

1. For the environment to the right, the agent tried 6 episodes from the start state A to one of the terminal states (C, D, and E), which are listed below:

Episode #1: state = A, action = R, new state = C, reward = +10
 Episode #2: state = A, action = L, new state = B, reward = 0
 state = B, action = R, new state = E, reward = -1000
 Episode #3: state = A, action = L, new state = B, reward = 0
 state = B, action = L, new state = D, reward = +200
 Episode #4: state = A, action = L, new state = B, reward = 0
 state = B, action = R, new state = E, reward = -100
 Episode #5: state = A, action = R, new state = C, reward = +25
 Episode #6: state = A, action = L, new state = B, reward = 0
 state = B, action = L, new state = D, reward = +400

Your task is to build the Q-table from these results. The Q-table has two states and two actions per state. Use learning rate = 0.5 and discount factor = 1. All entries of the Q-table are zero initially.

Episode #1

state = A, action = R, new state = C

$$Q(A,R) \leftarrow Q(A,R) + 0.5 [r - Q(A,R)] = 0 + 0.5 [10 - 0] = 5$$

Episode #2

state = A, action = L, new state = B

$$Q(A,L) \leftarrow Q(A,L) + 0.5 [r + \max(Q(B,L), Q(B,R)) - Q(A,L)] = 0 + 0.5 [0 + \max(0, 0) - 0] = 0$$

state = B, action = R, new state = E

$$Q(B,R) \leftarrow Q(B,R) + 0.5 [r - Q(B,R)] = 0 + 0.5 [-1000 - 0] = -500$$

Episode #3

state = A, action = L, new state = B

$$Q(A,L) \leftarrow Q(A,L) + 0.5 [r + \max(Q(B,L), Q(B,R)) - Q(A,L)] = 0 + 0.5 [0 + \max(0, -500) - 0] = 0$$

state = B, action = L, new state = D

$$Q(B,L) \leftarrow Q(B,L) + 0.5 [r - Q(B,L)] = 0 + 0.5 [200 - 0] = 100$$

Episode #4

state = A, action = L, new state = B

$$Q(A,L) \leftarrow Q(A,L) + 0.5 [r + \max(Q(B,L), Q(B,R)) - Q(A,L)] = 0 + 0.5 [0 + \max(100, -500) - 0] = 50$$

state = B, action = R, new state = E

$$Q(B,R) \leftarrow Q(B,R) + 0.5 [r - Q(B,R)] = -500 + 0.5 [-100 - (-500)] = -300$$

Episode #5

state = A, action = R, new state = C

$$Q(A,R) \leftarrow Q(A,R) + 0.5 [r - Q(A,R)] = 5 + 0.5 [25 - 5] = 15$$

Episode #6

state = A, action = L, new state = B

$$Q(A,L) \leftarrow Q(A,L) + 0.5 [r + \max(Q(B,L), Q(B,R)) - Q(A,L)] = 50 + 0.5 [0 + \max(100, -300) - 50] = 75$$

state = B, action = L, new state = D

$$Q(B,L) \leftarrow Q(B,L) + 0.5 [r - Q(B,L)] = 100 + 0.5 [400 - 100] = 250$$

2. For each sentence below, determine whether it is valid, unsatisfiable, or neither. Briefly explain your answers. You can also use the equivalence rules or truth tables to prove your answers.

(a) Smoke \Rightarrow Smoke

valid: This is equivalent to $\neg \text{Smoke} \wedge \text{Smoke}$, which is always true

(b) Smoke \Rightarrow Fire

neither valid nor unsatisfiable

(c) (Smoke \Rightarrow Fire) \Rightarrow (\neg Smoke \Rightarrow \neg Fire)

neither valid nor unsatisfiable (true if Smoke is true and Fire is true; false if Smoke is false and Fire is true)

(d) $\text{Smoke} \vee \text{Fire} \vee \neg \text{Fire}$
valid

(e) $((\text{Smoke} \wedge \text{Heat}) \Rightarrow \text{Fire}) \Leftrightarrow ((\text{Smoke} \Rightarrow \text{Fire}) \vee (\text{Heat} \Rightarrow \text{Fire}))$
valid: Both sides of the biconditional are equivalent to $\text{Smoke} \vee \text{Heat} \vee \neg \text{Fire}$

(f) $\text{Big} \vee \text{Dumb} \vee (\text{Big} \Rightarrow \text{Dumb})$
valid: This is equivalent to $\text{Big} \wedge \text{Dumb} \wedge \neg \text{Big} \wedge \text{Dumb}$

(g) $(\text{Big} \wedge \text{Dumb}) \vee \neg \text{Dumb}$
neither valid nor unsatisfiable (true if Dumb is false; false if Dumb is true and Big is false)

3. Consider the snapshot (shown to the right) of the **Minesweeper** game. Let A, B, C and D be four propositional symbols representing the existence of mines in their locations.

1	A	B	C
2	2	2	1
D	1		

(a) Try to write down propositional-logic sentences according to the game rule.

(b) Convert the sentences to CNF.

(c) Try to determine their truth values by repeatedly applying the resolution inference rule.

(a) $A, D, A \wedge D, A \wedge B \Leftrightarrow \neg D, A \wedge D \Leftrightarrow \neg B, B \wedge D \Leftrightarrow \neg A, A \wedge B \Leftrightarrow \neg C, A \wedge C \Leftrightarrow \neg B, B \wedge C \Leftrightarrow \neg A, C \Leftrightarrow \neg B, B \Leftrightarrow \neg C$

(b) $A, D, \neg A \vee \neg B \vee \neg D, \neg A \vee \neg B \vee \neg C, A \vee B, A \vee C, A \vee D, B \vee C, B \vee D, \neg B \vee \neg C$

(c) A, C, D are true, and B is false

4. Here you will practice the inference of the Wumpus world using first-order logic. Some notes about the model and symbols are:

Treat Wumpus as a constant object. (There is only one Wumpus.)

Use variables to represent the squares.

Predicates that are unary relations of the squares: *Breezy*, *Smelly*, *Pit*.

Predicates that are binary relations of the squares: *Adjacent*.

Use the function symbol *Position* for the square occupied by an object. (Of course, the only object that can occupy a square is Wumpus.)

(a) For the following rules of the Wumpus world, rewrite them into FOL sentences. You can use *Breezy*, *Smelly*, *Pit*, *Safe*, *At*, and *Adjacent* as the predicates. (*Adjacent* is a relation between two squares. *At* is a relation between an object and a square. Treat Wumpus as a constant object.)

A square with no Wumpus and no pit is safe. (This is the definition of the predicate *Safe*.)

A square is breezy if and only if it is adjacent to a pit.

A square is smelly if and only if it is adjacent to the Wumpus.

The Wumpus can only be in a square with no pit.

(b) Given the following information, prove that the Wumpus is at square (2,2) using resolution in FOL. You need to convert all the relevant sentences to CNF first.

The square (1,1) is safe.

The square (2,1) is smelly.

The square (1,2) is smelly.

$$(a). \forall x \text{ Safe}(x) \leftrightarrow \neg \text{Pit}(x) \wedge \neg \text{At}(\text{Wumpus}, x)$$

$$2. \forall x, y \text{ Pit}(y) \wedge \text{Adjacent}(x, y) \leftrightarrow \text{Breezy}(x)$$

$$3. \forall x, y \text{ At}(\text{Wumpus}, y) \wedge \text{Adjacent}(x, y) \leftrightarrow \text{Smelly}(x)$$

$$4. \forall x \text{ At}(\text{Wumpus}, x) \rightarrow \neg \text{Pit}(x)$$

(b) Safe(s11)

Smelly(s21)

Smelly(s12)

$$1. \text{ to } \begin{cases} \text{Safe}(x) \rightarrow \neg \text{Pit}(x) \wedge \neg \text{At}(\text{Wumpus}, x) & - \textcircled{1} \\ \neg \text{Pit}(x) \wedge \neg \text{At}(\text{Wumpus}, x) \rightarrow \text{Safe}(x) & - \textcircled{2} \end{cases}$$

$$\textcircled{1} \rightarrow \neg \text{Safe}(x) \vee \neg \text{Pit}(x)$$

$$\neg \text{Safe}(x) \vee \neg \text{At}(\text{Wumpus}, x)$$

$$\textcircled{2} \text{ Pit}(x) \vee \text{At}(\text{Wumpus}, x) \vee \text{Safe}(x)$$

$$2. \neg \text{Pit}(y) \vee \neg \text{Adjacent}(x, y) \vee \text{Breezy}(x)$$

$$3. \neg \text{At}(\text{Wumpus}, y) \vee \neg \text{Adjacent}(x, y) \vee \text{Smelly}(x)$$

$$4. \neg \text{At}(\text{Wumpus}, x) \vee \neg \text{Pit}(x)$$

$$5. \neg \text{Breezy}(x) \vee \text{Pit}(G(x)), \neg \text{Breezy}(x) \vee \text{Adjacent}(x, G(x))$$

$$6. \neg \text{Smelly}(x) \vee \text{At}(\text{Wumpus}, G(x)), \neg \text{Smelly}(x) \vee \text{Adjacent}(x, G(x))$$

$$\text{Smelly}(s21) + 6 \rightarrow \text{Adjacent}(s21, G(s1)), \text{At}(\text{Wumpus}, G(s1))$$

$$\text{Smelly}(s12) + 6 \rightarrow \text{Adjacent}(s12, G(s2)), \text{At}(\text{Wumpus}, G(s2))$$

$$\text{Adjacent}(s21, G(s2)) \rightarrow (G(s21) = s11) \vee (G(s21) = s31) \vee (G(s21) = s22)$$

$$\text{Safe}(s11) \text{ and } 1. \textcircled{1} \rightarrow \neg \text{At}(\text{Wumpus}, s11) \xrightarrow{\text{resolve}}$$

$$\text{At}(\text{Wumpus}, s31) \vee \text{At}(\text{Wumpus}, s22) \quad \left. \begin{array}{l} \text{Wumpus 只能在一格} \\ \text{Pit} \vee \text{At}(\text{Wumpus}, s22) \end{array} \right\}$$

$$\text{At}(\text{Wumpus}, s13) \vee \text{At}(\text{Wumpus}, s22) \quad \left. \begin{array}{l} \text{Pit} \vee \text{At}(\text{Wumpus}, s22) \end{array} \right\}$$