COMPUTER SCIENCE TRIPOS Part II – 2025 – Paper 8

9 Optimising Compilers (tcg40)

The following program in a C-like language is given to a compiler:

```
x = 42
a = x * y
b = a + z
c = b * 2

if (a > 0) {
  int d
  d = d - d
  e = d
}
print (e)
```

(*a*) (1-2) Name the two kinds of data-flow anomaly appearing in the code above. (3) Describe the data-flow analysis that can be used to identify them and (4) its corresponding data-flow equation(s). (5) State if this data-flow analysis is a forward or backward analysis and (6) explain how one can derive the direction (forward/backward) from the data-flow equation. [5 marks]

(*b*) Perform the data-flow analysis from part (*a*) on the code above and mark all data-flow anomalies that can be detected purely from the data-flow analysis without transforming the code. Include all details, e.g., list which properties you had to check to identify these data-flow anomalies. [5 marks]

(*c*) Draw the clash-graph of the code above (without removing any anomalies). Does the graph have specific properties that allow us to give an immediate upper bound for the number of colors needed to register-allocate the graph? Draw the smallest graph that requires 6 colors. [4 marks]

(*d*) In general, what is the computational complexity of graph colouring – of finding a colouring that uses the smallest number of colours? [2 marks]

(*e*) Describe a heuristic-based register-allocation approach and use it to allocate registers in the above program. Split ties by allocating variables earlier in the alphabet first. Variables assigned but unused should be allocated to a register named ‘zero’. Otherwise, use registers ‘r0’, ‘r1’, ‘r2’, and ‘r3’. [4 marks]