MC921A 2s21 Compiler Construction

Adm

- Theory & Lab sessions: Mon. (21h – 23h) and Wed. (19h – 21h)
- Instructor: Hervé Yviquel
- TAs: Vitória Dias and Guilherme Perrotta

Classroom wall

The course will use Google Classroom for announcements and posts about the course progress. Students are required to closely follow the messages posted on the wall, as they include very relevant courseware information.

Syllabus

Classes will use a set of slides and videos, available through the course agenda. If necessary, additional lecture notes as well as articles discussed in class will be made available. Classes will work asynchronously, and class times will be used for Q&A sessions. Support will be provided by the Professor and the TAs during the Q&A sessions and through Google Classroom.

Slides and videos are intellectual property of the books' authors, instructors or UNICAMP, and cannot be distributed without their previous authorization.

The course will be strongly based on the slides and videos which use material from the following books:

- Andrew Appel. Modern Compiler Implementation in Java.
- Aho, Sethi and Ullman. Compilers: Principles, techniques and tools.
- Keith Cooper and Linda Torczon. Engineering a Compiler.

A problem set from the above books and some of their corresponding solutions are provided as references to the level of questions in the exams. We strongly recommend that the students work on these problems.

Assignments

The final course grade will be based on 7 programming lab projects, and 2 theory exams.

Exams will be assigned through Google Classroom (GC), will start at the same time and day of a regular class, and will end 24h afterwards. The answers of the exam questions should strictly follow these guidelines.

Projects will use the GitHub Classroom environment, where each project has an associated template repository. Students have to pull the assignment templates locally to work, and push it for testing, and before the deadline for grading. The GitHub system will run the tests and automatically compute the assignment grade. To better understand how this process works please have a look at this video.

All test inputs for the projects are open, and there are no closed tests. The correct output for each test is open, and their evaluation will take into consideration not only execution correctness but also performance for some projects.

GitHub will automatically close the submission system after each project deadline, and there will be **no extensions**. Hence, we **strongly** recommend that the student submit its work even if the testing is incomplete.

A link to each project notebook, containing a detailed description of the project, programming guidelines, code snippets, etc. can be found in the appropriate entry of the course agenda.

Grades

Grades will be available at most 15 days after the project/exam due date. Regarding the calculation of the course final grade, the following rules apply:

- **Exams Average (E)**
  - The average of the exams is E = average(Ei), i = 1-2. Exams will start at the beginning of a Theory session day (i.e. 8h00), and must be submitted at most 24h after.
- **Projects Average (P)**
  - Rule for P1-P5 and P7: the grade of each project is computed as Pi = Ki/Ni * 10.0, where Ki is the number of correct tests, and Ni the total number of tests of project Pi.
  - Rule for P6: The grade of this project is computed as P6 = sum(Ci)/N6 + B, Ci = 1 if Ri/Si > 1.1, otherwise Ci = 0, where Ci represents the correctness of the test, N6 the total number of tests for P6, Ri the number of instructions in the output code of the non-optimized student compiler, and Si the number of instructions in the output code of the student compiler. The bonus B = sum(Di)/N6, where Di = 1, if Ri/Si > Wi, Di = 0 otherwise, where Wi is the reference compiler speedup.
  - Rule for P7: This project is optional and its grade only serve as a bonus.
  - The average of the projects is P = average(Pi, i = 1,6) + P7/10.
- **Course Average (A)**
  - The course average before the final exam is: A = E * 0.3 + P * 0.7
- **Final (F)**

    - Students with E < 5.0 are required to take the Theory Final Exam (TF), otherwise TF = E
    - Students with all Pi (i = 1,5) >= 5.0 (after removing the lowest grade) and P6 >= 5.0 do not need to take the Project Final Exam (PF) and PF = P.
    - Students with any Pi (i = 1,5) < 5.0 (after removing the lowest grade) or P6 < 5.0 are required to take the Project Final Exam (PF), where PF = average(Qi), i = 1-6, if Qi >= 5.0 where Qi is the corrected version of project Pi. Otherwise PF = min(Qi), i = 1-6.
    - Final Exam F = min(TF, PF), if TF or PF is smaller than 5.0. Otherwise, F = TF * 0.3 + PF * 0.7.
- **Course final grade (G)**
    - For those who have F < 5.0, G = F, otherwise G = (A + F)/2

Grade review requests must follow the rules below:

- Review requests must be made exclusively through this form.
- Review requests will be received only within 48 hours after the grade is released. After that, it will not be considered.
- The review will be done within 15 days after the request is received, and the result will be informed to the student via his/her DAC/Unicamp e-mail.

If the student misses any exam for personal reasons, it should use this form to upload a handwritten signed letter explaining the situation. Any missed exam will be automatically substituted by the grade in the Theory Final Exam (TF). A second missed exam will have no replacement.

Collaboration policy

Exams are individual assignments, and collaboration for their execution is not permitted. Any violation will be considered fraud.

Projects are group assignments (maximum 2 students per group). Groups can collaborate with the goal of understanding and discussing the assignment solution. Nevertheless, code sharing and copying are not allowed, and will be considered fraud.

Each submitted project will be checked for fraud using automatic tools. Only code from the currently submitted project, and not from previously submitted projects, will be considered for fraud evaluation. If the student takes the Project Final Exam (PF), the code from all his/her projects will be checked for fraud.

Frauds will not be accepted, G = 0.0 will be assigned to everyone involved, and the case will be brought to the Undergraduate Dean.

Module	#	Date	Syllabus	Slides	Videos	Exams	Project Available	Due
1 Lexical Analysis	1	9/8	Tokens and regular expressions	1.1 - 1.3	1.1 - 1.3		P1 Lexer	
	2	11/8	DFA and NFA	1.4 - 1.5	1.4 - 1.5			
	3	16/8	DFA-NFA simulation and conversion	1.6 - 1.7	1.6 - 1.7			
2 Synthatic Analysis	4	18/8	Parser tree introduction and ambiguity	2.1 - 2.2	2.1 - 2.2		P2 Parser	P1 Lexer
	5	23/8	LR(0) introduction and construction	2.3 - 2.5	2.3 - 2.5			
	6	25/8	LR(1) construction, ambiguity, and error recovery	2.6 - 2.8	2.6 - 2.8			
	7	30/8	LL(1) construction, ambiguity, and error recovery	2.9 - 2.11	2.9 - 2.11			
3 Semantic Analysis	8	1/9	Abstract Syntax Tree (AST)	3.1	3.1		P3 AST	P2 Parser
	9	6/9	Holiday	-	-			
	10	8/9	Symbol table and semantic analysis	3.3 - 3.4	3.3 - 3.4			
	11	13/9	Q&A Lab	-	-			
	12	15/9	Exame 1			E1		
4 Code Generation	13	20/9	IR, Trees and DAGs	4.1	4.1			
	14	22/9	Instruction selection maximal munch	4.2	4.2		P4 Semantic	P3 AST
	15	27/9	Instruction selection dynamic programming	4.3	4.3			
	16	29/9	Local register allocation	4.4	4.4			
	17	4/10	Address register allocation	4.5	4.5			
	18	6/10	Holiday	-	-			
	19	11/10	Linear IR and basic blocks	4.6	4.6			
	20	13/10	Stack-frame	4.7	4.7			
5 Data-flow Analysis and Basic Optimizations	21	18/10	Data-flow analysis introduction	5.1	5.1		P5 Codegen	P4 Semantic
	22	20/10	Reaching definitions and UD-chain	5.2 - 5.3	5.2 - 5.3			
	23	25/10	Course evaluation	-	-			
	24	27/10	Holiday	-	-			
	25	1/11	Basic optimizations	5.4 - 5.6	5.4 - 5.7			
6 Advanced Optimizations	26	3/11	Liveness Analysis and interference graph	6.1 - 6.4	6.1 - 6.4			
	27	8/11	Global register allocation	6.5 - 6.6	6.5 - 6.6		P6 DFA	P5 Codegen
	28	10/11	Holiday	-	-			
	29	15/11	Global register allocation with coalescing	6.7 - 6.8	6.7 - 6.8			
	30	17/11	Loop optimizations	6.9 - 6.12	6.9 - 6.12		P7 LLVM	
	31	22/11	Q&A Lab	-	-			
	32	24/11	Q&A Lab	-	-			
	33	29/11	Exame 2			E2		
7 The End	34	1/12	Q&A Lab	-	-			
	35	6/12	Q&A Lab	-	-			P6 and P7
	36	15/12	Theory Final Exam			TF		
	37	20/12	Project Final Exam					PF