Artificial Intelligence

Fiona Yan Liu

Department of Computing

The Hong Kong Polytechnic University

Learning Outcomes of Lecture 03

- Understand the related concepts of constraint satisfaction problem
- Understand the motivation of studying constraint satisfaction problem
- Master the strategies to speed-up the solving of CSP
- Understand the related concepts of games
- Understand the motivation of studying games
- Master a typical problem-solving method for games
- Master the strategies to speed-up the solution search

Agenda

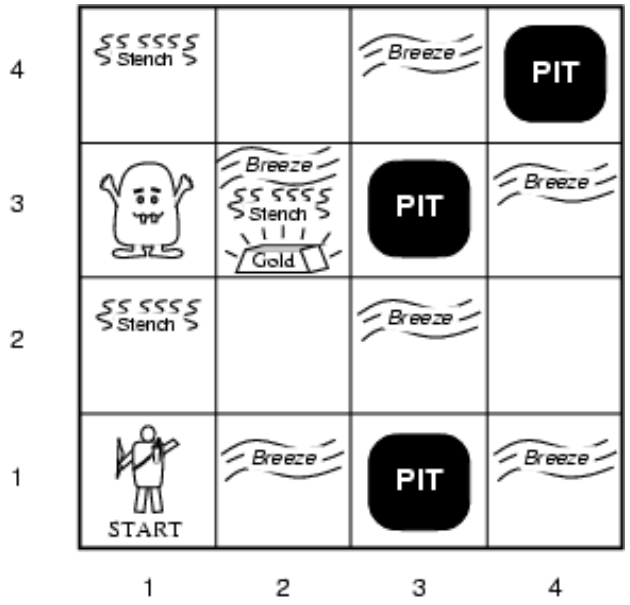
- Case study of Wumpus world
- Knowledge based agent
- Query
- Decision tree

Agenda

- Case study of Wumpus world
- Knowledge based agent
- Query
- Decision tree

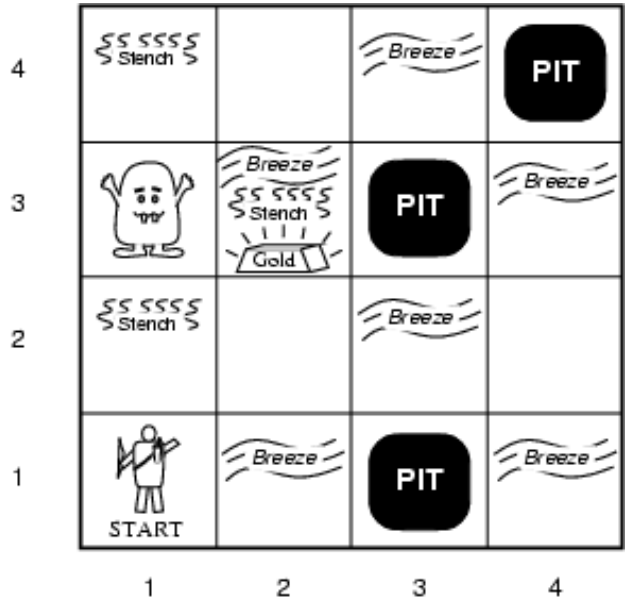
Wumpus World

- The player intends to have the gold
  - The player don't know the location of the gold
  - The player can move in the Wumpus world, one square/movement
  - The player can see the gold if they are in the same square
- Wumpus will eat the player if they met in the same square
  - Wumpus will not move
  - The player has one and only one weapon to kill the Wumpus
  - The number of Wumpus is unknown
- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow



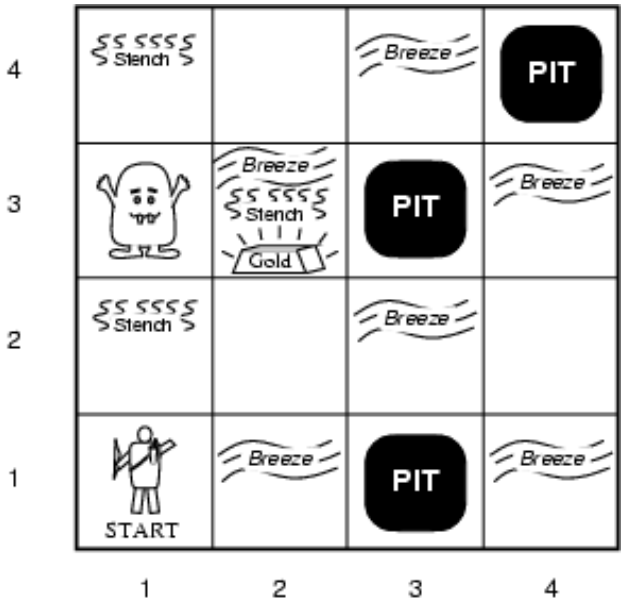
Environment and rational agent

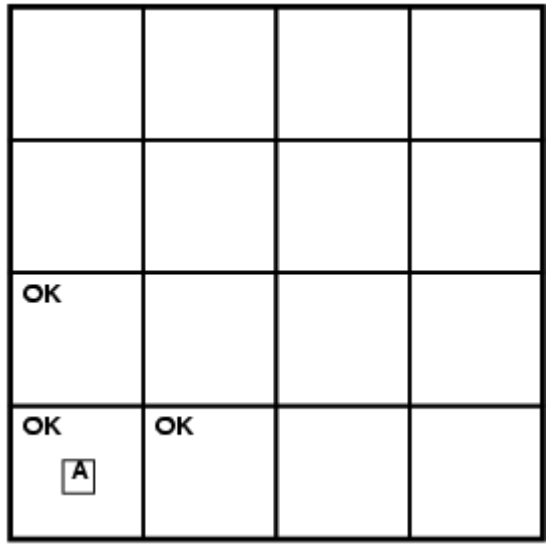
- Environment
  - Squares adjacent to Wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
- Rational agent
  - Shooting kills Wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square
- Sensors
  - Stench, Breeze, Glitter, Bump, Scream
- Actuators
  - Left turn, Right turn, Forward, Grab, Release, Shoot



Initial State

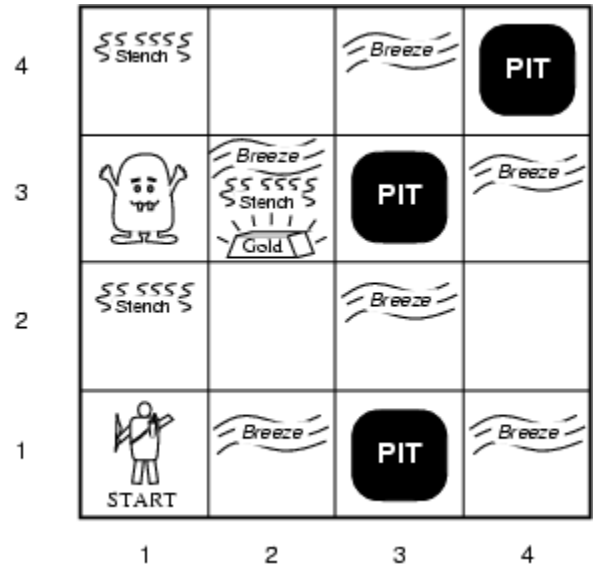
- State: [1,1]
- Perception: [None, none, none, none, none]
  - Sensors: [Stench, Breeze, Glitter, Bump, Scream]
- Action: Forward
  - Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot
- Result: [1,2]

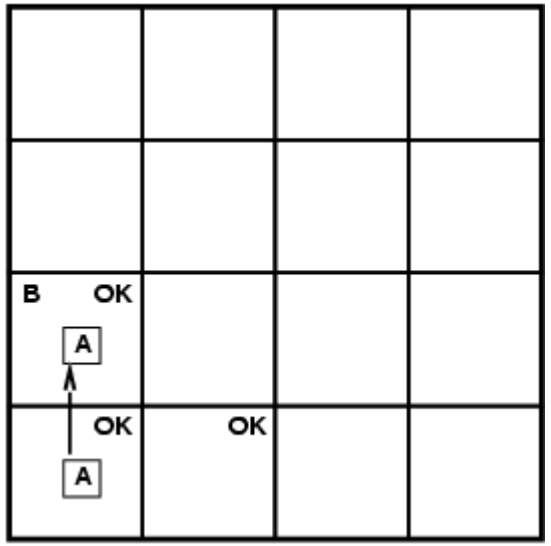




Exploring a Wumpus World

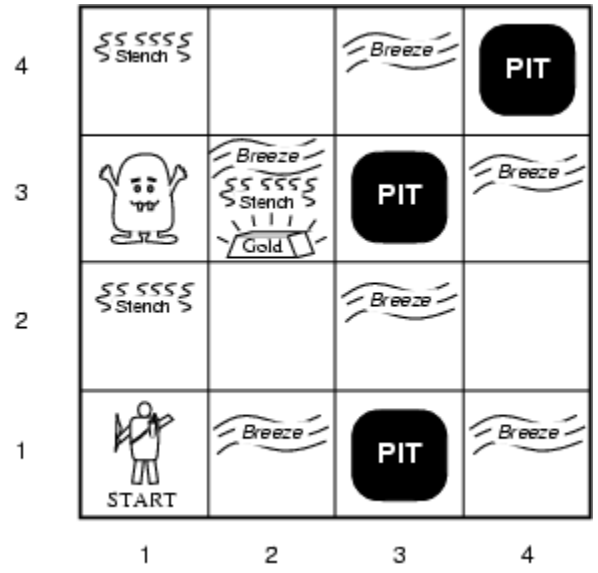
- Perceive: [Stench, Breeze, Glitter, Bump, Scream]
- [1,1]: [None, none, none, none, none]
- [1,2]: [Stench, none, none, none, none]

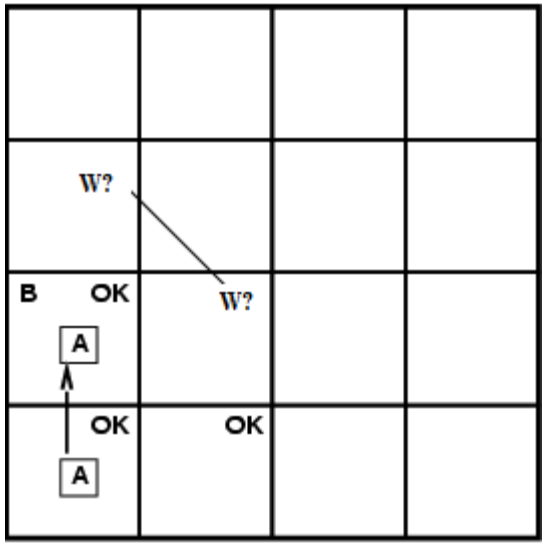




Exploring a Wumpus World

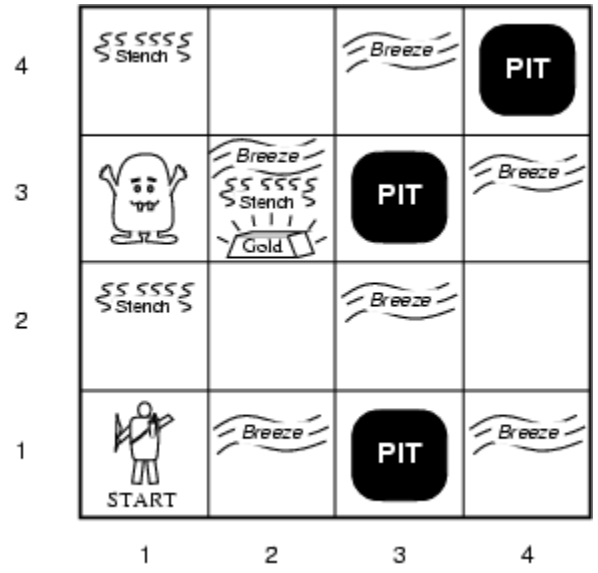
- Perceive: [Stench, Breeze, Glitter, Bump, Scream]
- [1,1]: [None, none, none, none, none]
- [1,2]: [Stench, none, none, none, none]

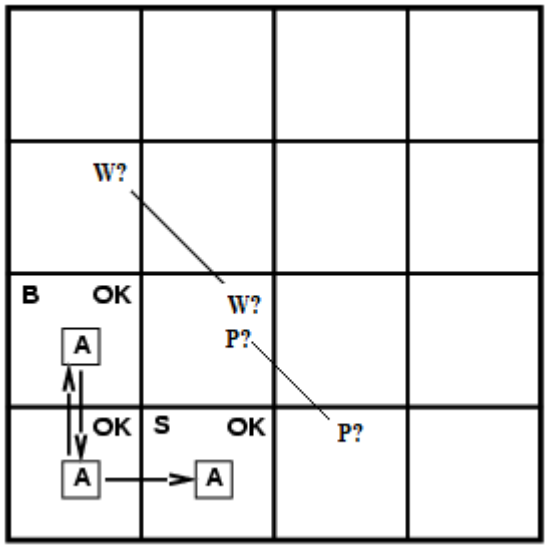




Exploring a Wumpus World

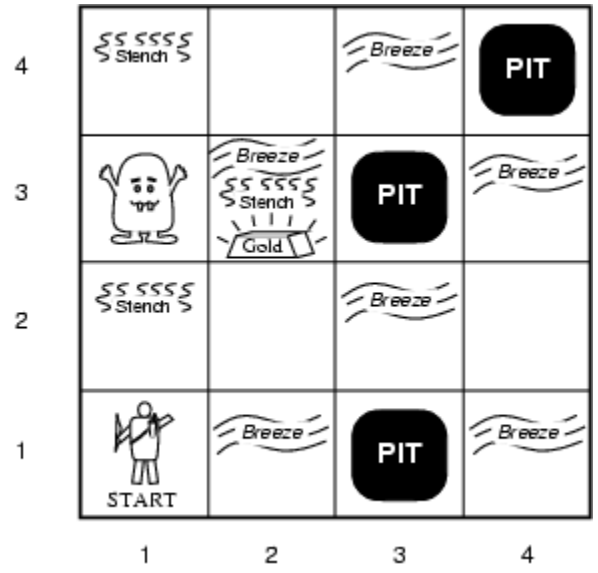
- Perceive: [Stench, Breeze, Glitter, Bump, Scream]
- [1,1]: [None, none, none, none, none]
- [1,2]: [Stench, none, none, none, none]
- [2,1]: [None, breeze, none, none, none]

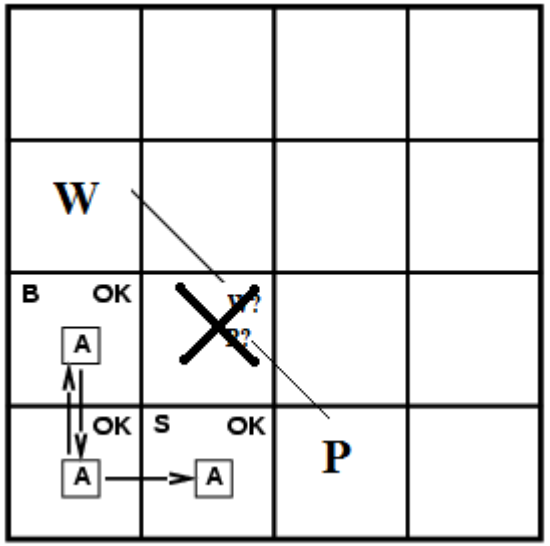




Exploring a Wumpus World

- Perceive: [Stench, Breeze, Glitter, Bump, Scream]
- [1,1]: [None, none, none, none, none]
- [1,2]: [Stench, none, none, none, none]
- [2,1]: [None, breeze, none, none, none]





Agenda

- Case study of Wumpus world
- Knowledge based agent
- Query
- Decision tree

Knowledge-based Agent

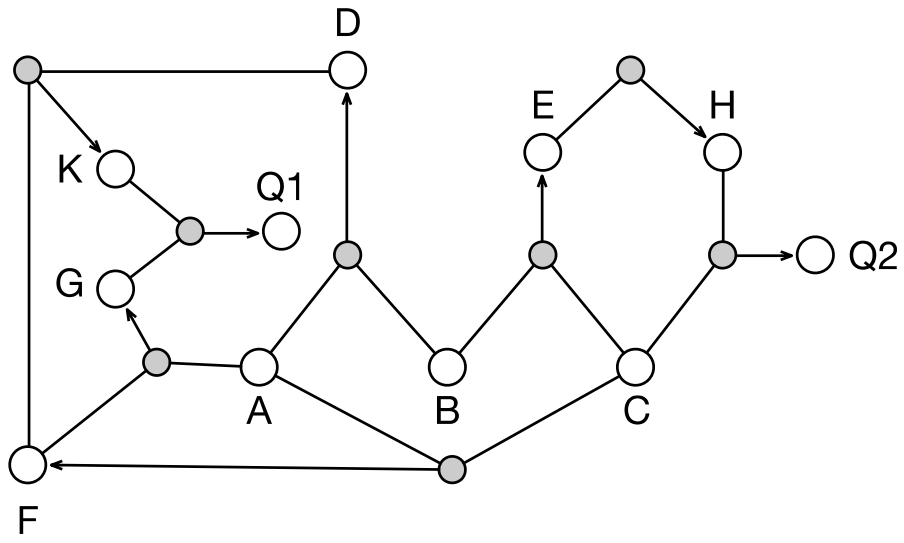
- Knowledge base
  - A set of sentences
  - Each sentence is expressed in a language called a knowledge representation language
- A presentation language is defined by its syntax
  - Specify the structure of sentences
  - And its semantics, which defines the truth of each sentence
- Knowledge-based Agent
  - Tell the knowledge base what it perceives
  - Ask the knowledge base what action it should perform
  - Tell the knowledge base which action was chosen
  - The agent executes the action

Agenda

- Case study of Wumpus world
- Knowledge based agent
- Query
- Decision tree

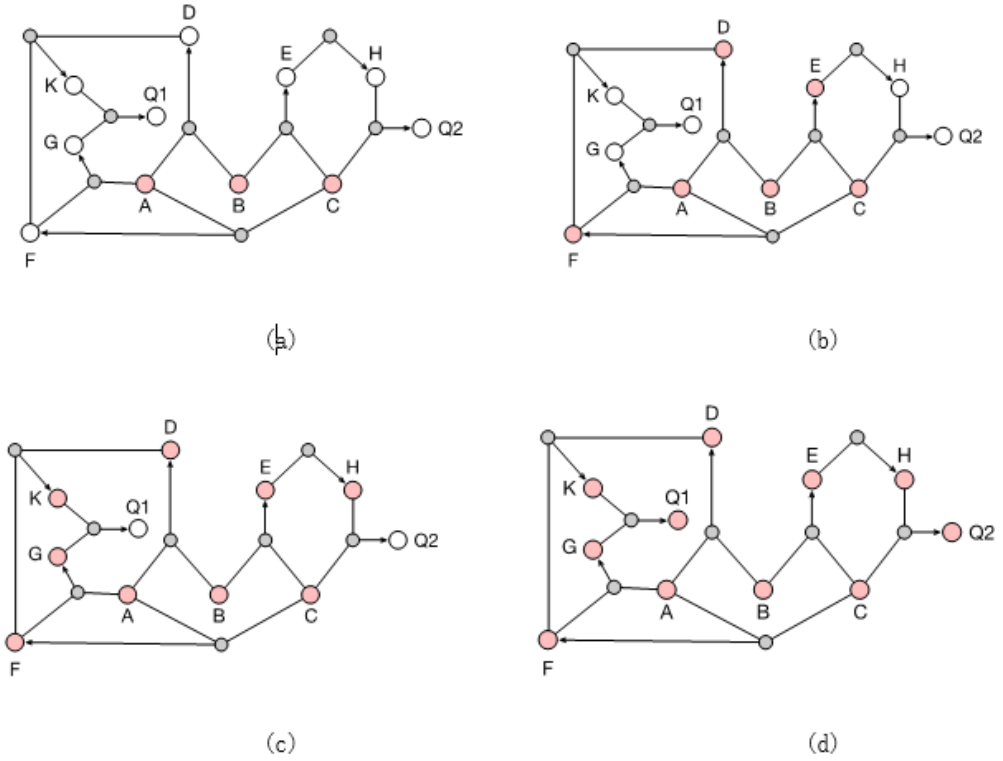
Query

- Knowledge-base
  1. A
  2. B
  3. C
  4. $A \wedge B \Rightarrow D$
  5. $B \wedge C \Rightarrow E$
  6. $A \wedge C \Rightarrow F$
  7. $A \wedge F \Rightarrow G$
  8. $D \wedge F \Rightarrow K$
  9. $G \wedge K \Rightarrow Q1$
  10. $E \Rightarrow H$
  11. $H \wedge C \Rightarrow Q2$
- Query
  - Q2



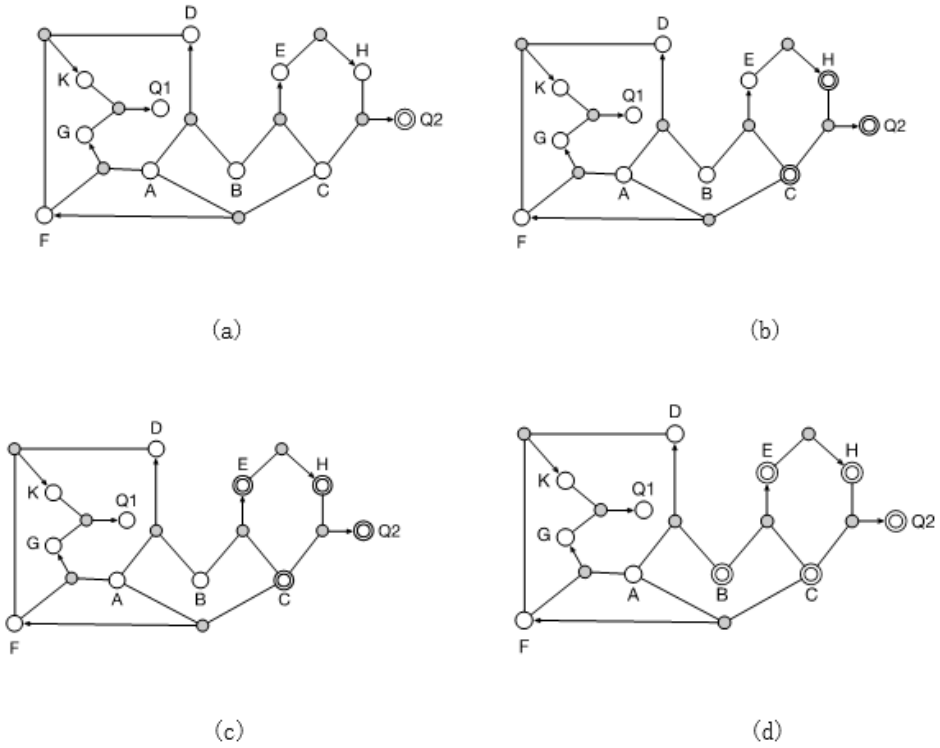
Forward Chaining

- Idea
  - fire any rule whose premises are satisfied in the knowledge base
  - add its conclusion to the knowledge base, until query is found
- Knowledge-base
  1. A
  2. B
  3. C
  4. $A \wedge B \Rightarrow D$
  5. $B \wedge C \Rightarrow E$
  6. $A \wedge C \Rightarrow F$
  7. $A \wedge F \Rightarrow G$
  8. $D \wedge F \Rightarrow K$
  9. $G \wedge K \Rightarrow Q1$
  10. $E \Rightarrow H$
  11. $H \wedge C \Rightarrow Q2$



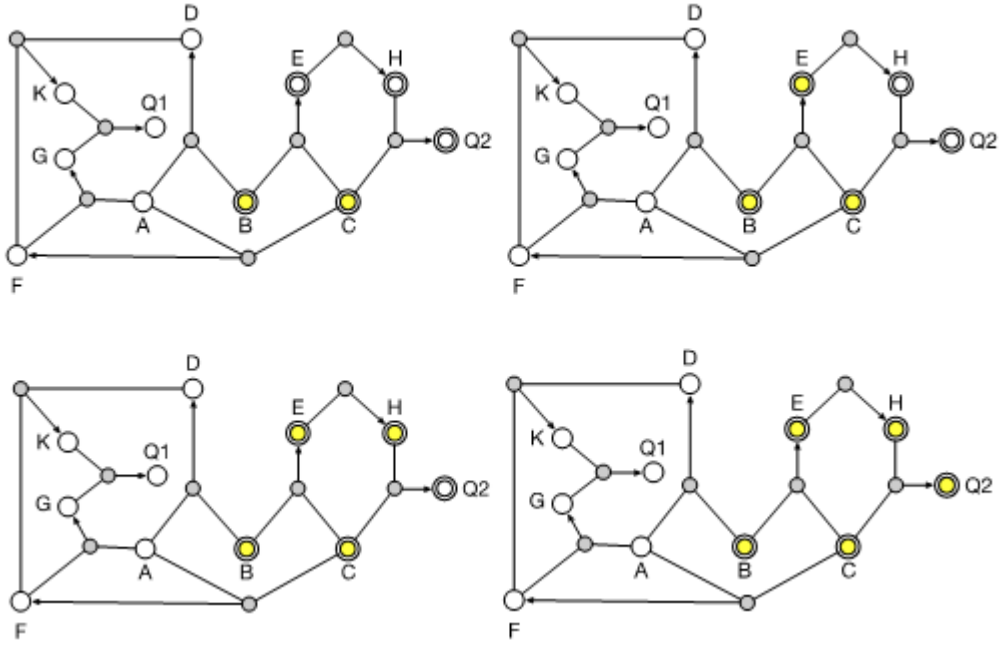
Backward Chaining

- Idea
  - work backwards from the query q:
  - Check the related sentences for the query
- Knowledge-base
  1. A
  2. B
  3. C
  4. $A \wedge B \Rightarrow D$
  5. $B \wedge C \Rightarrow E$
  6. $A \wedge C \Rightarrow F$
  7. $A \wedge F \Rightarrow G$
  8. $D \wedge F \Rightarrow K$
  9. $G \wedge K \Rightarrow Q1$
  10. $E \Rightarrow H$
  11. $H \wedge C \Rightarrow Q2$



Backward Chaining

- Idea
  - work backwards from the query q:
  - Validate the results
- Knowledge-base
  1. A
  2. B
  3. C
  4. $A \wedge B \Rightarrow D$
  5. $B \wedge C \Rightarrow E$
  6. $A \wedge C \Rightarrow F$
  7. $A \wedge F \Rightarrow G$
  8. $D \wedge F \Rightarrow K$
  9. $G \wedge K \Rightarrow Q1$
  10. $E \Rightarrow H$
  11. $H \wedge C \Rightarrow Q2$



Agenda

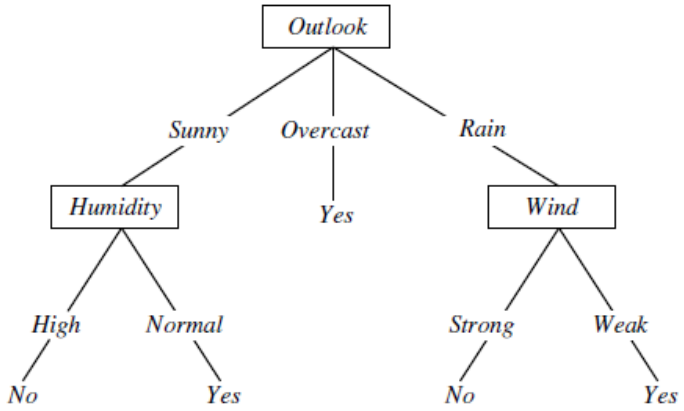
- Case study of Wumpus world
- Knowledge based agent
- Query
- Decision tree

Decision Tree

- Given a training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Each internodes test an attribute
- Each branch corresponds a attribute value
- Each leave node assigns a classification



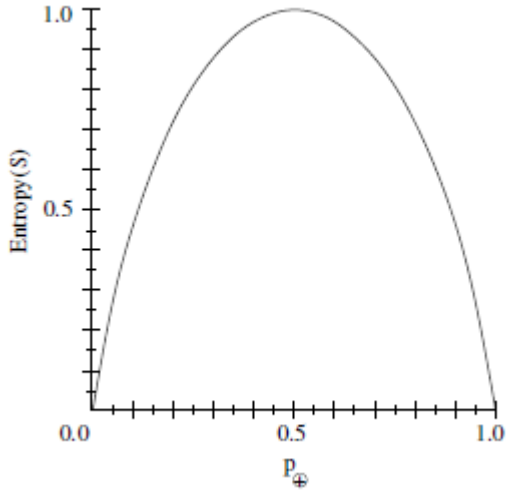
Choosing an Attribute

- *Entropy (s)*
  - Expected number of bits needed to encode class + or – of randomly drawn member of *S*

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

- *Gain (S,A)*
  - Expected reduction in entropy due to sorting on *A*

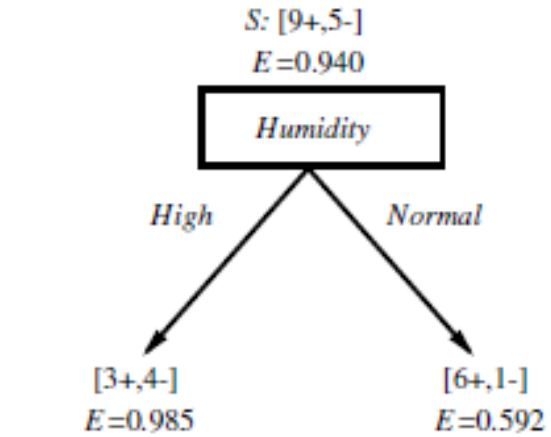
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Gain Calculation

$$E(S) = -p_+ \log_2 p_+ \; - p_- \log_2 p_- = -\frac{9}{14}\log_2 \frac{9}{14} - \frac{5}{14}\log_2 \frac{5}{14} = 0.94$$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



$$Gain(S, Humidity) = E(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} E(S_v) = 0.94 - \frac{7}{14}\left(-\frac{3}{7}\log_2 \frac{3}{7} - \frac{4}{7}\log_2 \frac{4}{7}\right)$$
$$- \frac{7}{14}\left(-\frac{6}{7}\log_2 \frac{6}{7} - \frac{1}{7}\log_2 \frac{1}{7}\right) = 0.94 - \frac{7}{14} * 0.985 - \frac{7}{14} * 0.592 = 0.151$$

Select the Best Feature

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S, Outlook) = 0.94 - \frac{5}{14}\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) - \frac{4}{14}\left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) - \frac{5}{14}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.94 - \frac{5}{14}*0.97 - \frac{4}{14}*0 - \frac{5}{14}*0.97 = 0.247$$

$$Gain(S, Wind) = 0.94 - \frac{8}{14}\left(-\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}\right) - \frac{6}{14}\left(-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}\right) = 0.94 - \frac{8}{14}*0.811 - \frac{6}{14}*1 = 0.048$$

Select the Best Feature

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S,T) = E(s) - \sum \frac{S_v}{S} E(S_v)$$

$$= -\left(-\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14}\right) - \frac{4}{14}\left(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right)$$

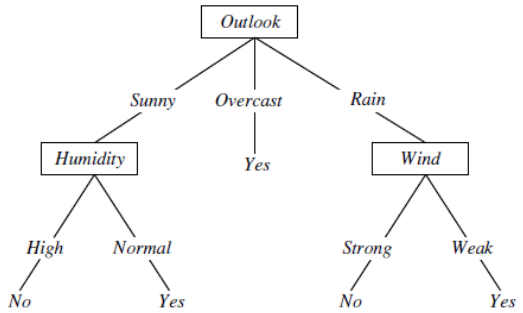
$$-\frac{6}{14}\left(-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}\right) - \frac{4}{14}\left(-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}\right)$$

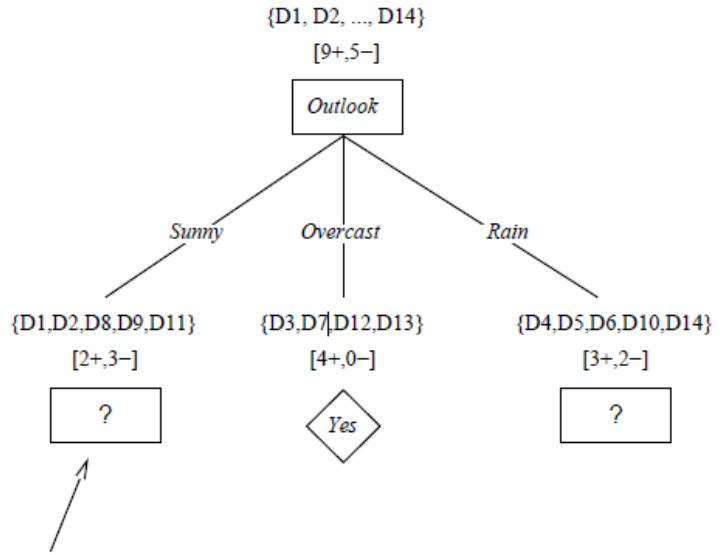
$$= 0.94 - \frac{4}{14} * 1 - \frac{6}{14} * 0.918 - \frac{4}{14} * 0.811 = 0.94 - 0.91 = 0.03$$

- *Gain(S, Outlook) = 0.247*
- *Gain(S, Temperature) = 0.03*
- *Gain(S, Humidity) = 0.151*
- *Gain(S, Wind) = 0.048*

Select Next Attribute of Sunny Branch

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No





Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny}, *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Select Next Attribute of Rain Branch

$$Gain(\text{Rain}, T) = E(s) - \sum \frac{S_v}{S} E(S_v)$$

$$= -\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) - \frac{3}{5}\left(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right) - \frac{2}{5}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right) = 0.97 - \frac{3}{5} * 0.918 - \frac{2}{5} * 1$$

$$= 0.02$$

$$Gain(Rain, H) = E(s) - \sum \frac{S_v}{S} E(S_v)$$

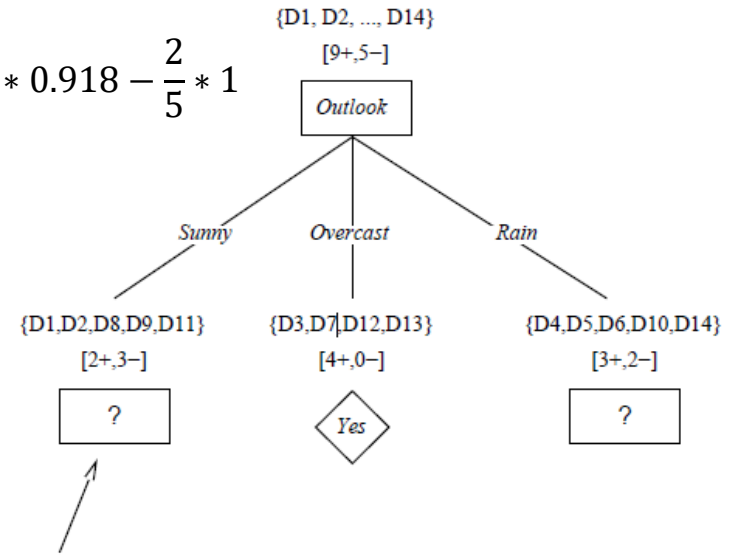
$$= -\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) - \frac{3}{5}\left(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right) - \frac{2}{5}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)$$

$$= 0.97 - \frac{3}{5} * 0.918 - \frac{2}{5} * 1 = 0.02$$

$$Gain(Rain, W) = E(s) - \sum \frac{S_v}{S} E(S_v)$$

$$= -\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) - \frac{3}{5}\left(-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right) - \frac{2}{5}\left(-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right)$$

$$= 0.97 - \frac{3}{5} * 0 - \frac{2}{5} * 0 = 0.97$$



S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny} , *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny} , *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny} , *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Decision Tree

- Given a training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Select the attribute with the highest information gain
- until each leave includes the data only from Yes or No

