6CCS3AIN AI Reasoning & Decision-Making: Introduction

Week 1 – Part A

Peter McBurney
Department of Informatics
King's College London
London

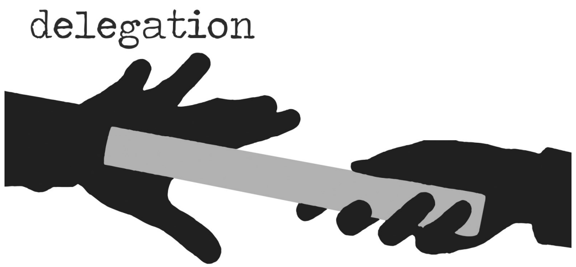
peter.mcburney@kcl.ac.uk

Week 1
2021

Computer Science is the science of delegation

Key question: *How do we delegate tasks to a machine?*

- **Hardware engineering**: How to design machines able to effectively & efficiently undertake the tasks we delegate to them?
- **Programming theory**: How do we best write instructions for machines to do what we what we want them to do?
- **Programming language theory**: How we design programming languages in which to write instructions for machines?
- **Software engineering**: How do we organize human programmers so that they can jointly delegate tasks to a single machine?
- **Distributed computing**: How do we organize machines so that a set of instructions can be executed jointly by multiple machines?
- **Verification & testing**: How can we check the instructions before we give them to a machine?



What is AI ?

- Artificial Intelligence is the design of machines (hardware or software) able to delegate tasks to other machines.
- Delegation normally requires thinking and planning, and sometimes doing.
- Artificial Intelligence is the methodology for careful and rigorous thinking about ways of knowing and ways of acting
  - After Seymour Papert, 1988 (image below).
- Since the 1950s, AI has developed a variety of methods:

  Search

  Statistical classification & neural networks

  Logic programming & argumentation

  Uncertainty formalisms & Bayesnets

  Multi-agent systems

  Machine Learning/ Deep Learning.

2020s: AI celebrates its first Millenium

- First search algorithm dates from 1020s
- Developed by Ibn Sina (aka Avicenna, c. 980 – 1037)
- Example of a syllogism:

  Premise 1: All men are mortal.

  Premise 2: Socrates is a man.

  Conclusion: Therefore, Socrates is mortal.

- Avicenna proposed a method for finding a proof of a Conclusion in a syllogism when some premises of the syllogism were missing, and drawing on a database of possible premises.

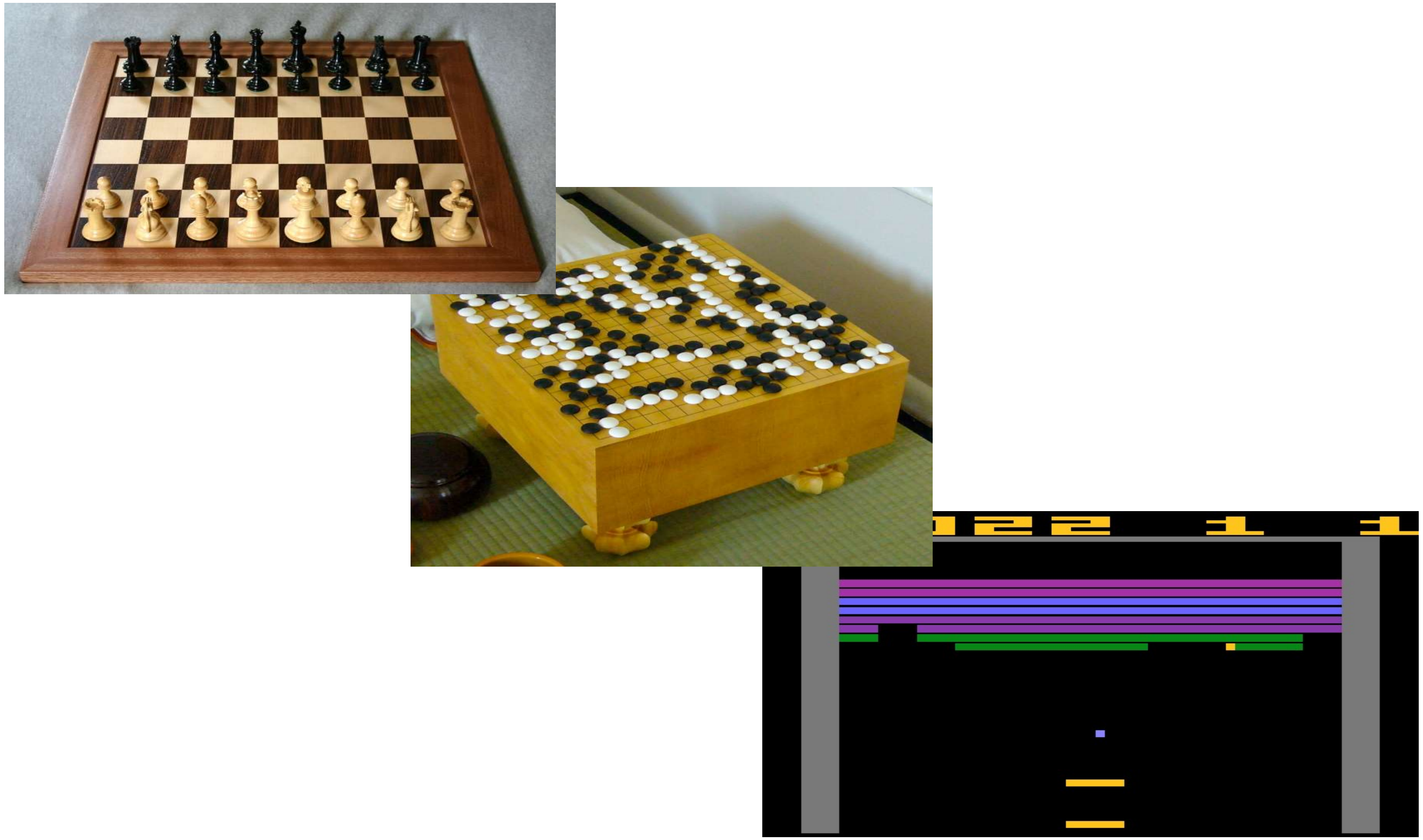
Source:
Wilfrid Hodges [2010]:
"Ibn Sina on analysis: 1. Proof search. or: abstract state machines as a tool for history of logic."
pp. 354-404, in: A. Blass, N. Dershowitz & W. Reisig (Editors):
Fields of Logic and Computation. LNCS volume 6300. Berlin, Germany: Springer.

Data-driven vs. Model-driven AI

- Why has AI suddenly become popular recently?
  - Large increase in data with the World-Wide-Web
  - Significant increases in computer processing power
  - New techniques in machine learning / deep learning

- Data-driven approaches vs. model-driven approaches
  - Machine Learning/ Deep Learning are usually data-driven
    - Patterns are found with no explanation as to why or what these mean
  - In model-driven approaches, the AI system has a model of the application domain
    - For example, a causal model connecting causes with effects.

AI and games: Chess, GO and Atari Breakout



Three strategies for designing AI

- Chess: Deep Blue (IBM) 1996
  - Software programmed with the rules of Chess and heuristics (rules of thumb) for strategy
  - By playing many games, it created experience of the value of different heuristics in different situations

- Go: Alpha Go (Deep Mind/Google) 2015
  - Software programmed with the rules of GO
  - Through playing many games (including with itself), it learnt the heuristics itself.

- Atari Video Game called Breakout (Deep Mind) 2013
  - Software not even given the rules
  - Software learnt the rules by observing human players
  - Software learnt the best strategies by playing many games.

- Note: In all three games, simulation models were used to generate data.

Complete vs Partial Information

- Note that traditional Chess is a game of Complete Information
  - Each player knows the rules, they see the moves of themselves and of the other player, and they see what happens after each move (for example, they see if a player loses a piece). They know the current status and history of the game, and they both see the results.

- Many games are games of Partial or Imperfect Information.
  - In Invisible Chess, players cannot see where all the pieces move to, for example, perhaps 2 pieces are hidden.
  - You only know where a hidden piece is sitting when one of your other pieces tries to land on the same square.
  - This game is studied as a model for submarine warfare.

- Most real-life applications of game theory involve some element of partial information.
  - In wartime, there are many uncertainties and the results of one's actions may not be known for a long time, or perhaps never.

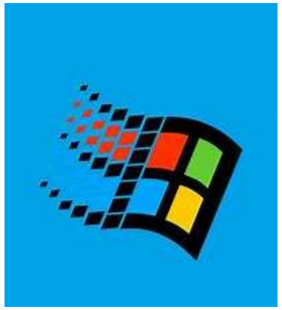
AI now has many different threads

AI has developed various techniques & approaches since 1950

- Search
- AI Planning
- Statistical classification & neural networks
- Logic programming & argumentation
- Uncertainty formalisms & Bayesian Belief Networks (Bayesnets)
- Autonomous Agents and Multi-agent systems
- Machine Learning & Deep Learning
- XAI

- Classification of AI systems by the USA Defence Advanced Projects Research Agency (DARPA):
  - Reasoning Systems (eg, using logic or argumentation)
  - Statistical Learning Systems (eg, neural nets, machine learning/deep learning)
  - Contextual and Adaptive Systems.

Some applications of AI



- Medical Expert Systems since the 1970s
  - These aimed to encode the expertise of human experts (eg, cancer specialists)
  - A single expert system could encode the expertise of many human experts
  - An expert system would typically perform better than any single human.
- Microsoft Windows Operating Systems have had some embedded AI since Windows95 (released in 1995)
  - Windows95 used an AI system to diagnose printer faults
  - The human user answered questions about the symptoms of the fault
  - An internal Bayesnet reasoned backwards to try to find the cause of the fault.
- Self-driving vehicles
  - Autonomous aircraft (drones)
  - Autonomous road vehicles
  - Autonomous ships.

Some current research challenges

- How to automatically generate explanations of AI systems?
  - Most machine learning & deep learning systems do not explain themselves
  - Under European financial regulations, human-understandable explanations are now required for any AI systems that significantly impact individuals and small/medium enterprises
  - Research area is called XAI (Explainable AI)
- How to make machine learning & deep learning systems robust?
  - Easily tricked by adversaries
  - Outcomes very sensitive to bias in data and edge cases
- How can machines assess human intentions?
  - Hard for humans to do, since we may even know our own intentions.
  - For example, automated negotiation systems need to assess how likely are the participants to keep promises that they make.
- How to combine the best of data-driven and model-driven approaches?
- How to successfully combine humans and AI systems?
  - Eg, one human controlling multiple AI systems
  - Multiple humans controlling one AI system
  - Vice versa.

In this course

In this course, we will focus on some techniques for reasoning about knowledge and action:

- Probability Theory
- Models using probability, such as Stochastic Processes
- Expected Utility Decision Theory
- Game Theory
- Argumentation Frameworks (formal models of argument)
- Models of collective decision-making (eg, in swarms)
- Ethical and legal considerations in AI, and AI Governance.

Intended Learning Outcomes

The Aim of this course is to provide a grounding in artificial intelligence techniques which are used:

- To represent knowledge about the world
- To make decisions on the basis of that information, and
- To update what is known about the world.

Learning Outcomes:

- AIN1: Demonstrate a sound knowledge and understanding of advanced techniques in artificial intelligence.
- AIN2: Judiciously apply these techniques to a range of subject-specific problems.
- AIN3: Implement these techniques in computer software.

Theoretical or Practical?

Source: Wikimedia

- Is this a theoretical or a practical course?

The answer is: BOTH.

- Computer Science, unlike the physical sciences,

is mostly the study of man-made objects.

- It is therefore common for practical implementations to be created before the theory is developed. For instance:
  - We had programmable textile looms from the 1720s, but no theory of programming until the 1960s. *(Image of Jacquard Loom.)*
  - We had working calculating machines by the 1840s, but no theory of computation until 1936 (Turing Machines).
  - We had an application of Blockchain - decentralized cryptocurrencies (Bitcoin) - in 2009, but we are still working on the theory of these.
- Practice often comes before Theory in Computer Science.

We need both theory and practice, and each helps the other.

Textbook Reference (Russell & Norvig)

- This week we cover material from:

  "Artificial Intelligence, A Modern Approach"
  (AIMA) by Russell & Norvig

  Chapter 2, from beginning to end of Section 2.4.5.

- If you have access to this book, it is worth reading the whole of Chapter 2.

- You may also find Chapter 1, which gives a brief history of Artificial Intelligence, interesting also.
  - Although the last part on the availability of very large datasets, is a bit out of date now.

Intelligent Autonomous Agents

An agent is a software system that:

- is situated in an environment,
- is capable of perceiving some or all of its environment, and
- is capable of acting in its environment

with the goal of satisfying its objectives.

According to Wooldridge & Jennings (1995), agents are

- Proactive (they try to achieve specific goals)
- Reactive (they respond to their perceptions of their environment)
- Social (they are aware of other entities in their environment)
- Autonomous (to a greater or lesser extent).

How to think about agents

"Agents are robots without any of the hardware."

- In programming, when you invoke an **object** it always does what it is programmed to do. (It might have a bug, so it may not necessarily do what the programmer intended it to do.)

- When invoked, an agent may or may not do what it is programmed to do, because of its autonomy.

"Agents are objects with attitude!"

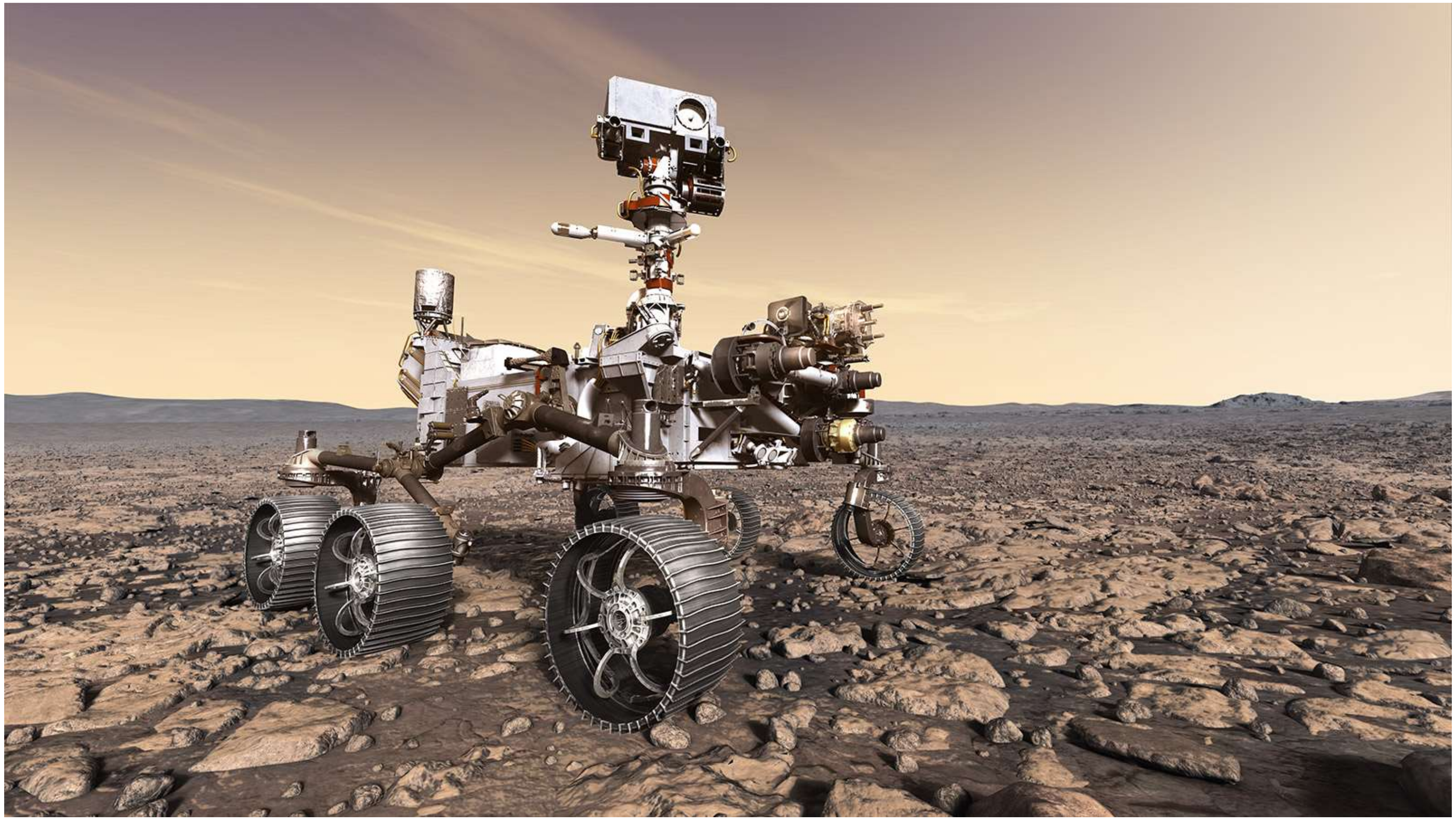
Questions to ponder

- Is a cat an agent?
- Is a dog an agent?
- Is a light-switch an agent?

Intelligent agents

- For intelligent agents, we talk about:
  - Its environment (where it is situated)
  - Its sensors (how it perceives the environment)
  - Its effectors or actuators (how it acts upon the environment)
- For a human agent:
  - Environment: the physical world
  - Sensors: eyes, ears, tastebuds, skin, etc, . . .
  - Effectors: hands, fingers, arms, elbows, head, legs, feet, voice, . . .
- For a self-driving car:
  - Environment: physical world on and around streets
  - Sensors: GPS, video cameras, microphones, touch sensors, lidar, . . .
  - Effectors: wheels, horn, lights, windscreen washers.

NASA Mars Rover



Source: NASA

Self-driving road vehicles

- Integrating real-time data feeds from different sensors
  - Vision, sound, lidar (all directions)
  - GPS data
  - Estimating speed and directions of other vehicles or people
  - Inferring intentions of others
- Information arriving very quickly and may be contradictory
- Encoding road laws & learning norms of behaviour
  - Norms differ from place to place
  - Rules may need to be broken
- Need to take decisions quickly
- Key risks:
  - Ethical trade-offs may be necessary
  - If there is legal action, there will be a need to explain these trade-offs.

Photo credit: Google 2015

How to design the controller for a self-driving car?

- We want the vehicle to achieve a certain goal
  - Eg, travel from point A to point B. (called “A-to-B-ing”)

- What should the car do at each point along the way?

- One common view of decision-making:
  - For each decision, identify all the possible alternative options.
  - Compare these options by some metric
  - Decide which option is the best option
  - Execute that option
  - Do this fast enough to be feasible (very fast for a self-driving vehicle).

Constraints on decisions

- Option Uncertainty
  - We may not have information on the possible alternative options
  - We may not know the potential consequences of different options
  - We may not know our own preferences
- Incommensurability
  - We may not be able to compare alternative options
  - Eg, Is it better to take a gap year or to head straight to university?
- Outcome uncertainty
  - We may not know what happens after we execute our selected option
- Resource constraints
  - Some problems are computationally very hard
  - “The magical number 7 plus or minus 2” (Miller’s Law)
- Time constraints
  - We usually have a fixed time (often very short) to make a decision.

Rational agents

- When computer scientists talk about “rational” agents, they usually mean agents acting in the real world under resource constraints.

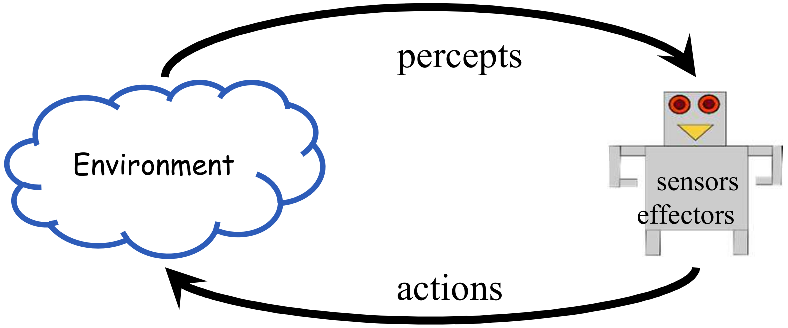
- Economists, on the other hand, usually mean an agent unencumbered with such constraints
  - With perfect knowledge of the present
  - Infinite memory of the past
  - Perfect foresight of the future
  - Infinite processing capabilities, etc.

- See the economic concept (see Wikipedia): *Homo Economicus.*

Do the right thing?

- The key problem we have is knowing the right thing to do.
- Knowing what to do can in principle be easy:
  - Consider all the alternatives and choose the "best".
- For example: See the Wikipedia entry on **Trolley Problems**.
- Why isn't this easy in practice?
- Need to be able to:
  - identify the alternatives,
  - decide which alternative will lead to the best outcome,
  - and do so quickly enough for the decision to be useful!

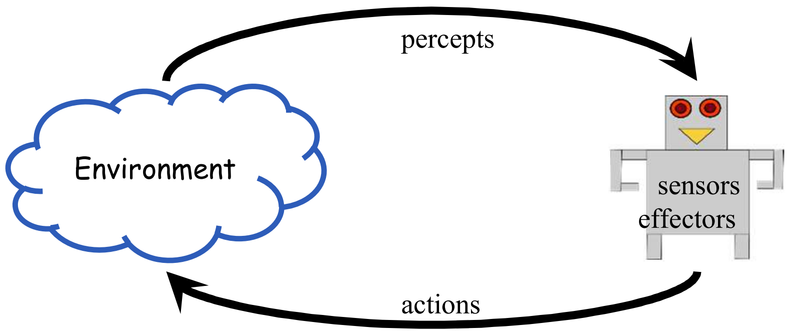
Ideal rational agent



For each sequence of percepts, an ideal rational agent will act to maximise its expected performance measure, on the basis of information provided by the sequence of percepts, together with any information built into the agent.

Note this does not preclude performing actions to find things out.

Ideal rational agent

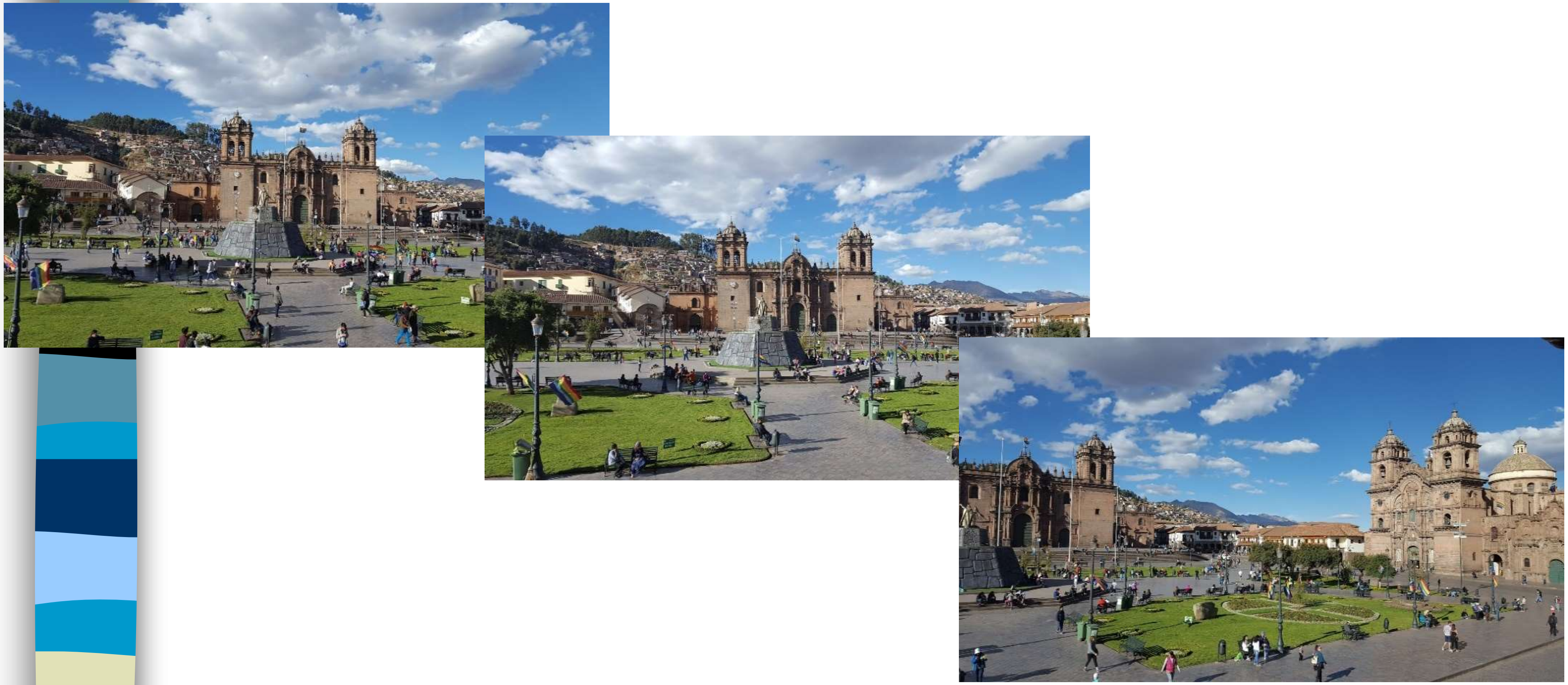


- Rational agent doesn't mean omniscient or clairvoyant agent.
- Rational agent doesn't mean perfect agent.

I'm walking along the Strand and see an old friend. There's no traffic near by and the green light is showing to allow me to cross the street. I don't have anywhere else I need to be, so being rational I start to cross the road.

Meanwhile, high above, a cargo door falls off a passing airliner, and before I reach my friend I am flattened. Was I irrational to cross the street?

A sequence of percepts

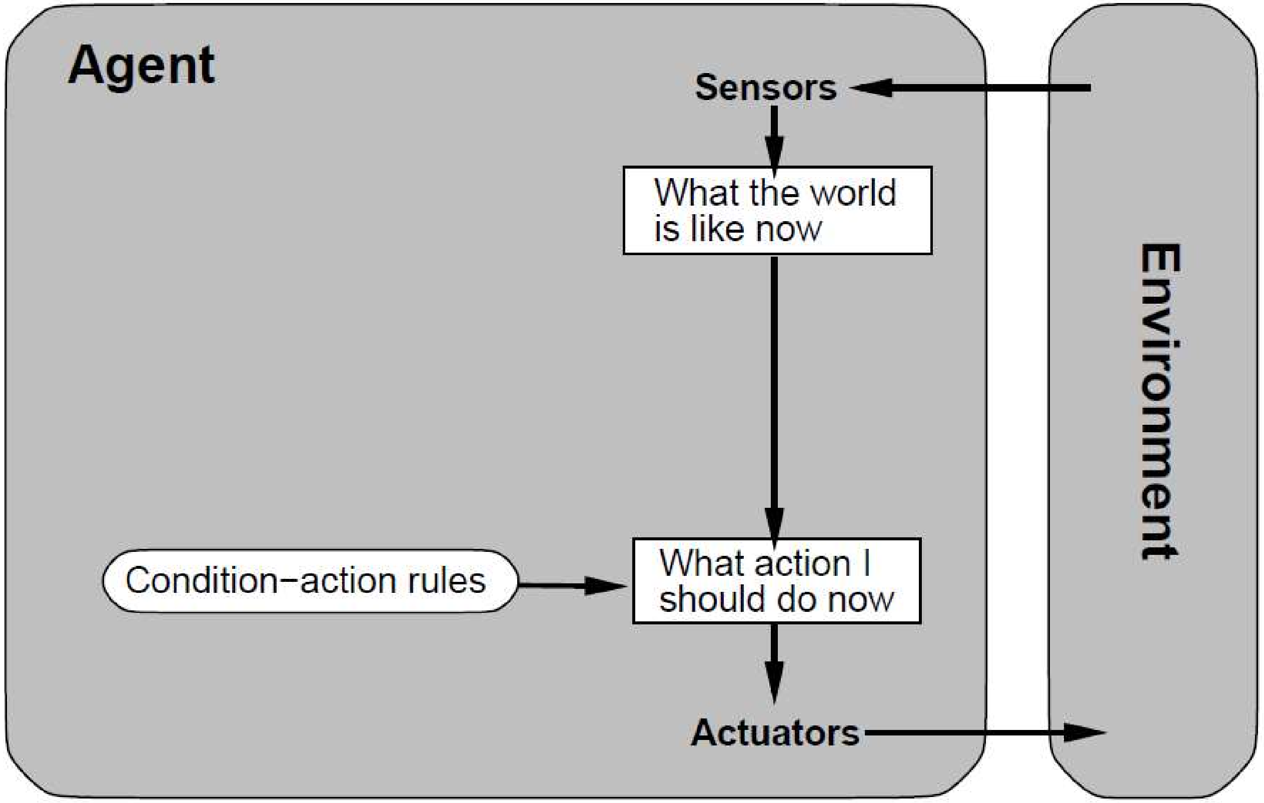


Challenges of knowledge representation

Some challenges:

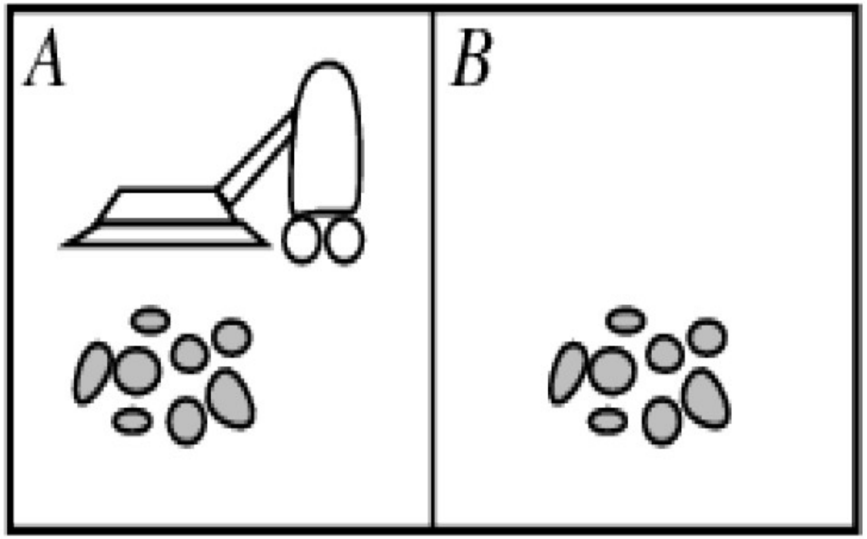
- What to represent & how to represent it
- Conflicts in incoming information (eg, audio vs image)
- Interpreting the percepts.

Simple reflex action



A simple agent maps each percept directly to an action.

Vacuum world: We have a simple vacuum cleaner



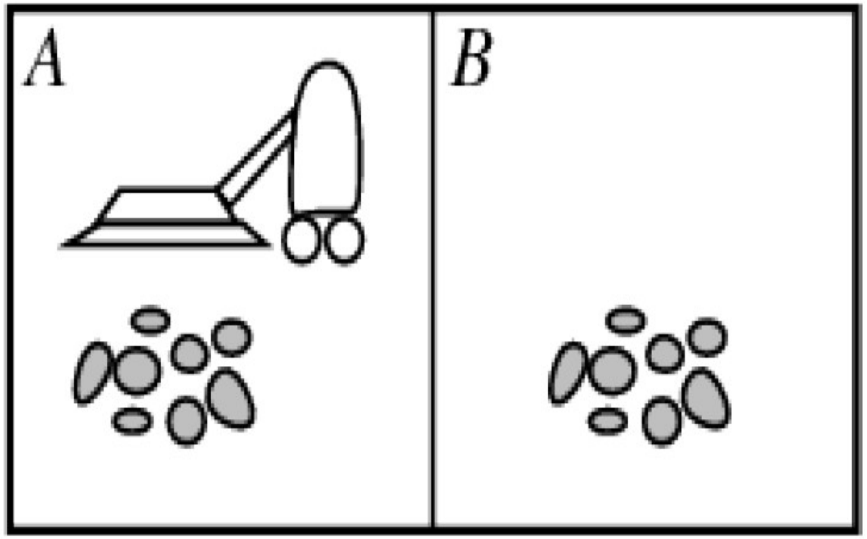
- 2 Sensors:
  - Location? (A or B)
  - Dirty? (Yes or No)
- 3 Actions:
  - Suck up dirt
  - Move right
  - Move left.

Exercise: Can you make a list of rules of the form

if condition then action

to control this simple vacuum cleaning agent.

Vacuum world: If-then rules for the vacuum cleaner



- 2 Sensors:
  - Location? (A or B)
  - Dirty? (Yes or No)

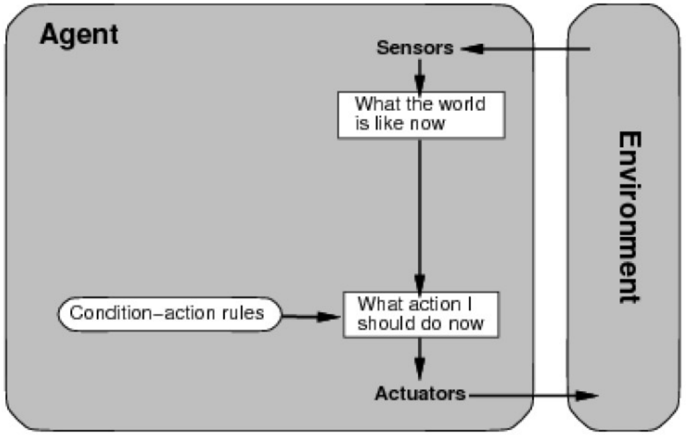
- 3 Actions:
  - Suck-up-dirt
  - Move-Right
  - Move-Left

if *Dirty = Yes* then *Suck-up-dirt*

else if *Location = A* then *Move-Right*

else if *Location = B* then *Move-Left*

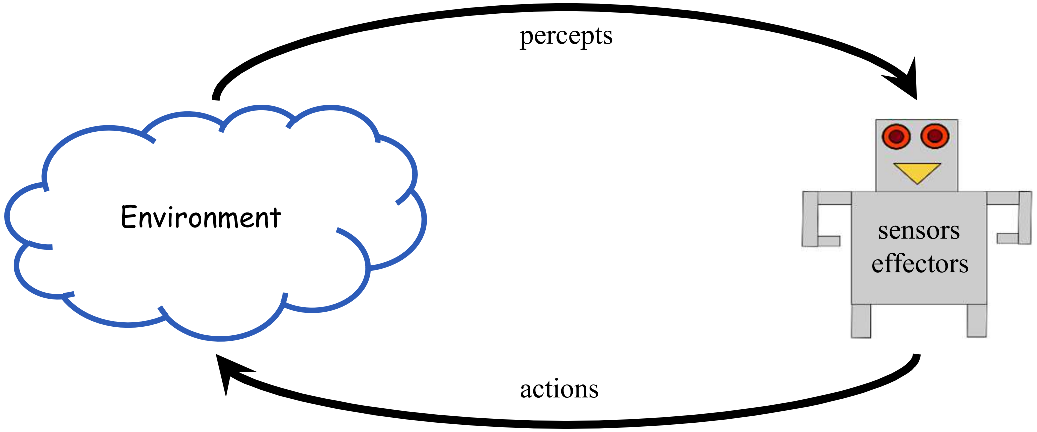
Simple reflex agent



```
function Simple-Reflex-Agent(percept) returns an action
        static: rules, a set of condition-action rules

        state ← Interpret-Input(percept)
        rule ← Rule-Match(state, rules)
        action ← rule.action
        return action
```

Most of the time things are not so simple



Accessible environments

Can the agent detect all aspects of the state that are relevant to deciding what to do?

An environment is accessible if the agent can obtain all accurate and up-to-date information about all the relevant aspects.

Deterministic and Stochastic Environments

Can the agent be certain what state will result from its actions?

A deterministic environment is one in which any action has a single guaranteed effect – there’s no uncertainty about the state that will result from performing an action.

An environment where we can quantify the non-determinism with probability is called stochastic.

Episodic and Sequential Environments

Can the agent’s decision about what to do affect future decisions?

In an episodic environment, performance of an agent is dependent on a number of discrete episodes, with no link between the performance of an agent in different episodes.

In a non-episodic or sequential environment, the current decision affects future decisions.

Dynamic vs Static Environments

Can the agent be sure that its environment will only change as a result of its own actions?

A dynamic environment has other processes operating on or in it, and hence may change in ways beyond agent's control.

A static environment remains unchanged except by the performance of the agent's own actions.

Discrete vs Continuous Environments

Does the agent's environment have a fixed, finite number of actions and percepts in it?

If yes, then it is discrete.

Otherwise it is continuous.

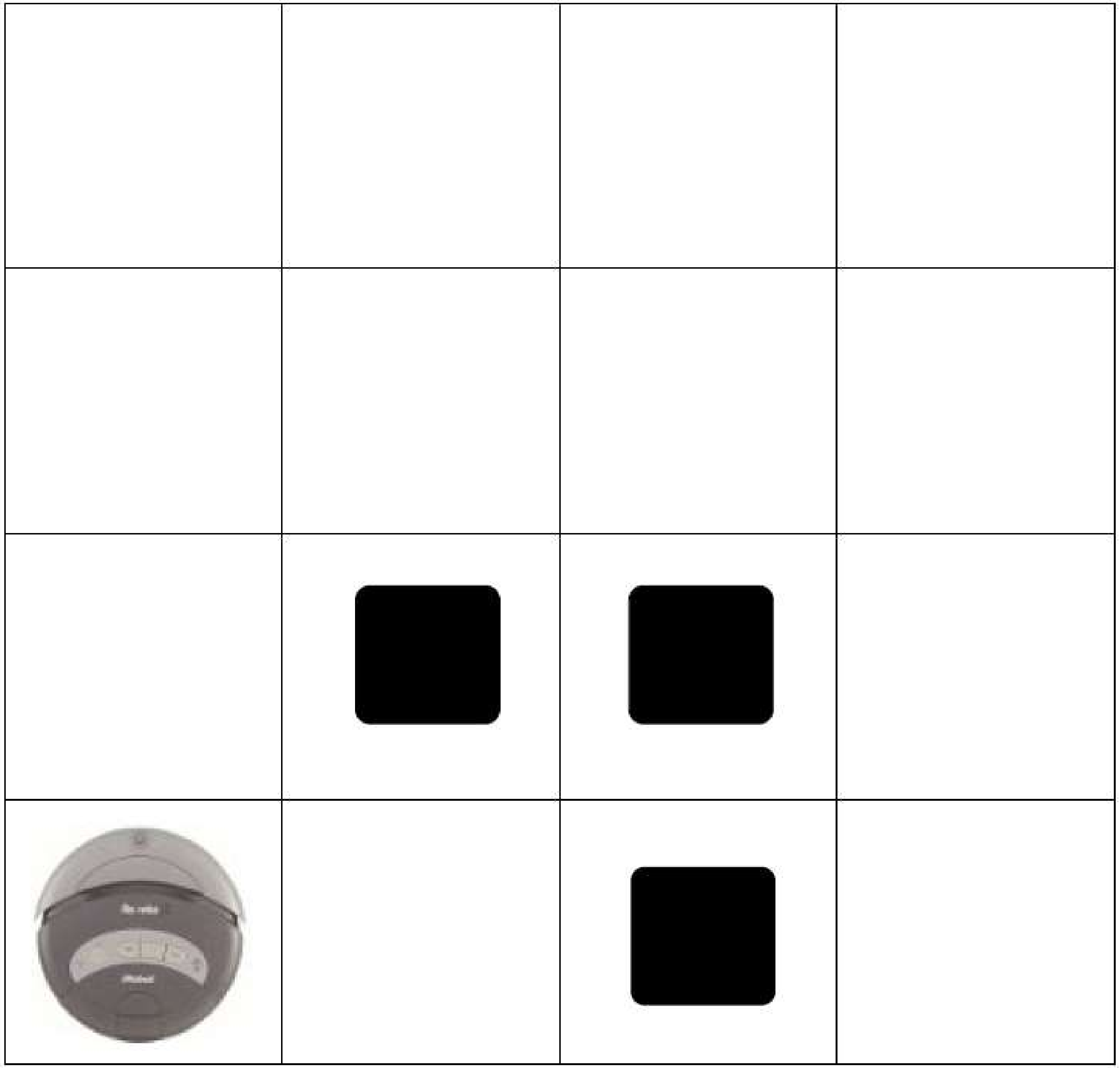
A run of an agent in an environment

When an agent acts in an environment it generates a run.

A run is a sequence of interleaved states and actions

s0, a1, s1, a2, s2, a3, s3, ….

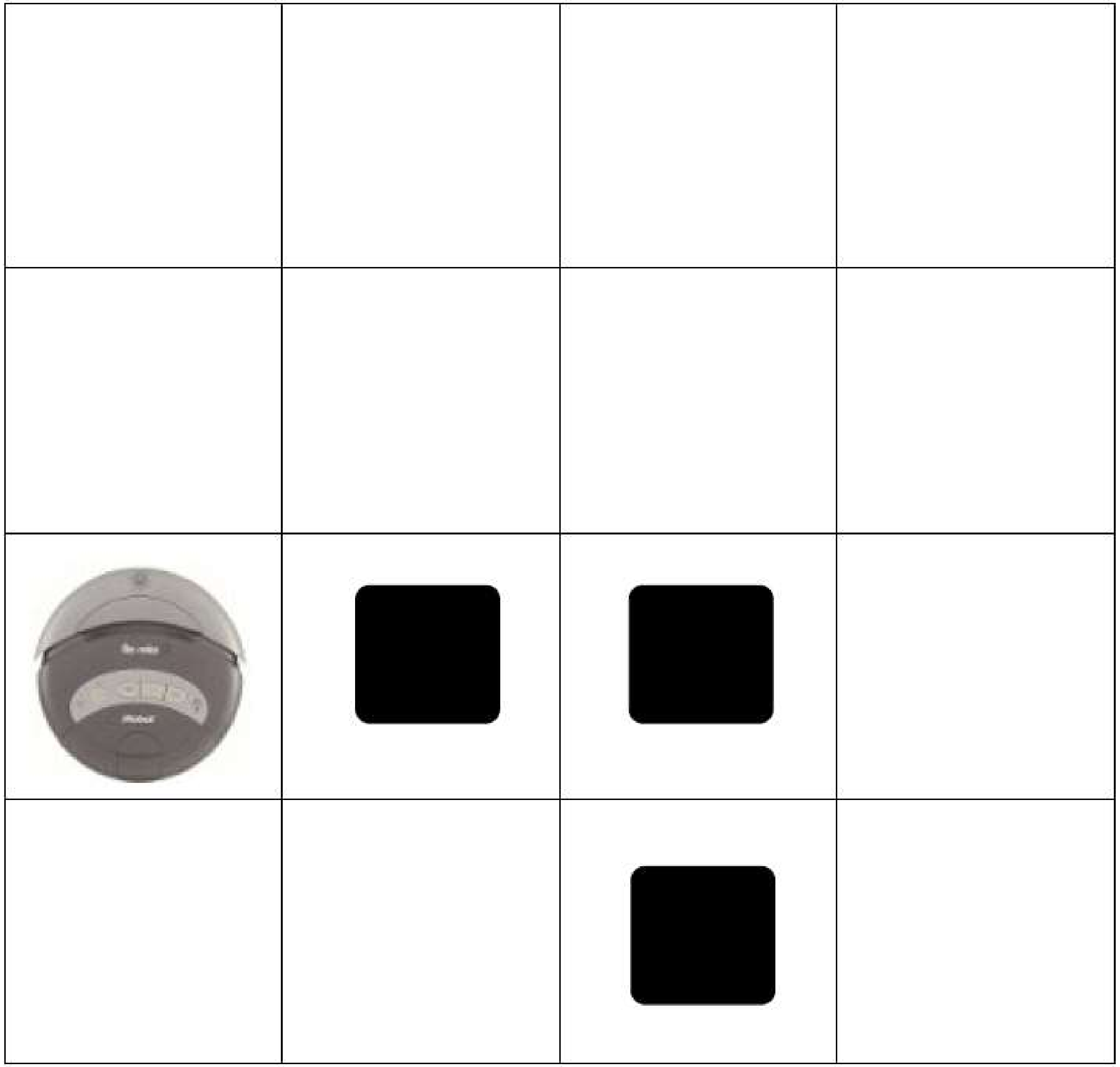
An agent in an environment – in bottom left square



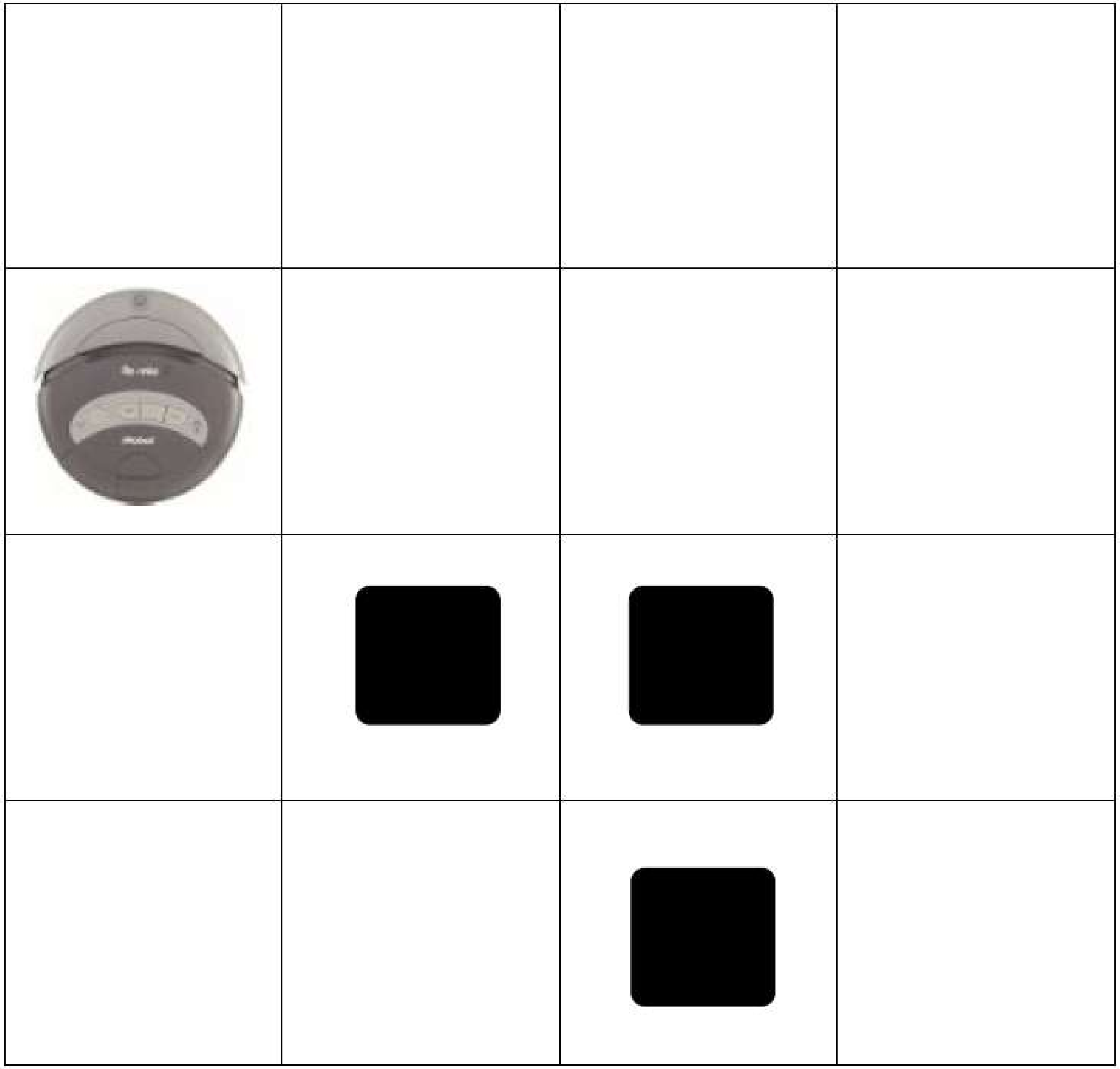
Agent

Obstacles
in 3 squares

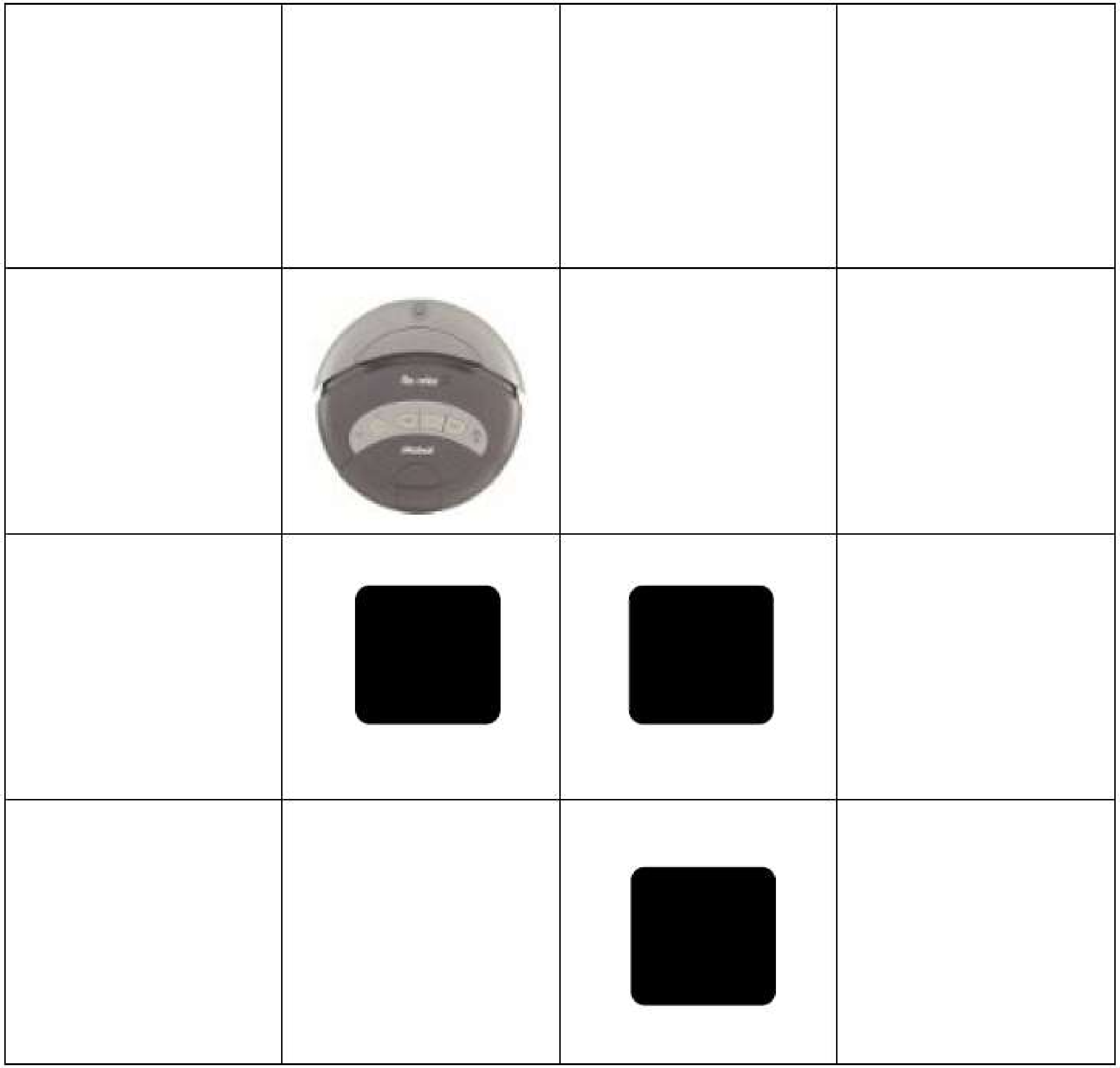
The agent moves one square North



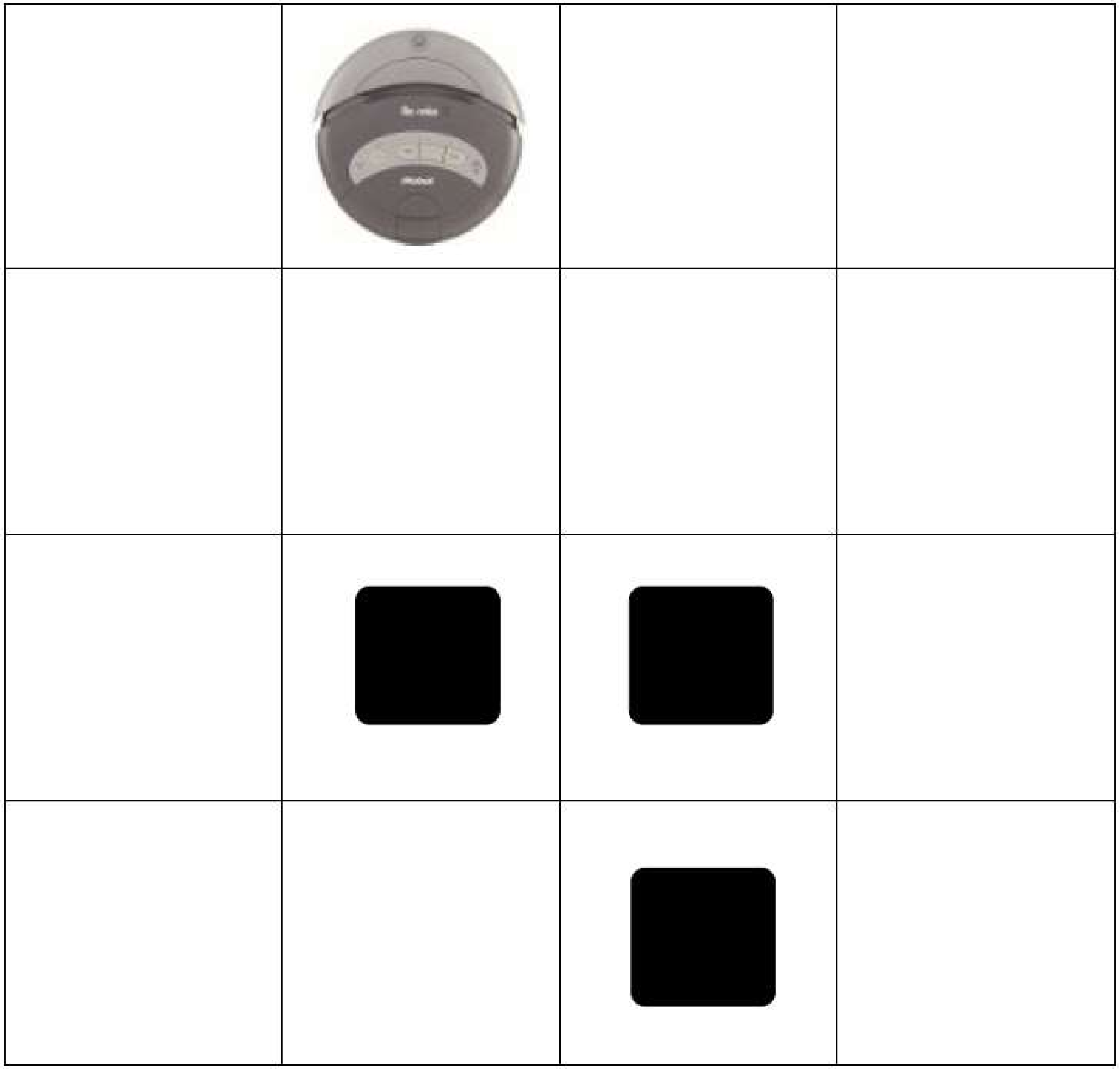
The agent moves one square North again



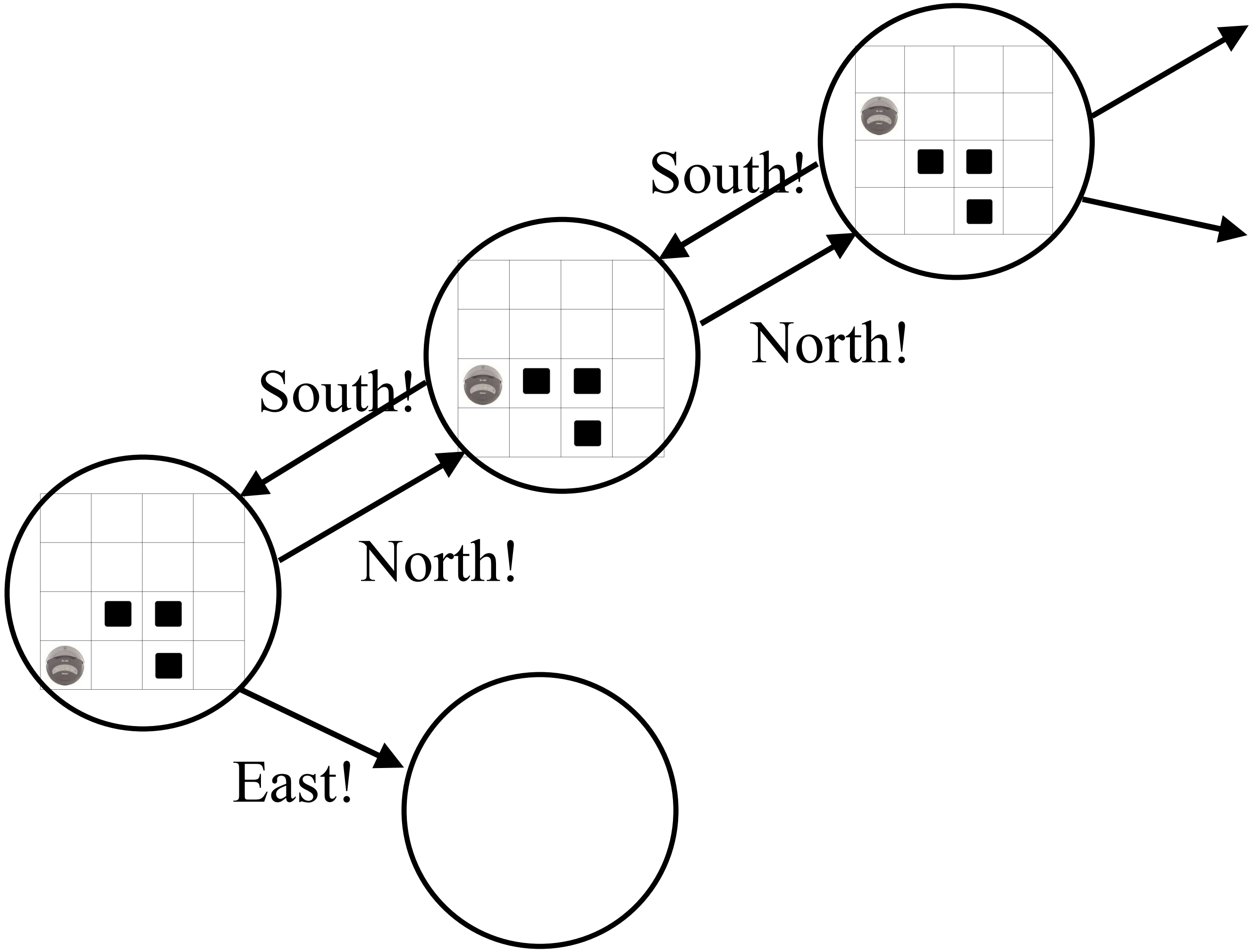
The agent moves one square East



The agent moves one square North

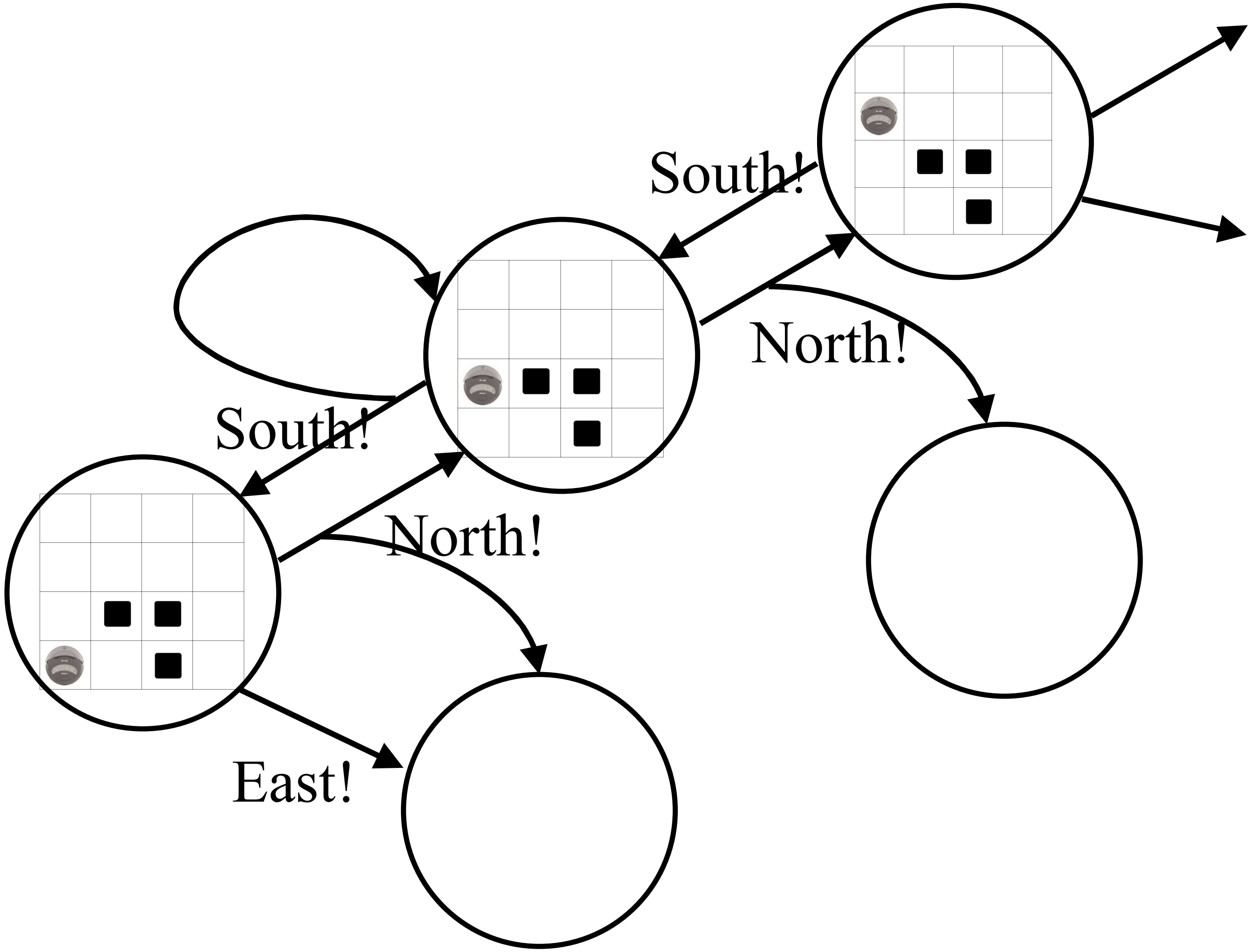


We can visualize the space of possible runs from start state



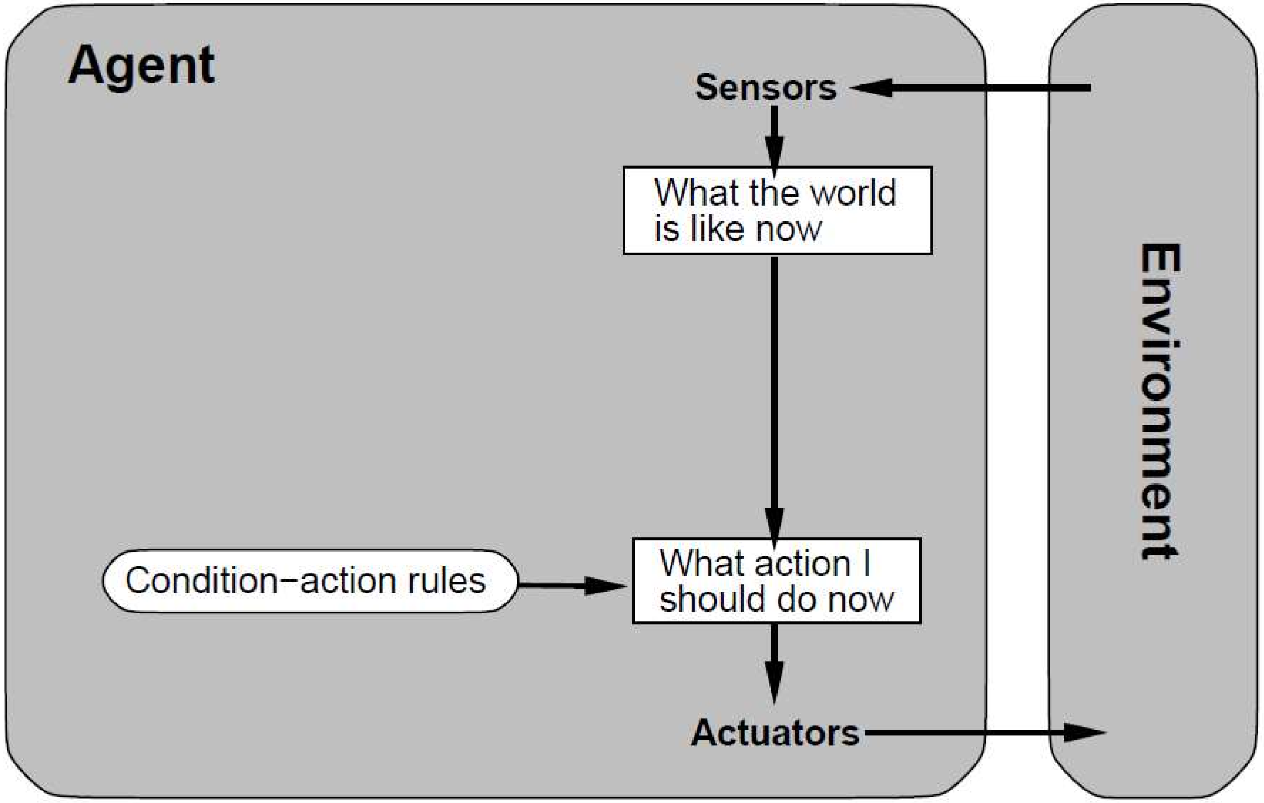
Exercise: What does the state space look like if the first move by the agent was to go one square East?

Runs in more complex environments



If the Environment is non-deterministic or dynamic, then the set of possible runs is more complex.

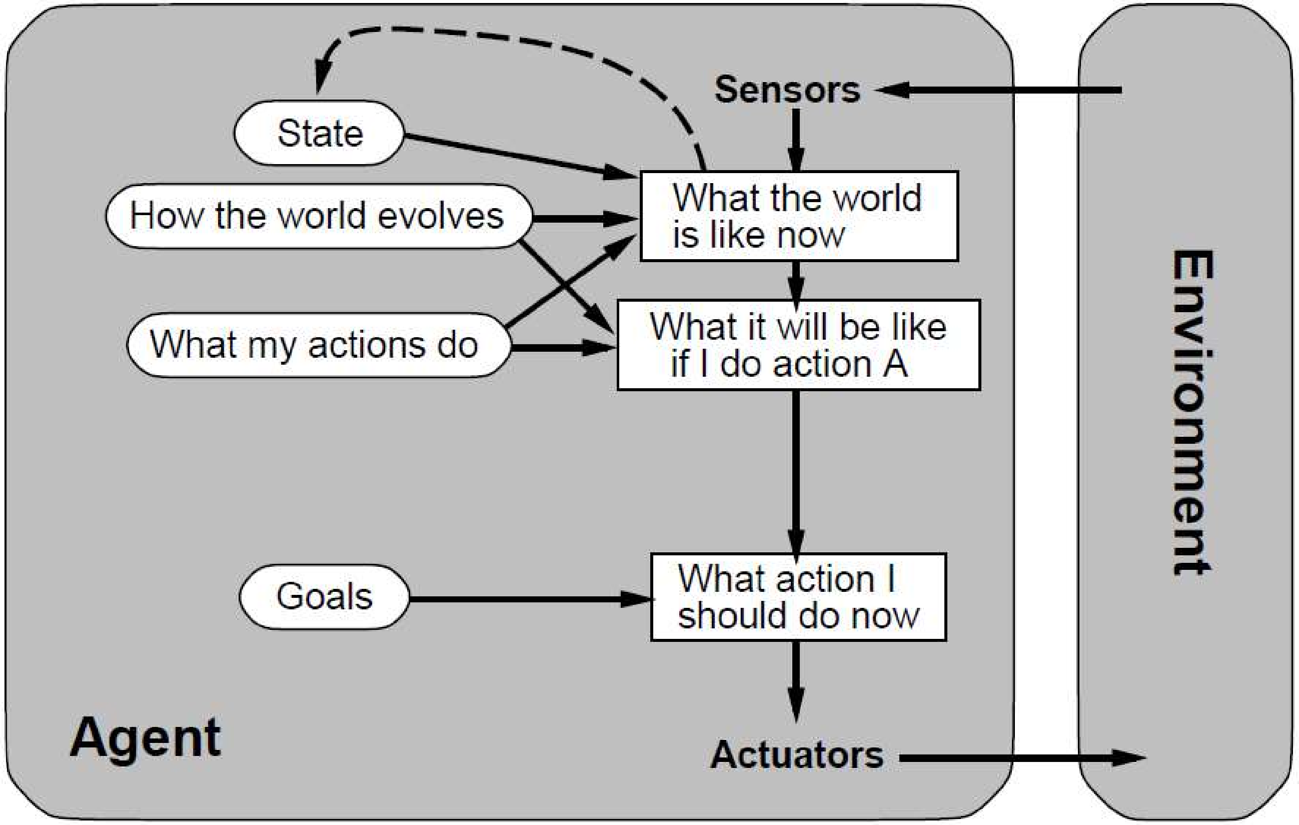
When simple agents aren’t enough



A simple agent maps percepts directly to actions.

This means we have to be able to make a correct decision based only on the current percept.

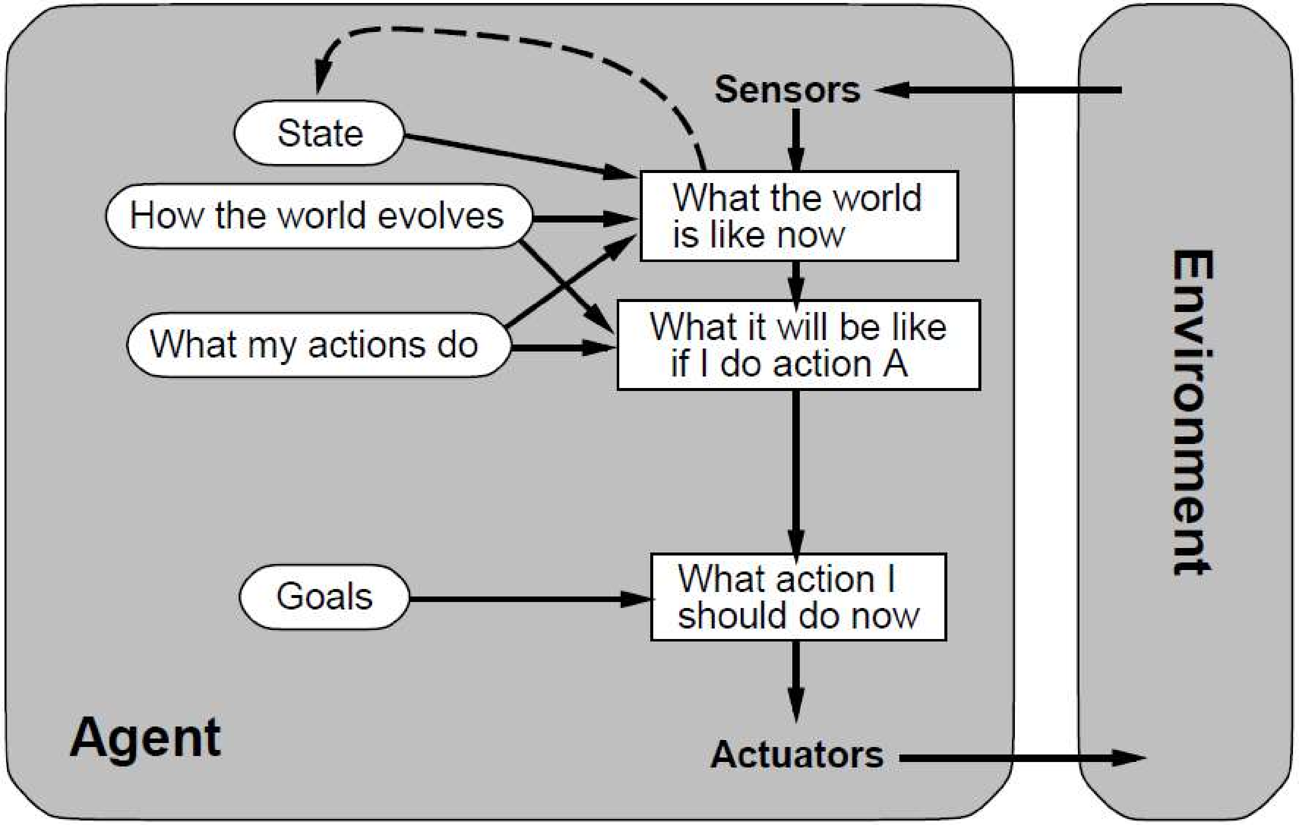
Agents with state and goals


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

New percept interpreted in light of existing knowledge about the state.

Knowledge about the world used to model inaccessible parts of the environment.

Agents with state and goals


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Knowledge about the world (the Environment) is used to predict the state resulting from an action.

Goal information used to identify desirable states.

Agents and goals

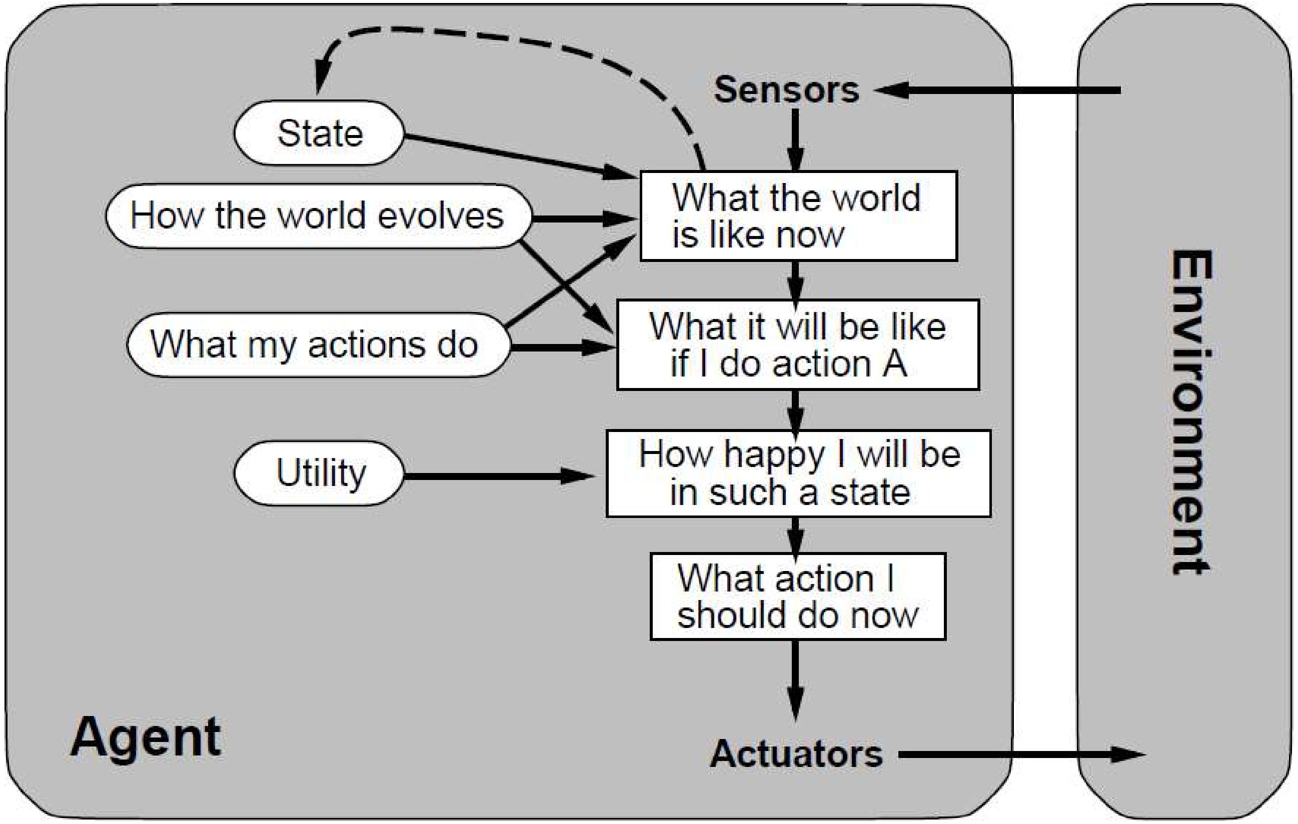
Suppose an Agent has a well defined set of goals.

- Is it easy for the agent to decide what to do?

- No – goals may conflict.

- And, even without conflict of goals, there may be lots of different means to achieve the same goal.
  - These different means may have different costs & benefits, and different consequences.
  - They may have different likelihoods of being successful.

Utility-based agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Utility is a quantitative value that measures how “good” a state is.

Utilities

- The **utility** of a state is a measure of the benefit or value of that state to a particular agent.
  - This notion is from Economics.

- If the state is certain to come about, and if we can calculate the value of the state to an agent, then we can determine the utility of the state.
  - utility(s1) for state s1

- In reality, it usually is not easy to calculate the quantitative value of utility.

Expected Utilities

- Most states are not certain to come about, but may only occur with some probability. In that case, if we can calculate the utility of the state (say S), and the probability that it will come about, we can calculate the **Expected Utility** of the state:

- Expected utility (s1) = E [U(s1)]
  = probability (s1) * utility(s1)
  = Prob(s1) * utility(s1)

If we have more than one state, we assume we can add up the pairs of probabilities multiplied by the utilities, as follows:

For S a set of states,

and where Σ (s ε S) indicates sum over all states in S

$$\text{E [U(}\mathrm{S}\text{)]} = \Sigma \text{ (s } \varepsilon \text{ } \mathrm{S}\text{)} \quad \text{Prob(state=s)*utility(s)}$$

Challenges with utilities

- Summing up over all states might be very costly in terms of processing resource or time, due to the number of states involved.

- Determining the utility of a state is not necessarily straightforward!

- Difficult to take a long term view.
  - In chess this is easy, because we get the reward only at the end, but in other settings this might not be easy.

Conclusions

- We have considered AI as the task of creating computer programs able to delegate to other computer programs.
- In the last 70 years, AI has developed many techniques, and Machine Learning is just one of these.
- In this lecture we have considered the notion of an intelligent agent, and looked at several different types of agent environments.
- We considered the notion (from Economics) of the utility (or value) of a state, and this idea will play an important part in the course.
- During this week, you should read and take notes on Chapter 2 of the textbook by Russell and Norvig.

Thankyou!

6CCS3AIN AI Reasoning & Decision-Making
Course Information
Lecture 1 – Part B

Peter McBurney
Department of Informatics
King’s College London
London

peter.mcburney@kcl.ac.uk

Week 1

Please wear a mask

(while indoors, unless you are exempt)

Are you vaccinated?
Have you had a test today?

Tests and vaccinations are
available on King's campuses.

Instructors

Dr Frederik Mallmann-Trenn
(he/him/his)
frederik.mallmann-trenn@kcl.ac.uk
Office Hours: Tuesdays 10.30 – 12.30
Weeks 2-7.

Professor Peter McBurney
(he/him/his)
peter.mcburney@kcl.ac.uk
Office Hours: Fridays 09:00-11:00
Weeks 1 and 8-10.

Teaching Assistants 2021

- Zixuan Dai
- Cristina Gava
- Charles Higgins
- Mackenzie Jorgensen
- Sophia Kalanovska
- Lin Li

Topics to be covered

Topics include:

- Knowledge representation
- Reasoning with uncertainty
- Decision making (simple, complex and multi-party)
- Game Theory
- Learning
- Argumentation
- Collective decision-making
- Ethics

Textbooks

- See the KEATS module page for information about the recommend textbooks.

Self-directed learning

This is 15-credit module, expecting 150-learning hours:

- Pre-recorded Lectures: teach concepts, processes, techniques, and show examples (20hrs)
- Labs / Tutorials: teach skills and expose you to examples (20hrs)
- Coursework: assesses your competence in these skills (20hrs)
- Self-directed learning: *90hrs*

Assessment

There are two components to the assessment this year:

- Final exam: 80%
- Practical coursework: 20%

Coursework

Practical Coursework (worth 20%)

- There will be 1 coursework exercise to complete.
- It will use Python (we expect you to learn it outside of the labs)
- Value of coursework: 20% (compulsory)
- Due date: TBD

- Exercises will be posted on KEATS.
- Watch for updates (and check version numbers).
- We will check for plagiarism in your code.

Small-Group Tutorials (SGTs)

- These cover questions to help you understand the material.
  (And some practice for the exam)
- Will often be open-ended, no-right-answer questions.
- Questions will be posted before the week of the Tutorial, and we will expect you to have worked on the questions before the tutorial.
- Answers will be posted after the week of the tutorial.
- Watch for updates on Keats (and check version numbers).

- For a topic discussed in lectures in Week k, the small-group tutorials will cover this material in Week k+1, and the answers to the tutorial questions will be released during Week k+2.
- Tutorial questions and answers are in the Keats section for Week k (ie, the week of the topic to which those questions and answers refer).

Labs / Practicals

- For the lab work, you will write code for Pacman to make decisions about what to do.
- You will explore steadily more complex environments.
- And by "more complex" we mean "harder"
- These will build towards the coursework.

- As in 2020, we will hold weekly online sessions (coursework clinics) where you can ask the TAs questions about the code and the coursework.

What we expect from you

- Listen to the online lectures and read the textbooks
- Take notes during lectures
  - Taking notes help you remember the material
  - Without notes, the slides will not be enough when you come to revise
- Attend all sessions: Q&A Session, small-group tutorials, lab clinics.
- Make use of the course materials (including the textbooks)
- Read around the subject/try things for yourself
- Plan your time carefully: the more you leave things until later, the more likely you are to struggle.
- Do as much learning on your own as you can, but know that sometimes you have to ask for a little extra help.

If you have questions . . .

1. Check the slides
2. Ask other students
3. Have a look at the reading materials (text books)
4. Check if someone has already asked the question in the forum
5. Ask the question yourself in the forum and wait a bit
6. Ask during tutorials or coursework clinics
7. Arrange to talk to one of the Lecturers in our office hours
8. Send us an email (we aim to answer within 3 working days).

The last two options will be much slower as there are more than 200 students.

Expectations of Behaviour

Staff and students are expected to behave respectfully to one another – during lectures, outside of lectures and when communicating online or through email.

We won't tolerate inappropriate or demeaning comments related to gender, gender identity and expression, sexual orientation, disability, physical appearance, race, religion, age, or any other personal characteristic.

If you witness or experience any behaviour you are concerned about, please speak to someone about it. This could be one of your lecturers, your personal tutor, a programme administrator, the diversity & inclusion co-chairs (Alfie Abdul-Rahman & Petr Slovak) at

informatics-diversity@kcl.ac.uk,

a trained harassment advisor, or any member of staff you feel comfortable talking about it to.

More info can be found at:

https://www.kcl.ac.uk/hr/diversity/dignity-at-kings

Announcements

- We will post message on the “Announcements” message board on KEATS.
- This sends you an email.

- Check your mail!

We know it is old technology, but most companies still use it extensively.

- It will save you waiting for things that won’t happen, or doing work you don’t need to do, or not looking at the corrected document, or . . .

Thankyou!