Handout 1: About the lecture

Issued: 19. 01. 2016

The lecturers of this course are

<table>
<thead>
<tr>
<th>lecturer</th>
<th>office</th>
<th>phone</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martin Steffen</td>
<td>8162</td>
<td>22 85 0417</td>
<td><a href="mailto:msteffen@ifi.uio.no">msteffen@ifi.uio.no</a></td>
</tr>
<tr>
<td>Birger Møller-Pedersen</td>
<td></td>
<td></td>
<td><a href="mailto:birger@ifi.uio.no">birger@ifi.uio.no</a></td>
</tr>
<tr>
<td>Stein Krogdahl</td>
<td></td>
<td></td>
<td><a href="mailto:steinkr@ifi.uio.no">steinkr@ifi.uio.no</a></td>
</tr>
<tr>
<td>Eyvind Axelsen (obligs)</td>
<td></td>
<td></td>
<td><a href="mailto:eyvinda@ifi.uio.no">eyvinda@ifi.uio.no</a></td>
</tr>
</tbody>
</table>

The webpage of the course is found under

http://www.uio.no/studier/emner/matnat/ifi/INF5150/v15/

It contains information about the course, slides will be available, as well as exercises, announcements, info about obligs, 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 corresponding 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.

Our course is based on the textbook [Louden, 1997], which is one of many possible textbooks. The book 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" \cite{Aho1986}, many generations of computer science students learned compiler concepts from that textbook (and still do).\footnote{And many textbooks published took the dragon book as inspiration and repackaged it as “compiler construction in XXX” (add the favorite language of the day/at your faculty).} It contains more theory than \cite{Louden1997} and treats the theoretical background in more depth.

A book that gave me a lot (for an different course I gave earlier) is \cite{Appel1998b} \cite{Appel1998c} \cite{Appel1998a} 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.

Another book (available in the library) I consulted when preparing the lecture and which made a good impression was \cite{Cooper2004}. Anyhow, as far as books go: we mainly rely on \cite{Louden1997}. In case some chapter(s) of other books are included in the pensum, we will explicitly state so and provide the material.

\textbf{References}


