Book of Magne

INF222 Crashcourse 2023

Sander Wiig

INF222 Crashcourse for v2023.
Some sections have been adapted from Anya's INF225 notes and course material.

UNIVERSITY OF BERGEN
Faculty of Mathematics and Natural Sciences

Institute for Informatics
University of Bergen
Norway
May 12, 2023

Contents

Preface

The goal of this script is to provide an overview of the course INF222 at the University of Bergen. It is by no means a comprehensive guide to the course, but rather a supplement to the lectures and exercises.
There is probably a whole lot wrong with the text and the code, so if you find any errors, please let me know by submitting a ticket or pull request on GitHub.

I wish you all good luck on the exam.

Contributors

The following people have contributed in some way to this book:

- **Magne Haveraaen** for going over the course outline, and for teaching me what I know.
- **Anya Bagge** for her excellent INF225 notes, and for being a great teacher.
- **Jørn Lode** for his amazing figures that illustrate how states work.
- **Ralf Lämmel** for his book on Software Language Engineering, which much of this book is based on.

1 What is a language

1.1 What is a programming language

A programming language

- is an artifical language(i.e made by us humans on purpose)
- used to tell machines what to do

More formally a programming language is a set of rules that converts some input, like strings, into instructions that the computer can follow. This is of course a very general description and it, therefore, follows that there are many different types of programming languages. IT therefore should come as little surprise that we group languages by features and properties.

1.1.1 Types of languages

There are many ways of grouping languages. They can be grouped by Purpose, typing, paradigm, Generality vs. Specificity, and many more. For now, we're going to group them by paradigm, and Generality vs. Specificity.

Generality vs. Specificity

Languages are usually grouped into two categories when based on their specificity.

- DSL
- GPL

Domain Specific Languages are as the name suggests languages with a "specific" domain. DSLs usually have limited scope and use. Examples are JSON and SQL. A Domain-Specific Language is a programming language with a higher level of abstraction optimized for a specific class of problems. Optimized for a certain problem/domain. DSLs can be further subdivided into external DSL(separate programming languages), and internal DLS(language-like interface as a library.)

General Purpose Languages however are more general and can be used to solve many different problems in many different situations. These languages have a wide array of uses and are usually what we think of when we hear the words programming language. Examples of GPLs are Java and Haskell.

Characteristic	**DSL vs. GPL**
Domain	DSLs have a small and well-defined domain
Size	GPLs are large, DSLs are usually small
Lifespan	GPLs last for years to decades, DSLs typicaly live for shorter periors.

Figure 1: Some more comparisons between GPLs and DSLs

Paradigm

We can also classify languages by programming paradigm, some of these are;

- Imperative Languages, i.e C
- Functional Languages, i.e Haskell
- Object-oriented Languages, i.e Java, C#, C++
- Logic Languages

Write something about this also *Write body*

Syntax and Semantics

All programming languages have two parts; the **Syntax**, and the **Semantics**.
Syntax is the study of *structure*, just as semantics is the study of *meaning*. Or in other words the syntax tells us *how* to write legal programs, the semantics tells us *what* those programs do.

1.2 Meta Programming

One of the harder things in the course is **Meta-Programming**. INF222 is usually the first time you've encountered meta-programming and it can be hard a hard concept to grasp. Meta-programming is programming *about* programming. More properly meta-programms treat other programms as data. When you see a datastructure like **Basic Signature Language** or **Basic Imperative Programing Language** in Haskell it represents a program.