

INTRODUCTION

What is **Artificial Intelligence**?

Introduction

- AI is a broad field. It means different things to different people.
- AI is concerned with getting computers to do tasks that require human intelligence.
 - Example 1 : Complex Arithmetic –Computers can do this very easily.
 - Example 2: Recognizing a face – People do easily, but computers was very difficult to automate.

Introduction (Cont.)

- **AI** is concerned with difficult tasks , which require complex and sophisticated reasoning process and knowledge.

Why to automate Human Intelligence?

(and to which degree is it possible?)

Why to automate Human Intelligence ?

- Reason 1: To understand human intelligence better: We may be able to test and refine theories of Human Intelligence by writing programs which attempt to simulate aspects of human behavior.
- Reason 2: To have smarter programs and machines; by studying human reasoning we may develop useful techniques for solving difficult problems.

Introduction (Cont.)

Science Fiction Human-like robots – whether such a goal is possible or even desirable – belongs to science fiction, but does have impact on the practical work of writing smarter programs and developing better models of human reasoning.

Introduction (Cont.)

- **AI** – for us is a technical subject; we put emphasis on “**Computational Techniques**” and less on psychological modeling and philosophical issues.
- **AI** is both a branch of science and a branch of engineering.
 - As **ENGINEERING**, **AI** is concerned with the concepts, theory and practice of building intelligent machines.

AI as a branch of Science and Engineering

Examples:

1. **Expert Systems** that give advice about specialized subjects; e.g., medicine, mineral exploration, etc....
2. **Question-Answering Systems** for answering queries posed in restricted, but large subset of English and other natural languages.
3. **Theorem Proving Systems.**
4. **Systems for program verifications.** It is a very important field of CS.

Knowledge in Intelligent Entities

“Intelligent entities seem to anticipate their environments and the consequences of their actions.”

We assume that they possess **knowledge** of their environments.

Knowledge in Intelligent Entities (Cont.)

- What is such knowledge?
- What forms can it take?
- How do entities use knowledge?
- How is knowledge acquired?

Knowledge in Intelligent Entities (Cont.)

We have:

- Procedural Knowledge.
- Declarative Knowledge.

We talk about and define:

- Knowledge Representation.
- Knowledge Base.

Forms of Knowledge

There are two major ways we can think about machine having knowledge about its world:

- IMPLICIT – Procedural
- EXPLICIT – Declarative

Procedural Knowledge

- Examples:
 - The knowledge represented by the actual running or execution of a program is **procedural** – It is difficult to extract the knowledge from the text of the program code for other uses. It is contained in the very procedures that uses it.
 - Spider knowledge about spinning the web and
 - Tennis knowledge used by a player are both **procedural**.

Declarative Knowledge

- Examples:
 - Tennis Knowledge as TAUGHT by the instructor is declarative knowledge.
 - Engineer designing a bridge is declarative.
- Declarative Knowledge – contains declarations about the world. Typically it is stored in **symbol structures** that are accessed by the procedures that use this knowledge.

Intelligent Machines

Intelligent Machines will need both procedural and declarative knowledge.

Declarative Knowledge – focus of AI

- AI focuses more on the declarative knowledge.
- One of the most standard books by N. Nilson (Stanford) **Logic as foundations of AI** is concerned only with declarative knowledge.

Reasons for preferring Declarative Knowledge

- Reasons for AI researchers to prefer declaratively represented knowledge :
 - Can be **changed** easily.
 - Can be used for several **different purposes**.
 - The knowledge base itself does not have to be repeated or designed for different applications.
 - **Can be extended by reasoning process** that derive additional knowledge.

Conceptualization

- The formalization of knowledge in declarative form begins with “Conceptualization”.
- The language of conceptualization is predicate calculus.
- Definition from the book [Logic as foundations of AI](#).

Conceptualization (Nilson Def.) (Declarative Knowledge.)

- Conceptualization – step one of formalization of knowledge in declarative form.
- $C = (\mathcal{U}, F, R)$
- \mathcal{U} – Universe of discourse; a finite set of objects.
- F – Functional Basis Set; Set of functions (defined on \mathcal{U}). Functions may be partial.
- R – Relational Basis Set; Set of relations defined on \mathcal{U} .

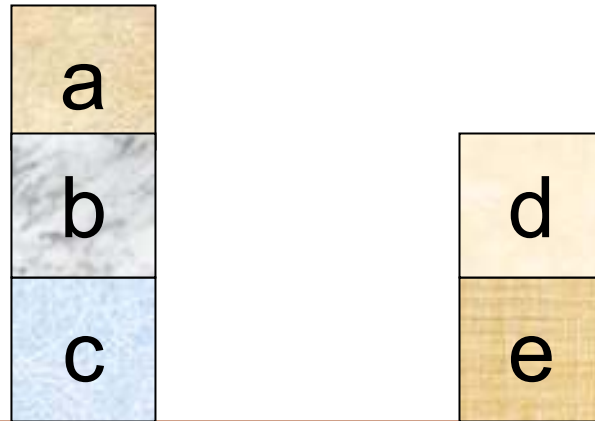
Conceptualization (Cont.)

- **R** – Relational Basis Set; Set of relations defined on \mathcal{U} .
- $\mathcal{R} \in \mathbf{R}$, $\mathcal{R} \subset \mathcal{U}^n$, $\# \mathcal{R} = n$

This is like in predicate logic:

$M = (\mathcal{U}, \mathbf{F}, \mathbf{R})$ is a Model. Where $\mathcal{U} \neq \emptyset$,
 $\mathcal{F} \in \mathbf{F}$, $f \in \text{FUN}$, $f_I = \mathcal{F}$, $f_I : \mathcal{U}^n \rightarrow \mathcal{U}$, etc.,
Satisfiability Model, etc., in Predicate Logic.

Example: Block World



(Example is continued on next slide.)

Example: Block World (Cont.)

- $\mathcal{U} = \{ a, b, c, d, e \}$
- \mathbf{F} – functions. $\mathbf{F} = \{h\}$
- **Intuitively:** h maps a block into a block on the top of it. ($h = \text{Top}$)
- **Formally:** $h = \{(b,a), (c,b), (e,d)\}$
- $h(b) = a$; $h(c) = b$; $h(e) = d$.
- h is a partial function and $h : \mathcal{U} \rightarrow \mathcal{U}$
- Domain of $h = \{b,c,e\} \subset \mathcal{U}$

Example: Block World (Cont.)

- **R** – Set of Relations.
- **R** = {Above, On, Table, Clear}
- Above $\subseteq \mathcal{U} \times \mathcal{U}$, On $\subseteq \mathcal{U} \times \mathcal{U}$
- Table $\subseteq \mathcal{U}$, Clear $\subseteq \mathcal{U}$

Example: Block World (Cont.)

- Intuitively: *Above* (x,y) iff x is anywhere above y.
(Semantics.)

FORMALLY (definition):

- *Above* = {(a,b), (b,c), (a,c), (d,e)}
(Syntax)

- Intuitively: *On* (x,y) iff x is immediately above y.

- FORMALLY (definition):

$$\text{On} = \{(a,b), (b,c), (d,e)\} \quad , \quad \text{On} \subseteq \mathcal{U} \times \mathcal{U}$$

Example: Block World (Cont.)

- Intuitively:
- $\text{Clear}(x)$ iff there is no block on top of x .

- Formally(definition)

$\text{Clear} = \{a, d\} \subseteq \mathcal{U}$
(one argument Relation.)

- Intuitively: $\text{Table}(x)$ iff x is resting directly on the table.

Formally(definition)

- $\text{Table} = \{c, e\} \subseteq \mathcal{U}$ (one arg. Relation.)

Example: Block World (Cont.)

- Observe:
- $\text{On} \subseteq \text{Above}$; $\text{Clear} \cap \text{Table} = \emptyset$.

We choose in our Conceptualization only these Relations and Functions but – depending on what we want to tell about our world – we can have less or more of them.

Example: Block World (Cont.)

- $On \subseteq \mathcal{U} \times \mathcal{U}$
 $On = \{(a,b), (b,c), (d,e)\}$ (Math. Definition)
- This is “Prolog” like statements:
 $On(a,b)$, $On(b,c)$ and $On(d,e)$ Facts in Prolog
It is equivalent to your definition
(declaration) of what “On” means, i.e.
- We write $On(a,b)$ for (a,b) element of On .

Intended Interpretation

- $On = \{(a,b), (b,c), (d,e)\}$
- We can also use other symbols, e.g. :
▪ $= \{(a,b), (b,c), (d,e)\}$ (Math. model)
- This is the same as:
▪ (a,b) , ▪ (b,c) and ▪ (d,e)
- **Intended Interpretation** of the symbol ▪ is
“x is immediately above y.”

Representation in Predicate Logic

- **Facts** about our Universe:

On(a,b)	Above(a,b)	Clear(a)
On(b,c)	Above(b,c)	Clear(d)
On(d,e)	Above(a,c)	Table(c)
Top(b,a)	Above(d,e)	Table(e)
Top(c,b)	Top(e,d)	

Reasoning: Resolution i.e., Prolog

- To be able to use Prolog we have to convert our Axioms (rules) in “non qualifier” form (Skolemization) and classical form.
- Resolution – Inference Engine (of Prolog)

Plan for Logic Part

1. Propositional Resolution.
(Proof of Correctness =
Completeness Theorem.)
2. Strategies (to go faster!)
3. Predicate Resolution- introduction

Major AI Areas

1. Game Playing:

In early 1950 **Claude Shannon (1950)** and **Alan Turing (1953)** were writing chess programs for von Neumann computers.

2. But, in fact **Shannon** had no real computer to work with, and

3. **Turing** was denied access to his own team's computers by the British government on the grounds that

4. **research into AI was frivolous !**

Major AI Areas

Game Playing (Cont.)

- Search as a Major AI Technique:
Search is a **problem solving** technique that systematically explores a space of **problem states**, i.e., stages of problem solving process.
 - Example:
Different board configurations in a game form a space of alternative solutions. The space is then searched to find a final answer.

Search as AI Technique (Cont.)

- Much of early research in State Space Search was done using common board games: checkers, chess, 16-puzzle.
- Games have well defined rules, and hence it is easy to generate the search space.
- Large space – Heuristic Search.
- 1984 book by Pearl , “Heuristics” – First Comprehensive Mathematical treatment of heuristic search.
- Heuristic Search is widely used now in Theorem Proving and Data Mining.

Major AI Areas (cont)

2. Automated Reasoning and Theorem Proving:

- Origin: Foundations of Mathematics.
- Mathematics can be considered as “axiomatic theory.”
- Hilbert Program (1910) – to formalize **all** of mathematics in such a way that a proof of **any** theorem can be found automatically.
- Gentzen(1934) – **positive** answer for first order logic. (Propositional Logic.)

Major AI Areas, Automated Reasoning and Theorem Proving (Cont.)

- Gödel (1933) – negative answer for arithmetic; incompleteness theorem.
- Robinson (1965) – Resolution.
- Program Verifications – use theorem proving techniques.

Major AI Areas: Expert Systems

3. Expert Systems:

- Obtaining knowledge from human experts, or **databases (automated rules generators)** and representing it in a form that computer may apply to similar problems. **Rule Based Systems.**
- Expert Systems grew into information systems.
- **Always has specific domain.**

Expert Systems (Examples)

- First Examples:
 - Dendral, Stanford 1960:
built to infer the structure of organic molecules from their chemical formulas.
 - MYCIN, Stanford 1970:
diagnostic system, plus prescribes treatment for Spinal Meningitis and bacterial infection in the blood. It was the first program to address the problem of reasoning with **uncertain** and/or **incomplete** information.
Still on the Web ! (Medical Information Systems.)

Expert Systems

(Our handout #1 – Modern Approach)

Managing Uncertainty in E.S: (Jerzy Busse, Kluwer)

1. Knowledge acquisition by using **Machine Learning**
2. **Rule Induction** from databases. (Rough Sets approach)
3. **Uncertainties in Quantitative approach:**
 - Bayes rules and network (probabilistic approach)
 - Belief networks. (probabilistic)
 - Dempster – Shafer Theory:
 - Bayesian Belief
 - Dempster Rules.

Managing Uncertainty in E.S.

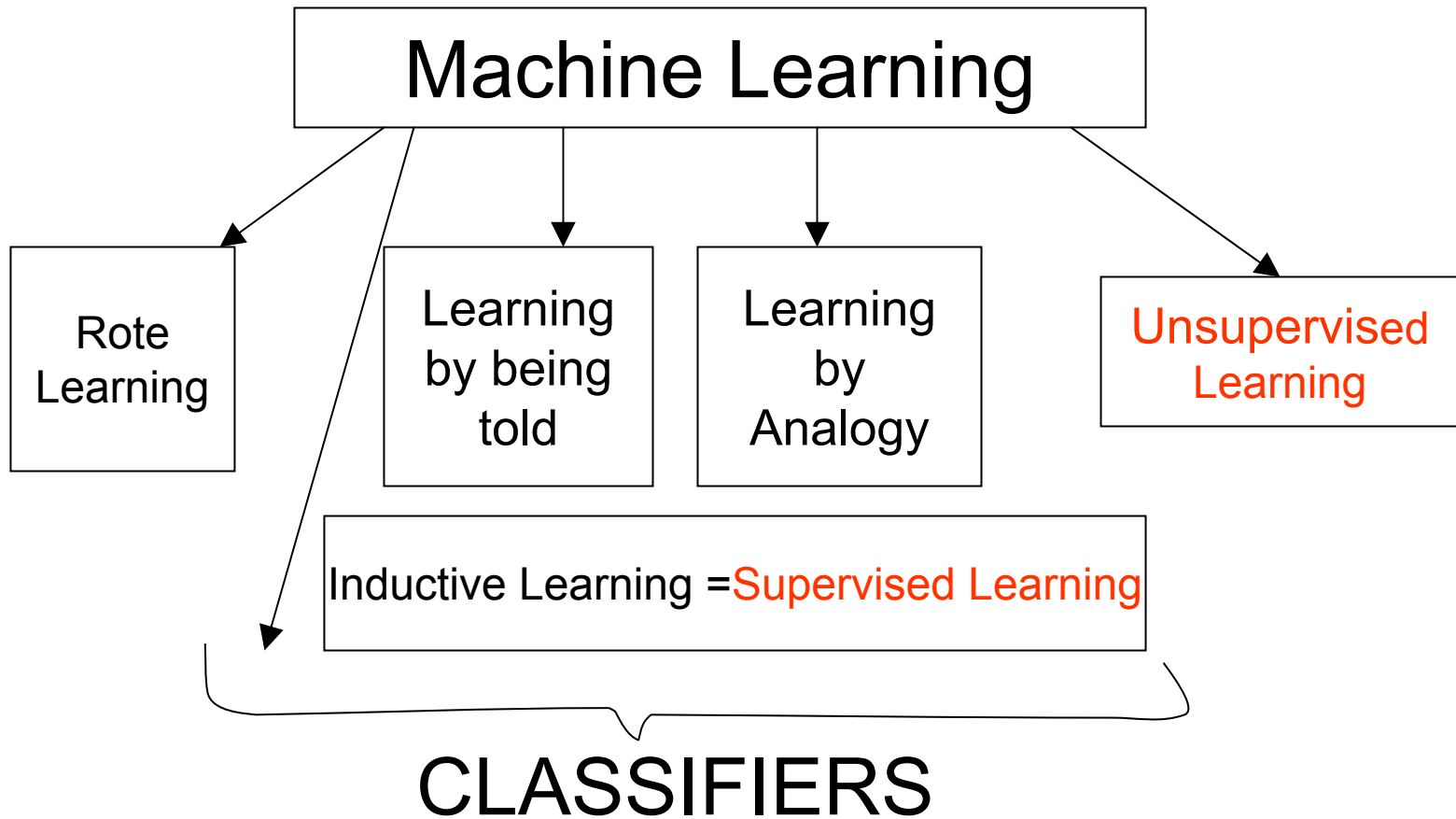
3. Uncertainties – Set Valued Approach:

- Fuzzy Sets (Zadek, 1965)
- Rough Sets (Pawlak, 1985)
- Machine learning / data mining techniques.

4. Uncertainties – Qualitative Approaches:

- Modal Logics.
- Non-monotonic logics.
- Default logic
- Plausible Reasoning.

Machine Learning



Expert Systems

MYCIN Story:

MYCIN asked if the patient was **pregnant** even though it has been told that the patient was **male**.

Expert Systems

- Modern Expert Systems always have Machine Learning Components. (Supervised)
- Inductive (Supervised) Learning in large databases is called Data Mining.
- Supervised Learning Techniques are:
 - 1) Genetic Algorithms. (Evolutionary)
 - 2) Neural Networks (Inductive Learning.)
 - 3) Decision Tree (Inductive Learning)
 - 4) Rough Sets
 - 5) Classification by Association

Other AI Areas

- Natural Language Processing.
- Natural Language Understanding
- Robotics
- Intelligent Visualization.
- Automated Theorem Proving.

Short History

- The Name, “AI” , was suggested in 1956 by McCarthy (at Dartmouth at that time, and then at Stanford, Yale) during a two month long workshop at Dartmouth.
- The Workshop was devoted to programs that could perform:
 - Elementary Reasoning Tasks
 - Proving Simple Theorems.
 - Answering Simple Questions.
 - Playing Board Games.
 - ALL Non computational (in a sense of numbers) tasks.

Short History (cont.)

- All together there were **10 people**. For the next 20+ years the field would be dominated by them, their students and colleagues at MIT, CMU (Carnegie-Mellon University) , Stanford and IBM.
- **Allen Newell** and **Herbert Simon** from **CMU** stole the show with **Logic Theorist (LT)** – first program to think non-numerically.

Short History (Cont.)

- LT proved most of the theorems in Chapter 2 of Russell and Whitehead's "Principia Mathematica".
- Herb Gelernter (Stony Brook) constructed first (1959) Geometry Theorem Prover.
- Now Theorem Proving is a separate field.

Short History (Cont.)

- 1952-1969 : Time of Early Enthusiasm and Great Expectations.
- 1952 :
Arthur Samuel wrote a tournament level checkers program.
- In February 1956 the program was demonstrated on national TV.
- A. Samuel , like Alan Turing had a hard time to obtain computer time; worked only at night.

Short History (Cont.)

- 1958 :
McCarthy moved from Dartmouth to MIT and invented LISP. (Second oldest programming language still in use; Which is the Oldest?)
- LISP is now being replaced by Prolog as a dominant AI language (in many areas.)
- McCarthy and his group also invented Timesharing and formed Digital Equipment Corporation (DEC) to produce time sharing computers.

Short History (Cont.)

- 1958 :
 - Marvin Minsky moved to MIT. He represented **Anti-logic** outlook.
 - McCarthy was Pro-logic. Hence **McCarthy** moved to **Stanford**.
 - McCarthy's Logic agenda was busted by Robinson's discovery of **Resolution** → (Kowalski) "**Prolog** logic programming", **founded SRI, Stanford Research Institution** – main area of research is general purpose methods for logical reasoning.

Short History (Cont.)

- 1969:
 - Green's Question – Answering and Planning Systems.
 - Shakey Robotics Projects; first integration of logical reasoning and physical activity.
- 1963:
 - J. Slagle's program SAINT was able to solve closed form integration problems. (first year calculus.)

Short History (Cont.)

- 1968 :
T. Evans program “Analogy” solved geometric analogy problems from IQ tests.
- 1967:
D. Bobrow’s program “student” solved some SAT problems.

Short History (Cont.)

- 1971:
D. Huffman's "vision" project did rearrangement of the blocks, put on top of the table, using a Robot hand that picked one block at a time.
- 1970:
P. Winston – first learning theory.

Short History (Cont.)

- 1972:
T. Winograd – first natural language understanding theory.
- 1974:
Planner of Scott Fahlman.
- 1966 – 1974:
A Dose of Reality !

Short History (Cont.)

- In 1958 H. Simon predicted that in 10 years a computer would be a chess champion and that it would prove **important** mathematical theorems,
But:
Many programs often were based mainly on simple syntactic manipulations.
- **ELIZA (1965) by Weizenbaum:**
 - Search on the web ! Weizenbaum is still alive.
 - The machine ELIZA had no understanding.

Short History (cont.)

- 1966:
All american governmental funding for machine translations were cancelled !
- 1973:
british government stopped AI support to all but 2 universities.

Short History (cont.)

- **Genetic Algorithms** formulated in **1958-59**, but computers were not yet up to it.
Now all over the place!
- The same happened to **Neural Networks** – mathematical model and theoretical research was rampant, but computers were not strong and fast enough to give meaningful results.
- **1980** – **back propagation (NN) algorithm** and first applications followed.

Knowledge-Based Systems the (1969-79)

- Narrow the area of expertise and then solve.
- Dendral (1969):
 - Buchanan, a philosopher turned Computer Scientist, and Joshua Lederberg (a nobel geneticist) at Stanford, brought forward the first successful knowledge-intensive system, “Dendral”.
 - Knowledge base is a large number of special purpose rules.
 - With Dendral, there is a clean separation of the knowledge base (Rules) and the reasoning component. (following McCarthy.)

Knowledge-Based Systems the (1969-79)

- MYCIN and certainty factors.
- Prospector (1979, Rutgers):
 - Provided recommendations of exploratory dwellings at geological sites.

Short History

AI becomes an Industry (1980-now)

- 1982:

First successful Expert System, “RI”, at DEC (McDermot) was made.

The Program helped configure orders for new Computer Systems and by 1986 was saving the company \$40 million a year.

Short History

AI becomes an Industry (1980-now)

- 1988:
 - DEC's AI group had 40 E.S.!
 - Du Pont had 100 E.S. in use and 500 in development, saving \$10 million a year.
 - Every major US corporation had (has) its own AI group.
- Information Systems – in all Industries and new University departments.

Short History

AI becomes an Industry (1980-now)

- 1981:
 - Japan announced “Fifth Generation” project.
 - The “Fifth Generation Project” used Prolog to achieve full-scale natural language understanding.
 - USA formed a company MCC (Microelectic and Computer Technology Corporation) to compete with Japan.

Others:

Cornegie Group, Inference, Intellicop,
Lisp Machines.

Short History (Cont.)

AI becomes an Industry (1980-now)

- Industry went from a few million in sales in 1980 to 2 Billion in 1988.
- For the “Fifth Generation Project” FGP, a progress had been made but the project failed.
- Prolog is just one of many programming languages.

PROLOG – Logic Programming

Short History

- 1964–65 :
Robinson, (Syracuse University),
introduced **Resolution**.
- 1968–70:
Kowalski, University of Edinburgh, England,
created first version of Prolog.
- David Warren (British) made the prolog
machine.
- **Stony Brook's D. Warren** was a president of
Association for Logic Programming. Prominent !

Philosophical Issues

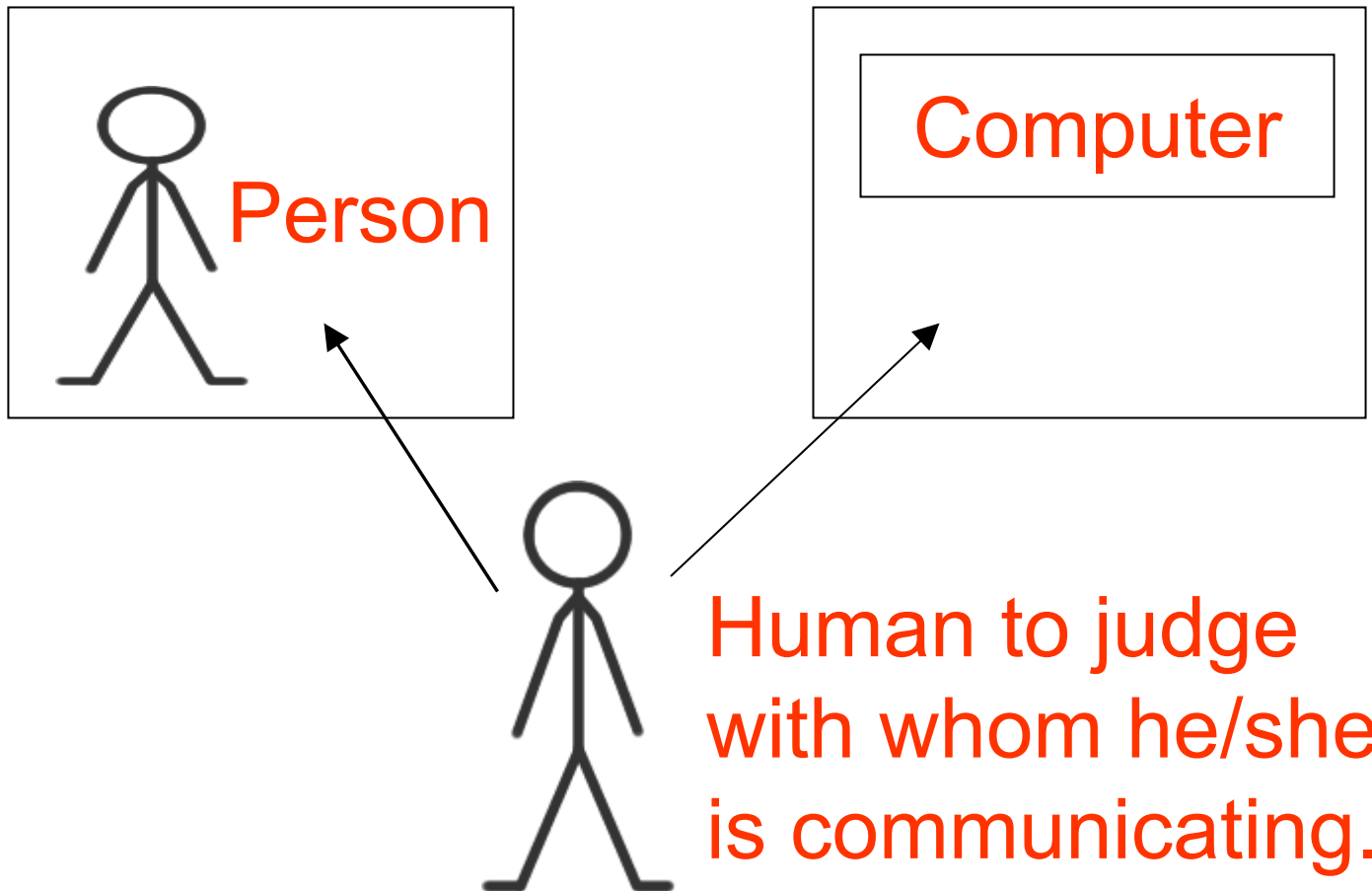
- AI research makes the assumption that human intelligence can :
 - 1) be reduced to the (complex) manipulations of symbols, and
 - 2) It does not matter what Medium is used to manipulate these symbols. (It does not have to be a biological brain.)

Philosophical Issues

- Turing's test (1950) and Charles Searle's Chinese room (1980) are the most known contributions to the philosophical debate.
- **Loebner Prize Contest** – competition started in 1991 & inspired by Turing test.
 - Best program – \$2,000.
 - Grand Prize - \$100,000. (No one claimed yet.)

Philosophical Issues (Cont.)

- Turing Test:



Philosophical Issues (Cont.)

- Turing test (Cont.):
 - The human can ask the other person/computer any question; including very subjective questions, like:
 - “What is your favorite color?” and
 - “What do you think of the privatization of the British Rail” ?
 - **Computer is Intelligent** – If it answers so well that the judge cannot distinguish it from a person.

Philosophical Issues (Cont.)

- Searle “Chinese Room” Critique:

If you have a huge rule book on Chinese you can “talk” Chinese even if you do not understand it; so, T. Test does not measure the Intelligence.

Questions – Answers Systems.

The Loebner Prize Contest

- 10 judges type questions at a computer which are responded by a person or a program.
- The best program receives \$2000.
- The Winning program : \$100,000.
 - No one won yet, but one program fooled 5 judges out of 10.

Questions – Answers Systems. The Loebner Prize Contest (Cont.)

- 1996 Winning Fragment:

P: No worries ! Hey! I did not come here to be subjected to bad language! How rude.

J2: Sorry, sorry. Did not mean to offend. How old are you?

The Loebner Prize Contest 1996 Winning Fragment (Cont.)

P: That is ok. Just don't do it again.
Well, should my age really matter?
Just for the record, I am 24. How old
are you then?

J2: I am 21.

P: If you are 21, it is your business! Would
you like to hear a joke? Knock, knock.

Questions – Answers Systems.

- Programs like these don't really understand the conversation. They use various **stores of amusing replies and use keywords in the question to select the reply.**
- Example: 1960 a program, "ELIZA" attempted to imitate a psychotherapist.