TDDD55 Compilers & Interpreters
TDDB44 Compiler Construction

2018

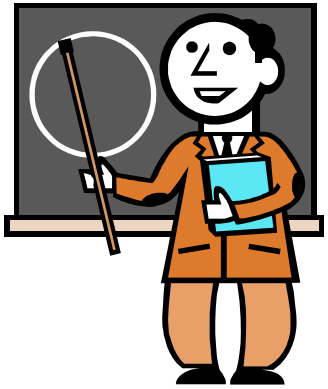
Organizational Issues

Martin Sjölund, IDA

Staff 2018

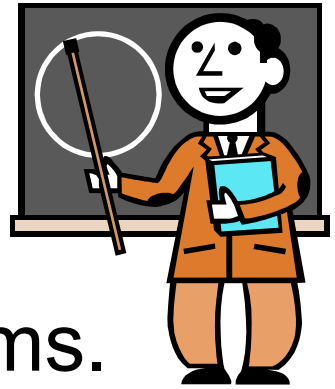
- Martin Sjölund (MS), Examiner, Lecturer, Tutorial assistant TDDB44/D55, Lab assistant TDDD55
- Jonas Wallgren (JW), Lecturer, Lab assistant TDDB44
- Jenny Lönn, Course secretary TDDB44
- Mikaela Holmbäck, Course secretary TDDD55
- Ola Leifler, Director of studies

Lecture Plan



- F1: Introduction (MS)
- F2+3: [opt. f. TDDB44] Short introduction to formal languages and automata (JW)
- F4: Lexical analysis; Symbol tables (JW)
- F5: Parsing; Top-Down Parsing (JW)
- F6: Top-Down Parsing cont., Bottom-Up Parsing (intro) (JW)
- F7: Bottom-Up Parsing [LR(0) items opt. f. TDDD55] (JW)

Lecture Plan (cont.)



- F8: Semantic analysis and internal forms. Syntax-driven translation. (PF)
- F9: Memory Management; Run-time organization (PF)
- F10: Code optimization (PF)
- F11: Code generation, general (MS)
- F12: [opt. f. TDDD55] Code generation for RISC and superscalar processors (MS)
- F13: Error management. Interpreters (MS)
- F14: Bootstrapping. Compiler Generators (MS)

Lessons/Tutorials

5 for TDDD55

4 for TDDB44

- Exercises on background theory (TDDD55)
- Preparation for the laboratory assignments
- Exam preparation session

Laboratory Assignments

- Separate for TDDD55 (2hp) and TDDB44 (3hp)
  - TDDD55: 1 lab group
    - Martin Sjölund (1)
  - TDDB44: 2 lab groups
    - Jonas Wallgren (2)

- Teams of size 2

- Register via webreg (linked from the course homepages)
  - Deadline for registration: Saturday November 17, 2018

Rules for examination of computer labs at IDA

- You are expected to do lab assignments in group or individually, as instructed for a course. However, examination is always based on individual performance.
- **It is not allowed** to hand in solutions copied from other students, or from elsewhere, even if you make changes to the solutions. If there is suspicion of such, or any other form of cheating, teachers are obliged to report it to the University Disciplinary Board.
- **Be prepared to answer** questions about details in specific code and its connection to theory. You may also be asked to explain why you have chosen a specific solution. This applies to all group members.
- **If you foresee problems** meeting a deadline, contact your teacher. You can then get some help and maybe the deadline can be set to a later date. It is always better to discuss problems, instead of, e.g., to cheat.
- **Any kind of academic dishonesty**, such as cheating, e.g. plagiarism or use of unauthorized assistance, and failure to comply with university examination rules, may result in the filing of a complaint to the University Disciplinary Board. The potential penalties include suspension, warning.
- Policy for handing in computer lab assignments at IDA
- **For all IDA courses** having computer lab assignments there will be one deadline during or at the end of the course. If you fail to make the deadline, you must retake the, possibly new, lab course the next time the course is given.
- **If a course deviates** from this policy, information will be given on the course web pages.

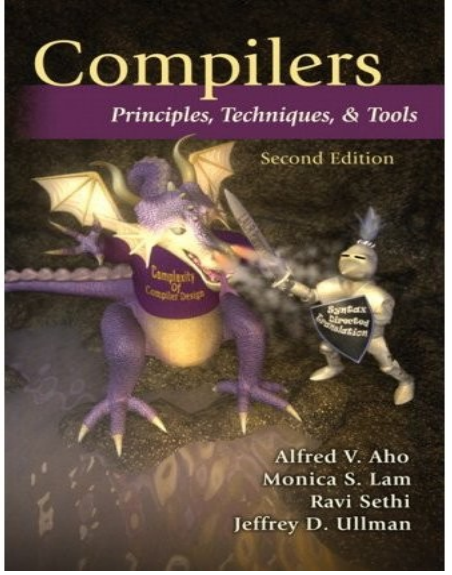
Policy for handing in computer lab assignments

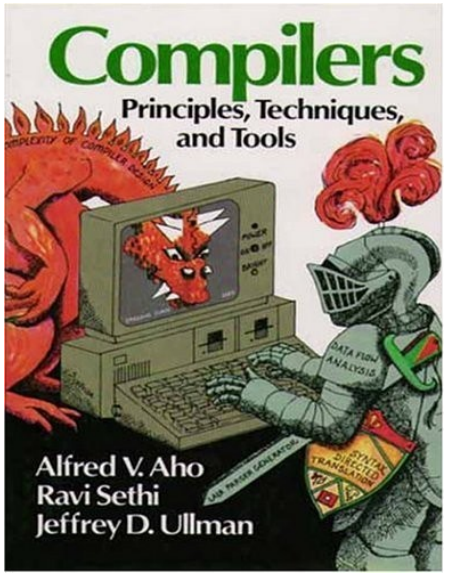
- **For all IDA courses** having computer lab assignments there will be one deadline during or at the end of the course. If you fail to make the deadline, you must retake the, possibly new, lab course the next time the course is given.
- **If a course deviates** from this policy, information will be given on the course web pages.

- Lab deadline: December 21 2018 (for TDDB44 bonus points)
- Recommended last day to hand in final labs is the last scheduled lab: 2018-12-20 (TDDB44) / 2018-12-20 (TDDD55). (Your teacher might be unavailable for example due to travelling)

- If you have only 1 (TDDD55) or 2 (TDDB44) labs remaining, it will be possible to correct the labs until the end of the exam period. After that, you will generally need to wait until the next time the course is given.

Literature





Mandatory (more or less):

- Aho, Lam, Sethi, Ullman: *Compilers Principles, Techniques, and Tools, Second Edition.* Addison-Wesley, 2006. (Also as paperback, 2007; new edition 2013?)
- Or the old, first edition (still ok) Aho, Sethi, Ullman: ..., 1986.

Course homepage:

- Lab Assignments
- Lecture notes
- Compiler Construction Exercises (exam preparation)

Exams

- Most assignments are the same for TDDB44 (3hp) and TDDD55 (2hp)
  - TDDD55: Formal languages / automata theory
  - TDDB55: Using an LR Parser
  - TDDB44: Creating an LR Parser
  - TDDB44: RISC/CISC
  - TDDB44: **Extra 3 points** on the regular exam if your labs are completed and approved before the deadline (and this is the first time you are registered on the course)

- For TDDD55 students, the exam can be challenging without adequate preparation. There is an extra tutorial session to prepare you for some of the harder exam assignments.
- The TDDB44 lab series also prepares you for the exam since the labs are much more extensive. Our experience is that TDDB44 students have an easier time preparing for the exam despite the assignments being more difficult.

For more information ...

See the course homepages,

- www.ida.liu.se/~TDDD55
- www.ida.liu.se/~TDDB44

- Schedule
- Reading directions
- References to additional literature
- Laboratory instructions
  - The lab skeletons are in /home/TDDD55 or gitlab.liu.se

What comes after this course?

- Join our compiler research team at PELAB and do a ***master thesis project*** in compiler technology!
  - Compiling for OO modeling languages (P. Fritzson, M. Sjölund)
  - Operational semantics based compiler generation (P. Fritzson)
  - Compiler bootstrapping, international open source compiler, www.openmodelica.org (P. Fritzson, M. Sjölund)
  - OO modeling language compilation on parallel machines (P. Fritzson)
  - Compilation & parallel programming on industry clusters (P. Fritzson)
  - Compiling for parallel / embedded systems (P. Fritzson, C. Kessler, M. Sjölund)
  - Code generation for embedded systems (C. Kessler)
  - Debugger technology (P. Fritzson, M. Sjölund)
  - … and more!

pelab
compiler technology