CS131 Week 6

Week 6 Lecture 1

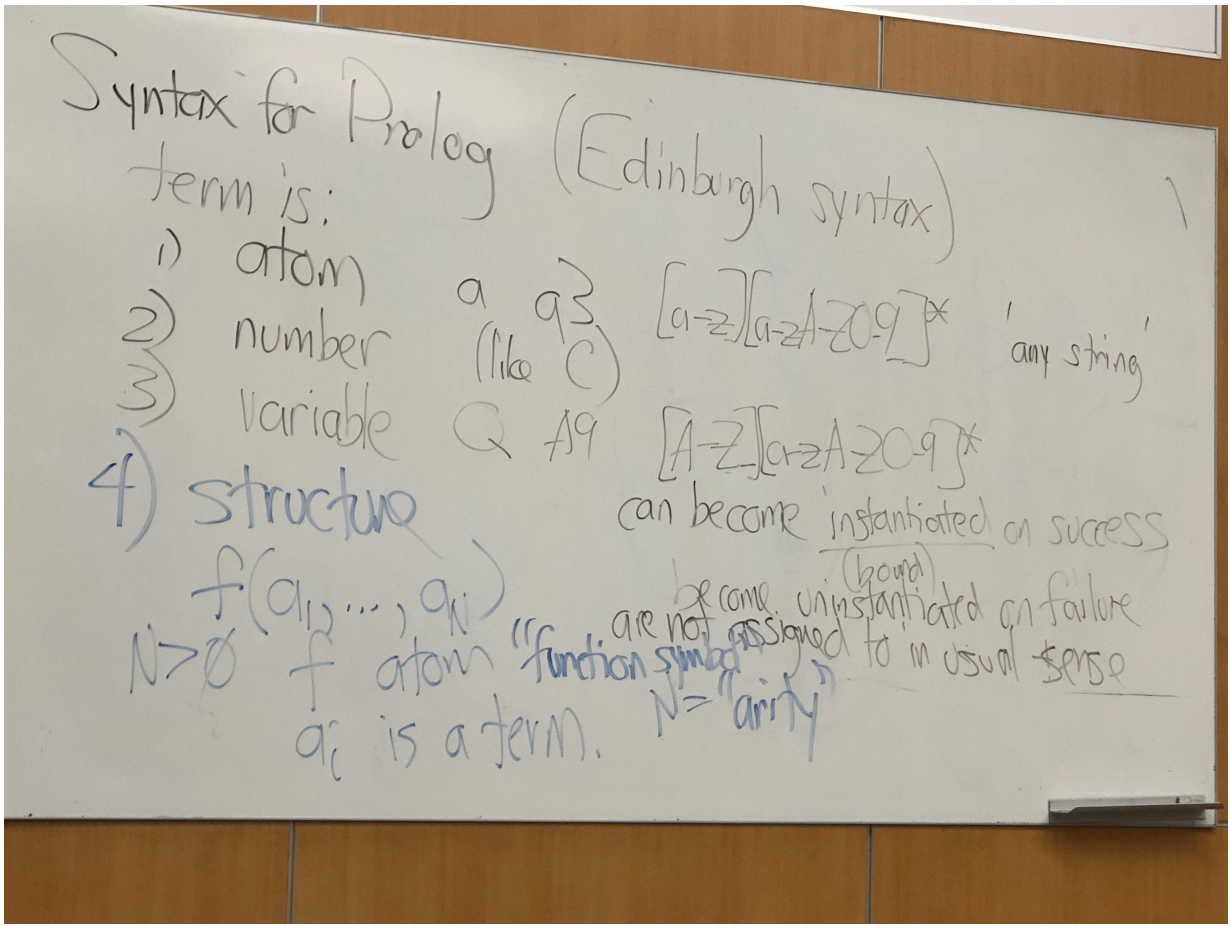
==========

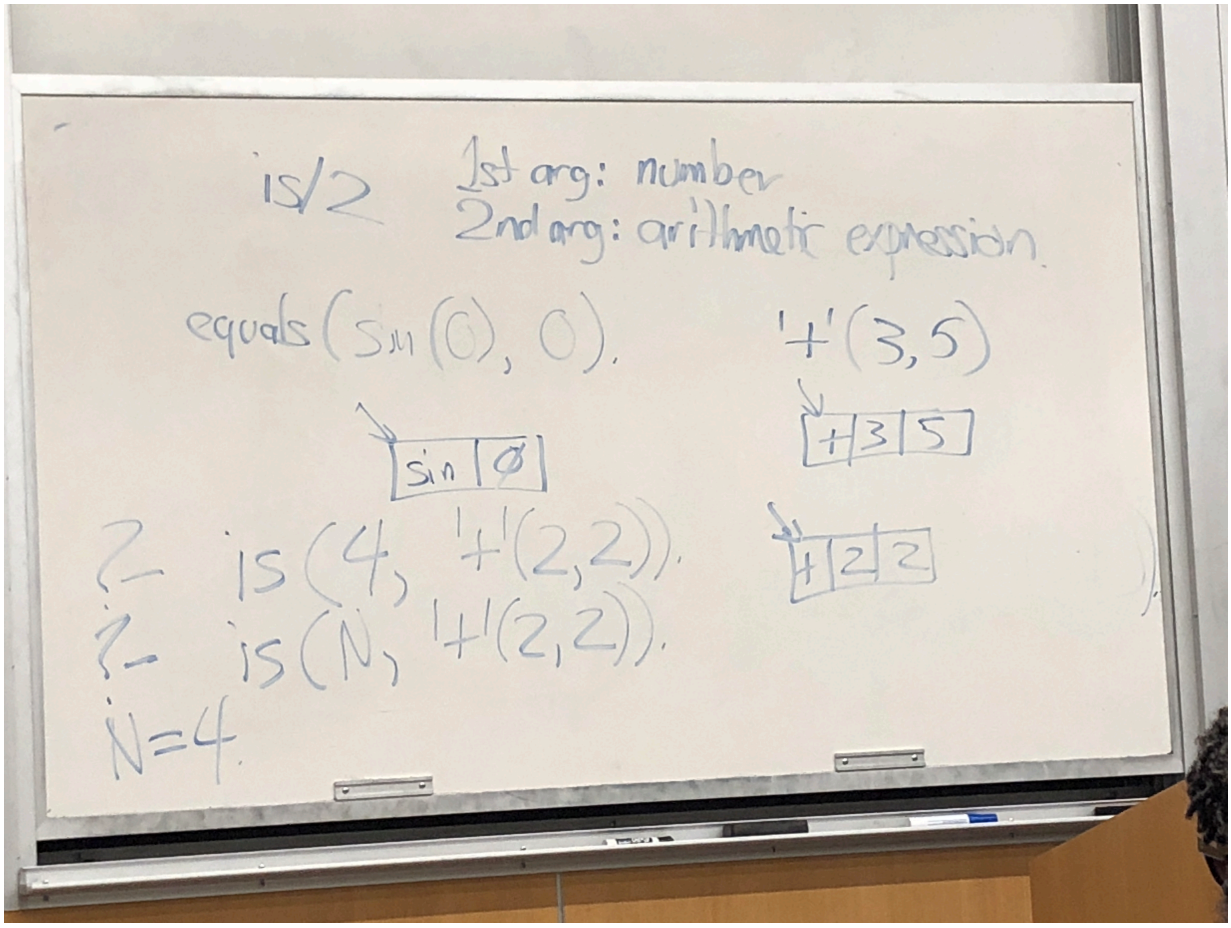
**2nd half of lecture*

Syntax for Prolog (Edinburgh syntax)
Term is:

1. atom
2. number
3. variable
4. structure

“function symbol”
N = “arity”





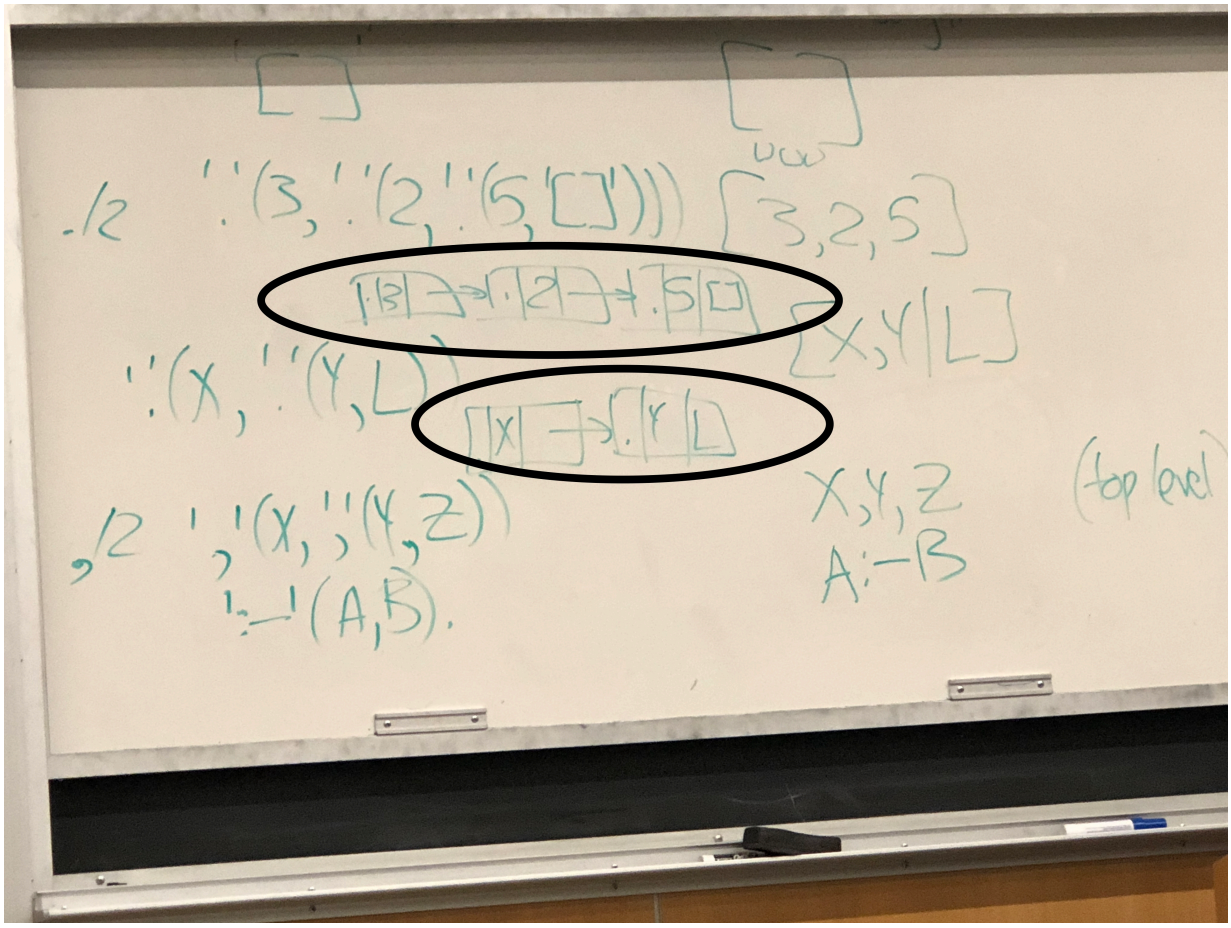
ground term: a term that doesn't contain any (logical) variables

?- is (1369, '*'(N, N)) <= problem: * (N, N) expects two numbers, but gets N

'+' (2, 2)
?- is (N, '+' (2, 2)).

Prolog syntactic sugar hints:

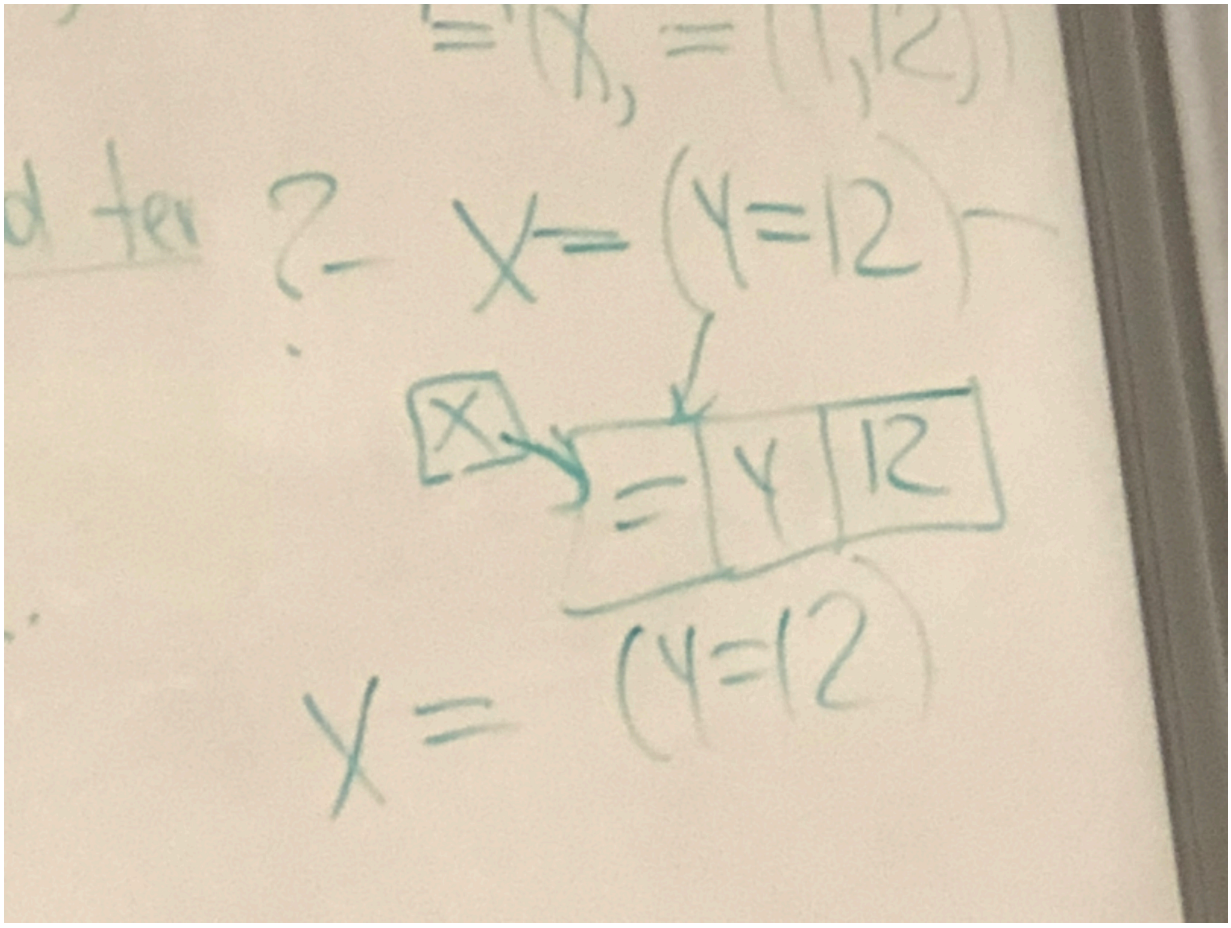
Core syntax	Syntactic Sugar
'[]'	[ ]
./2 '.' (3, '.' (2, '.' (5, '[]')))	[3,2,5]
'.' (X, '.' (Y, '[]'))	[X, Y | L]
,/2 ','(X, ',' (Y,Z))	X, Y, Z



Ending notes:
Non-associative operators
- :- // turnstyle
- = // equals

i.e. can't do `a :- b :- c :- d`

Unbound variables
Not known what the value is equivalent to
Can fill the unbound variables later
?- X = Y, Y = 2
^X, Y both unbound, bound to each other ^ Y and X now both 2

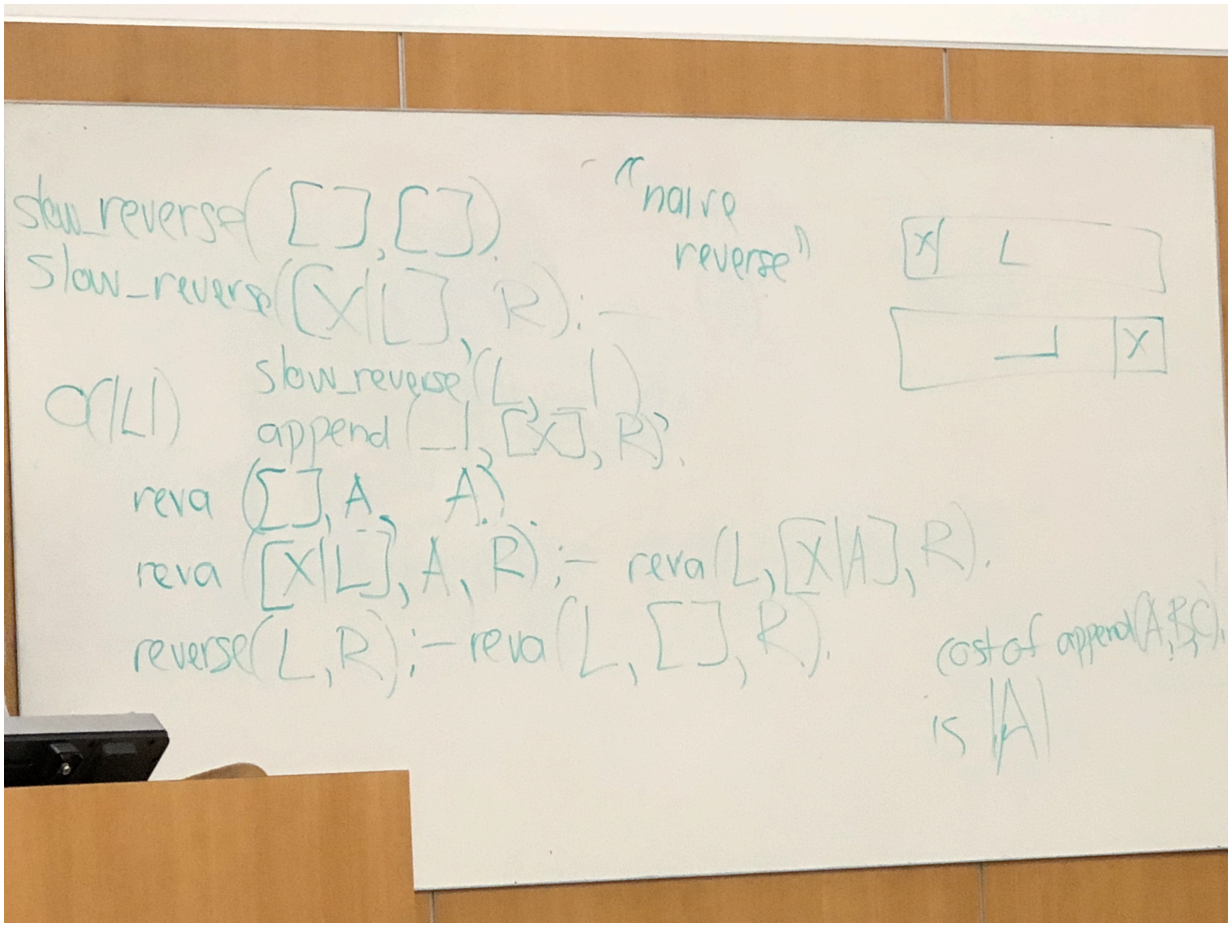


Predicate Examples

```
append_([],L2,L2).
append_([H1|T1],L2,[H1|TRes]):- append_(T1,L2,TRes).

% prefix L=[1,2,3,4,5]. so anything from  [],[1],[1,2],[1,2,3],[1,2,3,4], [1,2,3,4,5] will be a yes
prefix_(P,L) :- append_(P,_,L).
suffix_(S,L) :- append_(_,S,L).

dot([H1],[H2],HRes):-HRes is H1*H2.
dot([H1|T1],[H2|T2],Res):-HRes is H1*H2,
                    dot(T1,T2,TRes),
                    Res is HRes+TRes.
```



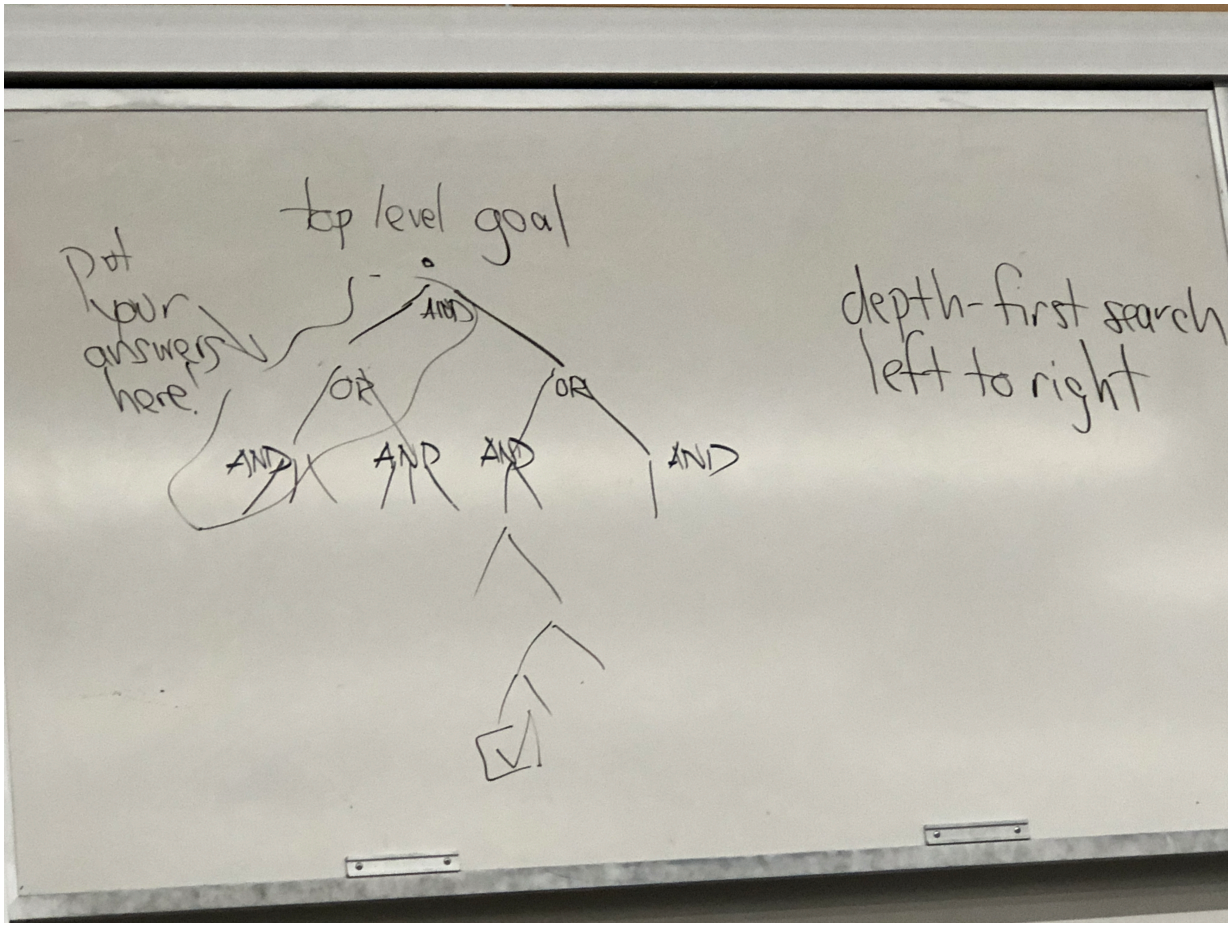
— 2/13/18

Week 6 Lecture 2

==========

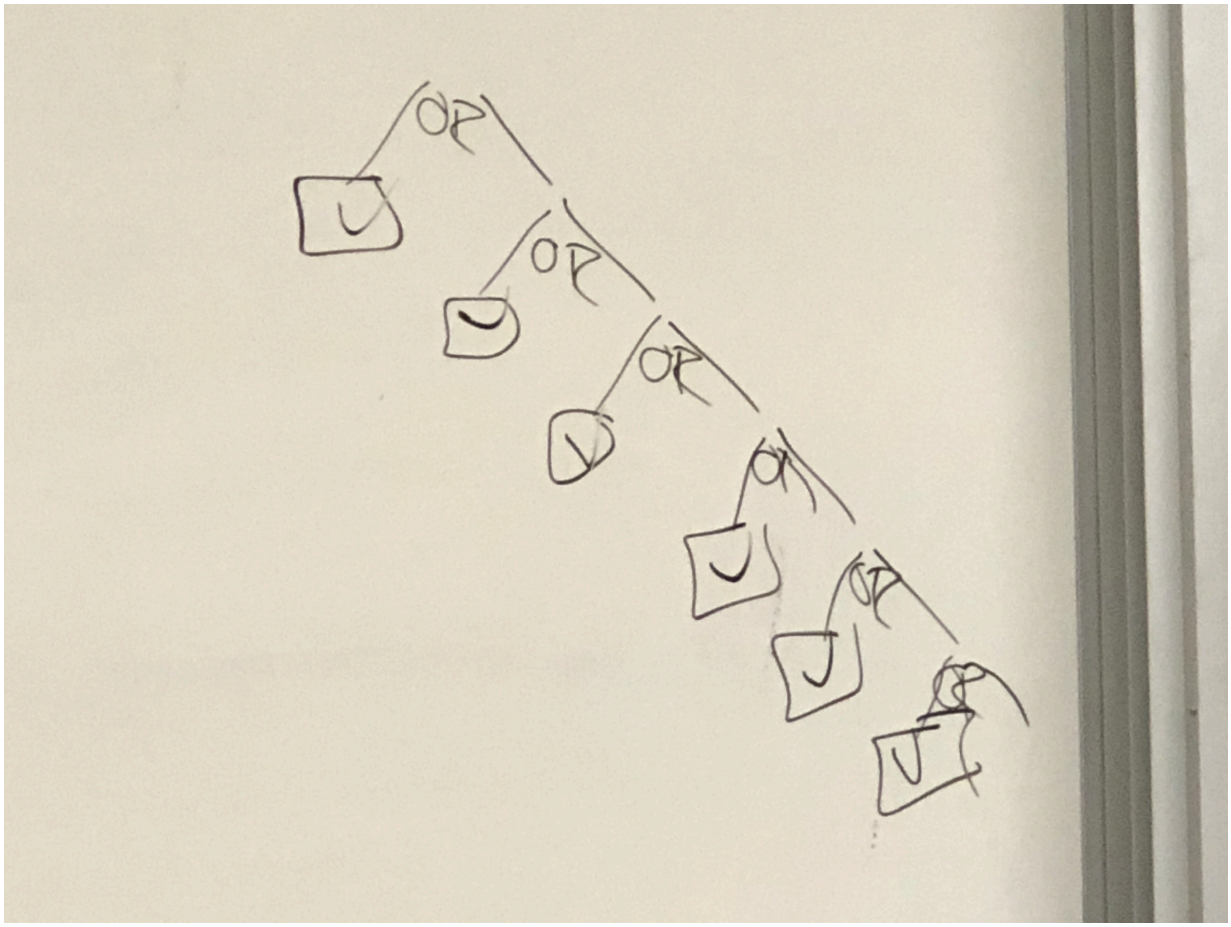
More Prolog!

- Facts are terms, no **logical connectives**
  - pr (cs32, cs131)
- Rules can use ':-' ',' 'if' 'and'
  - tcpr(A, B) :- pr(A, B)
  - tcpr(A, Z) :- pr(A, B), tcpr(B, Z)
- Queries (goals) ?- pr(A, B), tcpr(B, A). can use AND

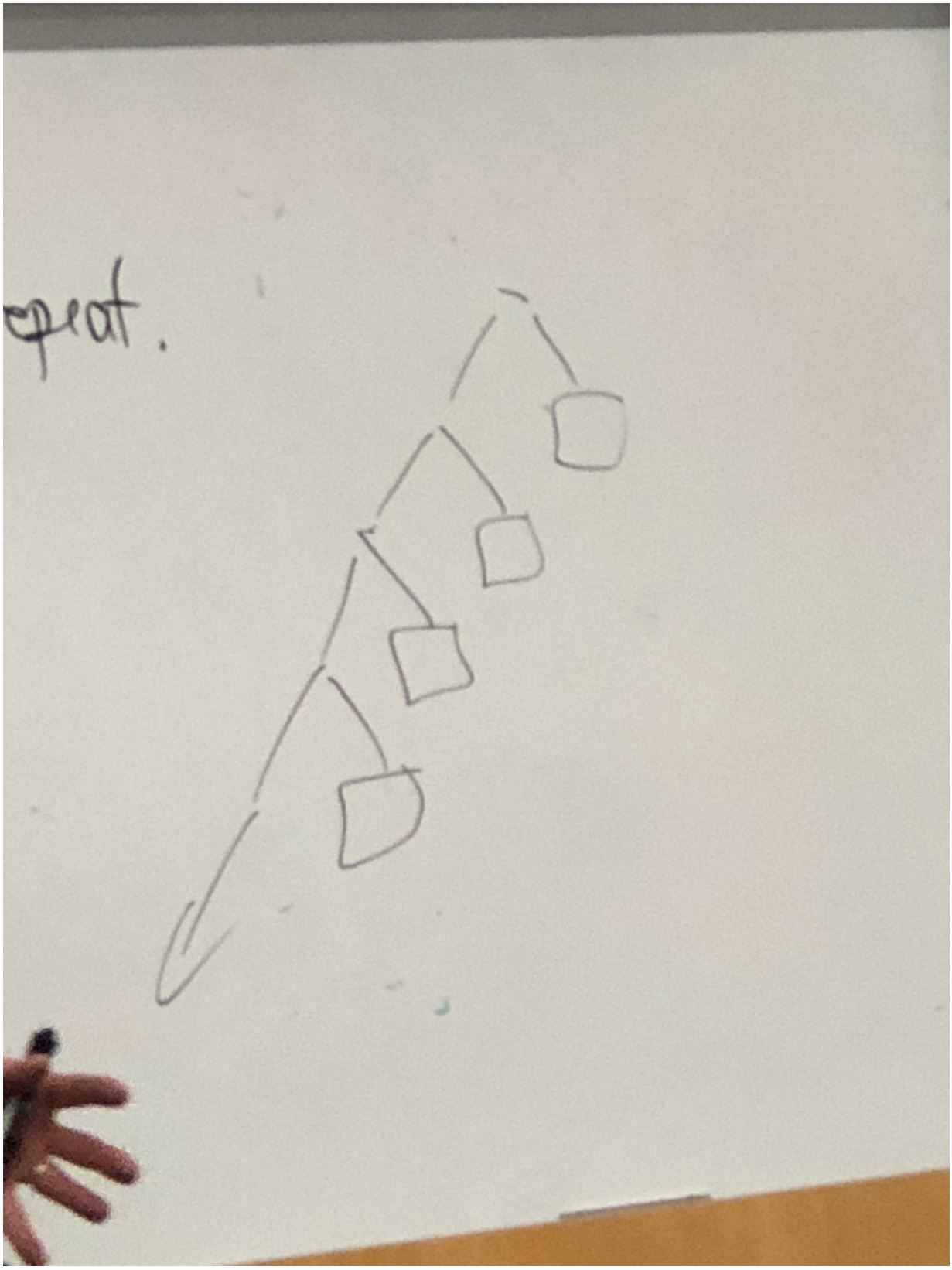


repeat.
repeat :- repeat

?- repeat.
yes
?- repeat, newline, fail.




```
badrepeat :- badrepeat.
badrepeat.
```



Debugging

- simplest way to debug is to **print**