Class Assignment Strategies

- **Team-Attack**: Team attack

- **Individualistic**: Search for possible

- **Political**: look at others and make decision based on who is winning, who is loosing, and conversation

- **Emotion and reasoning**: use aggression, honesty, and ruthlessness to vary agents behavior
Can you do this with just search?
Short-comings of just using search

- **Inflexibility:** You need to hard code every state
- **Performance:** exponential in the number of states
- **Observability:** No inference based on information seen
- **No inference and reasoning**
To address these issues we will introduce

- A knowledge base (KB): a list of facts that are known to the agent.
- A mechanism (Inference) to infer new facts from old facts
- Logic provides the natural language for this
Logic

Chapter 7 (some of the materials are from Welling at UCI)
Knowledge base:
- set of sentences in a formal language.

Declarative approach to building an agent:
- Tell the KB it what it needs to know.
- Ask the KB it what to do
Wumpus World PEAS description

- **Performance measure**
  - gold: +1000, death: -1000
  - -1 per step, -10 for using the arrow

- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - 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
Wumpus world characterization

- **Fully Observable** No – only local perception
- **Deterministic** Yes – outcomes exactly specified
- **Episodic** No – things we do have an impact.
- **Static** Yes – Wumpus and Pits do not move
- **Discrete** Yes
- **Single-agent?** Yes – Wumpus is essentially a natural feature
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a Wumpus world

Why is W in 3,1 and P in 1, 3?
We used logical reasoning to find the gold.

Logics are formal languages for representing information such that conclusions can be drawn.

Syntax defines the sentences in the language.

Semantics define the "meaning" of sentences;
  i.e., define truth of a sentence in a world.
Entailment means that one thing follows from another:

\[ \text{KB } \models \alpha \]

Knowledge base \( KB \) entails sentence \( \alpha \) if and only if \( \alpha \) is true in all worlds where \( KB \) is true.

E.g., the KB containing “the Giants won and the Reds won” entails “The Giants won”.
Entailment in the wumpus world

- Consider possible models for $KB$ assuming only pits and a reduced Wumpus world

- Situation after detecting

  nothing in [1,1], moving right, breeze in [2,1]
Wumpus models

All possible ways to fill in the ?’s.
$KB = \text{all possible wumpus-worlds consistent with the observations and the "physics" of the Wumpus world.}$
$\alpha_1 = \text{"[1,2] is safe"}, \ KB \models \alpha_1$, proved by model checking
$\alpha_2 = \"[2,2] is safe\", \text{ } KB \models \alpha_2$
Inference Procedures

- $KB \models_i \alpha = \text{sentence } \alpha \text{ can be derived from } KB \text{ by procedure } i$

- **Soundness**: $i$ is sound if whenever $KB \models_i \alpha$, it is also true that $KB \models \alpha$ (i.e., *no wrong inferences, but maybe not all inferences*)

- **Completeness**: $i$ is complete if whenever $KB \models \alpha$, it is also true that $KB \models_i \alpha$ (i.e., *all inferences can be made, but maybe some wrong extra ones as well*)
Propositional logic

Propositional logic is the simplest logic

The proposition symbols are sentences, $S$

- $\neg S$ (negation)
- $S_1 \land S_2$ (conjunction)
- $S_1 \lor S_2$ (disjunction)
- $S_1 \Rightarrow S_2$ (implication)
- $S_1 \Leftrightarrow S_2$ (biconditional)
Each model/world specifies true or false for each proposition symbol

E.g.

<table>
<thead>
<tr>
<th></th>
<th>$P_{1,2}$</th>
<th>$P_{2,2}$</th>
<th>$P_{3,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

Rules for evaluating truth with respect to a model $m$:

- $\neg S$ is true iff $S$ is false
- $S_1 \land S_2$ is true iff $S_1$ is true and $S_2$ is true
- $S_1 \lor S_2$ is true iff $S_1$ is true or $S_2$ is true
- $S_1 \Rightarrow S_2$ is true iff $S_1$ is false or $S_2$ is true
- $S_1 \leftrightarrow S_2$ is true iff $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true
Simple recursive process evaluates an arbitrary sentence, e.g.,

\[ \neg P_{1,2} \land (P_{2,2} \lor P_{3,1}) = \]
\[ true \land (true \lor false) = \]
\[ true \land true = \]
\[ true \]
### Truth tables for connectives

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$Q$</td>
<td>$\neg P$</td>
<td>$P \land Q$</td>
<td>$P \lor Q$</td>
<td>$P \Rightarrow Q$</td>
<td>$P \iff Q$</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

**OR:** $P$ or $Q$ is true or both are true.

**XOR:** $P$ or $Q$ is true but not both.

**Implication is always true when the premises are False!**
Let $P_{i,j}$ be true if there is a pit in $[i, j]$.

Let $B_{i,j}$ be true if there is a breeze in $[i, j]$.

\begin{align*}
\text{start:} & \quad \neg P_{1,1} \\
& \quad \neg B_{1,1} \\
& \quad B_{2,1}
\end{align*}

"Pits cause breezes in adjacent squares"

\begin{align*}
B_{1,1} & \iff (P_{1,2} \lor P_{2,1}) \\
B_{2,1} & \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1})
\end{align*}
Inference by enumeration

- Enumeration of all models is sound and complete.
- For $n$ symbols, time complexity is $O(2^n)$...
- We need a smarter way to do inference!
- In particular, we are going to infer new logical sentences from the data-base and see if they match a query.
Two sentences are logically equivalent iff they are true in same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

\[
\begin{align*}
(\alpha \land \beta) & \equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\lnot(\lnot \alpha) & \equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \rightarrow \beta) & \equiv (\lnot \beta \rightarrow \lnot \alpha) \quad \text{contraposition} \\
(\alpha \rightarrow \beta) & \equiv (\lnot \alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \leftrightarrow \beta) & \equiv ((\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)) \quad \text{biconditional elimination} \\
\lnot(\alpha \land \beta) & \equiv (\lnot \alpha \lor \lnot \beta) \quad \text{de Morgan} \\
\lnot(\alpha \lor \beta) & \equiv (\lnot \alpha \land \lnot \beta) \quad \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{align*}
\]
Validity and satisfiability

A sentence is **valid** if it is true in all models,  
e.g., $True$, $A \lor \neg A$, $A \Rightarrow A$, $((A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:  
$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is **satisfiable** if it is true in some model  
e.g., $A \lor B$, $C$

A sentence is **unsatisfiable** if it is false in all models  
e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:  
$KB \models \alpha$ if and only if $(KB \land \neg \alpha)$ is unsatisfiable  
(there is no model for which $KB=\text{true}$ and $\alpha$ is false)
Proof methods divide into (roughly) two kinds:

**Application of inference rules:**
Legitimate (sound) generation of new sentences from old.
- Resolution
- Forward & Backward chaining

**Model checking**
Searching through truth assignments.
- Improved backtracking: Davis--Putnam-Logemann-Loveland (DPLL)
- Heuristic search in model space: Walksat.
We like to prove:

\[ KB \models \alpha \]

\[ \text{equivalent to: } KB \land \neg \alpha \text{ unsatisfiable} \]

We first rewrite \( KB \land \neg \alpha \) into conjunctive normal form (CNF).

A “conjunction of disjunctions”

\[ (A \lor \neg B) \land (B \lor \neg C \lor \neg D) \]

- Any KB can be converted into CNF.
- In fact, any KB can be converted into CNF-3 using clauses with at most 3 literals.
Example: Conversion to CNF

B_{1,1} \iff (P_{1,2} \lor P_{2,1})

1. Eliminate \iff, replacing \alpha \iff \beta with (\alpha \implies \beta) \land (\beta \implies \alpha).
   
   (B_{1,1} \implies (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \implies B_{1,1})

2. Eliminate \implies, replacing \alpha \implies \beta with \neg \alpha \lor \beta.
   
   (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1})

3. Move \neg inwards using de Morgan's rules and double-negation:
   
   (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1})

4. Apply distributive law (\land over \lor) and flatten:
   
   (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1})
Resolution Algorithm

The resolution algorithm tries to prove: $KB \models \alpha$ equivalent to
$KB \land \neg \alpha$ unsatisfiable

Generate all new sentences from KB and the query.

One of two things can happen:

1. We find $\mathcal{P} \land \neg \mathcal{P}$ which is unsatisfiable. I.e. we can entail the query.

2. We find no contradiction: there is a model that satisfies the sentence $KB \land \neg \alpha$ (non-trivial) and hence we cannot entail the query.
1. \( P \lor Q \)
2. \( \neg P \lor R \)
3. \( \neg Q \lor R \)

Prove \( R \).
Proof by Resolution

1. **P v Q**
2. **¬P v R**
3. **¬Q v R**

**Knowledge Base**

1. **P v Q**
2. **¬P v R**
3. **¬Q v R**

**1, 2 Resolution**

4. **Q v R**
3. **¬Q v R**

(P v Q) and (¬P v R) = true

means either Q or R have to be true

Prove R.

P has to be True to False, then
If P is false then
Q has to be true

If P is true then
R has to be true

Therefore, if P has to be T or F, then Q is T or R is T, thus you can cancel literals
Proof by Resolution

1. \( P \lor Q \)
2. \( \neg P \lor R \)
3. \( \neg Q \lor R \)

Prove \( R \).

\( (Q \lor R) \) and \( (\neg Q \lor R) \) = true

means either \( R \) or \( R \) has to be true

\( Q \) has to be True to False, then
If \( Q \) is false then
\( R \) has to be true

If \( Q \) is true then
\( R \) has to be true

Therefore, if \( Q \) has to be \( T \) or \( F \), then \( R \) is \( T \), thus you can cancel literals
Horn Clauses: A Special Case

- Resolution can be exponential in space and time.
- If we can reduce all clauses to “Horn clauses” resolution is linear in space and time

- **Horn clause**: a clause with at most one positive literal
  - \( \neg A \lor \neg B \lor \neg C \lor \neg D \)

- **Definite clause**: a Horn clause with exactly one positive literal
  - \( \neg A \lor \neg B \lor C \lor \neg D \)
Horn clauses are closed under resolution

\[(\neg A \lor B \lor \neg C) \land (\neg B \lor \neg D \lor \neg E \lor F)\]

\[\neg A \lor \neg C \lor \neg D \lor \neg E \lor F\]
Forward Chaining

- Start with known facts and derive new knowledge to add to the knowledge base
- Agent can derive conclusions from incoming percepts
- Data-driven approach
Forward Chaining

- Horn clauses:
  - C1: \( \neg P_1 \lor \neg P_2 \lor P_4 \)  
  \( (P_1 \land P_2 \rightarrow P_4) \)
  - C2: \( \neg P_4 \lor P_5 \)  
  \( (P_4 \rightarrow P_5) \)

- Facts:
  - \( P_1, P_2 \)
Forward Chaining

Horn clauses:
- C1: $\neg P_1 \lor \neg P_2 \lor P_4$  \hspace{1cm} (P_1 \land P_2 \rightarrow P_4)
- C2: $\neg P_4 \lor P_5$  \hspace{1cm} (P_4 \rightarrow P_5)

Facts:
- $P_1$, $P_2$

Percepts $P_1$ and $P_2$ resolve with C1 to get $P_4$
Forward Chaining

Horn clauses:

- C1: \( \neg P_1 \lor \neg P_2 \lor P_4 \) \hspace{1cm} (\( P_1 \land P_2 \rightarrow P_4 \))
- C2: \( \neg P_4 \lor P_5 \) \hspace{1cm} (\( P_4 \rightarrow P_5 \))

Facts:

- \( P_1, P_2 \)

Percepts \( P_1 \) and \( P_2 \) resolve with C1 to get \( P_4 \)

Resolve \( P_4 \) with C2 to get \( P_5 \)
Horn clauses:

- C1: \( \neg P_1 \lor \neg P_2 \lor P_4 \)
- C2: \( \neg P_4 \lor P_5 \)

Knowledge Base

1. \( P_1 \land P_2 \rightarrow P_4 \)
2. \( P_4 \rightarrow P_5 \)
Forward Chaining

Horn clauses:

- C1: \( \neg P_1 \lor \neg P_2 \lor P_4 \)
- C2: \( \neg P_4 \lor P_5 \)

Fact: \( P_1 \) is seen

Knowledge Base
1. \( P_1 \land P_2 \rightarrow P_4 \)
2. \( P_4 \rightarrow P_5 \)

Knowledge Base
1. \( P_1 \land P_2 \rightarrow P_4 \)
2. \( P_4 \rightarrow P_5 \)
3. \( P_1 \)
Forward Chaining

Horn clauses:

- C1: \( \neg P_1 \lor \neg P_2 \lor P_4 \)

- C2: \( \neg P_4 \lor P_5 \)

Knowledge Base

1. \( P_1 \land P_2 \rightarrow P_4 \)
2. \( P_4 \rightarrow P_5 \)

Fact: \( P_1 \) is seen

Using 1, 3, 4 deduce \( P_4 \)

Fact: \( P_2 \) is seen

Using 2, 5 deduce \( P_5 \)
Goal-driven reasoning

Work backwards to see if query is true

If inconclusive, query is false

Efficient: only touches relevant facts or rules
Backward Chaining

Horn clauses:
- C1: \( \neg P_1 \lor \neg P_2 \lor P_4 \)  
  \((P_1 \land P_2 \rightarrow P_4)\)
- C2: \( \neg P_4 \lor P_5 \)  
  \((P_4 \rightarrow P_5)\)

Facts:
- \(P_1, P_2\)

Goal: \(P_5\)

Subgoal: prove \(P_4\)
Backward Chaining

Horn clauses:
- C1: $\neg P_1 \lor \neg P_2 \lor P_4$  
  $(P_1 \land P_2 \rightarrow P_4)$
- C2: $\neg P_4 \lor P_5$  
  $(P_4 \rightarrow P_5)$

Facts:
- $P_1, P_2$

Goal: $P_5$

Subgoal: prove $P_4$
- Sub-sub goal: prove $P_2$
- Sub-sub goal: prove $P_1$
Class Exercise

- Write this problem in terms of propositional logic
- Use resolution to show inference that
  - W is in 3, 1
  - P is in 1, 3
  (give the facts shown at different time steps)
- Show your work