

University of Cape Town
Department of Computer Science
CSC3005h Final Oral Solution
2006

Marks : 25

Time : 45 minutes

Instructions:

- Answer all questions.
-

COMPILERS

What are activation records?[2]

the layout of data needed to support invocation of a subprogram

For programs with recursion, where do we typically store activation records and why? [2]

on the stack – so each invocation of a subprogram has a unique environment

Name 2 advantages of using intermediate representations. [2]

separation of front/back ends, easier to apply optimisations to

What is constant folding? [1]

replacing calculations with only constants with the computed values

What is common subexpression elimination? [1]

factoring out repeated calculations and doing them only once

Why do we apply canonicalisation transformations to the IR tree? [2]

to resolve mismatches between IR and machine code and linearise code

Do we have to do basic block and trace analysis? Why/why not? [2]

No. the code will simply be longer/slower if we did not do this.

How does the dynamic programming algorithm for instruction selection work? [4]

a cost is calculated for each node, starting at the leaves and working up to the root

at each node, for each matching tile, the cost is that of the tile and the minimum costs of subtrees lower down in the tree not covered by it. we assign the lowest cost of all options to the node.

finally, when the root is covered, we generate instructions in reverse order.

Is this algorithm optimal or optimum? What does this mean? [2]

optimum. it means there is not better solution.

What is the point of liveness analysis? [2]

to determine which variables need to retain a value at the same instant

Finally, once we have liveness information, how does k-colouring register allocation work? [5]

build an interference graph.

beginning:

simplify non-move related nodes with $<k$ degree.

coalesce non-move related node which do not change colourability of graph, and start at beginning.

freeze move of low degree node, and start at beginning.

if no simplifies/coalesces/freezes, simplify potential spill node.

when graph is empty, select nodes and assign colour.

if conflict on potential spill, then it is actual spill. rewrite code to shorten live range and start all over again.