Week 2. The Historical Role of Computation

Neil Barton
Universität Konstanz

09. November 2020

Structure

My aim here is to identify (by looking at the history) some key themes that will be important for the rest of the course:

1. Theme: The same problem encoded in different ways can have a different level of difficulty.
2. Theme: We clearly have some notion of unambiguous instructions and effective procedure. But how to make this precise?
3. Theme: Sometimes computational difficulty can be useful.
4. Theme: We want computing devices to be flexible and compute multiple different operations.
5. Theme: What are the differences (should there be any) between human and machine computation?
6. Final Problem: The Entscheidungsproblem.

Why we need computation

- Mathematics (and especially versions of geometry and arithmetic) are very useful for keeping track of amounts of objects, and for basic engineering, and prediction applications.
- However, just working with perceptions, humans are not very good.
- Example: If you try to accurately sum two collections of objects presented (without time to use an algorithm to count) you'll be bad (accurate up to four, and tailing off as the numbers get larger[1]).
- So when working with larger numbers, we need some way of recording numbers and manipulating them in calculation.
- Similar points apply about reasoning with geometric figures.

[1] See [Dehaene, 1997] for examples of this kind.

Encodings of Problems

- Various cultures throughout the world developed quite sophisticated number systems e.g. Babylonians in what is modern day Iraq had a base-60 notation system, the Egyptians had numerical hieroglyphs, Roman-numerals are well-known, nowadays we use Arabic numerals, and Chinese and Indian mathematicians made some significant contributions to summing equations (among other things), and the (plausibly) earliest known mathematical artefact (the Lemombo bone) was discovered in Africa.
- As cultures developed and exchanged ideas, some number-systems were adopted for computation and others fell by the wayside.
- Conjecture: A substantial reason for this is that some systems are better for computation than others (e.g. Roman numerals are cumbersome, numeral systems with 0 are less ambiguous).
- We see here our first rough theme: The same problem encoded in different ways can have a different level of difficulty.

Algorithms

- Example. Determining if the cardinal size of two finite collections is equal.

Algorithms

- Example. Eratosthenes' sieve.

Algorithms

Rough Definition.

An algorithm (or procedure) is a finite set of unambiguous instructions to perform a specific task.

- Notice that this is not clearly a precise definition!
- Second theme: We clearly have some notion of unambiguous instructions and effective procedure. But how to make this precise?

Early cryptography

- Example. The Caesar cipher.

Early cryptography

- The idea is the following: Make something that's easy to unwind if you have a piece of information, but hard to unwind if you don't.
- The Caesar cipher actually isn't very good. But we can still identify:
- Overarching theme. Sometimes difficulty can be useful.

Machines and programmes

- Babbage designed the Difference Engine, a machine that could tabulate polynomial functions.
- However, the Difference Engine is designed for a single, very specific task.
- Overarching Theme. We want computing devices to be flexible in that they are able to compute multiple different operations.
- Babbage collaborated with Ada Lovelace on the Analytic Engine, a device capable of changing the operations performed (as well as the data being operated upon), using punch cards.
- In some ways Lovelace foresaw the versatility of computers.

Who/what computes?

- Before we introduce our last problem, let's consider who/what computes.
- For much of our history computation was a human activity.
- However, it appears as though machines can also compute.
- Overarching Theme. What are the differences (should there be any) between human and machine computation?

The Entscheidungsproblem

- We finish with one famous problem.
- We know that there are formal, mechanical rules for churning out theorems of first-order systems.
- But, given arbitrary formula ϕ and some set of axioms A, is there an effective procedure for determining whether there is a proof of ϕ from A?

References

Dehaene, S. (1997).
The Number Sense: How the Mind Creates Mathematics.
Oxford University Press.