

CS225 Programming Languages: Homework 3

Due Date: 11/18/21 by 11:59PM

Submission: Please submit your solutions to Problems 1-3 and 4b as one file, and your completed `bool-sc.ml` file to answer Problem 4a. All submissions must be made electronically in Blackboard.

Problem 1 (15 points). Provide a screen shot of the OCaml editor and interpreter combination you are using for the class.

Problem 2 (20 points). Recall that in OCaml, record fields can be constructed from expressions that need to be evaluated. For example, given:

```
type rt = { a : int; b : int }
```

we have:

```
{ a = 1 + 2; b = 3 * 7 } ⇓ { a = 3; b = 21 }
```

and it must be the case that either `1 + 2` is evaluated before `3 * 7` or vice-versa in any given implementation. Show which is the case on yours by devising an example that reveals it, and copy-paste or provide a screenshot of your interaction with the interpreter that demonstrates the evidence.

Problem 3 (30 points). Suppose you wanted to add an xor operation as a primitive to `BOOL`. To accomplish this, do the following:

- Propose a symbol for xor and extend the `BOOL` language of expressions to admit it as a binary operation.
- Extend the computational and contextual reduction rules of `BOOL` to provide an operational semantics for xor. You can specify either a left-to-right or right-to-left reduction semantics, but your semantics should be deterministic in the sense of Lecture Notes 9, Proposition 1.1.

Problem 4 (35 points). Review the Definition of the `BOOL` interpreter available here:

<https://github.com/uvm-plaid/CS225-public/blob/main/bool.ml>

Answer the following:

- (25 points) Modify the interpreter to support short-circuiting for the `And` and `Or` operations, and submit your code as the file `bool-sc.ml`.
- (10 points) Is your modified interpreter still correct in the sense of Lecture Notes 10, Theorem 1.1? If you say no, provide an example that illustrates the fallacy.