

COMP4403/7402 Compilers and Interpreters

Tutorial Exercises 1 (2021/1)*

School of Information Technology and Electrical Engineering, University of Queensland

February 11, 2021

For this tutorial you will need copies of the PL0 Concrete Syntax and PL0 Compiler Data Structures as used in the lectures.

1. For the following PL0 program:

```
var
  x: int; y: int; max: int;
begin
  read x; read y;
  if x < y then
    max := y
  else
    max := x;
  write max
end
```

- Describe the sequence of tokens supplied by the scanner (lexical analyser). Note that “int” is a pre-defined identifier, not a keyword in the language. See Figure 1 in the PL0 Concrete Syntax handout for the definitions of PL0 tokens. You don't need to do the entire program.
- Give the parse tree that results from parsing just the **if** statement in the above program, i.e. ignore everything up to and including the reads and everything from the write onwards. The top node in the parse tree should be for the nonterminal *IfStatement* and the tree should be consistent with the PL0 Concrete Syntax (see Figure 2 in the handout).
- Give the abstract syntax tree for the **if** statement within the program. The tree should be consistent with the tree nodes used in the compiler (see PL0 Compiler Data Structures handout).

2. In which compiler module would you expect the following to occur:

- An undeclared identifier is detected.
- An illegal character in the source is detected.
- The use of `']'` instead of `')` is detected.
- Unreachable code is eliminated.
- Run time checks for range overflow are inserted.
- Run time checks for range overflow are eliminated where it can be shown from the context that they are redundant.
- The use of a `','` instead of a `','` in a parameter list is detected.
- The insertion appropriate for a C language `#include file` macro command (which results in textual substitution of the contents of file at this point) is performed. (Warning: this is something of a trick question!)
- A type conflict in an expression is detected.
- A constant too large for the target machine is detected.
- A branch (jump) instruction that has as its destination an unconditional branch instruction is changed to go directly to the destination of the second branch.
- Whether the language has name equivalence or structural equivalence of types is an influence.
- Whether `;`'s are used as separators or terminators for statements.

*Copyright © reserved February 11, 2021

- (n) Code for coercing an integer to a real where this is required in a mixed mode expression (an expression with both integers and reals) is created.
- (o) The appropriate instance of an overloaded method required for a particular call is identified. Overloading of methods is said to occur when the same name is used for more than one method. The appropriate instance is identified using the types of the parameters.