Figure 7.211 A hybrid agent program for the wumpus world. It uses a propositional knowledge base to infer the state of the world and a combination of problem-solving search and domain-specific code to decide what actions to take.

by a unique binary number, we would need numbers with \(\log_2(2^n) = 2^n\) bits to label the current belief state. That is, exact state estimation may require logical formulas whose size is exponential in the number of symbols.

One very common and natural scheme for approximate state estimation is to represent belief states as conjunctions of literals, that is, \(1\text{-CNF}\) formulas. To do this, the agent program simply tries to prove \(X^t\) and \(-X^t\) for each symbol \(X\) (as well as each atemporal symbol whose truth value is not yet known), given the belief state at \(t - 1\). The conjunction of
Section 7.7. Agents Based on Propositional Logic

It is important to understand that this scheme may lose some information as time goes along. For example, if the sentence in Equation (7.4) were the true belief state, then neither $P_{31}$ nor $P_{2.2}$ would be provable individually and neither would appear in the $1$-CNF belief state. (Exercise 7.27 explores one possible solution to this problem.) On the other hand, because every literal in the $1$-CNF belief state is proved from the previous belief state, and the initial belief state is a true assertion, we know that entire $1$-CNF belief state must be true. Thus, the set of possible states represented by the $1$-CNF belief state includes all states that are in fact possible given the full percept history. As illustrated in Figure 7.21, the $1$-CNF belief state acts as a simple outer envelope, or conservative approximation, around the exact belief state. We see this idea of conservative approximations to complicated acts as a recurring theme in many areas of AL.

7.7.4 Making plans by propositional inference

The agent in Figure 7.20 uses logical inference to determine which squares are safe, but uses $A^*$ search to make plans. In this section, we show how to make plans by logical inference. The basic idea is very simple:

1. Construct a sentence that includes
   
   (a) $\text{init}^+$, a collection of assertions about the initial state;
   
   (b) $\text{Transition}^+$, the successor-state axioms for all possible actions at each time up to some maximum time $t$;
   
   (c) the assertion that the goal is achieved at time $t$: $\text{HaveGold} \land \text{ClimbedOut}^t$.