Basic Science for Software Developers

David Lorge Parnas
and
Michael Soltys

- Every engineer must understand the properties of the materials that they use.
- Properties of physical products:
  - Technological properties (e.g., rigidity)
  - Fundamental properties (e.g., Maxwell's or Newton's laws)

  Fundamental properties don't change with improved technology.
- Students must understand the fundamental limitations of the materials that they use, to be effective and competent engineers.
- Explaining the relevance of basic science is difficult: technological limitations are constantly used to compare products, and so they seem more real to students.

- For **software engineers** the materials used are computers and software.
- In this area too, the properties can be divided into two classes:
  - **Technological properties:** memory, processor speed, word length, precision, etc.
  - **Fundamental properties:** limits of computability, complexity, and the inevitability of noise in data.

  Technological properties change; fundamental properties don't.
- Misunderstandings: can we prove that loops terminate?

Basic Science Course at McMaster

- Finite Automata (finite number of states, no memory)
- Regular Expressions
- Context-Free Grammars
- Pushdown Automata (finite automata with a stack)
- Turing machines (computability)
- Rudimentary complexity (enough to discuss **P** & **NP**, and cryptography)

Complexity: build intuition

Challenge: so much of complexity is conjectures.

NP is the set of problems which have simple, verifiable solutions (and these solutions may be difficult to find).

Statement	**Interpretation**
P $\neq$ NP	hard problems exist
Avg-P $\neq$ Dist-NP	hard problems are easy to generate
There exists a one-way function	hard *solved* problems are easy to generate
There exists a trap-door function	Alice and Bob can *publicly generate* a hard problem for Carl

[a]

[a] R. Impagliazzo, "A personal view of average-case complexity", 1995.

Possible Worlds

- **Algorithmica:** $\mathbf{P} = \mathbf{NP}$, in this world everything is *easy*, once you learn how to do it.
- **Heuristica:** $\mathbf{P} \neq \mathbf{NP}$ but $\mathbf{Avg\text{-}P} = \mathbf{Dist\text{-}NP}$. Hard problems exist, but you never encounter them in "practice".
- **Pessiland:** $\mathbf{Avg\text{-}P} \neq \mathbf{Dist\text{-}NP}$ but one-way functions do not exist. So things are hard to solve, but not hard enough to allow for reliable cryptography.
- **Minicrypt:** Private-key cryptography is possible, there are pseudo-random number generators, digital signatures, zero-knowledge proofs.
- **Cryptomania:** All four statements in the previous table are true, and public-key cryptography is secure.