CSC 453 Milestone 8 (Typechecking)

Goal:

This milestone is to decorate/annotate an Tau AST with semantic types. This phase directly follows creating symbol bindings.

Specifications

The Tau language specification is a separate document.

The AST nodes are defined in `asts.py`.

Symbols and types are defined in `symbols.py`.

Create `typecheck.py`

You will create a new file called `typecheck.py` that will contain the code for annotating the AST and Symbols with their semantic types.

To accomplish this, you will walk the AST built by your parser (after doing symbol binding):

Check `asts.py` for AST nodes that include a `semantic_type` field. Similarly, check `symbols.py` for symbols that include a `semantic_type` field.

Adapt template to create `typecheck.py`

The file `stubs/typecheck_template.py` contains routines for walking the AST. The routines do not do anything other than walk the tree, but they make a good starting point for your code.

I recommend copying `template.py` to `typecheck.py` and modifying it to do what you need.

Notes

The Tau language specification is a separate document. It may not be complete. If you have questions, please ask on Piazza.

Feel free to publically discuss the language specification on Piazza. If you have questions, please ask.

Type Errors

Your type checker should report type errors by throwing an exception with a meaningful error message. This milestone will not explicitly check that you are doing this, but a later milestone will.

This milestone only requires that you correctly annotate a correct Tau program.

Difficulty

This milestone does not require a lot of code beyond the tree walker, which is provided. That said, it can be a little tricky to check and infer types for some nodes.

Start early and ask questions.

Standard Requirements

Your program must meet all the requirements outlined in the common requirements document.