# Knowledge-Based Systems

CS472/CS473 – Fall 2005

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)**

1960's **Microworlds – solving limited problems: SAINT (1963), ANALOGY (1968), STUDENT (1967), blocksworld invented.**

1962 **Perceptron Convergence Theorem is proved.**

Example ANALOGY Problem
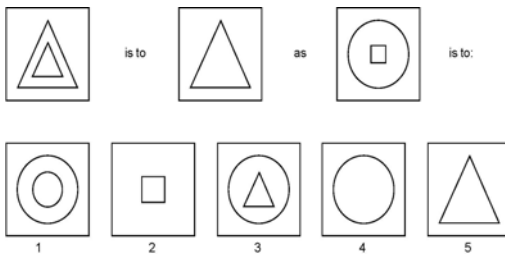
History of AI
1966 – 1974 Recognizing Lack of Knowledge

- **Herb Simon (1957): Computer chess program will be world chess champion within 10 years.**

- **Intractable problems, lack of computing power (Lighthill Report, 1973)**

- **Machine translation**

- **Limitations in knowledge representation (Minsky and Papert, 1969)**

➔ **Knowledge-poor programs**

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.**

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.

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 $\Rightarrow$ Brake)

(((YellowLight $\land$ Policeman) $\land$ (¬Slippery)) $\Rightarrow$ Brake)

(Policecar $\Rightarrow$ Policeman)

(Snow $\Rightarrow$ Slippery)

(Slippery $\Rightarrow$ ¬Dry)

(RedLight $\Rightarrow$ Brake)

Logic as a Knowledge Representation

- **Components of a Formal Logic:**
  - syntax
  - semantics (link to the world)
  - logical reasoning
    - entailment: $\alpha \vDash \beta$
      if, in every model in which α is true, β is also true.
  - inference algorithm
    - $KB \vdash \alpha$, 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*.

$$KB \vdash \alpha \text{ implies } KB \vDash \alpha$$

Completeness:

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

$$KB \vDash \alpha \text{ implies } KB \vdash \alpha$$

Why soundness and completeness important?

→ Allow computer to ignore semantics and "just push symbols"!

Propositional Logic: Syntax

- **Propositional Symbols**
  - A, B, C, …
- **Connectives**
  - $\land, \lor, \neg, \Rightarrow, \Leftrightarrow$
- **Sentences**
  - Atomic Sentence: True, False, Propositional Symbol
  - Complex Sentence:
    - (¬ Sentence)
    - (Sentence $\lor$ Sentence)
    - (Sentence $\land$ Sentence)
    - (Sentence $\Rightarrow$ Sentence)
    - (Sentence $\Leftrightarrow$ Sentence)

Example: Autonomous Car

Knowledge-base describing when the car should brake?

(PersonInFrontOfCar $\Rightarrow$ Brake)

$\land$ (((YellowLight $\land$ Policeman) $\land$ (¬Slippery)) $\Rightarrow$ Brake)

$\land$ (Policecar $\Rightarrow$ Policeman)

$\land$ (Snow $\Rightarrow$ Slippery)

$\land$ (Slippery $\Rightarrow$ ¬Dry)

$\land$ (RedLight $\Rightarrow$ Brake)

Observation from sensors:

YellowLight

$\land$ ¬RedLight

$\land$ ¬Snow

$\land$ Dry

$\land$ Policecar

$\land$ ¬PersonInFrontOfCar

Propositional Logic: Semantics

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

- **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:**
  - $\alpha \vDash \beta$ 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.