

# Homework 11 - Analysis Revision

Andrew Carter and Beryl Egerter

April 15, 2013

## 1 Overview of Claims & Introduction to Analysis

We will show that a student used different strategies to decide which order to look at functions between two problems. For an evaluation problem, the student evaluated the code in the order it would have been executed. For a debugging problem, the student started looking at the code at the top of the paper and continued down until reaching a likely candidate for the location of the bug. For this student, each strategy appear to help him understand unfamiliar constructs, which then allows him to solve the problem. We include two partial transcripts from a debugging and an evaluation problem to back these claims. The student whom we have transcribed has taken a couple of computer science classes at Harvey Mudd College, including Principles and Practices of Computer Science, Data Structures and Program Development, and has tutored extensively for the Data Structures and Program Development class.

## 2 Play-by-Play

### 2.1 Q1

1 Alright so, its got a main, so that gonna start. [...]

The student is going to start with the main function, which gets called at the bottom of the script.

Omitted is the transcript of the student continuing to follow the execution until the time he reaches function 2.

2 Function 2 returns function 1 with the same two arguments already passed to it,

3 and function 4, the result of -

The student is evaluating function 2. He notes that there is function 4, and he tries to determine what the resulting value should be.

4 Which doesn't have any implied arguments, thats interesting. Umm,

5 thats, odd [pause]

The student is looking for implied arguments to function 4.

6 lets see, so its calling functi-, ooh, it calling function 1.

7 There we go. Uhh, yes, so its calling function with a list a number, the array and 4,

8 and the function 4 as sort of a multiplier.

9 A function to apply.

The student continues to function 1 and notices how the argument is being used. He then deduces the correct semantics for the function call in function 2.

### 2.2 Q4

1 Alright so we have a board class which is gonna presumably represent the connect 4 board [...]

The student relates the board class to the game described in the problem statement.

- 2 it's going to create an array of arrays or a list of lists depending on what you call it in python [...]
- 3 i don't know if, if lists in python are dynamically allocated or not [...]
- 4 so i wonder what this is doing. I guess it would be creating - an array with at least six - er, seven secondary arrays in it [...]

The student mentions multiple possibilities for the data structure of the board member.

- 5 so drop, i guess that is where you drop a piece in - if column less than self.width um self.board.append player
- 6 oh ok so its doing [starts drawing on paper] like a - array of arrays essentially and then it just pops on a color [...] as they happen [...]

The student figures out what the board data structure looks like and how it changes.

- 7 so this is where you would need to do the check [points at drop function with pen]

The student says the drop function is where the bug could be fixed.

### 3 Interpretation

All line numbers in the following paragraphs refer to line numbers in the transcript of the corresponding problem.

The student is evaluating in execution order in Q1. The student is unfamiliar with the semantics of passing a function as an argument in Python. Initially, as demonstrated on line 3 of Q1, he tries to call the function which led to confusion since there were no arguments. On line 4 of Q1, he looks for implied arguments, further evidence that he is trying to call this function. However, by continuing in execution order to function 1, he is able to notice how the argument he had tried to call earlier is now being used as a function to apply, as stated in line 9 of Q1.

The student is debugging by reading from top down, stopping upon encountering a likely solution, in Q4. Upon reaching the creation of the board data member, the student's lack of experience with Python causes him to be confused about how the physical representation of a connect four board maps onto this data member. As he continues reading top down, the Python list functions used in the drop function help him determine which of the possibilities (previously enumerated on line 3 of Q4) were true for the board data member. This helped him map the physical representation onto the data member. This knowledge helps him fix the bug which had been given to him in relation to the physical connect four board.

The student used two different strategies in the debugging and evaluation question. These strategies helped him move past an unfamiliar concept each time. In other interviews, if the student did not move directly from function 2 to function 1 in Q1, it took much longer to connect function 4 passed as an argument as the applied function in function 1. In Q4, some other strategies we have seen have led the student to get stuck in other locations as they try to relate the code to the physical model.

### 4 Summary

We have shown that a student used different strategies to decide which order to look at functions between two problems. For the evaluation problem, the student evaluated the code in the order it would have been executed. In the process, he was initially stumped by the code passing a function as an argument, but realized what was happening by continuing to follow the path of execution. For the debugging problem, the student started looking at the code at the top of the paper and continued down until reaching a likely candidate for the location of the bug. As he debugged, he was initially unsure about the data structure used as the board data member, but as he continued down, he was able to find code that supported one of the possibilities he had enumerated earlier. For this student, each strategy appeared to help him understand unfamiliar constructs, allowing him to solve the problems.