Logical Agents
Recap of last class

• Principles of intelligence
• Agent architectures
  • SMPA architecture
  • subsumption architecture
Homework

- Design the control logic for a robot that has to drive around the Trottier building and collect empty soda cans
- Robot has:

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Effectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser rangefinder</td>
<td>three-wheel base</td>
</tr>
<tr>
<td>Compass</td>
<td>2-joint arm</td>
</tr>
<tr>
<td>IR beam-break (between fingers)</td>
<td>gripper</td>
</tr>
<tr>
<td>IR proximity sensors (around base)</td>
<td></td>
</tr>
<tr>
<td>Contact sensor (on hand)</td>
<td></td>
</tr>
</tbody>
</table>
Herbert the soda-can-collecting robot

- Laser-based table-like object finder – drives robot to table
- If robot stationary, arm control reaches out for coke can
- Laser-based coke-can object finder moves arm toward soda can
- When can breaks infrared beam between fingers, grasp reflex is activated to pickup can
Herbert

- No planning
- No representation of the environment
- No communication between modules
- Herbert could respond quickly to changed circumstances:
  - e.g., new obstacle, or object approaching on a collision course.
  - e.g., place a coke can in front of Herbert – he will pick it up. No expectations about where coke cans will be found.
Which way am I going?

• **Without a map:**
  • How can Herbert be sure to find its way to all parts of the environment?
  • How can Herbert be sure to bring cans back to “home?”

• [video](#)
Today’s Agenda

• Using logical reasoning as the basis of a knowledge-based agent
Consider...

- Reflex agents find their way from Montreal to Ottawa by dumb luck
- Chess program calculates legal moves of its king, but does not know that no piece can be on 2 different squares at the same time
- Representations we have seen for problem-solving agents is limiting
Knowledge-Based Agents

• combine general knowledge with current percepts to infer hidden aspects of current state
• Knowledge base (KB): set of sentences represented in a knowledge representation language; represents assertions about the world
• Inference rule: way to derive new sentences from existing ones
• we add new sentences to KB and query what is known by TELL and ASK operations
Wumpus World

- **Environment**
  - 4x4 grid of rooms
  - Agent starts in [1,1]
  - Gold and wumpus locations chosen randomly
  - Each square other than [1,1] can be a pit with P(0.2)

- **Actuators**
  - *Left turn, Right turn, Forward*
  - Agent dies if it enters a square containing a pit or live wumpus
  - Can climb out of the cave from square [1,1]

- **Sensors:**
  - Stench (S): in cells directly adjacent to wumpus (W)
  - Breeze (B): in cells directly adjacent to pit (P)
Inference at Play in the wumpus world

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td></td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram of the wumpus world grid.
That was easy

• Now how do we develop a logical agent to apply the same reasoning?
Propositional logic

negation: $\neg S$ is true iff $S$ is false

conjunction: $S_1 \land S_2$ is true iff $S_1$ is true and $S_2$ is true

disjunction: $S_1 \lor S_2$ is true iff $S_1$ is true or $S_2$ is true

implication: $S_1 \Rightarrow S_2$ is true iff $S_1$ is false or $S_2$ is true

biconditional: $S_1 \Leftrightarrow S_2$ is true iff $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\neg P$</th>
<th>$P \land Q$</th>
<th>$P \lor Q$</th>
<th>$P \Rightarrow Q$</th>
<th>$P \Leftrightarrow Q$</th>
</tr>
</thead>
<tbody>
<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>true</td>
<td>true</td>
</tr>
</tbody>
</table>
Standard Logical Equivalences

\[(\alpha \land \beta) \equiv (\beta \land \alpha)\]  \hspace{10pt} \text{commutativity of } \land
\[(\alpha \lor \beta) \equiv (\beta \lor \alpha)\]  \hspace{10pt} \text{commutativity of } \lor
\[(\alpha \land (\beta \land \gamma)) \equiv (\alpha \land (\beta \land \gamma))\]  \hspace{10pt} \text{associativity of } \land
\[(\alpha \lor (\beta \lor \gamma)) \equiv (\alpha \lor (\beta \lor \gamma))\]  \hspace{10pt} \text{associativity of } \lor
\[\neg(\neg\alpha) \equiv \alpha\]  \hspace{10pt} \text{double-negation elimination}
\[(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)\]  \hspace{10pt} \text{contraposition}
\[(\alpha \Rightarrow \beta) \equiv (\neg\alpha \lor \beta)\]  \hspace{10pt} \text{implication elimination}
\[(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))\]  \hspace{10pt} \text{biconditional elimination}
\[\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)\]  \hspace{10pt} \text{de Morgan}
\[\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)\]  \hspace{10pt} \text{de Morgan}
\[(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))\]  \hspace{10pt} \text{distributivity of } \land \text{ over } \lor
\[(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))\]  \hspace{10pt} \text{distributivity of } \lor \text{ over } \land
Inference in Propositional Logic: Modus Ponens

\[ \alpha \rightarrow \beta, \alpha \]

\[ \beta \]

means that whenever \( \alpha \rightarrow \beta \) and \( \alpha \) are given, we can infer \( \beta \)
**Wumpus world sentences**

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 $[i, j]$

“There is no pit in $[1, 1]$”:

$R_1$: $\neg P_{1,1}$

“a square is breezy if and only if there is an adjacent pit”:

$R_2$: $B_{1,1} \iff (P_{1,2} \lor P_{2,1})$

$R_3$: $B_{1,2} \iff (P_{1,1} \lor P_{2,2} \lor P_{1,3})$

percepts for first two squares visited:

$R_4$: $\neg B_{1,1}$

$R_5$: $B_{1,2}$
Reasoning in Propositional Logic

\[ R_2: B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

\[ R_6: (B_{1,1} \to (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \to B_{1,1}) \quad \text{bicond. elimin.} \]

\[ (P_{1,2} \lor P_{2,1}) \to B_{1,1} \quad \text{and-elimination} \]

\[ \neg B_{1,1} \to \neg (P_{1,2} \lor P_{2,1}) \quad \text{contrapositive equivalence} \]

\[ \neg (P_{1,2} \lor P_{2,1}) \quad \text{Modus Ponens with } R_4 \]

\[ \neg P_{1,2} \land \neg P_{2,1} \quad \text{de Morgan’s rule} \]

\[ R_7: \neg B_{2,1} \quad \text{percept} \]

What can you infer from \( R_7 \)?
Resolution

• If you know:
  • it’s raining or it’s snowing \((a \lor b)\)

• and you also know:
  • it’s not raining \((\neg a)\)

• then you can conclude:
  • it’s snowing \((b)\)
Resolution

unit resolution inference:

if \( y_i \) and \( m \) are complementary literals, i.e., \( y_i \land m = 0 \)

\[
\begin{array}{c}
\gamma_1 \lor \cdots \lor \gamma_{i-1} \lor y_i \lor \gamma_{i+1} \lor \cdots \lor \gamma_k, \\
\gamma_1 \lor \cdots \lor \gamma_{i-1} \lor \gamma_{i+1} \lor \cdots \lor \gamma_k
\end{array}
\]

Generalizes to full resolution rule:

\[
\begin{array}{c}
\gamma_1 \lor \gamma_2, \neg \gamma_1 \lor \gamma_3 \\
\gamma_2 \lor \gamma_3
\end{array}
\]
So continuing...

\[ R_{11}: P_{1,1} \lor P_{2,2} \lor P_{1,3} \]

\[ P_{1,3} \]

bicon. elim. of \( R_3 \) & MP with \( R_5 \)
resolution with \( R_1 \) and \( R_9 \)
Exercise

- New rule: The Wumpus cannot be in the same square as a pit
- Can you determine where the Wumpus is?
- What about a pit?
First-Order Logic (FOL)

- Propositional logic: propositions (sentences)
- FOL adds quantification ($\forall$, $\exists$) and predicates
Predicates

- assume that Spot and Fido are dogs
- then the predicate, $\text{Dog}(x)$
  - returns TRUE if $x$ is Spot or Fido
Unification

• Let $p$ and $q$ be sentences in FOL
• Let $U$ be a unifier, i.e., some set of substitutions of values for variables
• $\text{subst}(U,x)$ is the result of applying the substitutions of $U$ to sentence $x$
• If $\text{subst}(U,p) = \text{subst}(U,q)$ then $\text{UNIFY}(p,q) = U$
  • The unification of $p$ and $q$ is the result of applying $U$ to both of them.
Example

Dog(Spot)
Man(John)

UNIFY((Bites(Spot,y) ∧ Dog(x) ∧ Man(y)),
(Bites(z,John) ∧ Dog(z) ∧ Man(John))) =
{x/Spot; y/John; z/Spot}