Vanilla Meta-interpreter

$prove(G)$ is true when base-level body G is a logical consequence of the base-level KB.

$$prove(true).$$

$$prove((A \,\&\, B)) \leftarrow$$
$$\quad prove(A) \wedge$$
$$\quad prove(B).$$

$$prove(H) \leftarrow$$
$$\quad (H \Leftarrow B) \wedge$$
$$\quad prove(B).$$

Example base-level KB

$$live(W) \Leftarrow$$
$$\quad connected\_to(W, W_1) \ \&$$
$$\quad live(W_1).$$
$$live(outside) \Leftarrow true.$$
$$connected\_to(w_6, w_5) \Leftarrow ok(cb_2).$$
$$connected\_to(w_5, outside) \Leftarrow true.$$
$$ok(cb_2) \Leftarrow true.$$
$$?prove(live(w_6)).$$

Expanding the base-level

Adding clauses increases what can be proved.

- **Disjunction** Let $a; b$ be the base-level representation for the disjunction of a and b. Body $a; b$ is true when a is true, or b is true, or both a and b are true.

- **Built-in predicates** You can add built-in predicates such as N is E that is true if expression E evaluates to number N.

Expanded meta-interpreter

$$prove(true).$$

$$prove((A \,\&\, B)) \leftarrow$$

$$\quad prove(A) \wedge prove(B).$$

$$prove((A; B)) \leftarrow prove(A).$$

$$prove((A; B)) \leftarrow prove(B).$$

$$prove((N \text{ is } E)) \leftarrow$$

$$\quad N \text{ is } E.$$

$$prove(H) \leftarrow$$

$$\quad (H \Leftarrow B) \wedge prove(B).$$

Depth-Bounded Search

- Adding conditions reduces what can be proved.

% $bprove(G, D)$ is true if G can be proved with a proof tree

% of depth less than or equal to number D.

$$bprove(true, D).$$

$$bprove((A \,\&\, B), D) \leftarrow$$

$$bprove(A, D) \wedge bprove(B, D).$$

$$bprove(H, D) \leftarrow$$

$$D \geq 0 \wedge D_1 \text{ is } D - 1 \wedge$$

$$(H \Leftarrow B) \wedge bprove(B, D_1).$$

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 $KB \wedge (D_1 - D_0) \models G$.

$$dprove(true, D, D).$$

$$dprove((A \,\&\, B), D_1, D_3) \leftarrow$$

$$\quad dprove(A, D_1, D_2) \wedge dprove(B, D_2, D_3).$$

$$dprove(G, D, [G|D]) \leftarrow delay(G).$$

$$dprove(H, D_1, D_2) \leftarrow$$

$$\quad (H \Leftarrow B) \wedge dprove(B, D_1, D_2).$$

Example base-level KB

$$live(W) \Leftarrow$$

$$connected\_to(W, W_1) \ \&$$

$$live(W_1).$$

$$live(outside) \Leftarrow true.$$

$$connected\_to(w_6, w_5) \Leftarrow ok(cb_2).$$

$$connected\_to(w_5, outside) \Leftarrow ok(outside\_connection).$$

$$delay(ok(X)).$$

$$?dprove(live(w_6), [\,], D).$$