

AstToAst

Konrad Krämer

15.7.2021

Scope of AstToAst

The aim is to translate a tiny subset of R to Rcpp. The idea is that the user defines a function in R which is translated to an Rcpp function. It is required that the user defines the types of the arguments passed to the function. Thus, the final function **translate** gets two arguments the code itself which defines the body of the function and a vector containing the types. Allowed types are numeric (**num**), a numeric vector (**num_vec**) and a numeric matrix (**num_mat**). These functions can be used e.g. for optimization problems, as ode-function or maybe also other applications.

How does it work

The first step is to separate the different code lines from each other. From each code line the **R abstract syntax tree (AST)** is extracted. Afterwards, each line is checked whether a variable is assigned. Next, the right hand side (**rhs**) of the variable has to be examined. First, a list of the variables found at **rhs** is created. Furthermore, the following functions are allowed (at least at the current state): **+**, **-**, *****, **/**. There exists a precedence: matrix, vector, scalar. Thus, if a variable is found at **rhs** that is a matrix the variable to be assigned (**left hand side = lhs**) with the content of the **rhs** has also to be a matrix. If no matrix but a vector is found the variable at **lhs** has to be a vector. If no matrix or vector is found it has to be a scalar. These tests are important in case a variable switches the type.

Switching the type

Each variable consists of a class which contains a matrix, a vector and a scalar. Moreover, the class contains an int defining the type (**int type**) (0 = matrix, 1 = vector, 2 = scalar). If a switch-case is detected a new line before the line containing the switch is inserted. In this line the data stored in the old type is deleted and the int defining the type is set to the new value. In order to work properly the **=** operand has to be overloaded and during each assignment the **type** has to be checked.

Translating R AST to C++

Afterwards the R AST is translated to C++. First, all variables are defined. Next, the content of the input variables is stored in extra variables due to the fact that R does not know the class type.

Now the code has to be changed in order to enable compilation. The following changes have to be done:

- Change indices to indices - 1
-