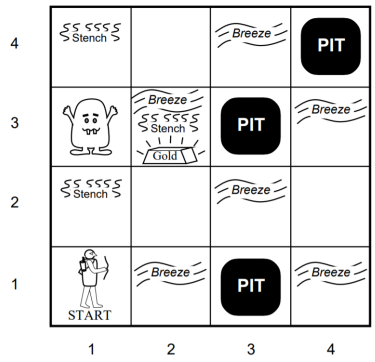


Figure 1: A Wumpus Wonderland

1 Discussion-Based Warm Ups

(a) Given the following, can you prove that the unicorn is mythical? How about magical? Horned?

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Do not attempt to formalize your solution here. Rather, turn to the people around you and reason through this question. How many possible worlds would we have to enumerate to give a formal answer?

As human reasoners, we can see from the first two statements, that if it is mythical, then it is immortal; otherwise it is a mammal. So it must be either immortal or a mammal, and thus horned. That means it is also magical. However, we can't deduce anything about whether it is mythical.

To provide a formal answer, we would have to enumerate the possible worlds ($2^5 = 32$ of them with 5 proposition symbols), mark those in which all the assertions are true, and see which conclusions hold in all of those.

(b) Determine which of the following are correct, and explain your reasoning:

(i) $(A \wedge B) \models (A \iff B)$

True, because the left-hand side has exactly one model that is one of the two models of the right-hand side.

(ii) $A \iff B \models A \vee B$

False, because one of the models of $A \iff B$ has both A and B false, which does not satisfy $A \vee B$

(iii) $A \iff B \models \neg A \vee B$

True, because the right hand side is $A \implies B$, one of the conjuncts in the definition of $A \iff B$

(iv) $(A \wedge B) \implies C \models (A \implies C) \vee B \implies C)$

True because the RHS is false only when both disjuncts are false, i.e., when A and B are true and C is false, in which case the LHS is also false. This may seem counterintuitive, and would not hold if is interpreted as "causes."

(v) $(A \vee B) \wedge \neg(A \implies B)$ is satisfiable.

True, model has A and $\neg B$.

(c) What is the difference between satisfiability and entailment (think about the purpose and requirements of each)?

From p. 250 in AIMA 3rd ed.: "A sentence is satisfiable if it is true in, or satisfied by, some model...

$\alpha \models \beta$ if and only if the sentence $(\alpha \wedge \neg\beta)$ is unsatisfiable. Proving β from α by checking the unsatisfiability of $(\alpha \wedge \neg\beta)$ corresponds exactly to the standard mathematical proof technique of reductio ad absurdum (literally, "reduction to an absurd thing").

Consider, for example, the agent's location, initially $[1, 1]$, and suppose the agent's unambitious goal is to be in $[2, 1]$ at time 1. The initial knowledge base contains $L^0_{1,1}$ and the goal is $L^1_{2,1}$. Now, SATPLAN will find the plan $[Forward^0]$; so far, so good. Unfortunately, SATPLAN also finds the plan $[Shoot^0]$. How could this be? To find out, we inspect the model that SATPLAN constructs: it includes the assignment $L^{2,1}_0$, that is, the agent can be in $[2, 1]$ at time 1 by being there at time 0 and shooting. One might ask, "Didn't we say the agent is in $[1, 1]$ at time 0?" Yes, we did, but we didn't tell the agent that it can't be in two places at once! For entailment, $L^{2,1}_0$ is unknown and cannot, therefore, be used in a proof; for satisfiability. Agents Based on Propositional Logic on the other hand, $L^{2,1}_0$ is unknown and can, therefore, be set to whatever value helps to make the goal true. For this reason, SATPLAN is a good debugging tool for knowledge bases because it reveals places where knowledge is missing. In this particular case, we can fix the knowledge base by asserting that, at each time step, the agent is in exactly one location. Satisfability is not necessarily guaranteed (exists in one of the models), but entailment is guaranteed.

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**
    $plan \leftarrow [Grab]$ + PLAN-ROUTE(*current*, {[1,1]}, *safe*) + [*Climb*]
  if *plan* is empty **then**
    $unvisited \leftarrow \{[x, y] : \text{ASK}(KB, L^{t'}_{x,y}) = false \text{ for all } t' \leq t\}$
    $plan \leftarrow$ PLAN-ROUTE(*current*, *unvisited* $\cap$ *safe*, *safe*)
  if *plan* is empty and ASK(KB, $HaveArrow^t$) $= true$ **then**
    $possible\_wumpus \leftarrow \{[x, y] : \text{ASK}(KB, \neg\, W_{x,y}) = false\}$
    $plan \leftarrow$ PLAN-SHOT(*current*, *possible_wumpus*, *safe*)
  if *plan* is empty **then** // no choice but to take a risk
    $not\_unsafe \leftarrow \{[x, y] : \text{ASK}(KB, \neg\, OK^t_{x,y}) = false\}$
    $plan \leftarrow$ PLAN-ROUTE(*current*, *unvisited* $\cap$ *not_unsafe*, *safe*)
  if *plan* is empty **then**
    $plan \leftarrow$ PLAN-ROUTE(*current*, {[1, 1]}, *safe*) + [*Climb*]
  $action \leftarrow$ POP(*plan*)
  TELL(KB, MAKE-ACTION-SENTENCE(*action*, t))
  $t \leftarrow t + 1$
  return *action*

A B C D E

---

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 $\leftarrow$ ROUTE-PROBLEM(*current*, *goals*,*allowed*)
  return A*-GRAPH-SEARCH(*problem*)

Figure 2: Hybrid-Wumpus-Agent from AIMIA 3rd ed. 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.

2 Wandering in Wumpus World

We bring together what we have learned in lecture as well as the ideas of search so far in order to construct wumpus world agents that use propositional logic. The first step is to enable the agent to deduce, to the extent possible, the state of the world given its percept history. This requires writing down a complete logical model of the effects of actions. We also show how the agent can keep track of the world efficiently without going back into the percept history for each inference. Finally, we show how the agent can use logical inference to construct plans that are guaranteed to achieve its goals.

Try it out: http://thiagodnf.github.io/wumpus-world-simulator/

Throughout this question, we will present several screenshots from the Wumpus World simulator linked previously. In each of these, assume that you *do* have an arrow on hand (as an extra exercise, consider how the answers might be different if you did not have an arrow). Also, note that the location of the explorer can be ignored. We just tried to place him somewhere where he wouldn't be blocking the text!

(a) Consider the following Wumpus World state:

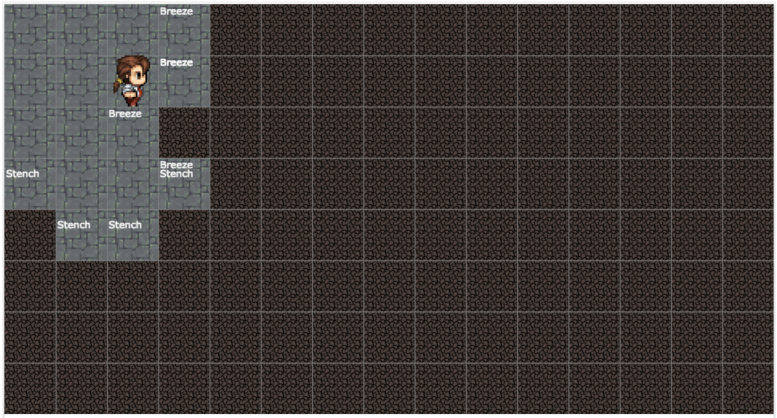
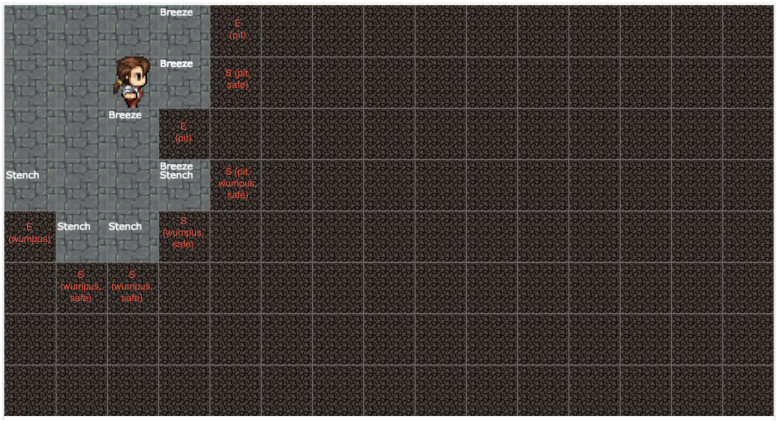


Figure 3: Entailment versus Satisfiability?

Based on our previous discussion around entailment and satisfiablity, identify locations where our knowledge base entails that there must be a Wumpus, Pit, or safe path. Additionally, identify locations where Wumpuses, Pit, and safe paths are not entailed but could be satisfied.

We've marked places where a pit, wampus, safe can be satisfied with S(type). Entailments are indicated with E(type).



(b) Now, refer to Figure 2 from Page 2, and take a moment to familiarize yourself with the pseudocode to understand how we might decide to act in Wumpus World. You'll notice that we have labeled the key decision-making portions of this code, and that different decisions need to be made given the state of our knowledge-base.

Match each of the following states to one of the labeled code chunks in the pseudocode, and explain your reasoning.

Figure 4: Which code chunk is applicable for each of these states?

1. C: There are two stenches within reasonable distance, therefore, based on satisfiability it is possible for a wumpus to exist in the unvisited square diagonal from our explorer. There are also no guaranteed safe spaces. However, lucky for us, we have an arrow that we can use in case of Wumpus!
2. B: We have found a square free from breeze or stench that is adjacent to an unvisited square. We know, based on our knowledge base, that this unvisited square must therefore also be safe and can visit it.
3. A: We have found gold, and we can grab it, and then plan the shortest, safest route out.
4. D: There is no guaranteed safe square, therefore, we must take a risk.

3 Axioms & Arrows

Up until now we have assumed that the plans we create always make sure that an actions preconditions are satisfied. Let us now investigate what propositional successor-state axioms such as $HaveArrow^{t+1} \iff (HaveArrow^t \wedge \neg Shoot^t)$ have to say about actions whose preconditions are not satisfied.

(a) First, let us consider what successor-state axioms are. How do they differ from action axioms, and why might we choose to use them?

(b) Show that the axioms predict that nothing will happen when an action is executed in a state where its preconditions are not satisfied.

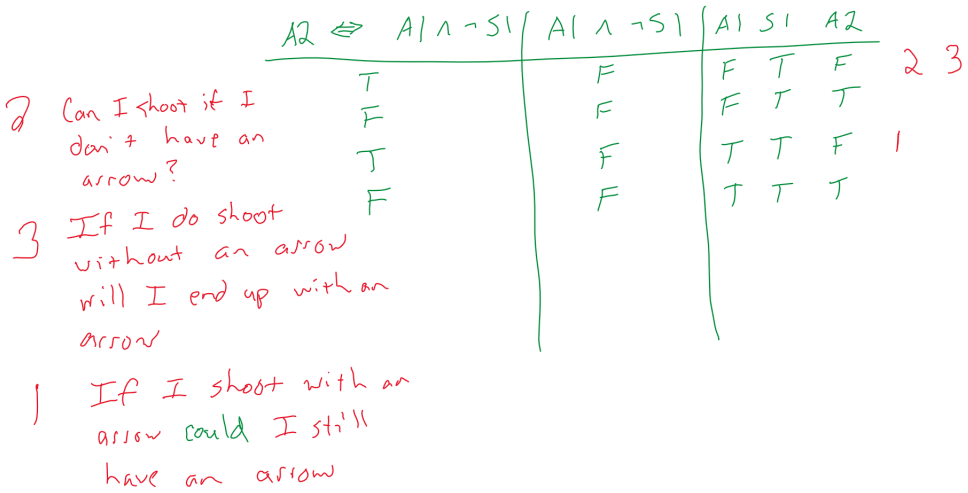
(c) Consider a plan p that contains the actions required to achieve a goal but also includes illegal actions. Is it the case that

$$\text{initial state} \wedge \text{successor-state axioms} \wedge p \models goal?$$

We recommend that you write a truth table and ask yourself the following questions when looking at the truth table

- Can I shoot if I don't have an arrow?
- If I do shoot without an arrow will I end up with an arrow?
- If I shoot with an arrow **could** I still have an arrow?

Solutions:



(a) Action axioms give us the conditions needed at the current state for an action to be carried out at the current state. Successor state axioms are axioms outlining what preconditions in the current state need to be true in order to ensure that the state at the next timestep will be as specified. By definition, it is an axiom that sets the truth value of F_{t+1} (where F is some fluent, or changeable variable in an environment) in one of two ways:

- The action at time t causes F to be true at $t+1$

- F was already true at time t and the action at time t does not cause it to be false.

It has the following schema: $F_{t+1} \iff ActionCausesF_t \vee (F_t \wedge_t)$.

We use successor state axioms to ensure that each state we compute is the result of legal action. Also, writing only in terms of action axioms can become overwhelming, if there are a lot of time-dependent variables to keep track of. Successor state axioms are what help us do this.

(b) We can illustrate the basic idea using the axiom given. Suppose that $Shoot^t$ is true but $HaveArrow^t$ is false. Then the RHS of the axiom is false, so $HaveArrow^{t+1}$ is false, as we would hope. More generally, if an action precondition is violated, then both $ActionCausesF^t$ and $ActionCausesNotF^t$ are false, so the generic successor- state axiom reduces to $F^{t+1} \iff False \vee (F^t \wedge True)$ which is the same as saying $F^{t+1} \iff F^t$, i.e., nothing happens.

(c) Yes, the plan plus the axioms will entail goal satisfaction; the axioms will copy every fluent across an illegal action and the rest of the plan will still work. Note that goal entailment is trivially achieved if we add precondition axioms, because then the plan is logically inconsistent with the axioms and every sentence is entailed by a contradiction. Precondition axioms are a way to prevent illegal actions in satisfiability-based planning methods.