CS 578 Programming Language Semantics Homework 1
Due 4:40pm, Tuesday, Apr. 9, 2024

You should do all problems, but only those marked with an asterisk (*) must be submitted. (You may submit the others for feedback if you wish.)

Solutions may be submitted on paper at start of class, or by email before class.

Preparing your submission using latex is strongly suggested, but if your handwriting is sufficiently legible, hand-written submissions are also acceptable.

Email solutions should be sent to `tolmach@pdx.edu`, with the subject line "CS578 HW1" and a single `.pdf` file as a simple attachment. (If your solution is handwritten, you must scan or photograph it, and produce `.pdf` for submission.)

All programs mentioned can be downloaded via the course web page; the relevant code for this assignment is in the `arith` directory.

1. Do Pierce 3.5.10.

2. Do Pierce 3.5.13. (Note the errata in the sample solution.)

3. Do Pierce 3.5.14. Use structural induction on `t`.

* 4. Do Pierce 3.5.17. Just do one direction ("if" or "only if") – take your pick! (Note the errata in the sample solution.)

5. Do Context Handout exercise 2.

For the remaining exercises, you may choose to use a different language (e.g. Haskell or Scala) in place of OCaml. If so, you are responsible for translating the underlying type definitions (in this case, just `term`) and supporting definitions into that language; you are not required to build a concrete syntax parser.

6. Write and test an OCaml function

```
size : term -> int
```

that implements definition 3.3.2. It will be easiest to put this in file `syntax.ml` and change `main.ml` to print out the size of each input term. Submit just the code for the `size` function.

* 7. Do Pierce 4.2.2. Just change the implementation of the `eval` function in `core.ml`. If no big-step rule applies, raise the `NoRuleApplies` exception; unlike in the small-step case, this should *not* be caught, but should instead propagate to the top level. Submit your changed `core.ml` file.