

CS151 Final Course Reflection

Hunter Guru and Anthony Weng

Spring Quarter, 2019

1 On Lecture

As a principal component of instruction, we felt like lectures in this class were generally very good. Material is introduced at a very digestible pace and made concrete with the use of numerous examples. We particularly appreciated the escalating style of examples; after the notation-based definition of new concepts and terms, very simple examples typically followed. Only after these simple examples were presented and mastered were more involved examples presented, a teaching style which we believe thoroughly helped with the learning in this course.

One critique we have regards the lectures of the latter portion of the quarter; specifically, Vinay's lectures. Although Vinay's lectures had interesting content, they felt disconnected from the course; at times, we wish that we utilized the time to continue building onto concepts as we had done earlier. As another possible alternative –similar to the class structure of CS106X in which we also had a significant final project– we think ending the course a week early and letting students work on their final projects would help future students develop higher-quality and more complete projects.

2 On Lessons, Readings, and General Material

An underutilized but valuable part of the course were the fleshed out notes, lessons, and readings. The readings were most useful in the earlier weeks of the course when we were familiarizing ourselves with the concepts in logic programming. The lessons and the interactive exercises were extremely helpful in solidifying our mastery over logic programming concepts. In future teachings of the course, it would be a great idea if more emphasis/visibility were given to these auxiliary resources. Additionally, the Epilog documentation was also extremely useful throughout the quarter.

As for general material, we spent the majority of our time learning logic programming through Sierra, which was a very worthwhile pedagogical tool, but we also think it would've been cool if we had received a formal introduction to Prolog through SWI Prolog. When push comes to shove, it seems like there are many logic programming prototypes built on SWISH, and knowing how to use

its libraries, features, and the general logic programming structures would more easily allow us to transition the intuition we spent sharpening throughout the course to more tangible projects; perhaps in the future, we could use the last week of class to learn some of these cool features. Vinay's article <http://web.stanford.edu/~vinayc/logicprogramming/html/prolog.html>, was really interesting, and we thought it would've been worthwhile to cover via additional demonstrations and lectures.

3 On TAs, Office Hours, Piazza (External Learning Resources)

As a whole, these external learning resources were very helpful. Questions on Piazza were usually answered very quickly, which greatly helped when we were working on assignments. There were often times when we regretted bashing our heads against a wall for hours trying to understand how to implement something, when simply asking for help on Piazza would've helped us far more. Piazza often pointed us in the right direction, giving us just the right amount of help, so we knew where to start building our intuition, without giving the answer away.

The TA (Abhijeet) was excellent – he was very accessible and enthusiastic about the material and was very willing to offer his time to help us and other students. The only comment here is that office hours were often understaffed; we often had to wait ~ 3 hours to have our questions addressed. While we appreciate the collaborative nature of the open-forum style of office hours facilitated by having everyone discuss their questions together, sometimes we like the option of just discussing our questions and leaving so we can continue working on the assignment; often times we felt as though we were bottle-necked by others' questions, and their inquiries didn't really spur conversations which helped our implementation issues. Additionally, the way that assignments were scheduled (they were generally due during the last day of the weekend) often prevented us from getting help at office hours before submitting our assignments; we think that it would be a good idea to get another set of office hours on Monday to allow students to work over the weekend and have the ability to ask some questions. Accordingly, the assignment due dates could be moved to Tuesday/Wednesday.

4 On Assignments

As a preface, the assignments of this course are where we have the most critique. We found all of the assignments very interesting and of distinct pedagogical benefit from lectures (i.e. assignments taught us things that we did not learn in lecture, rather than just being an exercise of the examples provided in lectures).

However, a recurring theme for the assignments was ambiguity regarding what was expected if we were to write a logic program, data/ruleset, etc. While we understand that this ambiguity may have been intended to provide us with

a certain degree of freedom, we believe examples in assignment prompts would go a long way in preventing unnecessary confusion. For example, simply the phrasing of Assignment 1's meta-vocabulary question caused us and the rest of the class a lot of confusion, and the way that the General Gameplayer question was phrased was also very confusing, as we thought we had to build a player for the tic-tac-toe rules we had generated.

Also —and this is our most significant critique of the course— the timing and scaling of assignments was not *at all* what we expected. CS 151 was/is advertised as a 3-unit course, but the hours we spent on the course scaled almost exponentially on a biweekly basis (perhaps 2.5^n where n is the index of 2-week periods that have passed). By Assignment 3, we were spending at least 15 hours of our time together, just grinding out the assignment. In terms of a lack of workload continuity between assignments, we believe the transition between Assignments 2 and 3 was the most difficult to manage. For future administrations of the class, we think either petitioning for CS 151 to be a 4-unit course or distributing smaller, but more frequent assignments (perhaps on a week-and-a-half basis) would be a great solution to this problem.

Additionally, we think that the assignments would be a lot more digestible if there was more variety amongst the problem set questions. We often found that the Zebra Problem, Blocks World, etc. were all very heavy "puzzle" tasks; the final project really opened our eyes, as we were able to use logic programming on practical problems that we found interesting; it really would've been interesting and motivating if there had been a computational law problem on one of the assignments (something that shows the applications of logic programming in a non-academic way).

It would also be a good idea for us to be able to get an idea of how well we are relatively performing in the course. Although we did receive our scores for Assignments 3 and 4, it would've been nice if we actually received our score for Assignment 3 before we submitted Assignment 4 (the style of mistakes we made on Assignment 4, perhaps may not have surfaced if we had received a flag on the mistakes we had made in Assignment 3). Additionally, simply receiving a percentage grade for our assignments isn't necessarily a good metric either, because we aren't sure how well we are performing in regards to the rest of the class. It would have been nice if the grade distributions were more clearly laid out, so that we could get a better idea of where we simply stood, especially in a course with no midterm exam.

5 Conclusion

We believe that CS 151 was an excellent course, in which we both learned a lot. Although we both have very different interests, (Anthony is interested in Economics and industry, while Hunter more so finds research and academia in Computer Science more interesting) both of us found things that we could takeaway and apply in our respective areas. CS 151 has the potential to be one of the more popular introductions to the theory track of the computer science

major, as it requires students to couple the course material with both practice (via the programming component) and direct theoretical reasoning (via logical rigor). It also is a phenomenal introduction to a more "analytical" sort of artificial intelligence, in which we got to build expert systems, through various if-then formulations. Please let us know if you would like any more feedback; we'd be happy to discuss any of our comments at more length.