Administrative Issues

- HW5: May 4th
- Essay: May 1st
  ABET requirement: you must get a passing score (37.5), otherwise you will receive Incomplete (I) for this course
- Final Review: May 1st
- Final exam: open notes and open book
  Thursday, May 8th, 2-4:30pm, GACB 105

Logic Programming: Prolog (III)

Loops and Control:

fail and cut (!)

Example

- Print all solutions of appending.
  printpieces(List) :- append(X,Y,List),
  write(X),
  write(Y),
  nl,
  fail.

  ?- printpieces([1,2]).
  [] [1,2]
  [1,2] []
  fail.

Example

- num(0).
  num(X) :- num(Y), X is Y+1.

  ;-) num(X).

- writenum(L,J) :- num(X),
  I is L,
  X <= J,
  write(X),
  nl,
  fail.

  ?- writenum(1,10). (Will this work?)
cut

Cut (using built-in predicate !) branches in the search tree (to avoid infinite loop).

• "freeze" the choice made when "!" is encountered.
• If "!" is reached on backtracking, the search of the subtree at the parent node of the node containing "!" stops, and the search continues with the "grandparent" node.

Example

• writenum(I,J) :- num(X),
  I =< X,
  X =< J,
  write(X),
  nl,
  X = J, !,
  fail.

?- writenum(1,10).

Where to cut exactly?

Where to cut exactly?
Where to cut exactly?

\[ X, \ldots, !, \ldots, Y \]

The first time when all the subgoals before the ! are solved

\[ \text{Backtracking doesn't come here} \]

Solving the goals after the !

Backtracking does go to these subtrees

The cut (!) goal always succeeds.

All the choices made before ! are frozen.

Along the path from the node where ! is introduced into the goals till the node where ! is reached (all previous goals satisfied), all the siblings of these nodes are pruned.
Exercise 1

- Duplicate the elements in a list, using the numbers of duplicates specified in another list.

- E.g.,

  ?- duplicate([1,5,3], [2,1,4], Result).

Result = [1,1,5,3,3,3,3].

```prolog
duplicate_single(H, 0, []).
duplicate_single(H, N, [H|U]) :- N>0, M is N-1, duplicate_single(H,M,U).
duplicate_list([], _, []).
duplicate_list([H|T], [N|Ns], Result) :- duplicate_single(H,N,U), duplicate_list(T,Ns,V), append(U,V,Result).
```

Exercise 2

- An example with predicates containing predicates

- Binary Tree

  e.g.,
  tree(1,tree(2,tree(4,tree(8,void,void),void),tree(5,void,tree(9,void,void))),
  tree(3,tree(6,void,tree(10,void,void)),tree(7,tree(11,void,void),void)));

  ordered(T): true if T is ordered
  (For each node n, all nodes in n's left subtree are smaller than n, and all nodes in n's right subtree are larger than n. Assuming no duplicates).

  ordered(T) :- ordered(T, Min, Max).
  ordered(tree(X, void, void), X, X).
  ordered(tree(X, L, void), Min, X) :- ordered(L, Min, Max), X>Max.
  ordered(tree(X, void, R), X, Max) :- ordered(R, Min, Max), X<Min.
  ordered(tree(X, L, R), Min, Max) :- ordered(L, Min, Max1), ordered(R, Min2, Max), X=Max1, X<Min2.

Exercise 3

```prolog
student(Amy). student(Bob).
take(Amy, CSE3302). take(Amy, CSE3303). take(Amy, CSE3304).
take(Bob, CSE3301). take(Bob, CSE3303). credit(CSE3301, 3). credit(CSE3302, 3).
credit(CSE3303, 4). credit(CSE3304, 2).
?- student(X), take(X, Y), credit(Y,Z).
?- !, student(X), take(X, Y), credit(Y,Z).
?- student(X), take(X, Y), !, credit(Y,Z).
?- student(X), take(X, Y), credit(Y,Z), !.
```

Lecture 23 – Prolog (II), Spring 2008 CSE3302 Programming Languages, UT-Arlington ©Chengkai Li, 2008