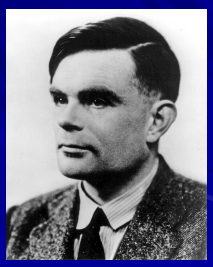
Computer Science meets Philosophy

Thinking Machines, Computational Models, and Moral Robots

Philosophical Frontiers of Computer Science

Professor Peter Millican
Hertford College, Oxford
peter.millican@hertford.ox.ac.uk
Faculties of Computer Science and Philosophy, University of Oxford



1

Organisation of this Handout

GENERAL OVERVIEW

1. Consider Computer Science and Philosophy! (3)
2. The many links between the subjects (21)
3. Computational models: a new form of enquiry (29)

THREE TOPICS ON THINKING MACHINES

4. Machine intelligence: the very idea! (39)
5. Artificial intelligence: an example (57)
6. Limits of computability: the Halting Problem (66)

2

1. Consider *Computer Science*, consider *Philosophy*, and consider *Combining Them*!

For more on the degree, see

http://www.cs.ox.ac.uk/compphil

http://www.philocomp.net/

For questions, email:

undergraduate.admissions@cs.ox.ac.uk

3

Degree Programme Desiderata

- *Interesting*, with lots of *variety* and *flexibility* (in case you discover 3 years from now that after 2 years of studying *X-ology* in great depth, you really want a change – and many students do!);
- *Empowering*, developing you to maximise your potential in a rapidly-changing world, where you might have to rethink your life plans repeatedly;
- *Economically valuable*, bringing you a wide range of excellent employment opportunities, with high salary *and* great conditions (ideally, using some of the skills you have practised for 3 years or more).

4

Applying for CS&P – Some Points

- Maths and Further Maths A-Level (or Higher Maths in IB etc.) are *strongly recommended*.
  - Apart from that, *do what you like!* Sciences are most common, but Art, Languages, Literature, Music, Religious Studies (etc.) are all fine.
- There's no disadvantage at all if you …
  - … have never studied Philosophy before;
  - … have only studied "sciences" in the sixth form;
  - … are applying for some other subject (e.g. pure CS) at other universities, and hence don't mention CS&P on your UCAS "Personal Statement" – *no problem!*

5

The Standard Conditional Offer

- A levels: A*AA
  - A* must be in Maths, Further Maths, or Computing/Computer Science;
  - If taking Further Maths A-level or AS-level, you should get at least an A.
- IB: 39 (including core points)
  - With 766 at HL, including 7 in HL Maths.
- Advanced Highers: AA/AAB
- Others: see https://bit.ly/2QTaUYG (UK) and www.ox.ac.uk/intquals (international).

6

Computer Science meets Philosophy

What Is Computer Science?

- Computers and the programs they run are among the most complex – and most sophisticated – products ever created.
- Computer science is about understanding, designing, analysing and using them effectively, answering questions such as:
  - What can computers do, and what are their limits?
  - How can they be made maximally efficient?
  - How far can we guarantee their reliability?
  - What natural processes can we use them to model?

7

Areas of Computer Science

- Theoretical Computer Science
  - For those who like mathematics and logic
- Software Engineering
  - For those who like designing things, and practical applications
- Artificial Intelligence
  - For those who want to explore new frontiers
- Also Computer Systems, Cyber-security, Databases, Graphics, Informatics, Networks, Scientific Computing, etc. …

8

Why Study Computer Science?

- Computers Shape the Future
  - They're everywhere, and constantly gaining ground in every field of life.
  - Understanding them, and being able to use them to develop entirely new applications, is personally *empowering*, economically *valuable*, and opens huge *creative potential*.
  - These skills are valued in almost every area of business, education, industry, and research, giving tremendous flexibility for the future.

9

Flexible Employment

- Computing also gives great opportunities for self-employment and flexible careers (e.g. working for others, but from home)
  - No need for special funding: you just need a machine and an internet connection, to link to a massive worldwide market!
  - Even as a student, a great learning experience:
    - Developing substantial software to be used by others, dealing with customers, solving problems in real time, scheduling, managing expectations and requirements, … all valuable skills.

10

Adding Philosophy to the Mix

- The skills required by Maths/Computing and by Philosophy significantly overlap:
  - Analysing hard problems into sub-problems, then linking the solutions to these together in logically watertight ways: this could apply equally to a computer program, or a philosophical argument.
  - Intellectual creativity – thinking through a problem that maybe nobody else has considered before.
- But adding Philosophy brings other – equally important – skills, as well as lots more *variety*, (with plenty of intellectual excitement and fun).

11

A Wealth of Fascinating Problems

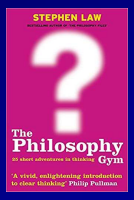
- Is there a God (and how can we know)?
- What is it for something to be "right" or "wrong"?
- What is the basis of political obligation?
- Are we "free", and does it matter?
- Is mathematics just a product of human thought?
- Do the paradoxes of modern physics matter?
- Could a machine think, or be free, responsible … ?
- Can we trust science (e.g. of climate change)?
- Are artistic judgments merely subjective?
- … *(etc. through all the disciplines)* …

12

Professor Peter Millican, Hertford College, Oxford *peter.millican@hertford.ox.ac.uk*

Computer Science meets Philosophy

Will I Enjoy Philosophy?



- There are lots of excellent books that you can dip into, for example Stephen Law's *Philosophy Gym*.
- If you'd like to sample some Oxford lectures, my 2018 *General Philosophy* lectures are available at:

  http://podcasts.ox.ac.uk/series/general-philosophy-2018

  Or find this link (and others) at:

  www.philocomp.net/home/oxford.htm

13

The Joy of Joint Honours

- There is a hugely enjoyable contrast between:
  - Solving technical problems, finding a "right answer" or creating a program that *does something*.
  - Grappling (often in company) with deep questions about "Life, the Universe, and Everything".
- *Don't be put off by the prospect of essays*: working out your thoughts on such fascinating topics is very rewarding, and very different indeed from writing school essays! (Schools tend to treat Philosophy as an "Arts" subject, but it's *much more* like science than literature.)

14

A Brilliant Combination for Both Study and Employment

- Computers are being increasingly applied to almost every new area of study and thought …
  - This requires *intellectual flexibility* and *imagination*, as well as technical skill.
  - Studying Philosophy is ideal for developing the ability to think "outside the box".
- Technical skills are not enough for high-level management positions …
  - Philosophy teaches you to *synthesise written material*, to *argue and present a case*, *understand other points of view*, to *write good reports* etc.

15

The Employers' View of *Computer Science and Philosophy*

"We believe that the degree is likely to produce high-value graduates who combine technical and analytical skills with rhetorical and literary skills. These are just the kind of people that industry, and we believe society generally, wants. The degree will produce graduates who can potentially become members of executive boards of companies, where clarity of thinking combined with knowledge of technology issues will be vital. … It will also produce software engineers with the skills to analyse, and intelligently explain and document, complex problems. We believe that the new degree is likely to excite and attract strong, ambitious candidates."

Oxford Computer Science External Advisory Board

(see www.philocomp.net for more)

16

A Multitude of Problems to Solve

- Consider, for example:
  - Climate change, ecosystem stress, extreme events, potential famine, mass migrations; Cultural conflicts (e.g. racial or religious), political instability, crises of authority, economic instability; Healthcare crises, drug resistance, epidemics.
- Add to these some specific AI risks:
  - Economic polarisation (from domination of "big tech" business, automated financial trading etc.); Political manipulation (from money, targeted advertising, etc.); Widespread unemployment (as people lose economic value to computers, self-driving cars, and robots etc.).

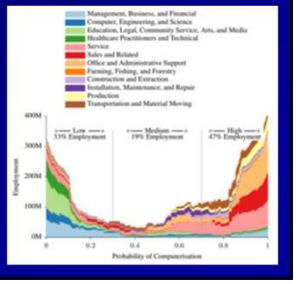
17

The *Futuremakers* Podcast



Our new podcast has 10 episodes on AI, ranging from its history to its many future risks and possibilities.

For example, one of them discusses with author Mike Osborne (and others) his influential and worrying recent paper *"The Future of Employment: How Susceptible are Jobs to Computerisation?"*



18

Professor Peter Millican, Hertford College, Oxford *peter.millican@hertford.ox.ac.uk*

Computer Science meets Philosophy

Not "Business as Usual"

- These new problems require a generation of thinkers who can *both* understand the relevant technologies, *and* think creatively "outside the box" to help find new solutions.
  - Philosophers specialise in this sort of creative, critical thinking about standard assumptions; and much of the modern world has built on their ideas (e.g. moving from the medieval to the scientific world-view, from religious authority and "divine right of kings" to democracy etc.).
  - But modern problems are so complex, that future philosophers will need AI to help solve them!

19

The Super-Flexible Oxford Course in Computer Science and Philosophy

- Year 1 Functional programming, Imperative programming, Design and analysis of algorithms, Discrete mathematics, Probability, *General philosophy*, Elements of deductive logic, Turing on computability and intelligence
- Year 2 50% Computer Science, *50% Philosophy*
- Year 3 25%-75% Computer Science, *25%-75% Philosophy*
- Year 4 0%-100% Computer Science, *0%-100% Philosophy*

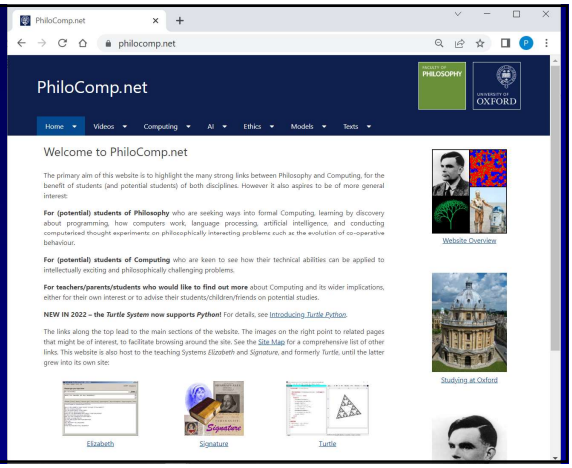
20

2. The Many Links Between Computer Science and Philosophy

For more on all of this, see

www.philocomp.net

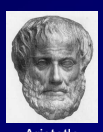
21

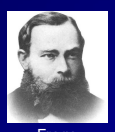


22

(a) Logic and Inference

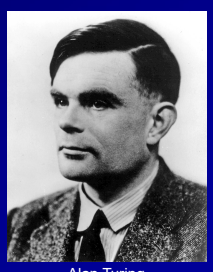
- There are some obvious links between the two disciplines. Logic is central to both:


Aristotle


Frege


Gödel

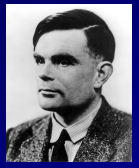

McCarthy


Alan Turing

23

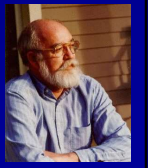
(b) Artificial Intelligence

– AI raises a host of philosophical questions:
- What is the nature of intelligence?
- Could a machine think?
- Could a computer be free, or responsible, or … ?


Alan Turing


Hubert Dreyfus


Maggie Boden

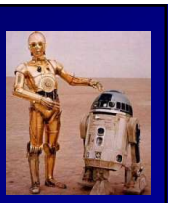

Dan Dennett

24

Professor Peter Millican, Hertford College, Oxford *peter.millican@hertford.ox.ac.uk*

Computer Science meets Philosophy

(c) Morality





- Information storage and transmission raise issues of security and privacy.
- These have greatly magnified as artificial intelligence (and particularly machine learning) has revealed new possibilities for targeted advertising, propaganda, and privacy violation.
- Increasing use of autonomous robots in transport, medicine, and other potentially life-threatening situations clearly requires formulation of ethical rules for their behaviour.

25

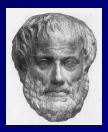
(d) Less Obvious Connections

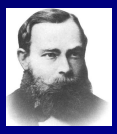
- *Aesthetics*: Can a computer create genuine art?
- *Metaphysics*: What kind of thing is a program?
- *Philosophy of Mathematics*: Should we accept computer-generated proofs we cannot fully grasp?
- *Philosophy of Science*: How far should computer models (e.g. of global warming) be trusted?
- *Philosophy of Social Science*: Can computer models revolutionise the study of Economics and Politics?
- *New Philosophical Fields*: Philosophy of Artificial Intelligence, Philosophy of Artificial Life, Philosophy of Computation, Philosophy of Information, ...

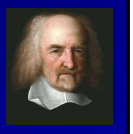
26

What Most Unites the Two Subjects

- *The dream to automate reasoning …*


Aristotle


Frege


Hobbes


Leibniz


Lovelace


Boole


Russell


Turing

27

Prostheses for the Mind

- Where information and methods of inference can be captured appropriately, the computer enables us to reason
  - *far more accurately,*
  - *much faster,*
  - *about anything at all!*
- This gives a wonderful new tool for thought-experiments, exploring possibilities that are too complex to calculate "by hand".



28

3. A New Form of Enquiry

- Mechanising deduction allows a novel form of investigation, where we:
  - Formally define some set of rules;
  - Implement those rules in a computer program;
  - Run the program to apply the rules entirely *mechanically* (hence *enforced honesty*);
  - See what happens, and use the outcome to inform our refinement of the rules.
- This combines *deductive* "reasoning" with *inductive* discovery and investigation.

29

Computer Science Across the Curriculum: Free Resources

- https://www.turtle.ox.ac.uk/csac
  - Scroll down to chapter 7, on "Chaos, Recursion, and Self-Similarity"
  - Select "Logistic equation" or "Logistic spider"
  - The relevant program will open in the "Turtle Online" system. Click on "RUN" to run it.
  - Other programs can be chosen from the address above, or by clicking on the "File" tab and using the "CSAC" menu.

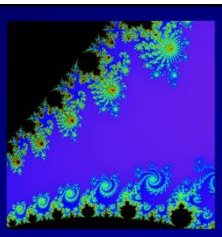
30

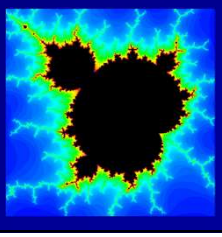
Professor Peter Millican, Hertford College, Oxford *peter.millican@hertford.ox.ac.uk*

Computer Science meets Philosophy

Exploring Chaos

- "Chaos" only came to light through computer models, but plays a major role in lots of natural phenomena.
  - Even the very simple "logistic equation" is chaotic.
  - Explore this, and fascinating mathematical objects such as the Mandelbrot Set, though our online programs;
  - Create "self-similar" patterns of great beauty ...





31

More at www.turtle.ox.ac.uk/csac

- Comprehensible programs to illustrate:
  - Animation (e.g. motion under gravity);
  - Rocket dynamics and shooting for orbit;
  - Cellular automata: "Life", epidemics, shells;
  - Chemistry: diffusion, Brownian motion;
  - Biology: evolution, co-evolution, flocking;
  - Physics: waves, quantum interference;
  - Games: knight's tour, nim, tic-tac-toe.
  - Chaos and iterated function systems ...

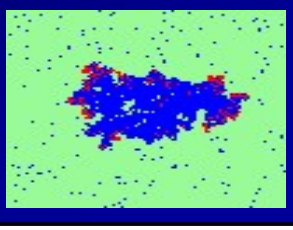
32

Shells, Ferns, Epidemics, ...



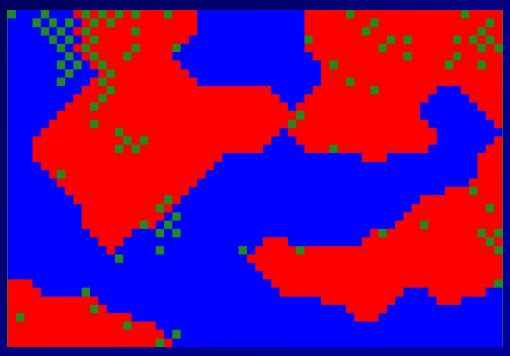






33

Schelling's Segregation Model

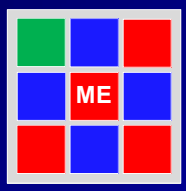


34

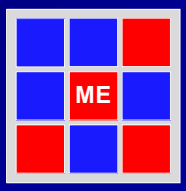
How the Model Works

A 60 x 40 grid with 100 random green spaces, but blues and reds evenly distributed everywhere else. Individuals are "unhappy" – and therefore move – if the other colour has a majority of 2 or more amongst their neighbours:

Happy:

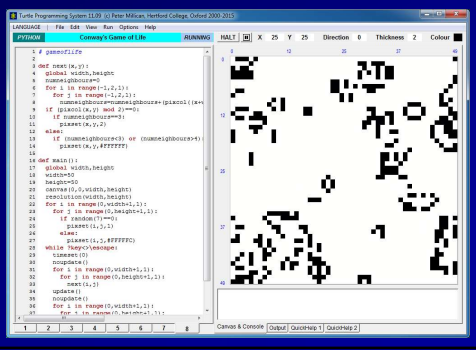


Unhappy:



35

Explore This and Other Interesting Models at www.turtle.ox.ac.uk

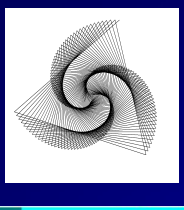


36

Professor Peter Millican, Hertford College, Oxford *peter.millican@hertford.ox.ac.uk*

Computer Science meets Philosophy

Code in BASIC, Java, Pascal or Python



PASCAL triangle_spin.tgp

```
1 PROGRAM triangle_spin;
2 VAR i: integer;
3 BEGIN
4  direction(90);
5  for i:=1 to 200 do
6   begin
7    forward(i*4);
8    right(121)
9   end
10 END.
```

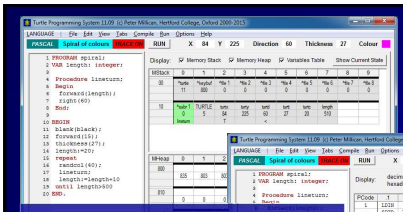
BASIC triangle_spin.tgb

```
1 REM triangle spin
2
3 DIRECTION(90)
4 FOR i%=1 TO 200
5  FORWARD(i%*4)
6  RIGHT(121)
7 NEXT
8 END
```

PYTHON triangle_spin.tgy

```
1 # triangle spin
2
3 def main():
4   direction(90)
5   for i in range(1,201,1):
6     forward(i*4)
7     right(121)
```

37



Computer Science

Your code is compiled into a form of *machine code*, which you can explore through the "Power User Menu", thus learning a lot about Computer Science. Above is the Memory display, and at the right, the machine code display.

38

4. Machine Intelligence: The Very Idea!

For more on all of this, see http://www.millican.org/papers/2013Turing.pdf: *"The Philosophical Significance of the Turing Machine and the Turing Test"*

39

Artificial Intelligence in the News

- Deep Blue beats World Chess Champion Garry Kasparov, May 1997
- IBM Watson wins *Jeopardy*, Feb 2011
- DeepMind's *Deep Q* system "teaches itself" to play a range of vintage Atari video games, Feb 2015
- Hawking, Musk, and Gates warn about the "existential threat" to humanity posed by "superintelligence"
- *AlphaGo* beats two top Go players, Lee Sedol (March 2016) and Ke Jie (May 2017), largely teaching itself how to play. *AlphaZero* then learns Go, Shogi and Chess entirely by self-teaching, beating the world computer chess champion after only 4 hours!

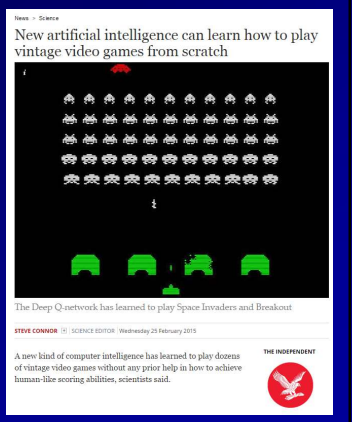
40

Deep Q:

Autonomous, Versatile Learning

The software learns to play very well, given information only about the pixels on the screen, its own controls, and the scores.

41



News > Science

New artificial intelligence can learn how to play vintage video games from scratch

The Deep Q-network has learned to play Space Invaders and Breakout

STEVE CONNOR | SCIENCE EDITOR | Wednesday 25 February 2015

A new kind of computer intelligence has learned to play dozens of vintage video games without any prior help in how to achieve human-like scoring abilities, scientists said.

THE INDEPENDENT



Deep Mind 4 Lee Sedol 1 (March 2016)

42

Computer Science meets Philosophy

"Maybe they can play games well by (finding and) following rules, but that isn't *GENUINE INTELLIGENCE*"

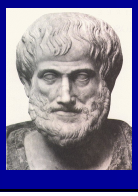
- This is a very natural objection, but raises the obvious question:

  what *is* "genuine intelligence"?
- In 1950, the mathematician and computing pioneer Alan Turing suggested an influential answer to this, drawing on the results of his earlier 1936 discoveries. I'd like to suggest that this involved a major "paradigm shift" …

43

Modes of Explanation (until 1600)

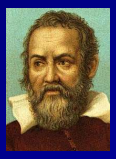
- Purposive Design (God)
  - Things in the world (e.g. animals, plants, minerals) take the form they do because they were *designed* to be that way.
- Purposive Action (Aristotle)
  - Things in general (humans, stones, water, fire, planets etc.) behave as they do because they are striving to achieve some desired state, or to avoid some abhorrent situation.



44

The Scientific Revolution (1609-87)

- Mechanism (Galileo, Descartes)
  - Physical objects move through *inertia* and *mechanical contact*. (Only human behaviour is governed by reason.)



- Mathematical Instrumentalism (Newton)
  - The action of gravity is not *intelligible* to us in the way that mechanism is, but we can predict its effects mathematically, in terms of forces that generate acceleration.



45

The Naturalistic Turn (1739-1859)

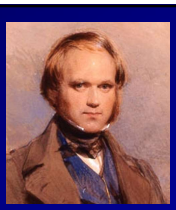
- Naturalist Psychology (Hume)
  - Human behaviour is governed more by emotions and imagination than by reason: it is more "animal-like" than "god-like".



- Evolution (Darwin, Wallace)
  - Biological organisms take the form they do owing to inheritance of characteristics and competition for survival and reproduction.

46

Darwin and Evolution



- Before Darwin, it was generally assumed that sophisticated adaptation of an animal or plant to its environment must be explained in terms of *purpose* (i.e. God's design).
- What Darwin showed was that such adaptation could be explained in a way that did not involve purpose, but instead combined *random variation* with *natural selection*. The variations that are best for surviving and multiplying tend to become more common in succeeding generations).

47

Breaking Paradigms

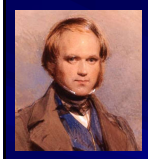
- Advances like Galileo's and Darwin's involve a fundamental change in our thinking:
  - How we see our place in the universe (e.g. we are not at the centre; we are continuous with the animals rather than quite separate from them).
  - Our understanding of the possibilities of scientific explanation (e.g. through mathematical forces rather than strivings; or by *inherited variation* and *selection* rather than rational design).
- So it's not surprising that these great advances were made by philosophically minded thinkers.

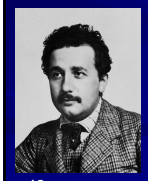
48

Professor Peter Millican, Hertford College, Oxford *peter.millican@hertford.ox.ac.uk*

Computer Science meets Philosophy

Darwin, Einstein, and Hume





- Darwin read Hume's *Enquiries*, *Essays*, *Natural History of Religion*, *History*, *Dialogues*, and biography. Indeed he was reading and noting Hume on "the reason of animals" when he came up with his theory of evolution (1838-9).
- Einstein said that "the critical reasoning required … [to question] … the absolute character of time … was decisively furthered … by David Hume's writings". "I studied [his] treatise on understanding … with eagerness and admiration shortly before finding relativity theory."

49

Modern Physics (1905-30)

- General Relativity (Einstein)
  - Space and time are integrated, and matter affects their structure, "curving" space in a way that generates gravitational movement.
- Quantum Mechanics (Bohr, Heisenberg)
  - Phenomena are described in terms of interacting "wave functions", which enable prediction but which cannot be understood as implying determinate underlying "states".

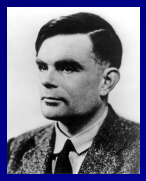
50

The Weirdness of Physics

- Both Relativity and Quantum Mechanics illustrate how different the world can be from our "intuitive" understanding of things.
- Alan Turing's invention of formal computation led to more counter-intuitive discoveries:
  - About the foundations of mathematics: there are some mathematical questions for which there is no possible method of solution.
  - About the nature of "thinking" itself: we need to be open to the possibility of inanimate thought.

51

Turing's Discovery (1936)



- Information processing can be understood in terms of symbolic inputs and outputs, governed by explicit and automatic processes with a limited range of operations (as defined by a "Turing machine").
- Hence information does not presuppose an "understanding" mind.
- It seems that *automated processes (with these very limited operations) can solve any information processing problem that we know how to solve*.

52

A New Paradigm

- We naturally see most things in the world as either (consciously) *purposive* or *inanimate*.
- "Intelligence" involves sophisticated processing of information to some end – so it seems to imply *purposiveness*, and hence consciousness.
- But since Turing, we have processes that involve sophisticated *information processing* without *conscious purpose*, just as Darwin brought us sophisticated *adaptation* without *intentional design*. For discussion, see:

  http://www.millican.org/papers/2013Turing.pdf

53

- Hence we can (and should) distinguish
  sophistication of information processing
  from
  phenomenology ("what it's like from the inside").
- These can often come apart:
  - Dogs can desire as strongly as we do (but that doesn't make us judge them intelligent);
  - Being great at maths or logic doesn't require a deep and sensitive inner life (consider Mr Spock);
  - Experts are often less "conscious" than novices;
  - Our intuitive judgements of difficulty are often badly wrong. Dogs can run and catch a ball, but this needs vastly more computation than arithmetic.

54

Computer Science meets Philosophy

Can Machines be Intelligent?

- If we do indeed distinguish *sophistication of information processing* from *phenomenology*, then it's clear that "intelligence", as we normally consider and talk about it, is more a measure of the former than the latter.
- In our new world of unconscious – but highly sophisticated – information processors, it makes sense to allow our concept of "intelligence" to evolve accordingly.
- So Turing is vindicated: we can legitimately allow for "intelligence" *without phenomenology*.

55

Could Machines Take Over?

- We have no reason to suppose that intelligent machines will suddenly achieve consciousness and decide to compete with (or eliminate) us.
- But the exploitation of intelligent machines could easily lead to a dangerous imbalance in *human* power (e.g. financial trading, targeted advertising, political manipulation, widespread unemployment, etc.).
- This *huge*, *unprecedented* technological change – potentially impacting on so many aspects of our lives – will require plenty of careful, critical reflection on how we organise society and its tensions. So ...
  - *Society needs philosophers as much as ever!*

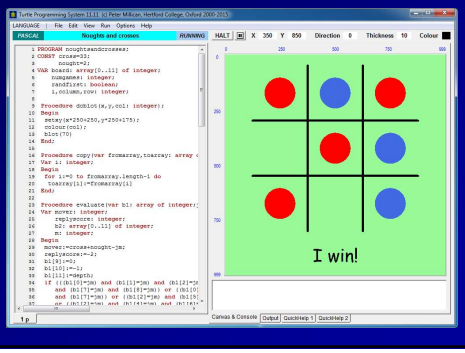
56

5. Artificial Intelligence: An Example

- Before Alan Turing invented the computer (in his famous 1936 paper "On Computable Numbers"), it was assumed that *intelligent* information processing could only be performed by a conscious agent.
- Now we have many examples of such processing done by computers.
  - The next slides show how the *Turtle System* noughts and crosses program calculates its moves, so that it can never be beaten.

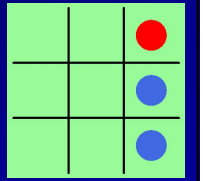
57

Illustrating the Power and Fun of Programming: Noughts and crosses



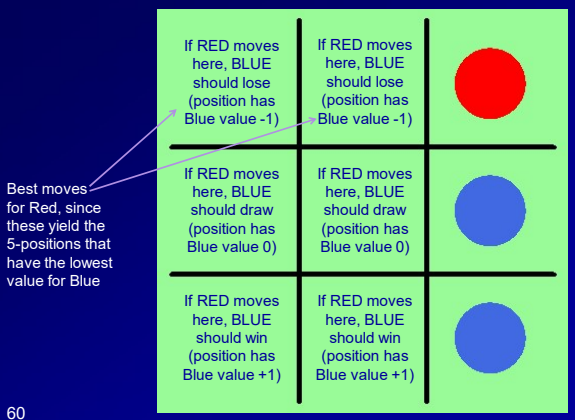
58

- Playing infallibly requires a reliable method for assessing the "value" of any position (+1=winning; 0=drawing; -1=losing).
- Consider this position, with Red to move:



  - It's a "6-space-position" (let's say "6-position" for short).
  - After Red makes her move, her opponent, Blue, will be faced with a 5-position in which to make his reply.
  - Suppose there is a method for calculating the value (to Blue) of any 5-position: how can Red use this *now* to work out her best move?

59



60

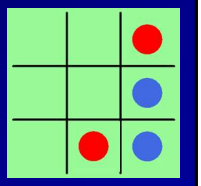
Computer Science meets Philosophy

"Minimaxing"

- Red winning is the same as Blue losing (and vice-versa).
  - The value of a position to Red is the inverse of its value to Blue (e.g. -1 as opposed to +1).
  - So Red tries to find moves which *maximise* the value to her, and *minimise* the value to Blue.
  - Red, faced with a 6-position, chooses the move which produces the 5-position which is *worst* (or equal-worst) for Blue.
- OK, but how does she work *that* out?

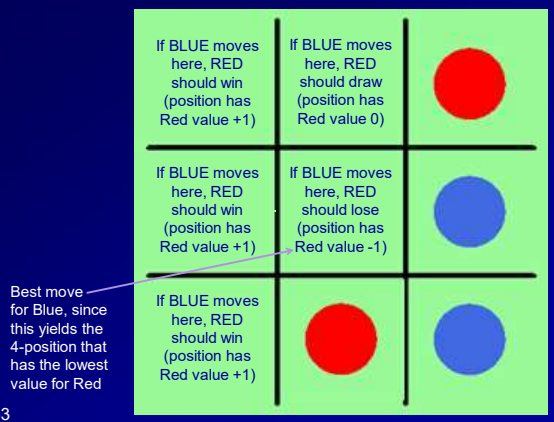
61

The same kind of problem, a move later …



- Suppose Red moved bottom centre. We would reach this position, with Blue to move:
  - It's a 5-position.
  - After Blue makes his move, his opponent, Red, will be faced with a 4-position in which to make her reply.
  - Suppose there is a method for calculating the value (to Red) of any 4-position: how can Blue use this *now* to work out his best move?

62



63

The Magic of Recursion

- We have seen how to evaluate a 6-position if we can evaluate a 5-position,
  - … and in the same way we can evaluate a 5-position if we can evaluate a 4-position,
  - … and a 4-position if we can evaluate a 3-position, and a 3-position if we can evaluate a 2-position, and a 2-position if we can evaluate a 1-position, and a 1-position if we can evaluate a 0-position!
- The only other thing we need is to be able to recognise a FINAL position as winning for Red, winning for Blue, or drawn.

64

- So to evaluate a position with X to move:
  - Has X already lost (because the opponent has a line of three)? If so, its X-value is -1.
  - Is the position full up (with all nine spaces filled)? If so, its X-value is 0.
  - If neither of these, then construct in turn all of the positions (with Y to move) that can arise from the current position (i.e. try out in turn all of X's possible moves in the current position) …
    - Evaluate each of these new positions *from Y's point of view* (resulting in evaluations Y_1, Y_2, and Y_3, say), and then assign – as X-value of the *current* position – the *inverse* of the *lowest* of Y_1, Y_2, and Y_3.
- Choose the move with the best X-evaluation.

65

6. Limits of Computability

- In his 1936 paper, Turing proved that some problems cannot be solved by *any* systematic method (i.e. any *algorithm*).
- He did this by defining a simple kind of machine (a *Turing machine*) which – he argued – can perform any information processing that is precisely specifiable (e.g. so that a human could do it).
- Then he proved that there are some things that a Turing machine cannot possibly do.
  - This is all part of the fascinating first year CompPhil course, "Turing on Computability and Intelligence".

66

Computer Science meets Philosophy

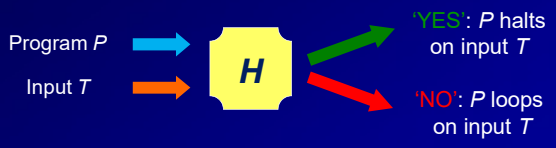
The Halting Problem

- Suppose we have the text of a computer program *P* (in a standard programming language), taking input from a text file, *T*.
- We would like to have a program *H* which will examine *P* and *T*, and then reliably work out the answer to this question:
  - *Will program P, when run with input T, eventually halt, or will it never terminate?*
- *Turing proved – approximately – that H is impossible (using Turing machines).*

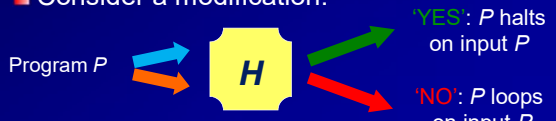
67

Proof by Contradiction

- Suppose we have such a program *H*.



- Consider a modification:



68

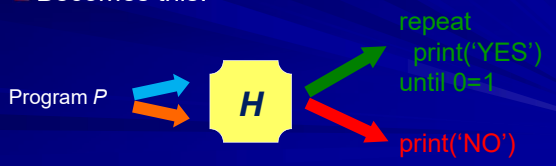
- Clearly if the initial program is possible, then the modified program must be possible too, because the only difference is that instead of taking *both* a program file *P and* a text file *T* as input, it uses the program file in place of *T* (hence its verdict is based on whether or not the specified program file *P* halts when given its own text as input).
- Having modified the *input* to our initial program, we now make an equally straightforward modification to its *output*. Here we assume that the output is generated by means of a simple "print" statement, which we modify in one case so that it goes into an infinite loop …

69

- We now modify our program so that this:

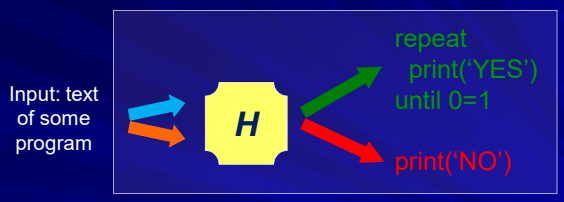


- Becomes this:



70

The Contradictory Program Q



- Does program *Q* halt when given itself as input? If it does, then it doesn't, and if it doesn't, then it does! So *Q* is an impossible program, and therefore *H* must have been impossible too.

71

Further Links

- For a brief explanation of the context of Turing's work on computability (e.g. Hilbert's programme and Gödel's theorem), see:
  - www.philocomp.net/home/hilbert.htm
  - www.philocomp.net/home/turing.htm
- For artificial intelligence links, see:
  - www.philocomp.net/ai.htm
- For some of the wide variety of philosophically illuminating computer models, see:
  - www.philocomp.net/models.htm

72