# Coding standards

To keep the abstract and concrete PKB API in sync, we created a variable table, statement table and procedure tables. These tables are vectors mapping variable names to indexes of the vectors. So that a API method like BOOLEAN isModifies(STMT\_NUM s1, INDEX varIndex)understands that INDEX is the mapped value of a certain variable name, where INDEX is just an integer in C++ type.

# Revamping for Assignment 1

In Assignment 1 our team has revamped the internal structures of the PKB in order to make the access to the various tables, in the PKB, with a smaller time complexity. Our previous approach was using a vector of integer pairs to store the relationships such as *Follows, Modifies, Uses* and *Parent*. An example of how the *Follow* table was previously represented is shown in Fig. 1. The vector holds elements which are pairs of statement numbers which obey the *Follow* relationship. For example, in index 3, statement number 4 is followed by statement number 5. This way, methods such as getFollowingStmt(4) returned the second element of each element which stores a first element of 4.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1, 2 | 2, 3 | 3, 4 | 4, 5 | 7, 8 | 10, 11 | 13, 14 | 15, 16 | 16, 17 |

Fig. 1.

As explained above, the complexity to retrieve an element from the above data structure is O(n).

In our revised version of the PKB, we have decided to change the internal structure of all the relationship storage to 2D vectors storing boolean. This way the time complexity is now O(1) to retrieve an element from the table. Fig. 2 shows the representation of the *Follow* table after the change of the data structure.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4 | 5 | 6 |
| 4 | 0 | 1 | 0 |
| 5 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 |

Fig. 2.

The vertical and horizontal indexes refer to statement numbers. For example, in this case getFollowingStmt(4) returns the statement numbers which have a TRUE (1) in the horizontal row of index 4. It is statement number 6 in this case.

# Benefits of current architecture in revamping PKB’s internal structure

Due to the abstract class of PKB, any internal changes to PKB does not affect the other components such as the Query Evaluator or the Parser which interacts with the PKB in the program.

# Further Improvements made

To further improve the system, all previous return statements where a vector of integers were returned, are now changed to a reference of the vector. Since the method receiving the vector of integers does not modify the vector, it is more efficient to return a reference. As such, the speed of the program is also improved.