Feedback MCS project part 1 (2017-2018)

Simon Marynissen

January 10, 2018

1 Global comments

Happy new year, and best wishes for 2018! This year, we received 124 submissions for the first part of the project. Out of those, 100 passed, 22 failed (with a non-zero score) and 2 received a zero. The average grade on the project was 14/20, with grades ranging from 0 to 20. A distribution of the different grades can be found below.

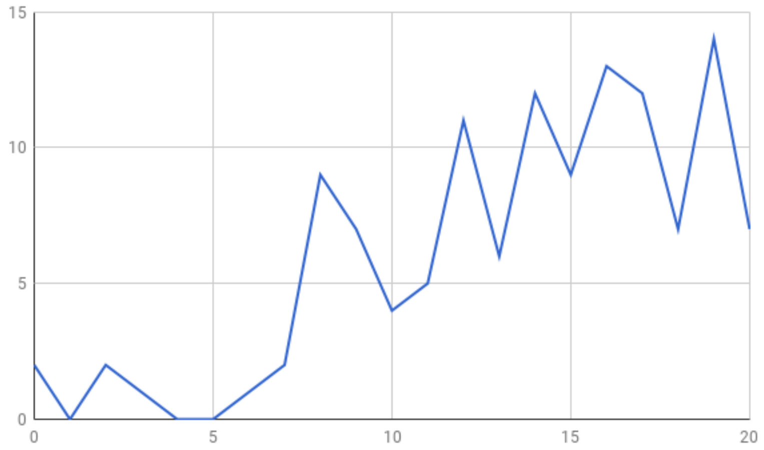

Figure 1: The Universe

Grading consisted of three different parts:

- 60% of your score was determined by correctness of the main theory. This part of the project is entirely tested by automated testing procedures.

- 30% of your score was determined by the verification you had to write.
- 10% of your score was determined by quality (readability) of your code.

2 Correctness (60%)

In order to test correctness of the main theory, we created some tests for your theory. These test can be classified in two categories: SAT and UNSAT tests.

For the SAT tests, we augmented your theory with some extra restrictions to test if specific scenarios are possible by your theory. For instance, sat test 4 tests if a particular game can end, and sat test 3 checks if it is possible to return to the initial positions after some moves. There are 12 SAT tests.

For the UNSAT tests, we tested specific scenarios that should not be possible. Several of them are of the form "the students' definition of predicate x differs from our definition of predicate x". For instance the test UNSAT 1 checks if the colouring is correctly defined. Other of these UNSAT tests test if a given constraint is not respected. For instance UNSAT test 9 checks if there is a row moved that contains a locked block. There are 12 UNSAT tests

The final score for this bunch of tests is determined as follows: both for the UNSAT and SAT tests, a score on 6 is calculated: 0.5 per passed test. The lowest of these two scores is taken as the final score. The reason for this type of grading is that the empty theory will always get the maximum score on the SAT tests, but a zero on UNSAT tests, while trivially unsatisfiable theories (for instance, a theory containing a contradiction), always get zero on the UNSAT tests, but the maximum score on the SAT tests. This way of grading ensures that these edge cases get a zero score, while good theories get a high score.

2.1 Remarks

We initially ran this test with a time limit of 3600s and a memory limit of 4gb. For several solutions these limits were not high enough. Since efficiency was not a grading criterion, we then raised the limits to 8gb. However, for some solutions this turned out to still not be enough. For these rare cases, an individual grading has been manually made.

3 Verifications (30%)

The average point for this part is 10.5/20. In verifications 1,4 and 5, you needed to check whether you could find a model with the property. In verifications 2,3 and 6, you needed to check whether you could NOT find a model which violates the property. (Think about the difference between: You can win the game (which you can prove by giving an example) You always win the game (which you can prove false by giving a counter-example, an example doesn't tell you anything) If your procedure was wrong, you get a zero for this question. Otherwise this is 1 point for each verification.

3.1 Remarks

A lot of students did not return true/false in the verification procedures, although, it was mentioned in the project description and commented inside the procedures in the provided skeleton. Fortunately, no points were subtracted for this.

4 Readability (10%)

The last bunch of points was given on the basis of quality of your code and documentation. Contrary to the other 90%, this is a subjective criterion. We especially payed attention to documentation of the code. For instance: is it explained in text what certain constraints or definition are intended to mean? In general, this was ok. The average score was 14/20. However, occasionally, we encountered submissions where the comments were restricted to

// definition of predicate x

before each definition or submission in which auxiliary predicates/types came without intended interpretation. In such cases, it is hard to understand the code and hence, not a lot of points were given to such submissions.

5 Further questions

In principle, we assume that this document and the model solution should suffice to make you understand why you obtained a certain score. In the rare case where this does not suffice, you can send an e-mail to `simon.marynissen@cs.kuleuven.be` and `ingmar.dasseville@cs.kuleuven.be`. If you want your e-mail to be taken seriously, please make sure your message clearly shows you have read this document, did an effort to understand where you went wrong and ask specific, directed questions. E-mails simply stating "I believe my solution is good and only got X/20. Why is this?" will not be answered.

Good luck with the exams!

Simon Marynissen
January 10, 2018