RESEARCH PARADIGMS IN COMPUTER SCIENCE

Dr. Garrett Dancik

https://cse.sc.edu/~mgv/csce190f14/three_paradigms_of_computer_science.pdf

Example: factorial function

```
// precondition: a non-negative integer (n) is ready to be specified
// postcondition: returns n! = n*(n-1)*(n-2)*...*1, with 0! = 1.
int factorial(int n) {
        int prod = 1;
        for (int i = n; i>1; i--) {
                prod *= i;
        }
        return prod;
}
```

- Can we prove that this function is *correct*?
- What if the function is called with a negative number?
- What if the function is called with a very large number? How *reliable* is the function?

The Rationalist Paradigm

- Computer science is a branch of mathematics, treats programs on a par with mathematical objects, and seeks certain, a priori knowledge about their ‘correctness’ by means of deductive reasoning.
  1. Computers are mathematical machines.
  2. Computer programs are mathematical expressions….They describe with unprecedented precision and in every minutest detail the behavior, intended or unintended, of the computer on which they are executed. …
  3. Programming is a mathematical activity…its successful practice requires determined and meticulous application of traditional methods of mathematical understanding, calculation and proof. (Hoare 1986)

The Rationalist Paradigm

- Sometimes, program correctness be defined and evaluated formally
  - The program finds the factorial of a positive integer
  - The program detects whether the face of person *y* appears in any given picture (face detection)
  - The program takes a regular expression (a string of text) and returns a list of World Wide Web documents sorted by their 'relevance' to this expression
- Correctness in practice (formal verification)
  - Coq proof assistant: https://coq.inria.fr/
  - Example article: http://www.sciencedirect.com/science/article/pii/S0167642306002048

The Rationalist Paradigm

- …but sometimes, program correctness can **NOT** be defined and evaluated formally
- Example: Microsoft Word For Windows 1.1 code
  - http://www.computerhistory.org/atchm/microsoft-word-for-windows-1-1a-source-code/
- Verification questions:
  - Will the program never terminate unexpectedly?
  - Will the program restrict unauthorized persons from accessing sensitive data?
  - Will the program execute with visibly identical outcome regardless of the operating system used?
  - Will the program not cause the space shuttle to explode
    - Huh? http://www.around.com/ariane.html

The Technocratic Paradigm

- Computer science is a branch of engineering, and includes aspects of software engineering, software design, software architecture, software maintenance and evolution, and software testing.
- Reliable knowledge about programs emanates only from experience, whereas certain, a priori ‘knowledge’ emanating from the deductive methods of theoretical computer science is either impractical or impossible in principle.
- “[But] while executing a program-script in various circumstances…can discover certain errors, no number of tests can establish their absence” (Three Paradigms in CS, Amnon H. Eden)

Technocratic Paradigm

- The argument of **complexity** demonstrates that deductive reasoning is impractical for large programs.
- "How then do engineers manage to create reliable structures? … They have a mature and realistic view of what "reliable" means; in particular, the one thing it never means is "perfect." There is no way to deduce logically that bridges stand, or that airplanes fly, or that power stations deliver electricity." (DeMillo et. al 1979)
- Knowledge about programs (i.e., reliability) comes from testing

- Example: http://www.computerhistory.org/atchm/microsoft-word-for-windows-1-1a-source-code/

The scientific paradigm

- Computer science is a branch of natural sciences, on a par with “astronomy, economics, and geology” (Newell & Simon 1976).
- Since many programs are unpredictable, or even ‘chaotic’, the scientific paradigm holds that a priori knowledge emanating from deductive reasoning must be supplanted with a posteriori knowledge emanating from the empirical evidence by conducting **scientific experiments**.
- Since program-processes are temporal,non-physical, causal, metabolic, contingent upon a physical manifestation, and nonlinear entities, the scientific paradigm holds them to be on a par with mental processes.

The scientific paradigm

- Experiments with programs go beyond establishing reliability. Computer programs can also be used as tools in discovering and empirically establishing laws of nature, e.g., through simulations, testing, or artificial intelligence
- In the scientific paradigm, we test a hypothesis (claim)
  - Algorithm A has a faster running time than Algorithm B
  - Method A is better at identifying faces than Method B
  - Constructing a pillar that partially obstructs a door increases the number of individuals who safely exit a room in the case of a fire.