CS3243: Introduction to Artificial Intelligence

Semester 2, 2019/2020

Teaching Staff

- Lecturer: Yair Zick
  Email: zick@comp.nus.edu.sg
  Office: COM2-02-60
- Co-lecturer: Daren Ler
  Email: dler@comp.nus.edu.sg
  Office: COM2-02-65

Teaching Staff

- TAs:
  - Mohit Agrawal (e0508702@u.nus.edu)
  - Hu Zikun (e0338273@u.nus.edu)
  - Lew Yin Hui Stephanie (dcslyhs@nus.edu.sg)
  - Arpan Losalka (arpan@u.nus.edu)
  - Tohar Lukov (e0384144@u.nus.edu)
- Undergraduate TAs
  - Vineeth Buddha (e0325373@u.nus.edu)
  - Li Zeyong (e0032604@u.nus.edu)
  - Teh Zi Cong Nicholas (e0148237@u.nus.edu)

Teaching Resources: LumiNUS

http://luminus.nus.edu.sg/

- Lesson Plan
- Lectures, Tutorials, Supplementary Materials, Homework
- Discussion forum
  - Any questions related to the course should be raised on this forum
  - Emails to me will be considered public unless otherwise specified
- Announcements
- Homework submissions
- Webcasts

A ‘Tasting Menu’ of AI

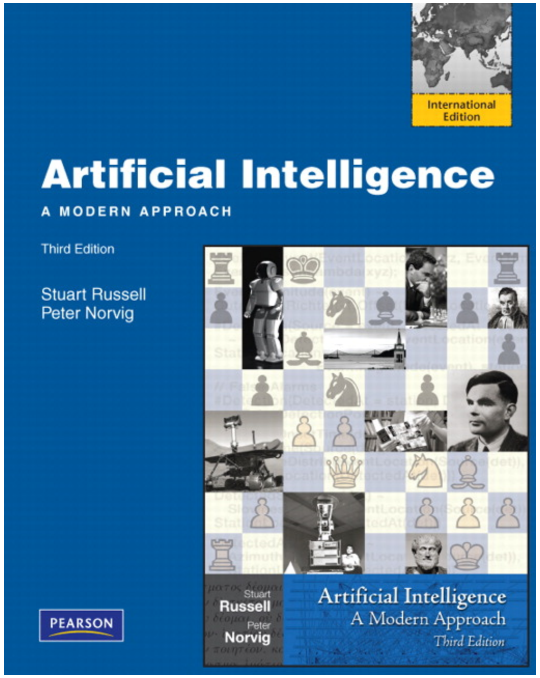
Foundational concepts of AI	Who?
• Search • Game playing • Logic • Uncertainty • Probabilistic reasoning	• Undergraduates • beginning graduate students. • CS orientation, or by permission.

Beyond CS3243

Machine Learning	Search & Planning	Logic	... And more!
CS3244	CS4246	CS4248	CS4261
CS5242	CS5338, TBA	CS6207	CS6208
CS5339		CS4244	CS6281
CS5340			
CS5344			

Readings

- Textbook:
  - Russell and Norvig (2010). Artificial Intelligence: A Modern Approach (3rd Edition ⟵ Important!)
  - Online Resources, Code, and ERRATA: http://aima.cs.berkeley.edu/
  - We will not cover entire book! But it makes for an interesting read…



Syllabus

- Introduction and Agents (chapters 1, 2)
- Search (chapters 3, 4, 5, 6)
- Logic (chapters 7, 8, 9)
- Uncertainty (chapters 13, 14)
- Learning (chapters 18,21)

Assessment Overview

What	When	Grade Percentage
Midterm Exam (during lecture, NO make-up)	5 March 2020	20%
Final Exam	2 May 2020 (afternoon)	50%
Class Project	Week 7 and Week 10	25%
Tutorials + Attendance	-	5%

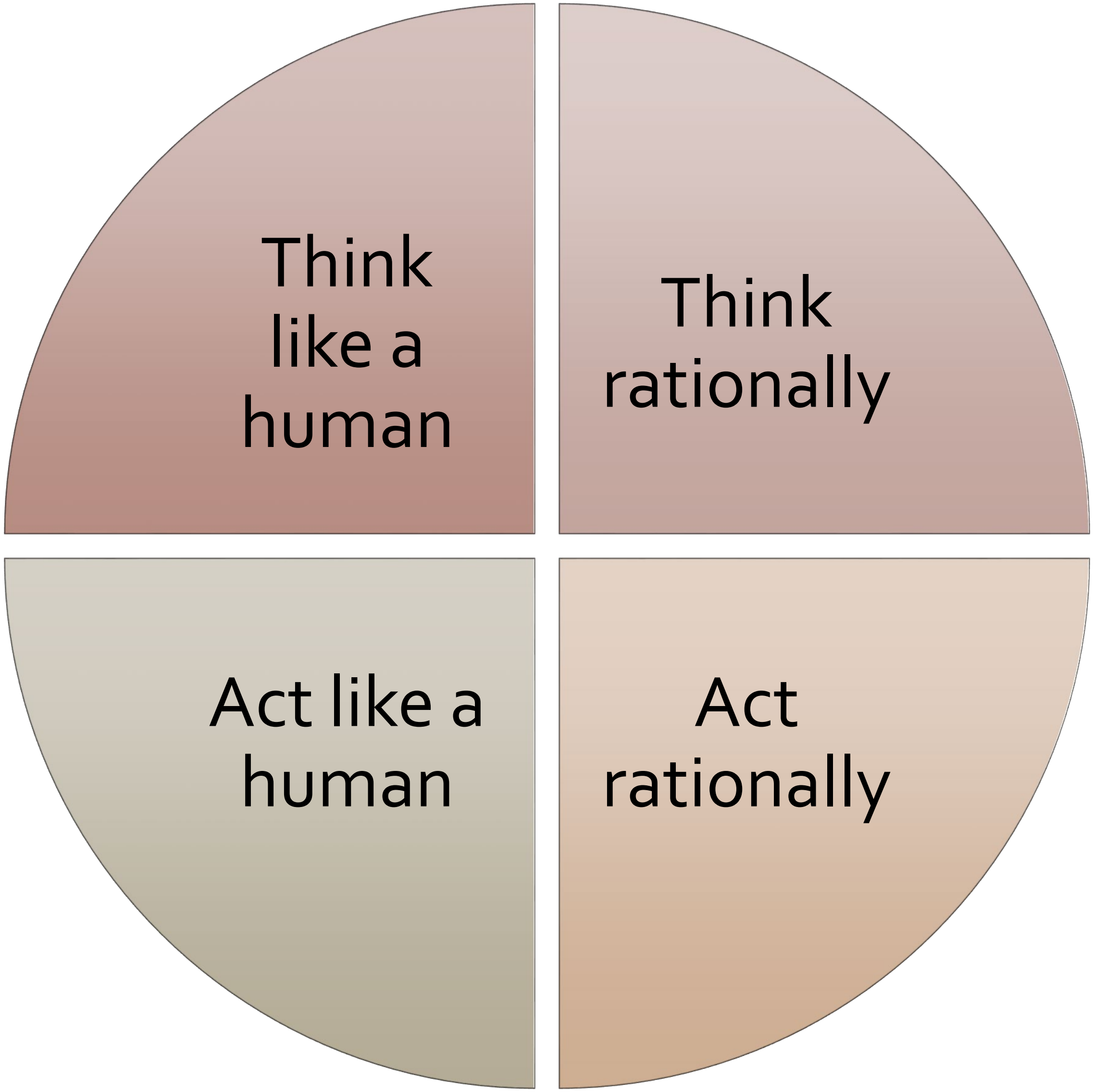
Introduction

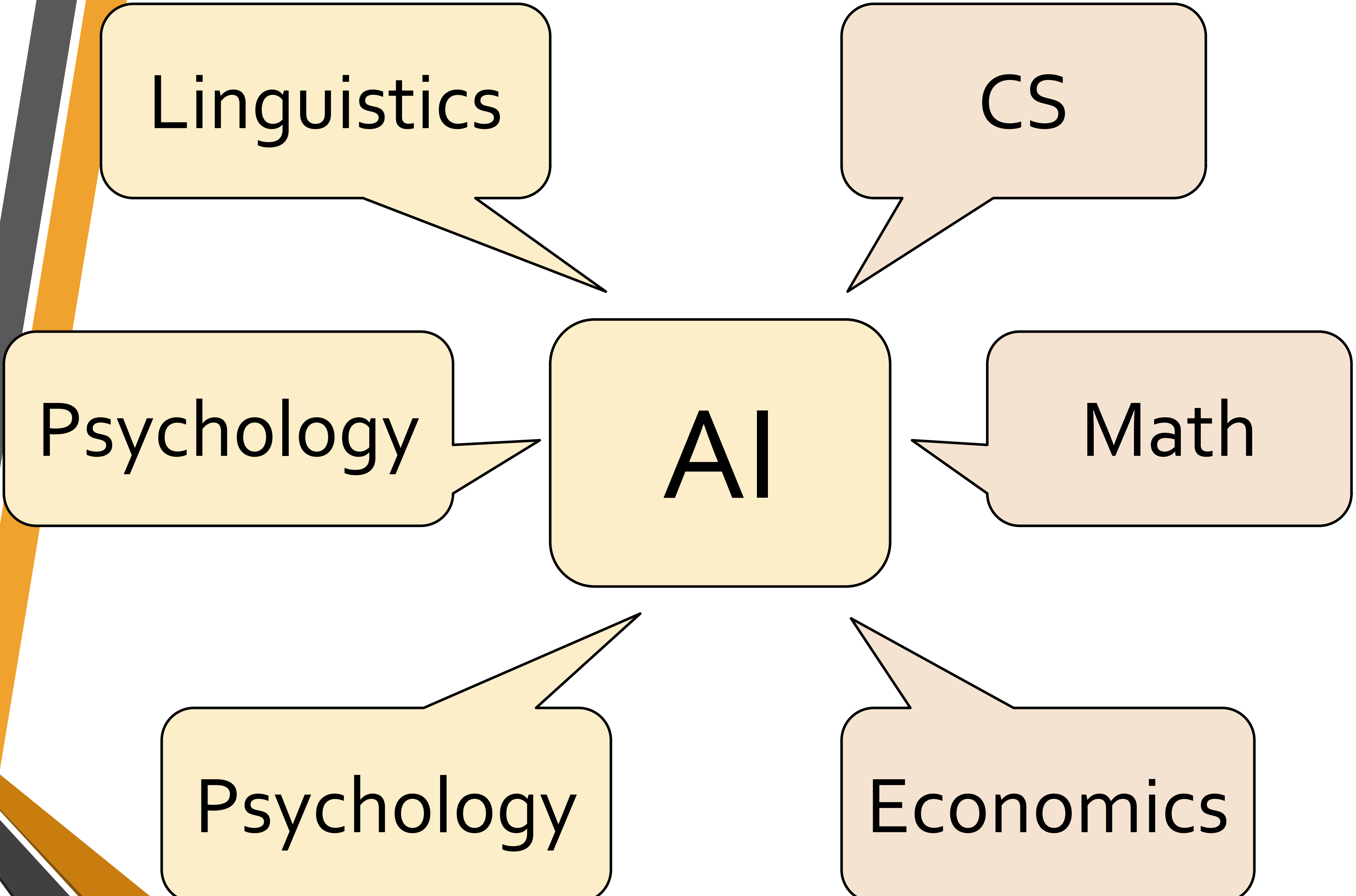
AIMA Chapter 1

What is AI?

Making Computers (at least as) Good as Humans in Human Tasks

- … but not necessarily in the same way a human would!
  - Birds and planes fly very differently
  - AI solves problems very differently.
  - I think!


Think like a human
Think rationally
Act like a human
Act rationally


Linguistics
CS
Psychology
AI
Math
Psychology
Economics

Philosophy

- Ethics
- Logic
- Learning
- Rationality
- Theory of the Mind

Computer Science

- Theory of Computing
- Hardware
- Control Theory
- Dynamic Systems

Mathematics

- Formal representation
- Probability
- Statistics

Economics

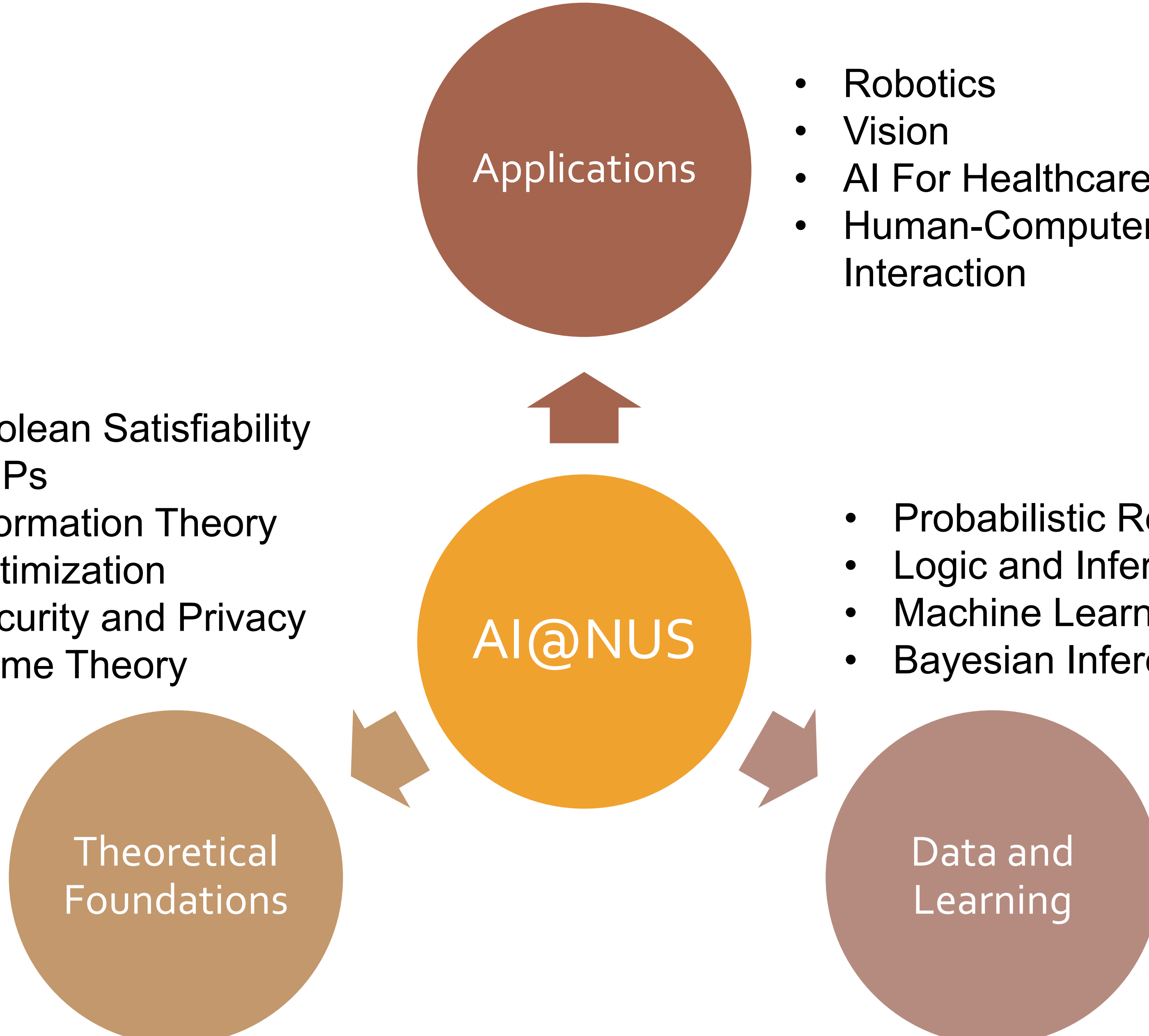
- Game theory
- Decision theory
- Fair Division
- Utility theory

Psychology

- Perception and motor control
- Experiments

Linguistics

- Knowledge representation
- grammar



- Robotics
- Vision
- AI For Healthcare
- Human-Computer Interaction

- Boolean Satisfiability
- CSPs
- Information Theory
- Optimization
- Security and Privacy
- Game Theory

- Probabilistic Reasoning
- Logic and Inference
- Machine Learning
- Bayesian Inference

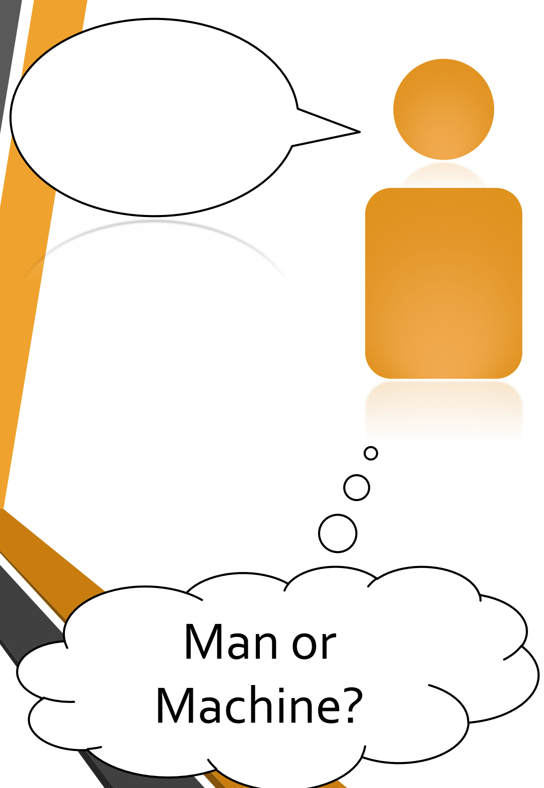
AI is Getting Better at Gameplay

"A computer once beat me at chess, but it was no match for me in kickboxing" – Emo Philips

Year	Game	Program	Developer	Techniques
1994	Checkers	Chinook	U. Alberta	Rule Based + search
1997	Chess	Deep Blue	IBM	Search + randomization
2008	Limit Texas Hold'em	Polaris (Cepheus 2015)	U. Alberta	Agent based modeling, game theory
2011	Jeopardy	Watson	IBM	NLP, Information retrieval, data analytics
2015	2 player No Limit Texas Hold'em	Claudico (later Libratus)	Carnegie Mellon Univ.	Game Theory, Reinforcement Learning
2016	Atari Games	DeepMind	DeepMind	Deep Learning
2016	Go	AlphaGo	DeepMind	Deep Learning, search
2018	DOTA 2	OpenAI Five	OpenAI	RL, Deep learning, search
2019	Starcraft 2	AlphaStar	DeepMind	RL, Deep learning, search

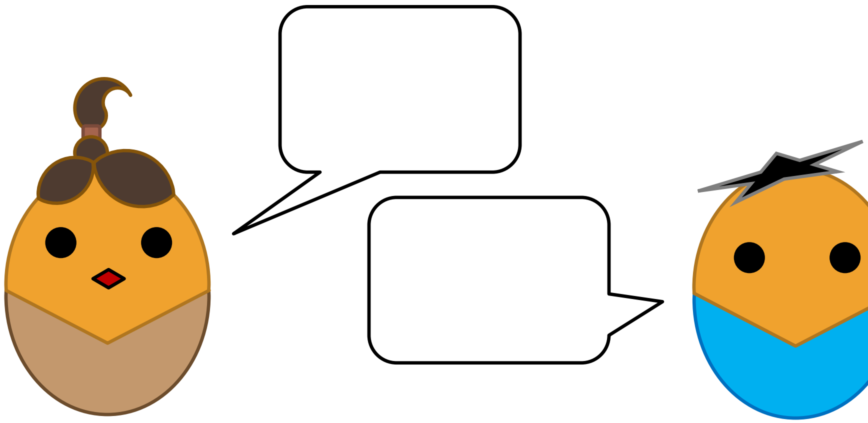
When Can a Machine Truly Think?

- Turing (1950). Computing Machinery and Intelligence:
  “Can machines think?” → “Can machines behave intelligently?”
- Operational test for intelligent behavior: The Imitation Game



Is the Turing Test a Good Idea?

- What are the advantages of the Turing test?
- What are the disadvantages?



- Think about 1-2 of each
- Talk to the person next to you about it
- We'll move to entire class after

Winograd Schema

The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.

Who [feared/advocated] violence?

Answers: The city councilmen/the demonstrators.

Winograd Schema

The trophy doesn't fit into the brown suitcase because it's too [small/large].

What is too [small/large]?

Answers: The suitcase/the trophy.

Winograd Schema

- A sentence with an ambiguous pronoun (he/she/it/they/their ...).
- Need to decide what the pronoun refers to (offered two options).
- This depends on **context**.
- Context can be easily flipped with a **single word**.

The firemen arrived [after/before] the police because they were coming from so far away.

Who came from far away?

Answers: The firemen/the police.

Winograd Schema Challenge

- You are given m Winograd schema, with the context word chosen uniformly at random.
- Design an AI that can correctly resolve a significant number of them.
- What is a trivial lower bound on the number of schema one can solve?

A Single Test for Intelligence?

- Difficult to resolve
- Tests tend to be
  - over-specified
  - very subjective
- Result will be debatable

Acting Rationally: Rational Agent

- Rational behavior: doing the "right thing"
- What is the "right thing" to do? Expected to achieve best outcome
  - Best for whom?
  - What are we optimizing?
  - What information is available?
  - Unintended effects

- Break through wall to get a cup of coffee
- Prescribe high doses of opiates to depressed patient
- kill human who tries to deactivate robot

Rational Agents

- An agent is an entity that perceives and acts
- **This course:** designing rational agents
- An agent is a function from **percept histories** to **actions**, i.e., $f: P^* \rightarrow A$
- We seek the best-performing agent for a certain task; must consider computation limits!

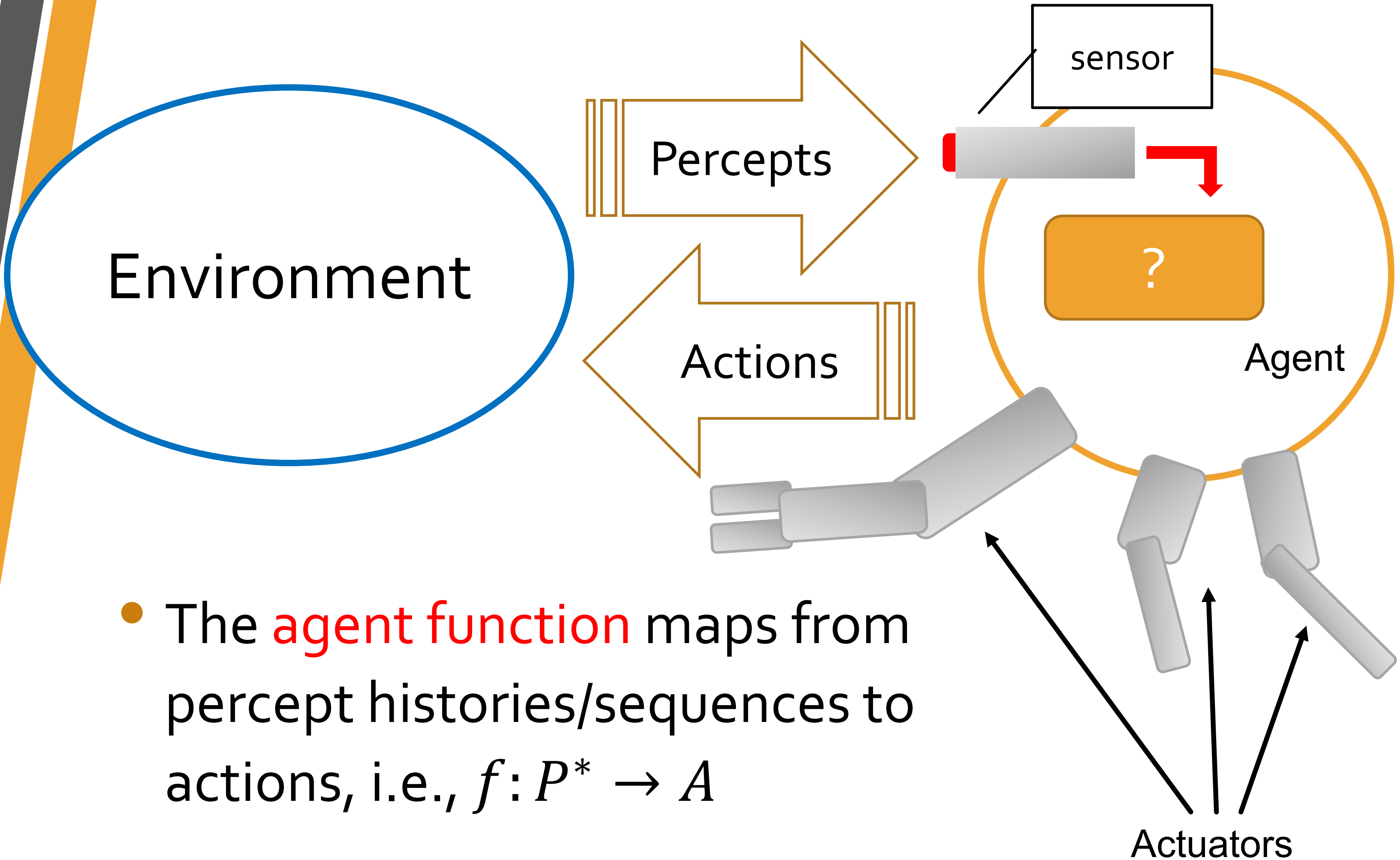
design best program given resources

Intelligent Agents

AIMA Chapter 2

Agents

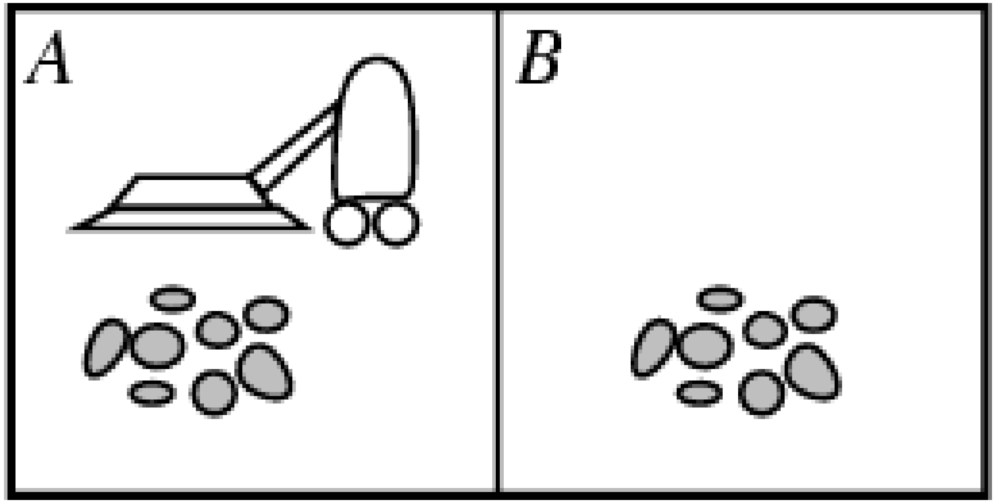
- Anything that can be viewed as perceiving its environment through sensors; acting upon that environment through actuators
- **Human agent:** eyes, ears, skin etc. are sensors; hands, legs, mouth, and other body parts are actuators
- **Robotic agent:** cameras and laser range finders for sensors; various motors for actuators



- The agent function maps from percept histories/sequences to actions, i.e., $f: P^* \rightarrow A$
- The agent program runs on the physical architecture to perform f

agent = architecture + program

Vacuum-Cleaner World



- Percepts: location and status, e.g., $[A, \text{Dirty}]$
- Actions: $\text{Left}, \text{Right}, \text{Suck}, \text{NoOp}$

Vacuum-Cleaner Agent Function

Percept Sequence	**Action**
$[A, \text{Clean}]$	Right
$[A, \text{Dirty}]$	Suck
$[B, \text{Clean}]$	Left
$[B, \text{Dirty}]$	Suck
$[A, \text{Clean}], [A, \text{Clean}]$	Right
$[A, \text{Clean}], [A, \text{Dirty}]$	Suck

Rational Agents

- An agent should strive to “do the right thing”, based on what it can perceive and the actions it can perform. The right action: maximize agent success.
- Performance measure: objective criterion for measuring success of an agent's behavior
- Vacuum-cleaner agent:
  - amount of dirt cleaned
  - time taken
  - electricity consumed
  - noise generated

Perhaps a bit of everything?

Rational Agents

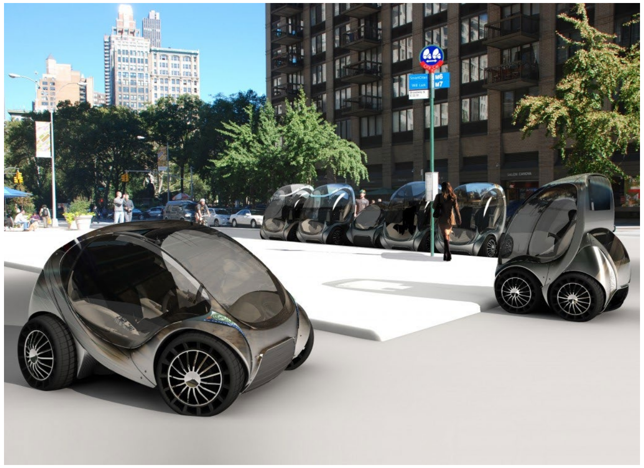
- Rational Agent:
  - For each possible percept sequence, select an action that is expected to maximize its performance measure…
  - given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rational Agents

- Rationality $\neq$ omniscience (all-knowing with infinite knowledge)
- Agents can perform actions that help them gather useful information (exploration)
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

Specifying Task Environment: PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- Consider, e.g., the task of designing an automated taxi driver:
    - Performance measure
    - Environment
    - Actuators
    - Sensors



Specifying Task Environment: PEAS
Automated Taxi

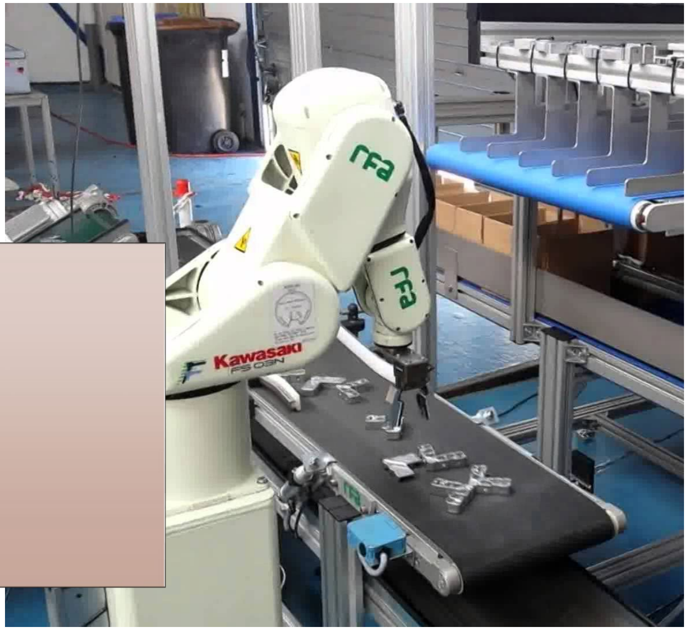
Performance Measure
- Safe
- Fast
- Legal
- Comfort
- Revenue

Environment
- Roads
- Other traffic
- Pedestrians
- Customers

Actuators
- Steering wheel
- Accelerator
- Brake
- Signal
- Horn

Sensors
- Camera
- Sonar
- Speedometer
- GPS
- Engine sensors

Specifying Task Environment: PEAS
Part Picking Robot



Performance Measure

- % parts in correct bins

Environment

- Conveyor belt
- parts
- bins

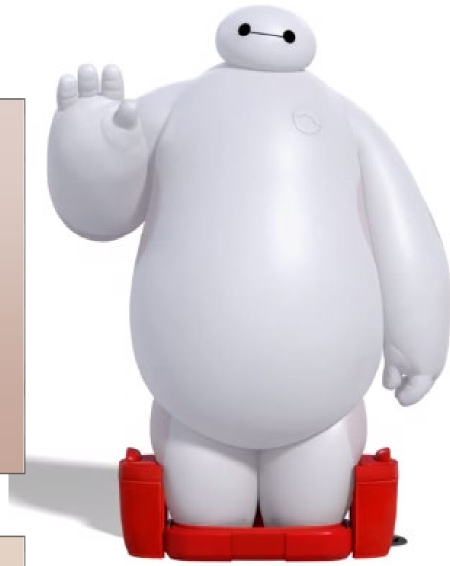
Actuators

- Jointed arm
- hand

Sensors

- Camera
- joint angle sensors

Specifying Task Environment: PEAS
Medical Diagnosis System



Performance measure

- Healthy patient
- cost
- lawsuits

Environment

- Patient
- hospital
- staff

Actuators

- Screen display (questions, tests, diagnoses, treatments, referrals)

Sensors

- Keyboard
- Medical Readings
- Medical History

Specifying Task Environment: PEAS
Interactive English Tutor

Performance measure
- Student's score on test

Environment
- Set of students
- Testing agency
- Chat platform

Actuators
- Screen display (exercises, suggestions, corrections)

Sensors
- Keyboard entry

Properties of Task Environments

Fully observable (vs. partially observable):

- sensors provide access to the complete state of the environment at each point in time.

Deterministic (vs. stochastic)

- The next state of the environment is completely determined by the current state and the action executed by the agent.

Episodic (vs. sequential)

- The choice of **current** action does not depend on actions in past episodes.
- Alternatively: **current** action does not affect future decisions

Properties of Task Environments

Static (vs. dynamic)

- The environment is unchanged while an agent is deliberating.

Discrete (vs. continuous)

- A finite number of distinct states, percepts, and actions.

Single agent (vs. multi-agent)

- An agent operating by itself in an environment.

Properties of Task Environments

Task Environment	Crossword puzzle	Part-picking robot	Taxi driving
Fully observable	Yes	No	No
Deterministic	Yes	No	No
Episodic	No	Yes	No
Static	Yes	No	No
Discrete	Yes	No	No
Single agent	Yes	Yes	No

Properties of task environment largely determine agent design. World is partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Agent Functions and Programs

- An agent is completely specified by the <u>agent function</u> mapping percept sequences to actions
- One agent function (or a small equivalence class) is <u>rational</u>
- Aim: Find a way to implement the rational agent function concisely

Agent Functions and Programs

Solution: write all possible percepts and optimal actions in a table.

All done!

Table-Lookup Agent

function TABLE-DRIVEN-AGENT(*percept*) **returns** *action*
  static: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, fully specified

  append *percept* to the end of *percepts*
  action ← LOOKUP(*percepts*, *table*)
  return *action*

- Drawbacks:
  - Huge table to store
  - Take a long time to build the table
  - No autonomy: impossible to learn all correct table entries from experience
  - No guidance on filling in the correct table entries

Agent Types

- Four basic types in order of increasing generality:
  - Simple reflex agent
  - Model-based reflex agent
  - Goal-based agent
  - Utility-based agent

Reflex Agent

- Passive: only acts when observes a percept
- Updates $state$ based on $percept$ only.
- Easy to implement

Motion sensor triggered.

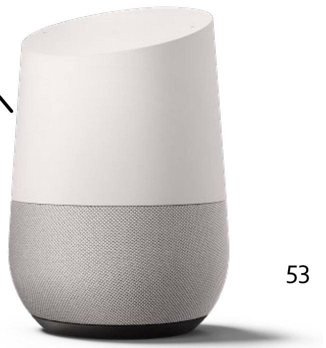
$$state \leftarrow WELCOME\ MODE$$

Model Based Reflex Agent

- Passive: only acts when observes a percept
- $state$ is updated based on percept, current $state$, most recent $action$, and model of world.

Motion sensor triggered. I am on ACTIVE MODE. I AUTHORIZED ACCESS. ⇒ HUMAN IN HOUSE

$$state \leftarrow WELCOME\ MODE$$



Goal Based Agent

- Has **goals**, acts to achieve them (not passive).
- $state$ is updated based on percept, current $state$, most recent $action$, and model of world.

Goal: make human happy (and buy products from my vendor)

Motion sensor triggered. I am on ACTIVE MODE. I AUTHORIZED ACCESS. $\Rightarrow$ HUMAN IN HOUSE

$$state \leftarrow WELCOME\ MODE(1)$$

Utility Based Agent

- Has **utility function**, acts to maximize it.
- $state$ is updated based on percept, current $state$, most recent $action$, and model of world.

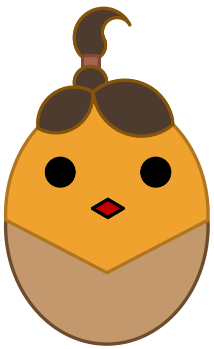
$$\max \alpha \times happiness + \beta \times purchases$$

Motion sensor triggered. I am on ACTIVE MODE. I AUTHORIZED ACCESS. ⇒ HUMAN IN HOUSE

$$state \leftarrow WELCOME\ MODE(17)$$

Learning Agent

Performance Standard



17 sodas is unacceptable!

Human did not like 17 sodas ☹

Update α, β configuration

Critic

Learner

Problem Generator

Maybe 170 sodas this time?

Performance Element

(think the utility agent architecture from the last slide)

Exploitation vs. Exploration

- An agent operating in the real world must often choose between:
  - maximizing its expected utility according to its current knowledge about the world; and
  - trying to learn more about the world since this may improve its future gains.

Exploitation vs. ***Exploration***