

CS1319: PLDI - Assignment 4

Hrsh Venket & Santripa Sharma

October 2023

1 Augmenting the Grammar

We begin our efforts on the semantic analyser/TAC generator for nanoC by augmenting the grammar to add attributes to existing symbols, and add new dummy/marker symbols, to help us generate the final code. Towards this end:

- We first define our attributed grammar, specifying which attribute is associated with each terminal/non-terminal symbol. The list of attributes is available at the top of the bison file (in the union), with comments.
- Next, we augment the grammar with some symbols, particularly, the marker and the guard symbol, both of which produce only epsilon, but have associated actions or attributes. A description of the action is available in the bison file alongside each rule.

2 Setting up data structures

We create the various compile-time data structures we require for translation, as well as functions for those structures. The declarations can be found in the header file, along with comments explaining their purpose (when not self-evident).

3 Writing actions

We write semantic actions for filling up our compile-time structures, and generating our translated TAC. One important note, is that in a few rules, we have used mid-rule actions, to either facilitate conversion of non-boolean expressions into boolean expressions (endowing them with true/false lists), used in the case of value-based conditionals, or in function definitions, to allow us to inspect/inherit attributes from sibling symbols in the parse tree.