[CPSC 539B] HW4

TIL: a type-directed optimizing compiler for ML

February 2019

1 Critique

The paper by Tarditi et al. is focusing on a new approach to compiling Standard ML (SML) based on 4 technologies: 1) intensional polymorphism 2) tag-free garbage collection 3) conventional functional language optimizations and loop optimization. This preserves types upto 80 percent of the compilation process. This paper is the work right before F to TAL, and here, we can see that the closure conversion is different from how presented in F to TAL.

The important property supported by TIL is that all optimization and key transformations are performed on the typed intermediate languages and this is the reason they have named their compiler approach as TIL. The reason they do so, because its easier to optimize the codes when their type information is available.
I think that the work in this case differs from the work done in F to TAL is after the closure conversion part where we can see that the during the conversion to an untyped language, we annotate the variables with representation information that tells the garbage collector what kind of values the variables must contain. This seems like too much effort.
It felt like a lot of things were missing in the paper, for eg. they mention about the different forms of languages starting Lmli to Ubform. However, they have not provided any syntax or semantics for the same. Maybe they were popular languages and everyone already knows about them.
However, I think this paper was a strong influence on the the F to TAL paper.

2 Open Questions:

I am having trouble understanding the relation between garbage collection and types. In the previous paper (F to TAL), they had made a similar statement. Here too they say repeatedly how important types are during optimization in order to do garbage collection. What are the problems that can be caused in garbage collection?