Report - Phase 4: Intermediate Code Generation

Yongun Seong 2017-19937

For this fourth phase of the project, I implemented a translation routine to convert the parsed and validated AST into an IR, more specifically a three-address-code form. This translation is performed recursively starting at the root of the AST, each node appending its TAC form onto a CCodeBlock handle passed to each resursive call.

Unlike the previous phases, I chose to implement this phase in a top-down manner. That is, I implemented the `ToTac` procedure for the root CAstScope first, and worked my way downwards to each statement and expression type. I did this because in the cae of this IR, the most complex parts were the bottom-most expressions, especially the array indexing node, while the higher-level language constructs were easy to translate.

As with previous phases, I noticed bugs introduced in the previous phases while testing my implementation. Notably, I had introduced a bug while type checking array indexing, and it was rejecting perfectly valid programs. I fixed this bug by allowing array indexing code to treat pointers to arrays the exact same way as the array themselves. Simultaneously, I ensured the `ToTac` procedure correctly handled both pointer and non-pointer accesses to arrays by generating IR like `t <- &A` when working with non-pointers.

I noticed that my implementation's output differs from the reference implementation in some ways. Some of these differences were benign, such as minor divergences in label numbers and names. In other parts, however, I am still unsure of their significance, and may need to further modify my code in the future, especially when we start generating code from the IR. This was especially noticeable in the types around array indexing, where the reference implementation used integers and void pointers as opposed to pointers to arrays as reported by my implementation. Also, I am still not fully confident in my implementation's ability to distinguish integers and longints, so plan on paying special attention to it in future phases.