## Today
- Prolog Proof Trees

## Assignments
- HW-8, R-8 due Thurs
- HW-9, R-9 out

## Assignments
- Quiz on Thurs
Proof trees

A proof tree is a way to view query evaluation (Prolog computation)

- specifies a “trace” of a Prolog program
- follows evaluation order (with unification and resolution)

The KB:

1. r(a).
2. s(X) :- r(X).

List of goals to “solve”

\[
\text{solve} [s(a)]
\]

Result of resolution step

\[
\text{nothing} \quad \text{solve} [r(a)]
\]

One child per program clause

\[
\text{solve} []
\]

Finished with clauses to solve (success!)
**Proof tree definition**

A proof tree ...

- Contains **nothing** and **solve** nodes
- Each **nothing** node is a leaf (no children)
- Each **solve** node contains a list of terms:
  - if empty list, the **solve** node is a leaf
  - otherwise the **solve** node has one child per program clause (in order)
    - if a clause does not unify with the head of the query list, the child is a **nothing** node
    - if it does unify, the query is replaced (unfolded) in the list using resolution
- The **root** is a **solve** node with the initial query terms (goal)
Another example

The KB:

1. p(X) :- q(X), r(X).
2. q(a).
3. q(b).
4. r(a).

?- p(X).

The proof tree:

- Note we often put bindings on the edges (e.g., X = a)
- Prolog evaluation performs a depth-first left-to-right search
And another example

The KB:

1. \( p(X) :- p(X), q(X) \).
2. \( q(a) \).
3. \( p(a) \).

\( ?- p(X) \).

Q: See a potential problem?

The proof tree:

- Creates an **infinite** proof tree!
And one last one

The KB:

1. \( p(a) \).
2. \( q(a) \).
3. \( p(X) :- p(X), q(X) \).

\(?- p(X) \).

The proof tree:

- Also creates an \textbf{infinite} proof tree ... but generates solutions!
  - What is wrong with the recursion? ... doesn’t make “progress”
Variable naming and proof trees

Be careful with variable names

\[ r(X, Y) \implies r(X, Z), s(Z, Y). \]

\[
\begin{aligned}
\text{solve} \ [r(X, Y)] \\
\quad \mid \\
\text{solve} \ [r(X, Z), s(Z, Y)] \\
\quad \mid \\
\text{solve} \ [r(X, Z1), s(Z1, Z), s(Z, Y)]
\end{aligned}
\]

- We cannot use Z here again!
- So we rename to Z1 ...
More examples:

- **“Joins”**

  \[
  \begin{align*}
  r(a,b). \\
  s(b,c). \\
  q(X,Z) & : = r(X,Y), s(Y,Z). \\
  \\
  \text{?- q(a,Y).}
  \end{align*}
  \]

- **Transitive closure**

  \[
  \begin{align*}
  e(a,b). \\
  e(b,c). \\
  tc(X,Y) & : = e(X,Y). \\
  tc(X,Y) & : = e(X,Z), tc(Z,Y). \\
  \\
  \text{?- tc(a,Y).} \quad \% \text{what is reachable from a?}
  \end{align*}
  \]