Assignments and Other Evaluation Components for PoPL 2020

Ramprasad S Joshi

October 30, 2020

1 Introduction

The purpose of take home evaluation components other than routine exams in this course has always been to help you grasp the concepts and help you become independent of textbook and manual memorization. It cannot of course eliminate the need for updating one's knowledge about language specifications or for using manuals as handbooks during programming practice. Often the high level abstract concepts like "object oriented *approach*", or this or that programming *paradigm*, are typically read and said as if they are just symbolic and ritual, devoid of real meaning or relevance, like mottos of organizations. But in our domain, these terms arise in the practice of programming by the most serious and creative programmers. The relevance and meaning precedes coining the abstract terms themselves. And they cannot be understood in their full glory and impact without sharing some of the practice that goes into this development.

Therefore, and regardless of the current situation and the particular mode and methods adopted this semester, I have decided to have those components over and above the exams. However, this semster, we have already experienced the additional stress and the new logistical challenges arising out of increased evaluation components. I am fully aware and concerned.

To balance things, I have decided to plan the remaining month of the semester so as to eliminate the need to prepare and practice for the other exams (T3 and Compre) separately in addition to the assignments component. That is, the content and the preparation will be integrated for all these remaining components.

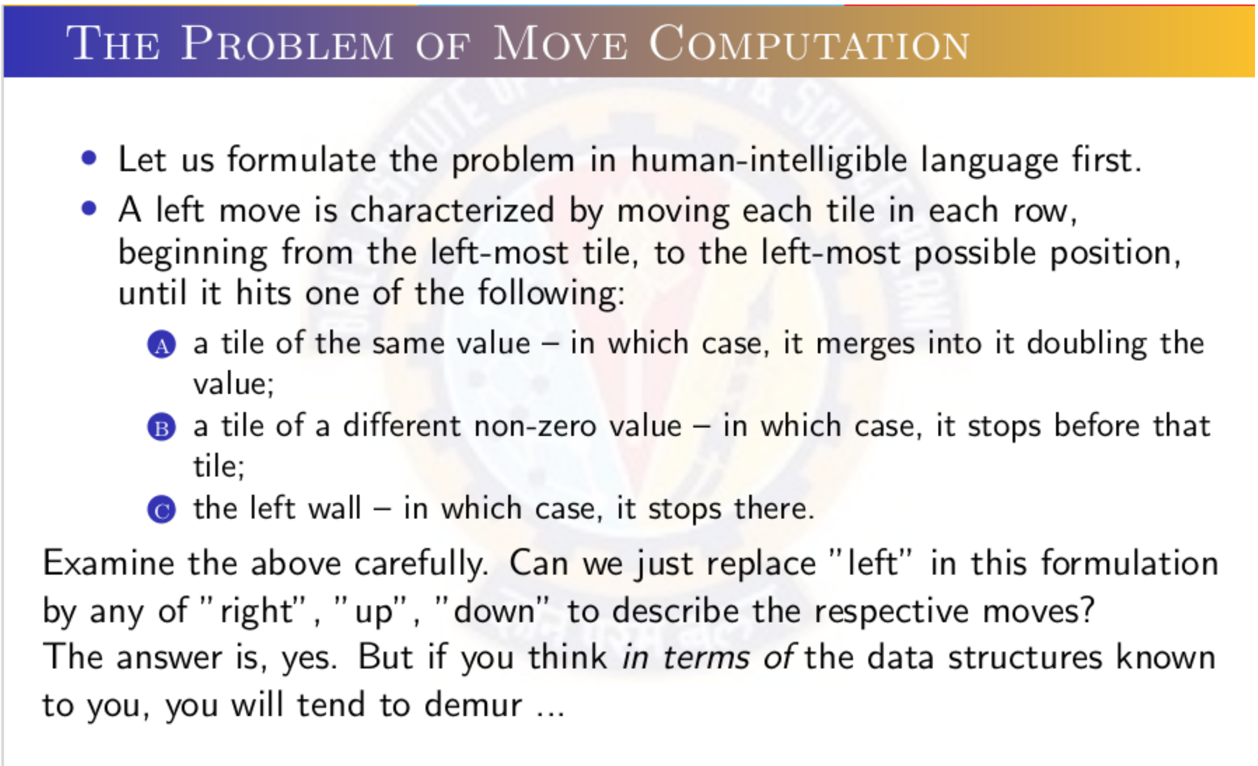
The assignment is around the 2048 thematic game that I had introduced at the beginning. Implementing some of the solutions to the basic problems associated with developing a 2048 game playing agent in various design paradigms is the core task.

2 The Central Problem

Recall our initial discussion on this (Figure 1). The design problems to be solved and decisions to be made for this purpose will be numerous. Some are listed here:

Storage Structure This was discussed at length in the first month of the semester. To decide which of the following: a matrix, or a collection of arrays, or a single array, or some other elementary aggregate data structure (list structures, trees, graphs, etc.): is suitable. This decision depends on some of the considerations that will be known better only after some trial and error, on the criteria that follow here.

In-Place Update Model or Cloning of States? In the larger scheme of things for playing the game, several lines of state sequences possible may need to be examined. Will you generate each line on the stack like $y = f(f(f(f(f(x)))))$ type recursion or you want to be able to create and maintain such lines in dynamic memory beyond procedure activations?



The Problem of Move Computation

- Let us formulate the problem in human-intelligible language first.
- A left move is characterized by moving each tile in each row, beginning from the left-most tile, to the left-most possible position, until it hits one of the following:
  - Ⓐ a tile of the same value – in which case, it merges into it doubling the value;
  - Ⓑ a tile of a different non-zero value – in which case, it stops before that tile;
  - Ⓒ the left wall – in which case, it stops there.

Examine the above carefully. Can we just replace "left" in this formulation by any of "right", "up", "down" to describe the respective moves?
The answer is, yes. But if you think *in terms of* the data structures known to you, you will tend to demur ...

Figure 1: The 2048 Game Programming Problem Core Task

Reusable Code As discussed in Figure 1, do you want to build a type hierarchy such that moves are only combinations of lower level operations on components of the full game state data structure? There will be a trade-off in this. The more of a puritan "OO" design you want to do, reusing everything that is reusable, and separating scopes well, the more will be the flexibility and readability and maintainability of the design. Of course, your effort will be more and the machine level efficiency of the executable generated may take a hit unless you guide the compiler deftly in optimization. In this regard, there are dozens of design choices at each stage.

Functional Design of Operations? Do you want the moves to become concepts separated from the storage structure of the game state? This will be one more higher level, and make the design even more "OO".

Using Functional OO Languages Can you quickly transfer your `C / C++ / Java` OO design to Python or Lisp or Haskell?

3 Process of Development and Evaluation

Make groups, or go solo. The group size is unrestricted, but be prepared to get the credit divided evenly thinly among large groups. I will use my judgment (coming from my industrial experience) to choose how to apportion credit, and will not let you try your usual give and take amongst yourselves.

The first task is to commit to your grouping. This will require you to identify each other and reach agreement on cooperation at a personal level as well as some broad design approaches. I will set up a quiz to record these decisions, sometime next week beginning; by that time you should have finished these discussions and come to some decision. Along with the groupings, you need to also tell me broad contrours of your design. The quiz form is such that each member needs to answer the design decision questions as well as member identification accurately and consistently. This will reduce the possibility of unfair sharing and also of eleventh hour rush towards the semester end and resultant malpractice. This quiz is **not** evaluative.

3.1 Evaluation Scheme

Will be updated. But what I have currently planned is a quiz that will extend the initial grouping and design quiz into an evaluative form. This second quiz, though evaluative, will be more like an automated text-based interview. You will not answer anything therein based on somebody else's authority or understanding, but your own facts and interpretations and insight from your project experience during this assignment.

And you will get proportionate rewards in the T3 and Compre in the form of questions that actually test the insights gained in this process.

4 Conclusion

This assignment spec is open to discussion at 5 pm today (and later too). It is in the spirit of this 2-unit course that is supposed to be a hands-on course in preparation for compiler construction next semester.