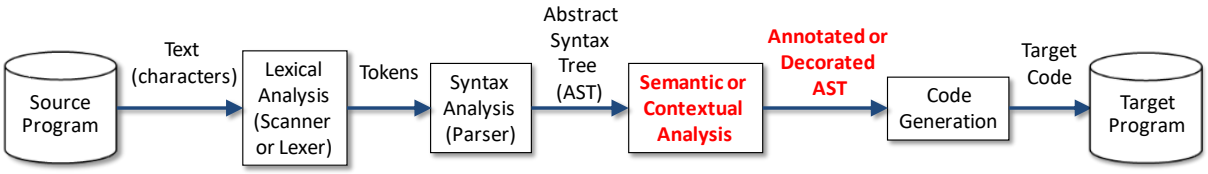
Programming Language Design
Department of Computer Science
Dr. Francisco Ortin
francisco.ortin@mtu.ie

# Laboratory 8 – Type Checking Phase

The objective of this laboratory is to perform type checking of C--. This is the second and last phase of the semantic analysis. The output must be the AST decorated / annotated with type information.




## Description

Our compiler must type-check any C-- input program. All the expression nodes in the AST must be annotated with their type, inferred by the semantic analyzer. If a type error exists, an informative message should be printed in the console, including line and column numbers.

Different case scenarios of wrong programs are provided:

- Input-wrong-1.txt: Type conversions.
- Input-wrong-2.txt: Common operations.
- Input-wrong-3.txt: Function invocations.
- Input-wrong-4.txt: Function invocations.

The following constraints must be fulfilled by the type-checking phase. For expressions:

- The +, -, * and / binary operators can be applied to two characters (returning an integer), two integers (returning int) and two doubles (returning double).
- The % operator can only be applied to two characters and two integers. Int is returned.
- Logical operators (&&, || and !) can only be applied to integers and return int.
- Comparison operators (>, <, >=, <=, == and !=) are applicable to two chars, two integers or two doubles, and they always return int.
- Unary minus can be applied to char (returning int), int (returning int) and double (returning double).
- The cast operator can only be applied to built-in types (void is not included).
- The [ ] operator requires the first operand to be an array and the second one to be an integer.
- For function invocation, the number of arguments must be the same as the number of parameters. The type of each argument must be the same as the type of the corresponding parameter. Recall that parameter and return types must be built-in (void is only valid for the return type).
- Any other combination represents a semantic error and must return ErrorType.

For statements:

- Two expressions can be assigned if they have the same built-in type (void is not included) and the left-hand side expression is an l-value.
- Only built-in types can be read and written. Expressions to be read must be l-values.
- The conditions in while and if statements must be int.
- The type of the returned expression must be the same as the return type declared in the enclosing function.

## Go ahead

Read the code in all the input-wrong.txt files and make sure you understand the semantic error to be displayed by your compiler.

Take the `TypeCheckingVisitor` class implemented in lab 06 and extend it to provide type checking (plus the existing l-value detection). Think carefully about how to design the type system (there exist plenty of combinations of operations and types). Follow the design guidelines explained in the lectures.

Make sure your compiler detects all the semantic errors. For each semantic error, the line and column should be printed, along with the corresponding message.

For input programs with no errors, make sure that types are appropriately annotated for all the expressions in the AST (use Introspector for that purpose).