

## Prolog - Part II

- Turing complete language, using BACKWARDS REASONING
- consult(program.pl)
- All rules defining predicate must show up successively
- don't assume direction of evaluation
- Recursively defined predicates

① Ball throwing example:

- base case:  $\text{reaches}(X, Y) = \text{throws}(X, Y)$
- step case:  $\text{reaches}(X, Z) = \text{throws}(X, Y), \text{reaches}(Y, Z)$

① Lists in Prolog:  $[\text{Head} | []] = [\text{Head}]$

$$[A | [B | \text{Tail}]] = [A | B, \text{Tail}]$$

- member\_of( $X, [\text{Head} | \text{Tail}]$ )  
 $X = \text{Head}$       don't care about this variable

- member\_of( $X, [_\text{Head} | \text{Tail}]$ )  
member\_of( $X, [\text{Head} | \text{Tail}]$ )

- nonmember\_of( $X, [\text{Head} | \text{Tail}]$ )  
different( $X, \text{head}$ )  
nonmember( $X, [\text{Tail}]$ )

- BASE: nonmember( $X, []$ ) = True

① Will it stop?  $\rightarrow$  append false  
visiting ALL possible answers  $\rightarrow$  [enforces backtracking]  
 $\rightarrow$  Remember:  $X \wedge \perp \equiv \perp$

① Consequences of non-termination  
 $\rightarrow$  All recursive goals loop if  $\neg$  instantiated variables  
 $\rightarrow$  Debugging: replacing constants with variables doesn't help

① Accumulators:  
 $\rightarrow$  add additional arguments to pass history to recursive goals  
 $\rightarrow$  Queries can't stop if set of paths is infinite  
② Consider only acyclic paths

① Narrowing answer set: easy way to improve termination properties

