The Art of Programming

'Introduktion till system i teknik och samhälle', HT 2008

Pavel Krčál

Overview

- Machines
- Algorithms
- Computers as Multi-Purpose Machines
- Programming
- This Mini-Project

Machines

- How do humans operate machines?

Multi-Purpose Machines

- Operating means more than just pushing a button.
- Instructions
- Recipes
- Playlists
- Movements
- Arithmetical operations

Instructions for Multi-Purpose Machines – Algorithms

Algorithms – History

- Al-Khwārizmī, Persian astronomer and mathematician
  - On Calculation with Hindu Numerals, 825 AD, Arabic
  - translated into Latin in the 12th century: Algoritmi de numero Indorum.
  - systematic solution of linear and quadratic equations (the Algebra book)
- David Hilbert (1862-1943)
  - 1920 – Hilbert's Program: mechanization of mathematics
  - Shown impossible by Kurt Gödel, 1931
- Alonso Church (1903-1995), Alan Turing (1912-1954)
  - Mathematical characterization of the notion of algorithm
  - Lambda calculus, Turing machines

Algorithms

- An algorithm:
  - a finite list of well-defined instructions (such that anybody/a machine can follow them and always get the same result)
  - a description of a procedure
- The ultimate purpose is mechanization of different activities that people do
  - Then we can let the machines do the job for us.
    - Al-Khwārizmī – linear and quadratic equations
    - Hilbert – mathematical proofs
    - Calculations in general
- An abstract thing, living in the world of abstract ideas
  - Not a concrete list of instructions on a sheet of paper.
Programming

- Entering algorithms into computers
- Computers: multi-purpose machines, nowadays everywhere

Requires mastering the algorithmic part:
- We have to know what do we want to enter.
- We have to know what instructions are well-defined for computers.

Then: operating a rather complex machine in a rather complex way
Craft/Art – how to operate the computer so that we enter the algorithm we intended to enter

How to Learn Programming?

- **Algorithmics:**
  - Mathematics, abstract thinking
  - Models of computation
  - Proofs of correctness, complexity

- **Craft/Art:**
  - Understanding computers as multi-purpose machines, basic principles
    - Digital machines
    - CPU, memory
    - Simple instructions: arithmetics, memory manipulations, Input/Output
  - A lot of practice – mastering of different tools
    - Programming languages
    - Libraries
    - Debugging

This Mini-Project

- **How will you (start to) learn programming:**
  - Lego robot mini-project
  - Programming an embedded computer in a Lego robot

<table>
<thead>
<tr>
<th>Requires</th>
<th>What do you gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very little of abstract thinking (no previous training required)</td>
<td>Hands-on experience</td>
</tr>
<tr>
<td>Very little of understanding computers/basic principles (Simon + help during the project)</td>
<td>An initial idea about the programming process</td>
</tr>
<tr>
<td>No practice – this you gain during the project by trial and error</td>
<td>(algorithms, computers as programmable machines, debugging)</td>
</tr>
</tbody>
</table>

- **Advices:**
  - Failing (and recovering from the failure) is better than an immediate success!
  - FISK – först idén sen koden

Requires

- What do you gain

- Very little of abstract thinking (no previous training required)
- Very little of understanding computers/basic principles (Simon + help during the project)
- No practice – this you gain during the project by trial and error

- Hands-on experience
- An initial idea about the programming process (algorithms, computers as programmable machines, debugging)