Redesign for Computer Systems and Organization at UVA

CS4991 Capstone Report, 2024

Tyler Rasmussen
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
hjy4kh@virginia.edu

ABSTRACT
Among computer science students and educators it is commonly known that lower level computer language courses such as the Computer Systems and Organization classes at the University of Virginia are the most difficult. The difficulty of the course content warrants its separation into three courses instead of two. This evaluation of contemporary computer science curriculums and ABET and other accreditations highlight the overloaded nature of these classes. I propose that material be offloaded from these courses to less rigorous courses such as moving logical operators and bitwise operations to Discrete Mathematics and Theory 1 (DMT1). Future work will involve the exploration of offloading further material from CSO and implementing and evaluating pilot courses.

1. INTRODUCTION

The Accreditation Board for Engineering and Technology (ABET) sets a series of criteria that computer science curriculums must meet in order to be accredited. It is important for universities to meet these requirements in order to maintain credibility and provide an adequate education. Criteria five requires "Exposure to computer architecture and organization, information management, networking and communication, operating systems, and parallel and distributed computing." (ABET, 2022). It is clear that the computer science curriculum must include computer courses closer to the silicon and circuit level. UVA meets these criteria primarily through the Computer Systems and Organization classes CSO1 and CSO2 respectively (UVA CS Department, 2024). However, many students feel that these are the most difficult courses in the CS curriculum. The exceptional rigor of CSO1 and CSO2 often frustrates students and impedes the educational process and learning of the material.

This is in contrast to DMT1, which many students believe to be one of the least rigorous computer science courses. However, both courses require mathematical logic and knowledge of logical operators.

Therefore I propose a change in curriculum to combine CSO1, CSO2, and DMT1 in order to equally distribute the rigor of the CS curriculum and create an improved curriculum.

2. RELATED WORKS

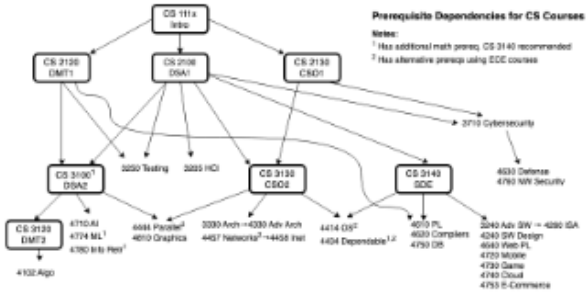
Ghergulescu, et. al. (2014) develops an algorithm to create ideal work hour distributions for CS curriculum modules. The data for the model was collected from undergraduate students in the UK and Ireland. The authors hope to be able to incorporate 21st century skills into the curriculum of the National College of Ireland. If this model is used to influence curriculum at UVA then data of student

hours must be collected in order to properly redistribute the curriculum contents more equally among classes. Blumenthal (2022) is an in depth analysis of how ABETs accreditation affects Bachelors of Science Computer Science (BSCS) curriculum. Lecture hours and credit hours were taken into account and accredited programs and non-accredited programs were examined. Blumenthal found that accredited programs on average required a few more CS related credits than non-accredited programs. It can therefore be concluded that increasing CS related credits is beneficial for a CS curriculum.

3. PROPOSAL DESIGN'

The core of this proposal involves reconstructing DMT1, CSO1, and CSO2 so that the overlap between logical operators is included in DMT1 and leaves CSO1 to start off with ISA's and Assembly. Figure 1 shows the current prerequisites of the UVA CS curriculum. The only change to the prerequisites would be that CSO1 would then require DMT1 as a prerequisite or corequisite.

Figure 1. UVA's CS Course Chart



(UVA CS Department, 2024)

The new proposed DMT1 course would still start with logical operators, rules, and truth tables, but would also include content from CSO1 that uses logical operators (UVA CS Department, 2024). This would allow students to practice bitwise operations and bit fiddling using these logical skills. The next step would be to go into circuits or continue with the DMT1 curriculum in set theory. In this way some of the rigor from CSO1 can be shifted to DMT1. An alternative to placing logical operations and bit fiddling in DMT1 would be to place them in the Introduction to Computer Science course. This would allow for the same curriculum structure as is currently implemented, as depicted in Figure 1. However, according to Blumenthal (2022, p 35:21), it is more common for these topics to be taught in independent courses rather than inside an introductory course. It is better to be cautious of overloading the intro course.

Blumenthal furthermore includes the minimum required lecture hours for Computer Architecture and Organization as 16 hours. However, UVA also includes networking and communication, operating systems, and parallel and distributed programming. This brings the combined minimum lecture hours to 46. Furthermore, UVA includes teaching the programming languages of C and C++ and introduction to version control in these courses further exacerbating the required hours for these topics.

During a lecture, a CSO2 professor once stated "We are covering this topic in class because it is required and we didn't know where else to put it" (paraphrased). This demonstrates how overloaded CSO1 and CSO2 are. They are not only focused on Computer Architecture and Organization but also related topics such as networking, cybersecurity, parallel computing, and operating systems.

Figure 2. Recommended minimum LectureHrs, KUs, and CEUs for ABET CS 13’

Hrs, KUs, and CEUs for ABET CS 13’

ID	Knowledge Area	Lecture Hrs.	Elective KUs	CEU
AL	Algorithms and Complexity	28	3	1.3
AR	Architecture and Organization	16	3	0.74
CN	Computational Science	1	5	0.05
DS	Discrete Structures	41	0	1.91
GV	Graphics and Visualization	3	6	0.14
HCI	Human-Computer Interaction	8	8	0.37
IAS	Information Assurance and Security	9	7	0.42
IM	Information Management	10	9	0.47
IS	Intelligent Systems	10	8	0.47
NC	Networking and Communication	10	1	0.70
OS	Operating Systems	15	6	0.70
PBD	Platform-Based Development	0	5	0.00
PD	Parallel and Distributed Computing	15	7	0.70
PL	Programming Languages	28	11	1.30
SDF	Software Development Fundamentals	43	0	2.00
SE	Software Engineering	28	9	1.30
SF	Systems Fundamentals	27	1	1.26
SP	Social Issues and Professional Practice	16	7	0.74

(Blumenthal, 2022 p 35:21)

By moving some of these topics from CSO1 and CSO2 more lecture hours can be dedicated to more complex computer organization topics students commonly struggle with such as page tables and cache tables. Less rigorous CS courses such as DMT1 can dedicate more time to logical operators and bitwise operations to more evenly distribute the rigor of the CS curriculum.

4. ANTICIPATED OUTCOMES

By redistributing topics in CSO1 and CSO2 into other CS classes such as DMT1 at UVA professors will be able to dedicate more lecture time and material to difficult topics. As a direct result students will be able to spend more time on difficult topics and increase their comprehension. This will allow for better grades and greater educational satisfaction for students. Furthermore, fewer limited resources such as office hours will be needed, providing increased access for students in extra need of help.

Implementation of this proposal will also result in reduced student fatigue and improved mental health. One friend switched majors from CS to Systems Engineering in order to avoid taking CSO2. It is evident that the overloaded nature of these two courses negatively affect the mental health of students. Students who do not take into consideration the rigor of these courses often struggle in their course load and schedule. The anticipated outcome of these changes to the UVA CS Curriculum is the mitigation of these negative student experiences.

5. CONCLUSION

This proposal advocates for a strategic overhaul of the CSO curriculum at UVA to alleviate student struggles and enhance learning outcomes. Reorganizing course content and distribution, will foster a more equitable academic load, improve comprehension, and alleviate student stress, ultimately elevating the quality of computer science education at UVA.

6. FUTURE WORK

The next phase of this proposal involves the exploration of other potential ways to offload material from CSO1 and CSO2 as well as develop and implement pilot courses that reflect the proposed changes in curriculum structure. This pilot phase will provide valuable insights into the feasibility and effectiveness of offloading course material from CSO1 and CSO2 to other CS courses. The pilot courses will then be evaluated with feedback from instructors and students. This approach ensures that any adjustments made are evidence-based and contribute to the ongoing improvement of the computer science curriculum at UVA.

REFERENCES

Blumenthal, R. (2022). Alignment among Normative, Prescriptive, and Descriptive Models of Computer Science Curriculum: The Effect of ABET Accreditation on CS Education. ACM

Transactions on Computing Education, 22(3). https://doi.org/10.1145/3513141

Ghergulescu, I., Stynes, P., & Pathak, P. (2014). A model for designing learning experiences for computer science curriculum. 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Frontiers in Education Conference (FIE), 2014 IEEE, 1–4. https://doi.org/10.1109/FIE.2014.7044100

ABET. (2022, March 12). 2023-2024 criteria for Accrediting Computing Programs. ABET.org. https://www.abet.org/wp-content/uploads/2023/05/C001_CAC-Criteria_2023-2024.pdf

UVA CS Department. (2024, April 9). Home. UVA CS Advising Guide. https://uvacsadvising.org/