Ask-the-user meta-interpreter

% *aprove*(G) is true if G is a logical consequence of the base-level KB and yes/no answers provided by the user.

$$
\begin{aligned}
&aprove(true).\\
&aprove((A \,\&\, B)) \leftarrow aprove(A) \wedge aprove(B).\\
&aprove(H) \leftarrow askable(H) \wedge answered(H, yes).\\
&aprove(H) \leftarrow\\
&\quad askable(H) \wedge unanswered(H) \wedge ask(H, Ans) \wedge\\
&\quad record(answered(H, Ans)) \wedge Ans = yes.\\
&aprove(H) \leftarrow (H \Leftarrow B) \wedge aprove(B).
\end{aligned}
$$

Meta-interpreter to collect rules for WHY

% $wprove(G, A)$ is true if G follows from base-level KB, and A is a list of ancestor rules for G.

$$
\begin{array}{l}
wprove(true, Anc). \\
wprove((A \,\&\, B), Anc) \leftarrow \\
\qquad wprove(A, Anc) \wedge \\
\qquad wprove(B, Anc). \\
wprove(H, Anc) \leftarrow \\
\qquad (H \Leftarrow B) \wedge \\
\qquad wprove(B, [(H \Leftarrow B)|Anc]).
\end{array}
$$

Delaying Goals

Some goals, rather than being proved, can be collected in a list.

- To delay subgoals with variables, in the hope that subsequent calls will ground the variables.
- To delay assumptions, so that you can collect assumptions that are needed to prove a goal.
- To create new rules that leave out intermediate steps.
- To reduce a set of goals to primitive predicates.

Delaying Meta-interpreter

% $dprove(G, D_0, D_1)$ is true if D_0 is an ending of list of delayable atoms D_1 and $KB \wedge (D_1 - D_0) \models G$.

$$
\begin{aligned}
& dprove(true, D, D). \\
& dprove((A \,\&\, B), D_1, D_3) \leftarrow \\
& \qquad dprove(A, D_1, D_2) \wedge dprove(B, D_2, D_3). \\
& dprove(G, D, [G|D]) \leftarrow delay(G). \\
& dprove(H, D_1, D_2) \leftarrow \\
& \qquad (H \Leftarrow B) \wedge dprove(B, D_1, D_2).
\end{aligned}
$$

Example base-level KB

$live(W) \Leftarrow$
$\quad connected\_to(W, W_1)\ \&$
$\quad live(W_1).$
$live(outside) \Leftarrow true.$
$connected\_to(w_6, w_5) \Leftarrow ok(cb_2).$
$connected\_to(w_5, outside) \Leftarrow ok(outside\_connection).$
$delay(ok(X)).$
$?dprove(live(w_6), [\,], D).$

Meta-interpreter that builds a proof tree

% $hprove(G, T)$ is true if G can be proved from the base-level KB, with proof tree T.

$$
\begin{aligned}
& hprove(true, true). \\
& hprove((A \,\&\, B), (L \,\&\, R)) \leftarrow \\
& \qquad hprove(A, L) \wedge \\
& \qquad hprove(B, R). \\
& hprove(H, if(H, T)) \leftarrow \\
& \qquad (H \Leftarrow B) \wedge \\
& \qquad hprove(B, T).
\end{aligned}
$$