

Final Report for Typerchecker Project

Reminder

Since I didn't completely follow the project instruction, there might be some changes for other files like `syntax`. Please do not immediately grab out `type` and `typing` for evaluation, I recommend the test within this directory directly.

Outcome and Performance

I finally got successful tests for almost all the test files for task T1 and T0. Some file might get error like `shallowing_T01.fw` which used to be right before my implementation of minimization. But most of the T1 task tests should be good enough for **Minimization**, **Well-Formedness** and **Fundamental Typing**. Time is not enough for me to fully implement T2 type function and existential type

My implementation

Basically my implementation contains **Minimization**, **Well-Formedness** and **Fundamental Typing**:

1. Minimization: Here we didn't choose to implement the original `minimize_type` function which stupidly re-calculate the suffices (or so-called id). My approach is completely different: creating new **counter** bindings inside environment which assigns specific counter for each new type variable. Everytime we erect cvar, we only have to use `cvar_new` and `cvar_def_new` function to directly update the counter and generate new internal type variable.
2. Well-Formedness: Here we didn't completely follow the instruction which implies to implement `wf_ctyp` function by implementing `subst` and `subst_typ` first, we directly named another function `new_subst_typ` to finish type reductions and finally integrate it into `wf_ctyp` to help us attain the well-formed types
3. Fundamental Typing: Here we literally follow the instruction, coded `type_typ`, `type_exp` and `type_decl` to ensure the basic running of our testing. Nothing very special to bring up, but lots of non-sense printing message there might influence the beauty and still have some pattern matching incomplete but probably fine with our testing.

Feeling

Very terrible at very beginning since I spent tons of time discussing the content and code with other students. However, I gradually started to get a feeling about the logic and structure implied there. This implementation really benefits me a lot by clarifying a variety of knowledge I couldn't pragmatically understand before on the notes or textbooks. Although I think I would not get a good score on this project, I still honestly love this project and willing to proceed if I get rest time after this semester.