Midterm study guide

This midterm will have a more theoretical vibe. You will also be expected to write OCAML code on paper, as well as read OCAML code, but the majority of the exam will be devoted to explaining the theoretical concepts we have been discussing.

1. Be able to describe very precisely the syntax rules and evaluation semantics for all the language constructs we have considered up to this point.

2. Define and describe the differences between the static environment and the dynamic environment. When is each used?

3. Explain the concept of function closure, explain how it differs from functions, show examples of its importance.

4. Be able to follow the process of type inference to determine the most general type for a function.

5. Be able to solve recursion problems on paper, especially list-processing problems.

6. Explain the mechanisms that allow OCAML to treat all functions as functions of exactly one variable, and yet be able to handle functions of "zero" or "two or more" variables.

7. Define when we say that an expression is in tail position. When do we say that a function is tail-recursive?

8. Define and explain the difference between variable mutation and structural mutation.

9. Describe the mechanisms that make "curried functions" possible.

10. What does "shadowing a binding" mean? Explain by an example. How does it differ from overwriting the binding?

11. What are the two fundamental ways of creating new types from existing ones (tuples vs variants)? What are their differences? What are the closest analogs in C++ or Java?

12. What is the use of the "option" type? Give examples.

13. Explain the use of the "unit" type and value $()$.

14. What are differences between OCAML's references and C/C++'s pointers?