Striegnitz
     - [http://www.learnprolognow.org](http://www.learnprolognow.org)

   - An introduction to Prolog
     - Michel Loiseleur & Nicolas Vigier
     - [http://boklm.eu/prolog/page_0.html](http://boklm.eu/prolog/page_0.html)
Chapter 2
Unification and Proof Search

This chapter has two main goals:

1. To discuss unification in Prolog, and to explain along the way, we'll introduce =/2, the built-in predicate for standard equality.
2. To explain the search strategy Prolog uses: modus ponens.

2.1 Unification
   Examples
   The occurs check
   Programming with unification
2.2 Proof Search
2.3 Exercises
2.4 Practical Session

Chapter 3
Recursion

This chapter has two main goals:

1. To introduce recursive definitions in Prolog.
2. To show that there can be mismatches between procedural meaning.
3.1 Recursive Definitions
   Example 1: Eating
   Example 2: Descendant
   Example 3: Successor
   Example 4: Addition
3.2 Rule Ordering, Goal Ordering, and Termination
3.3 Exercises
3.4 Practical Session
Recursion

• Recursive definition:
  – (informal) something is defined in terms of itself
  – Circular: *musicality*
  – Recursive:
    • repeated application of the same rule to subparts
  – Example (factorial):
    • 0! = 1
    • n! = n * (n-1)! for n>0
Recursion

• Example (length of a list):
  – length of [] = 0
  – length of [X,...] is 1 + length of [...]
  – length of [1,2,3]
    1.  = 1 + length of [2,3]
    2.  = 1 + 1 + length of [3]
    3.  = 1 + 1 + 1 + length of []
    4.  = 1 + 1 + 1 + 0
    5.  = 3
Recursion

• Example (length of a list):
  – length of [] = 0
  – length of [X,...] is 1 + length of [...]

• In Prolog:
  – length_of([],0).
  – length_of([X|L],N) :- length(L,M), N is M+1.

• Prolog built-in:
  – X is <math expression>
Recursion

• Example (factorial):
  – 0! = 1
  – n! = n * (n-1)! for n>0

• In Prolog:
  – factorial(0,1).
  – factorial(N,NF) :- M is N-1, factorial(M,MF), NF is N * MF.

• Problem: *infinite loop*

• Fix: 2\textsuperscript{nd} case only applies to numbers > 0
  factorial(N,NF) :- N>0, M is N-1, factorial(M,MF), NF is N * MF.
Recursion

• **Example 1: Eating**
  • Consider the following knowledge base:
    is_digesting(X,Y) :- just_ate(X,Y).
    is_digesting(X,Y) :-
      just_ate(X,Z),
      is_digesting(Z,Y).

    just_ate(mosquito,blood(john)).
    just_ate(frog,mosquito).
    just_ate(stork,frog).

• **Query:**
  ?- is_digesting(stork,mosquito).
Recursion

• **Example 2: Descendant**
  child(anne,bridget).
  child(bridget,caroline).
  child(caroline,donna).
  child(donna,emily).
  descend(X,Y) :- child(X,Y).
  descend(X,Y) :- child(X,Z), descend(Z,Y).

• **Query:**
  ?- descend(anne,donna).
Recursion

• **Example 3: Successor**
  – 0 is a numeral.
  – If X is a numeral, then so is succ(X) “successor of”

• **In Prolog:**
  – numeral(0).
  – numeral(succ(X)) :- numeral(X).

• **Query:**
  ?- numeral(succ(succ(0))).

• **Query:**
  ?- numeral(N).
Recursion

• **Addition:**
  - $0 + X = X$
  - $X + Y = 1 + (X-1 + Y)$

• **Example:**
  1. $3 + 2$
  2. $= 1 + (2 + 2)$
  3. $= 1 + (1 + (1 + 2))$
  4. $= 1 + (1 + (1 + (0 + 2)))$
  5. $= 1 + (1 + (1 + 2))$
  6. $= 1 + (1 + 3)$
  7. $= 1 + 4$
  8. $= 5$

```prolog
add(0, Y, Y).
add(succ(X), Y, succ(Z)) :-
    add(X, Y, Z).
```
Goal Ordering

• **Example 2: Descendant**
  child(anne,bridget).
  child(bridget,caroline).
  child(caroline,donna).
  child(donna,emily).
  descend(X,Y) :- child(X,Y).
  descend(X,Y) :- child(X,Z), descend(Z,Y).

• **Swap the conjunctive definition order:**
  descend(X,Y) :- descend(Z,Y), child(X,Z).

• **Query:**
  ?- descend(anne,donna).

- Prolog procedural order: left to right