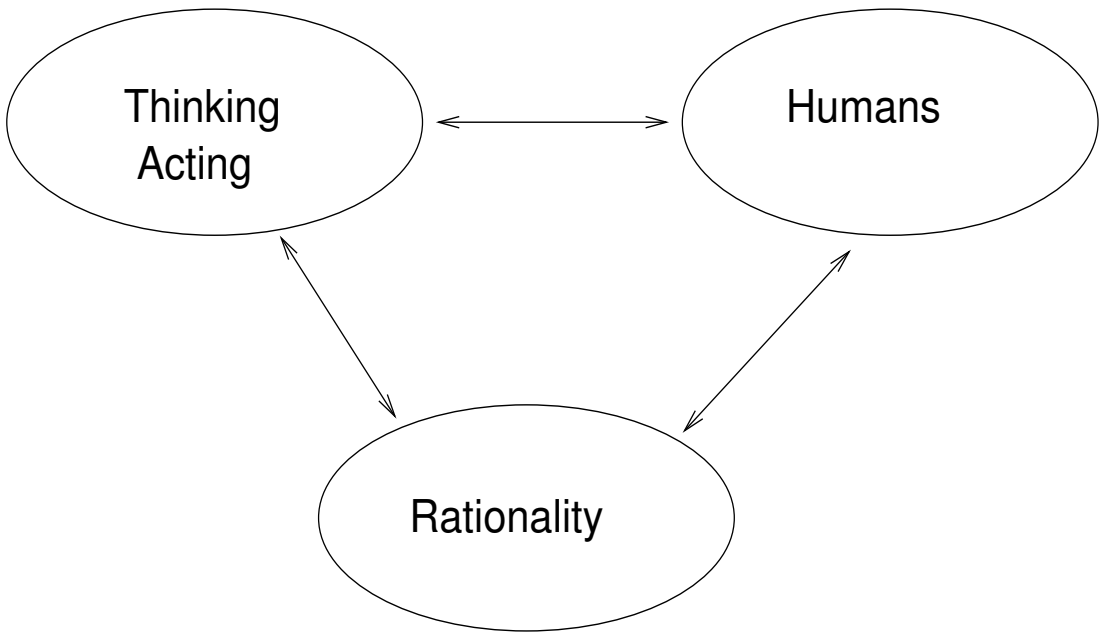
Lecture One: Introduction - Artificial Intelligence, Intelligent Agents, Knowledge Representation and Reasoning

301315 Knowledge Representation and Reasoning

What is AI - AI Definition

Definition

AI is a computer program or machine that can do something *intelligently*.



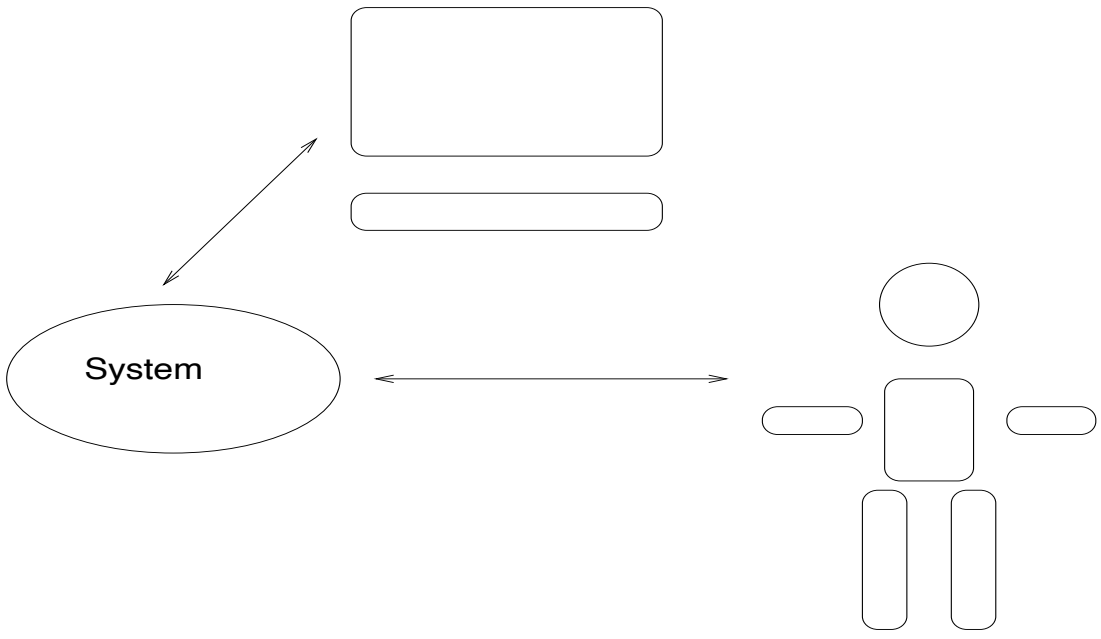
What is AI - AI Definition

AI (systems) can do the following things:

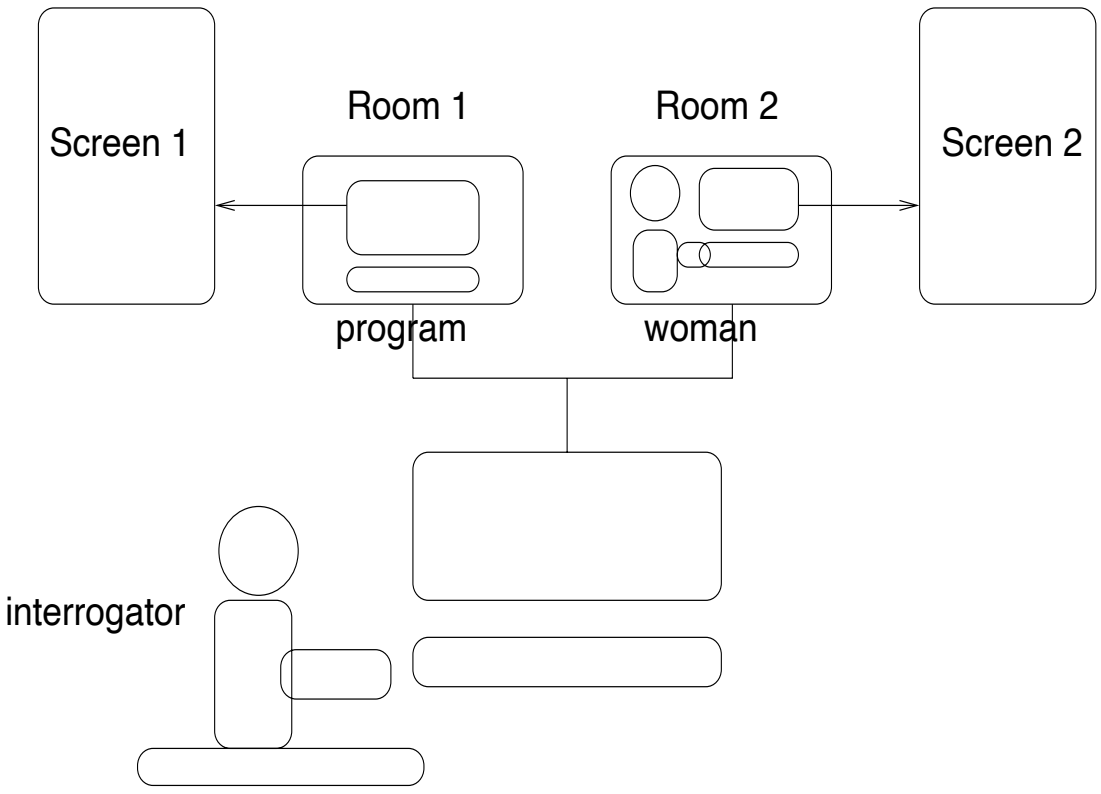
- Understanding natural languages
- Reasoning
- Learning
- Vision
- Motion

What is AI - Turing Test

But, what is the fundamental issue of AI?



What is AI - Turing Test



What is AI - Turing Test

- Interrogator can input questions from computer;
- Computer in Room 1 can receive the question, and answer the question, display its answer on Screen 1;
- Computer in Room 2 can receive the question. The woman types her answer into the computer in Room 2, and the computer then displays the woman's answer on Screen 2;
- The woman in Room 2 always answers the question Truthfully;
- The computer in Room 1 can lie!

What is AI - Turing Test

- Interrogator: question_1, question_2, $\cdots$ question_k;
- Computer (Room 1): answer_1, answer_2, $\cdots$, answer_k;
- Woman (Room 2): answer_1', answer_2', $\cdots$, answer_k'.

Will the interrogator know which answers from computer and which answers from the woman?

If the interrogator cannot distinguish, then we say that the program in the computer (in Room 1) passes the *Turing Test*.

Question: Does such program exist?

The answer is *Yes*!

What is AI - Why AI

- A goal of AI is to learn how to build software components of intelligent agents capable of reasoning and acting in a changing environment
- To exhibit intelligent behaviour, an agent should have a mathematical model of
  1. its environment
  2. its own capabilities
  3. its goals

  and it must have algorithms for achieving its goals

What is AI - Why AI

- AI can give us insights into the nature of thought. For the first time in history, researchers are applying a new tool - the computer - to study and test their ideas by designing intelligent agents
- AI can help us with software engineering
- AI can give us things of practical value such as decision support systems, intelligent search engines, and robots
- AI can serve as a connection between disciplines, both within computer science and between CS and science, logic, philosophy, and linguistics

Intelligent Agents - What is an Agent

- agent is an entity that observes and acts on its environment and directs its activity toward achieving goals; i.e., almost any program
- intelligent agent - performs complex reasoning tasks that lead to nontrivial behavior
- adaptive agent - adapts to changes in its environment
- autonomous agent - independent of human control

Intelligent Agents - What is an Agent

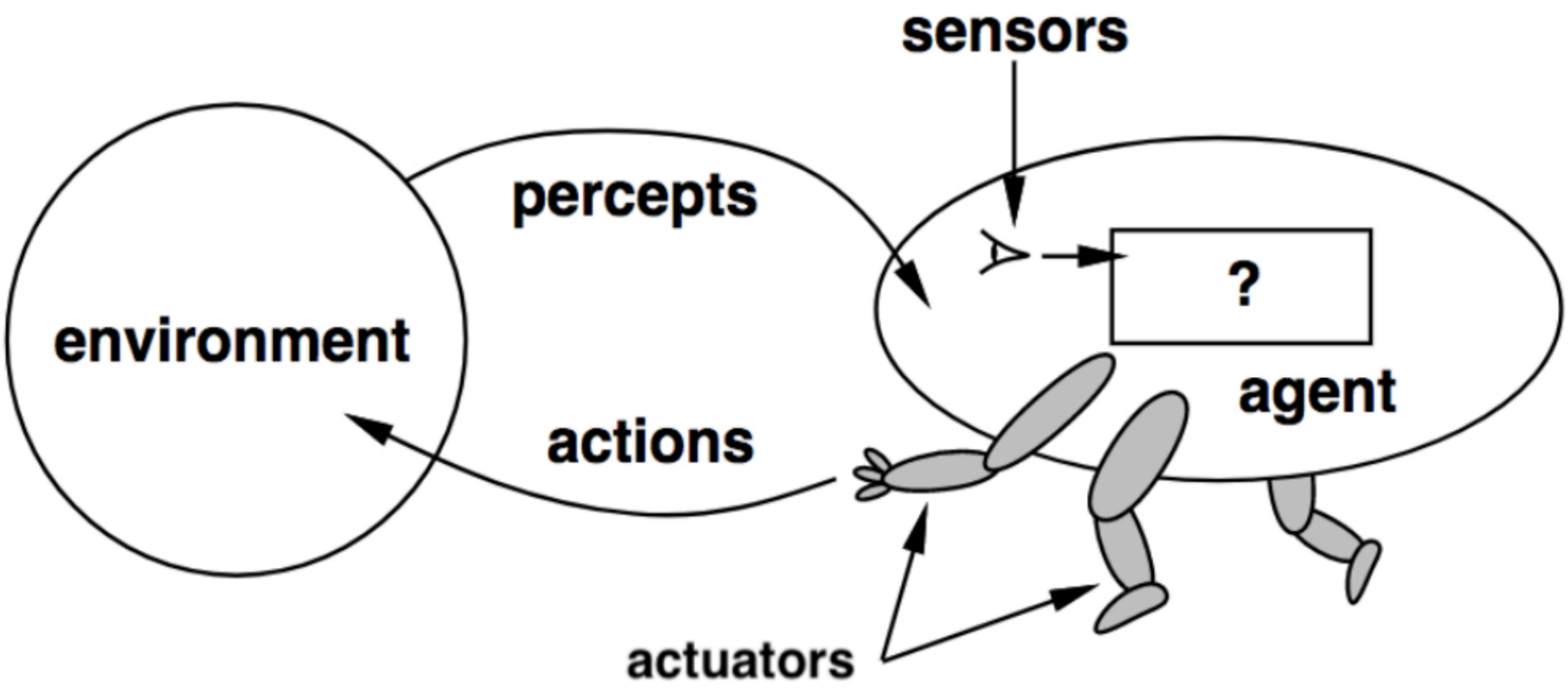


Figure: An agent interact its environment through its sensors and actuators.

Intelligent Agents - Agent Model

- a language for representing the agent's knowledge
- reasoning algorithms for things like
  - learning
  - planning
  - diagnostics

  Often based on sophisticated search
- agent architecture - structure combining different submodels of an agent into one coherent whole

Intelligent Agents - An Agent's Workflow

An Agent does the following

1. observe the world:
   - make sure that what you sense makes sense
   - update what you know based on what you sense
2. select a goal
3. search for a plan to achieve the goal
4. carry out the beginning part of the plan, update your view of the world based on what you did, return to step 1

What is KR - KR Definition

- *Knowledge*: proposition/statement/assertion/ like "John *knows* that Peter will come today"
- *Representation*: A symbolic language to represent what an agent believes or/and knows about the external world and/or his own mind
- *Reasoning*: A formal manipulation of the symbols representing a collection of believed propositions (statements/assertions) to produce representation of new ones.

What is KR - KR Definition

Definition

Knowledge Representation and Reasoning (KR) is a field of AI dedicated to representing information about the world in a form that a computer system can utilise to solve complex tasks such as diagnosing a medical condition, having a dialog in a natural language. It incorporates findings from logic to automate various kinds of reasoning.

What is KR - Why KR

A knowledge base Σ contains the following two assertions:

$$fly(X) \leftarrow bird(X)$$
$$bird(tweety)$$

With proper logic reasoning mechanism, we should conclude that *tweety* can fly, i.e., $\Sigma \vdash fly(tweety)$.

Now, we extend Σ by adding two new assertions:

$$not\_fly(X) \leftarrow emu(X)$$
$$emu(tweety)$$

Then from the extended knowledge base, Σ', should we conclude *tweety* cannot fly? i.e., $\Sigma' \vdash not\_fly(tweety)$

What is KR - Why KR

10 min classroom exercise

Example

Please think of a reasoning problem from your daily life scenario.

What is KR - Why KR

KR will enable us to build a system to achieve a number of reasoning tasks:

- We can add new tasks and easily make them depend on previous knowledge
- We can extend existing behaviours by adding new beliefs/knowledge
- We can debug faulty behaviour by localising the erroneous beliefs (assertions) of the system
- We can concisely explain and justify the behaviour of the system

The Role of Logic - Logic, Computer Science, and AI

- Theoretical computer science developed out of logic and computational theory: the correctness of programs, algorithm complexity, and model checking
- Theorem proving
- KR is logic based: commonsense reasoning, nonmonotonic logics, reasoning about change, belief revision and system update

The Role of Logic - Commonsense Reasoning

Example

Emu flies

The Role of Logic - Commonsense Reasoning

Example

Let a knowledge base Σ contain the following assertions:

$$fly(X) \leftarrow bird(X)$$
$$not\_fly(X) \leftarrow emu(X)$$
$$\bot \leftarrow fly(X), not\_fly(X)$$
$$bird(tweety)$$
$$emu(tweety)$$

Then we have:

(1) $\Sigma \vdash fly(tweety)$
(2) $\Sigma \vdash not\_Fly(tweety)$
(3) $\Sigma \vdash \bot$ - This is a contradiction!

The Role of Logic - Commonsense Reasoning

Example

How can we solve this contradiction? Let Σ^* consist of the following rules:

$$fly(X) \leftarrow bird(X), not\ not\_fly(X)$$
$$not\_Fly(X) \leftarrow emu(X)$$
$$\bot \leftarrow fly(X), not\_fly(X)$$
$$bird(tweety)$$
$$emu(tweety)$$

We will have $\Sigma^* \vdash not\_fly(tweety)$, but $\Sigma^* \nvdash fly(tweety)$, no contradiction now.

Logic Programming

We have some basic facts as follows:

$father(john, sam).$
$mother(alice, sam).$
$gender\_of(john, male).$
$gender\_of(sam, male).$
$gender\_of(alice, female).$

Then the following two rules define the relation *parent*:

$parent(X, Y) \leftarrow father(X, Y).$
$parent(X, Y) \leftarrow mother(X, Y).$

Logic Programming

Finally, we have the following rule to define relation *child*:

$$child(X, Y) \leftarrow parent(Y, X).$$

Now we can ask question to this program Π consisting of all facts and rules above.

- "Is Sam a child of John? ", i.e., $\Pi \vdash child(sam, john)$? (Yes)
- "Who are Sam's parents?", i.e., find such X such that $\Pi \vdash parent(X, sam)$ (we will find that $X = john$ and $X = alice$)

Logic Programming

Example

A logic program representing a family tree:

```
%% A Family ASP Program

male(jack).
male(oliver).
male(ali).
male(james).
male(simon).
male(harry).
female(helen).
female(sophie).
female(jess).
female(lily).
```

Logic Programming

```
parent_of(jack,jess).
parent_of(jack,lily).
parent_of(helen,jess).
parent_of(helen,lily).
parent_of(oliver,james).
parent_of(sophie,james).
parent_of(jess,simon).
parent_of(ali,simon).
parent_of(lily,harry).
parent_of(james,harry).
```

Logic Programming

```
%% Rules
father_of(X,Y):- male(X),
    parent_of(X,Y).

mother_of(X,Y):- female(X),
    parent_of(X,Y).

grandfather_of(X,Y):- male(X),
    parent_of(X,Z),
    parent_of(Z,Y).

grandmother_of(X,Y):- female(X),
    parent_of(X,Z),
    parent_of(Z,Y).
```

Logic Programming

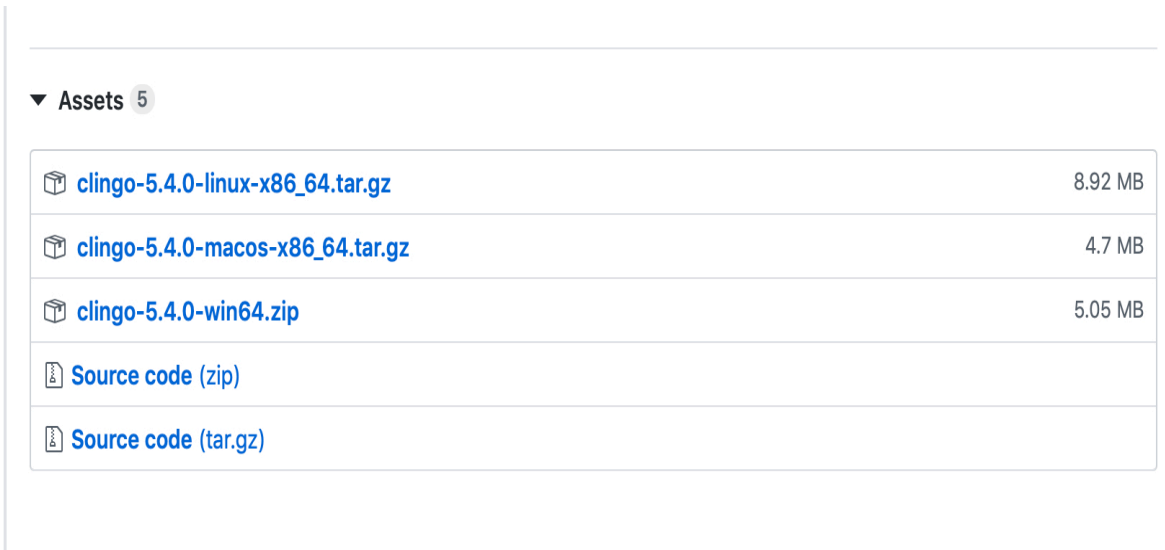
```
sister_of(X,Y):-
    female(X),
    father_of(F,Y), father_of(F,X), X != Y.

sister_of(X,Y):- female(X),
    mother_of(M,Y), mother_of(M,X), X != Y.

aunt_of(X,Y):- female(X),
    parent_of(Z,Y), sister_of(Z,X).
```

Logic Programming

```
brother_of(X,Y):- male(X),
    father_of(F, Y), father_of(F,X), X != Y.

brother_of(X,Y):- male(X),
    mother_of(M, Y), mother_of(M,X), X != Y.

uncle_of(X,Y):-
    parent_of(Z,Y), brother_of(Z,X).

ancestor_of(X,Y):- parent_of(X,Y).
ancestor_of(X,Y):- parent_of(X,Z),
    ancestor_of(Z,Y).
```

Tutorial and Lab Exercises

1. Installation of Answer Set Programming system `clingo` on your personal computer:
   - Go to `https://github.com/potassco/clingo/releases`
   - Download the binary source codes from the link that fits to your system:



   - Install it on your laptop

Tutorial and Lab Exercises

2. Do an extensive search, to describe what are the main ideas behind AI program AlphaGo. What is the key difference between AlphaGo and its successor AlphaGo Zero?
3. Carefully study the family tree logic program displayed on slides 26-30, list all new facts that can be derived from this family tree program.