QECDT Topics Presentation: Complexity theory

Dominic Moylett

University of Bristol

dominic.moylett@bristol.ac.uk

November 18, 2015

Complexity theory in a nutshell

"How hard can it be?", *Jeremy Clarkson*

What is complexity theory?

Complexity theory is the study of how difficult it is to solve a problem with a computer.

What is complexity theory?

Complexity theory is the study of how difficult it is to solve a problem with a computer.

How do we measure difficulty?

What is complexity theory?

Complexity theory is the study of how difficult it is to solve a problem with a computer.

What is a problem?

What is complexity theory?

Complexity theory is the study of how difficult it is to solve a problem with a computer.

What is a computer?

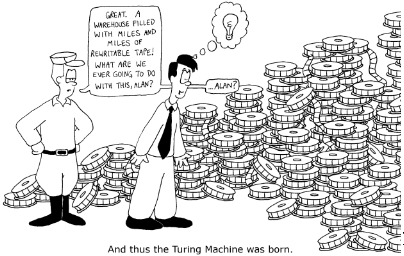
Structure of part one

- What is a computer?
- What is a problem?
- How do we measure difficulty?

Summary of part one

- What is a computer? *Deterministic Turing Machine, Non-Deterministic Turing Machine*
- What is a problem? *Deciding if a word is in a language, verifying that a word is in a language*
- How do we measure difficulty? *Upper bound of time for an input of length n*

End of part one



And thus the Turing Machine was born.[1]

[1] http://www.cs.utah.edu/~draperg/cartoons/2005/turing.html

Structure of part two

- Putting it all together!
- ...only to get another (very difficult) problem.
- How might we try to solve this new problem?

Exercise Left for the Student

Does $P = NP$?

(NB: You should probably refer to the literature before trying to solve this.)

The P versus NP problem

Arguably first proposed by Gödel in a letter to von Neumann in 1956.[2]

First stated formally by Cook in 1971.[3]

Solving the problem will earn you a million dollars, courtesy of the Clay Mathematics Institute.[4]

Aaronson has called it the most important Millennium Problem, as the answer could make the other problems significantly easier or harder to solve.[5]

[2] https://ecommons.cornell.edu/bitstream/handle/1813/6910/89-994.pdf
[3] http://dl.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=805047
[4] http://www.claymath.org/millennium-problems/p-vs-np-problem
[5] http://www.thenakedscientists.com/HTML/interviews/interview/1001376/

The easy side: $P \subseteq NP$

Recall that any TM is by definition non-deterministic.

Likewise, any polynomial-time TM is also non-deterministic.

Hence $P \subseteq NP$.

The easy side: $P \subseteq NP$

Alternative proof (using verification):

Let *TM* M decide L in polynomial time. Define V as follows:

$V(w, c)$:
if $M(w)$ *accepts* **then**
| **accept**
end
else
| **reject**
end

V verifies L in polynomial time. Hence $P \subseteq NP$.

The harder side: Is $NP \subseteq P$

Another way to think of this problem: *If a problem can be easily verified, can it be easily solved?*

How might we answer this question?

Why not look at the hardest problems in NP?

If $P = NP$, then even the hardest problems in NP will be solvable in polynomial time.

And if $P \subset NP$, then these are the problems that won't have a polynomial time solution, as could be checked by lower-bound analysis.

But how can we determine the hardest problems in NP?

Summary of part two

- Putting it all together! *P*, *NP*
- ...only to get another (very difficult) problem. *Are easy to verify problems easy to solve?*
- How might we try to solve this new problem? *NP-Complete problems*

What else is there?

Recall our three questions from part one:

- What is a computer?
- What is a problem?
- How do we measure difficulty?

What if we answered these differently?

What is a computer?

Probabilistic Turing Machines: BPP, RP

Parallel Computing: NC

Talking to another, more powerful computer: MA, IP

Quantum computers: EQP, BQP

Time travel: P_{CTC}

What is a problem?

Computational problems: *NP-Hard*

Counting problems: $\#P$

Complementary problems: *co-NP*

How do we measure difficulty?

Exponential time: EXP

Linear time: LIN

Space complexity: $PSPACE$, $EXPSPACE$

Sublinear working space: L

This is only the beginning

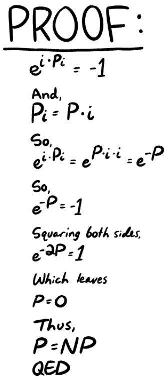
There are many more complexity classes out there, and very quickly relating them in a simple equation like this:

$$P \subseteq NP$$

Becomes this:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE = IP = P_{CTC} \subseteq EXP \subseteq EXPSPACE$$

The end



[6]

[6] http://www.smbc-comics.com/?id=3919

The end

7

[7] http://www.smbc-comics.com/?id=3919