AutoMATES: Six Month Report

Program Analysis Work Progress

Refactoring of front-end:

The main work progress was done for the refactoring in “Fortran code - XML AST - Python Code” generation process. The refactoring was desired due to the issues with Open Fortran Parser (OFP) generating AST in XML form with no strict commonalities, which caused inefficiency and mess in translate.py as well as the pyTranslate.py to determine the different syntaxes for the same object types. For example, <format> element appears under (1) declaration tag, (2) loop tag, or (3) statement tag disregarding the scope, but simply based on the position of where format appears in the Fortran source code. The new program rectify.py fixes those (1) scope ignoring (2) incommon syntax (3) redundant elements issues and produces a new XML AST for translate.py. Thus, translate.py does not have to handle any cases to fix the above mentioned issues, and focus its main purpose of generating the pickle file for pyTranslate.py. Simply, measuring the progress with the line of XML refactored, the rectify.py reduced the the lines by 30%~40% from the original XML. For example, the original PETASCE_markup.xml holds 4,177 lines whereas the new XML only holds 2,858 lines. The benefit of the refactoring in the following programs is that it reduces the number of recursions and loops that were needed in order to traverse down the nested elements.

Enhancement of GrFN Specifications and added support for new FORTRAN language features in GrFN

- Continued progress in generating GrFN specification files accompanied by its lambda file such that it abstracts the multi-module structure of the original FORTRAN code into a multi-file GrFN JSON data structure.
- The Python-to-GrFN converter now handled multi-module files more elegantly with a separate GrFN and lambda file for each FORTRAN module.
- Started refactoring and cleanup of the code-base of the Python-to-GrFN converter. This will allow for more seamless integration of newer FORTRAN language features to the GrFN JSON and lambda file.
- Laid groundwork for the automatic support for inline comments. This will not necessarily be integrated in the GrFN file and might be provided through other means as required.
- Started integration of pyTorch with the automatically generated lambda files. This integration will allow processing at a much higher speed and scale than previously supported.
- Continued discussion and brainstorming about how arrays and multidimensional data structures can be represented in GrFN among other features such as DELETE, SAVE and GOTO. With such language features being incrementally represented by GrFN, each time we come one step closer to specifying FORTRAN codes in all of its generality.