History of AI
1943 – 1969 The Beginnings
1943 McCulloch and Pitts show networks of neurons can compute and learn any function
1950 Shannon (1950) and Turing (1953) wrote chess programs
1951 Minsky and Edmonds build the first neural network computer (SNARC)
1956 Dartmouth Conference – Newell and Simon brought a reasoning program “The Logic Theorist” which proved theorems.
1952 Samuel’s checkers player
1958 McCarthy designed LISP, helped invent time-sharing and created Advice Taker (a domain independent reasoning system)
1962 Perceptron Convergence Theorem is proved.

Example ANALOGY Problem

— Knowledge-Based Systems/Agents

Key components:
- Knowledge base: a set of sentences expressed in some knowledge representation language
- Inference/reasoning mechanisms to query what is known and to derive new information or make decisions.

Natural candidate:
- logical language (propositional/first-order)
- combined with a logical inference mechanism

How close to human thought?
- In any case, appears reasonable strategy for machines.

Knowledge Representation

- Human intelligence relies on a lot of background knowledge
  - the more you know, the easier many tasks become
  - “knowledge is power”
  - E.g. SEND + MORE = MONEY puzzle.
- Natural language understanding
  - Time flies like an arrow.
  - Fruit flies like a banana.
  - The spirit is willing but the flesh is weak. (English)
  - The vodka is good but the meat is rotten. (Russian)
- Or: Plan a trip to L.A.
Example: Autonomous Car

State: k-tuple
(PersonInFrontOfCar, Policeman, Policecar, Slippery, YellowLight, RedLight)

Actions:
Brake, Accelerate, TurnLeft, etc.

Knowledge-base describing when the car should brake?
(PersonInFrontOfCar ⇒ Brake)
(((YellowLight ∧ Policeman) ∧ (~Slippery)) ⇒ Brake)
(Policecar ⇒ Policeman)
(Snow ⇒ Slippery)
(Slippery ⇒ ~Dry)
(RedLight ⇒ Brake)

Logic as a Knowledge Representation

• Components of a Formal Logic:
  − syntax
  − semantics (link to the world)
  − logical reasoning
    • entailment: α ⊨ β
      if, in every model in which α is true, β is also true.
  − inference algorithm
    • KB ⊨ α, i.e., α is derived from KB
    • should be sound and complete

Soundness and Completeness

Soundness:
An inference algorithm that derives only entailed sentences is called sound or truth-preserving.

KB ⊨ α implies KB ⊨ α

Completeness:
An inference algorithm is complete if it can derive any sentence that is entailed.

KB ⊨ α implies KB ⊨ α

Why soundness and completeness important?
 Allow computer to ignore semantics and “just push symbols”!

Propositional Logic: Syntax

• Propositional Symbols
  − A, B, C, …

• Connectives
  − ∧, ∨, ~, ⇒, ⇔

• Sentences
  − Atomic Sentence: True, False, Propositional Symbol
  − Complex Sentence:
    • (¬Sentence)
    • (Sentence ∨ Sentence)
    • (Sentence ∧ Sentence)
    • (Sentence ⇒ Sentence)
    • (Sentence ⇔ Sentence)

Propositional Logic: Semantics

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>~P</th>
<th>P ∧ Q</th>
<th>P ∨ Q</th>
<th>P ⇒ Q</th>
<th>P ⇔ 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>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

• Model:
  − Assignment of truth values to Symbols
  − Example: m= {P=True, Q=False}
    • Note: Often called “assignment” instead of “model”, and “model” is used for an assignment that evaluates to true.

• Entailment:
  − α ⊨ β if and only if, in every model in which α is true, β is also true.

• Model Checking:
  − To test whether α entails β, enumerate all models and check truth of α and β.