CS 3MI3: Fundamentals of Programming Languages

Due on Wednesday, December 6, 2023 at 11:59pm EST

Dr. Jacques Carette

Idea

The goals of this assignment are:

1. Do a typechecker implementation where you have to take care of doing unification yourself.
2. In other words, implement Algorithm W for the Hindley-Milner type system.

Logistics

Your laguage will be the lambda calculus (i.e. variables, abstraction aka lambda and application) with let-in, pairs, unit, booleans and natural numbers.

Specifically, the syntactic forms, values, evaluation rules and typing rules from Types and Programming Languages:

- Figures 3-1, 3-2
- Figures 8-1, 8-2
- Figure 9-1 without typing annotation on lambda
- Figure 11-2, 11-4, 11-5, with the modified polymorphic let from section 22.7 of the textbook.

For extra certainty, a `Skel.hs` file is provided with datatype definitions (that you must use as is) and stubs for the functions you need to implement.

The Tasks

1 Typing [60 points]

Take the language as specified above and implement (in Haskell) all the functions whose body is currently `undefined`.

You will need to implement *unification* to do this. You may borrow various pieces (including capture-avoiding substitution) from the tutorials where this helps. Make sure to cite that you do so!

Note that the textbook implements a *more complex* algorithm.

2 Testing [40 points]

As with previous assignments, you should provide a properly documented test suite for your code.

You may put some pseudo-tests in comments for things that do not work (but should) with appropriate comments. Those can be worth part marks as they indicate your partial understanding.

Submission Requirements

- Must be handed in as a `.zip`, `.gz`, or `.7z` file. Other archive formats will **not** be accepted, resulting in a score of 0. The archive should be called `A5_macemailid.zip` (with your email address, I am "carette", substituted in).
- The name of the file **does** matter.
- Code or tests which **produces errors** is worth **0** marks for the code (including testing) portion of the assignment. Let us know which platform you used (as a comment or in a README).
- Marks will be deducted if you have junk in your archive (such as object files, `.DS_Store` files, pointless subdirectories, etc.). Stack project files and cabal files are exempt from this.
- If you looked things up online (or in a book) to help, document it in your code. If you have asked a friend for help, document that too. "Looked things up online" includes all AI tools and the courses' own slides, tutorials, etc. Put this in a README file (as plain text, Markdown, or HTML). (*For this assignment, this is manditory, as it is built-in that you will have to look things up.*)