Exercise Sheet 3 for COMP6209 (C++ Template MetaProgramming

March 2016

You should use C++ templates to generate code as detailed below. Please attempt all of the questions before the exercise class and bring your solutions along.

1. Write a C++ template that is parameterised on an integer \( N \), and provides a function bigHello that prints “Hello World” \(!N\) times to standard out.

   Try this both using a loop, and then with static loop unrolling. In either case, the code should statically calculate \(!N\).

2. Write a C++ template \( \text{MIN} \) that statically evaluates the minimum of two integer values. Can we generalise this to \( N \) values?

3. Write a number of C++ templates to represent formulas of propositional logic over a single proposition variable \( p \). That is, provide templates to represent a propositional variable, boolean literals, and, or, implies, and not. In each formula template, provide a static function called \( \text{eval} \) that accepts a boolean and returns the evaluation of the formula with value of the propositional variable \( p \), being the provided boolean.

   Write a template called \( \text{POS} \) that determines whether all occurrences of the propositional variable are in positive positions. To calculate this, we see that a variable \( p \) is positive in formula \( F \) if it occurs under an even number of negations in \( F \). Note that an occurrence on the left hand side of an implication is also a negation.

   Can you generalise this solution to allow multiple propositional variables? Discuss or implement this.