



INF 5110: Compiler construction

Spring 2023

Handout 1

30. 01. 2023

Handout 1: About the lecture

Issued: 30. 01. 2023

The lecturer of this course is

lecturer	office	web	email
Martin Steffen	9th floor	https://martinsteffen.github.io	msteffen@ifi.uio.no

The *webpage* of the course is found under

<http://www.uio.no/studier/emner/matnat/ifi/INF5110/v23/>

It contains information about the course, slides will be available, as well as exercises, announcements, info about obligations, etc. A proposed weekly schedule is already there, but it may undergo adaptations as we go. If slides for forthcoming lectures have not yet been uploaded, you may find similar slides under the corresponding link for earlier versions of this course, as there will be a large overlap (but also adaptations).

Literature

Compiler construction is a classical, central, and mature field in computer science. It is, on the one hand, based on deep and fundamental principles, and on the other hand, it is obviously of great practical importance (and will remain so). There is, consequently, a wealth of material, ranging from basic graduate-level textbooks to advanced monographs. Despite being well-established, it is still a field of active research and new results concerning various advances in theory and practice are constantly published.

A book I can recommend is [Cooper and Torczon, 2004] for a modern presentation. A lot of material is taken for this course is taken from [Louden, 1997], which is fine too. Actually it makes not too much of a difference for the material covered. The course covers anyway mostly canonical material and [Cooper and Torczon, 2004] seems a bit more modern (and also more comprehensive if one wants to read more background). In any case, we won't cover the full books but will make a selection. The other book [Louden, 1997] is available also on the web, see

<http://www.cs.sjsu.edu/~louden/cmptext/>,

which contains also further information and especially a *list of errata* for various editions of the book.

The good news is, since compiler construction at the level we are encountering it, is to a certain extent stable and well-established, as said, the concepts, terminology, the structuring of

a compiler in various phases, the underlying theory, even notation, etc. is more or less *standard*. So, what we will teach as parser, or abstract syntax tree etc., is pretty much standard across various books (modulo details and perhaps notational conventions) and across various compiler courses for computer science across the world. So feel free to consult also other books or sources, as there are many available.

One classic book still worth a look is “the Dragon book” [Aho et al., 1986], many generations of computer science students learned compiler concepts from that textbook (and still do). It contains more theory than [Louden, 1997] and treats the theoretical background in more depth. We might take material from that book when covering code generation (and make the material it available).

A book that gave me a lot (for an different course I gave earlier) is [Appel, 1998b] [Appel, 1998c] [Appel, 1998a] when designing concretely the *structure* of a compiler (including interfaces between different “stages”). The book exists for 3 different (flavors of) programming languages (C, ML, and Java). At least in the first part of the book(s), it is less concerned with, for instance, teaching all different nuances of technologies how one could solve, for example, the parsing phase, but making explicit *one* particular (and standard) concrete choice for various phases.

References

- [Aho et al., 1986] Aho, A. V., Sethi, R., and Ullman, J. D. (1986). *Compilers: Principles, Techniques, and Tools*. Addison-Wesley.
- [Appel, 1998a] Appel, A. W. (1998a). *Modern Compiler Implementation in C*. Cambridge University Press.
- [Appel, 1998b] Appel, A. W. (1998b). *Modern Compiler Implementation in Java*. Cambridge University Press.
- [Appel, 1998c] Appel, A. W. (1998c). *Modern Compiler Implementation in ML*. Cambridge University Press.
- [Cooper and Torczon, 2004] Cooper, K. D. and Torczon, L. (2004). *Engineering a Compiler*. Elsevier.
- [Louden, 1997] Louden, K. (1997). *Compiler Construction, Principles and Practice*. PWS Publishing.