Chapter 7. Logical Agents

Given a complete set of successor-state axioms and the other axioms listed at the beginning of this section, the agent will be able to ASK and answer any answerable question about the current state of the world. For example, in Section 7.2 the initial sequence of percepts and actions is

\[ \neg \text{Stench} \land \neg \text{Breeze} \land \neg \text{Glitter} \land \neg \text{Bump} \land \neg \text{Scream} \implies \text{Forward} \]
\[ \neg \text{Stench} \land \text{Breeze} \land \neg \text{Glitter} \land \neg \text{Bump} \land \neg \text{Scream} \implies \text{TurnRight} \]
\[ \neg \text{Stench} \land \text{Breeze} \land \neg \text{Glitter} \land \neg \text{Bump} \land \neg \text{Scream} \implies \text{Forward} \]
\[ \neg \text{Stench} \land \text{Breeze} \land \text{Glitter} \land \text{Bump} \land \text{Scream} \implies \text{TurnRight} \]
\[ \text{Renee} \land \neg \text{Breeze} \land \text{Glitter} \land \text{Bump} \land \text{Scream} \implies \text{Forward} \]

At this point, we have \( \text{ASK}(KB, I_{12}^6) = \text{true} \), so the agent knows where it is. Moreover, \( \text{ASK}(KB, W_{t_{12}}) = \text{true} \) and \( \text{SIC}(KB, P_{t_{12}}) = \text{true} \), so the agent has found the wumpus and one of the pits. The most important question for the agent is whether a square is OK to move into, that is, the square contains no pit nor live wumpus. It's convenient to add axioms for this, having the form

\[ \text{OK}_{2_{12}}^{t_{12}} \iff (P_{x,a} \land W_{x,a} \land \text{WumpusAlive}) \]

Finally, \( \text{ASK}(KB, OK_{2_{12}}^{t_{12}}) = \text{true} \), so the square [2, 2] is OK to move into. In fact, given a sound and complete inference algorithm such as DPLL, the agent can answer any answerable question about which squares are OK—and can do so in just a few milliseconds for small-to-medium wumpus worlds.

Solving the representational and inferential frame problems is a big step forward, but a pernicious problem remains: we need to confirm that all the necessary preconditions of an action hold for it to have its intended effect. We said that the Forward action moves the agent ahead unless there is a wall in the way, but there are many other unusual exceptions that could cause the action to fail: the agent might trip and fall, be stricken with a heart attack, be carried away by giant bats, etc. Specifying all these exceptions is called the \textit{qualification problem}.

There is no complete solution within logic; system designers have to use good judgment in deciding how detailed they want to be in specifying their model, and what details they want to leave out. We will see in Chapter 13 that probability theory allows us to summarize all the exceptions without explicitly naming them.

7.7.2 A hybrid agent

The ability to deduce various aspects of the state of the world can be combined fairly straightforwardly with condition-action rules and with problem-solving algorithms from Chapters 3 and 4 to produce a \textit{hybrid agent} for the wumpus world. Figure 7.20 shows one possible way to do this. The agent program maintains and updates a knowledge base as well as a current plan. The initial knowledge base contains the \textit{atemporal} axioms—those that don't depend on \( t \), such as the axiom relating the breeziness of squares to the presence of pits. At each time step, the new percept sentence is added along with all the axioms that depend on \( t \), such
as the successor-state axioms. (The next section explains why the agent doesn’t need axioms for future time steps.) Then, the agent uses logical inference, by ASKing questions of the knowledge base, to work out which squares are safe and which have yet to be visited.

The main body of the agent program constructs a plan based on a decreasing priority of goals. First, if there is a glitter, the program constructs a plan to grab the gold, follow a route back to the initial location, and climb out of the cave. Otherwise, if there is no current plan, the program plans a route to the closest safe square that it has not visited yet, making sure the route goes through only safe squares. Route planning is done with A* search, not with ASK. If there are no safe squares to explore, the next step—if the agent still has an arrow—is to try to make a safe square by shooting at one of the possible wumpus locations. These are determined by asking where ASK(KB, ¬W) is false—that is, where it is not known that there is not a wumpus. The function PLAN-SHOT (not shown) uses PLAN-ROUTE to plan a sequence of actions that will line up this shot. If this fails, the program looks for a square to explore that is not provably unsafe—that is, a square for which ASK[KB, OK] returns false. If there is no such square, then the mission is impossible and the agent retreats to [1, 1] and climbs out of the cave.

### 7.7.3 Logical state estimation

The agent program in Figure 7.20 works quite well, but it has one major weakness: as time goes by, the computational expense involved in the calls to ASK goes up and up. This happens mainly because the required inferences have to go back further and further in time and involve more and more propositional symbols. Obviously, this is unsustainable—we cannot have an agent whose time to process each percept grows in proportion to the length of its life! What we really need is a constant update time—that is, independent of time.

The obvious answer is to save, or cache, the results of inference, so that the inference process at the next time step can build on the results of earlier steps instead of having to start again from scratch.

As we saw in Section 4.4, the past history of percepts and all their ramifications can be replaced by the belief state—that is, some representation of the set of all possible current states of the world. The process of updating the belief state as new percepts arrive is called state estimation. Whereas in Section 4.4 the belief state was an explicit list of states, here we can use a logical sentence involving the proposition symbols associated with the current time step, as well as the atemporal symbols. For example, the logical sentence

\[ \text{WumpusAlive} \land L_{2,1} \land B_{2,1} \land (P_{1,1} \lor P_{2,2}) \]  

represents the set of all states at time 1 in which the wumpus is alive, the agent is at [2, 1], that square is breezy, and there is a pit in [3, 1] or [2,2] or both.

Maintaining an exact belief state as a logical formula turns out not to be easy. If there are n fluent symbols for time t, then there are 2^n possible states—that is, assignments of truth values to those symbols. Now, the set of belief states is the powerset (set of all subsets) of the set of physical states. There are 2^n physical states, hence 2^{2^n} belief states. Even if we used the most compact possible encoding of logical formulas, with each belief state represented

---

12 We can think of the percept history itself as a representation of the belief state, but one that makes inference increasingly expensive as the history gets longer.