

# CSE 341: Section 7, May 8th -- Prolog

## Prolog Programming Level 1

### Q1. Positive

Write a rule positive that succeeds if all elements of a list are positive, and fails otherwise.

### Q2. Reverse

Write a Prolog rule to reverse a list.

### Q3. Last

Write a rule last to find the last element of a list

### Q4. Sum

Write a Prolog rule to sum the numbers in a list. (You can assume that the list consists of numbers.)

## Derivation Trees

### Q1. Member

Draw the derivation tree for the following goal (The rules are included below for reference):

?- mymember(A,[1,2,3]).

mymember(X,[X|\_]).

mymember(X,[\_|Ys]) :- mymember(X,Ys).

### Q2. Reverse

Draw the derivation tree for the following goal (The rules will be discussed in section):

?- reverse([1],R).

## Prolog Programming Level 2

### Q1. Deduplicate

Write remove\_dupl that removes duplicates from a list (and orders the elements in the de-duplicated list by the order of the elements' last occurrence in the original list).

### Q2. Set Operations

Write rules set\_diff and set\_int to find the set difference/intersection of 2 sets. (Assume input to be valid sets). (Recall that for two sets A and B, set difference  $A \setminus B$  is all the elements in A that were not in B)

### Q3. Take

Write take that takes N elements from a list. (Assume  $N \leq$  number of elements in the list)

### Q4. MyMax

Write a rule my\_max that finds the maximum element of a list of numbers. (Hint: Use clpr)

```
:- use_module(library(clpr)).
```