PartB-I

13.25 Implement a hybrid probabilistic agent for the wumpus world, based on the hybrid agent in Figure 7.20 and the probabilistic inference procedure outlined in this chapter.

After each move, the safety probability for each square that is not provably safe or fatal, and choose the safest if there is no unvisited safe square.

function HYBRID-WUMPUS-AGENT(*percept*) **returns** an *action*
  inputs: *percept*, a list, [*stench*, *breeze*, *glitter*, *bump*, *scream*]
  persistent: *KB*, a knowledge base, initially the atemporal "wumpus physics"
    t, a counter, initially 0, indicating time
    plan, an action sequence, initially empty

  TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
  TELL the *KB* the temporal "physics" sentences for time *t*
  safe ← {[*x*, *y*] : ASK(*KB*, $OK^t_{x,y}$) [redacted]}
  [redacted]
    plan ← [*Grab*] + PLAN-ROUTE(*current*, {[1,1]}, *safe*) + [*Climb*]
  if *plan* is empty **then**
    unvisited ← {[*x*, *y*] : ASK(*KB*, $L^{t'}_{x,y}$) [redacted] for all $t' \leq t$}
    plan ← PLAN-ROUTE(*current*, *unvisited* ∩ *safe*, *safe*)
  if *plan* is empty and ASK(*KB*, $HaveArrow^t$) [redacted] **then**
    possible_wumpus ← {[*x*, *y*] : ASK(*KB*, ¬ $W_{x,y}$) = *false*}
    plan ← PLAN-SHOT(*current*, *possible_wumpus*, *safe*)
  if *plan* is empty **then** // no choice but to take a risk
    not_unsafe ← {[*x*, *y*] : ASK(*KB*, ¬ $OK^t_{x,y}$) [redacted]}
    plan ← PLAN-ROUTE(*current*, *unvisited* ∩ *not_unsafe*, *safe*)
  if *plan* is empty **then**
    plan ← PLAN-ROUTE(*current*, {[1, 1]}, *safe*) + [*Climb*]
  action ← POP(*plan*)
  TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
  t ← *t* + 1
  return *action*

function PLAN-ROUTE(*current*, *goals*, *allowed*) **returns** an action sequence
  inputs: *current*, the agent's current position
    goals, a set of squares; try to plan a route to one of them
    allowed, a set of squares that can form part of the route

  problem ← ROUTE-PROBLEM(*current*, *goals*, *allowed*)
  return A*-GRAPH-SEARCH(*problem*)

Figure 7.20 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.

function ASK(KB, $ok^t_{x,y}$)
  inputs: the probability of the reachable squares to contain or not contain a pit
  P[x,y] = p($P_{1,1}$, ..., $P_{n,n}$) # product rule
  return Max(P)