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*
  $safe \leftarrow \{[x,y] : \text{ASK}(KB, OK^t_{x,y}) = true\}$
  if ASK($KB, Glitter^t$) = *true* **then**
    plan ← [*Grab*] + PLAN-ROUTE(*current*, {[1,1]}, *safe*) + [*Climb*]
  if *plan* is empty **then**
    $unvisited \leftarrow \{[x,y] : \text{ASK}(KB, L^{t'}_{x,y}) = false \text{ for all } t' \leq t\}$
    plan ← PLAN-ROUTE(*current*, *unvisited* ∩ *safe*, *safe*)
  if *plan* is empty and ASK($KB, HaveArrow^t$) = *true* **then**
    $possible\_wumpus \leftarrow \{[x,y] : \text{ASK}(KB, \neg W_{x,y}) = false\}$
    plan ← PLAN-SHOT(*current*, *possible_wumpus*, *safe*)
  if *plan* is empty **then** // *no choice but to take a risk*
    $not\_unsafe \leftarrow \{[x,y] : \text{ASK}(KB, \neg OK^t_{x,y}) = false\}$
    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*))
  $t \leftarrow t + 1$
  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 SEARCH(*problem*) // *Any search algorithm from Chapter 3*