COMP 412, Fall 2014
Practice Exam for the Third Hour Exam

NOTE: This practice exam is longer than the actual exam. The actual exam has four questions. Since practice exams are not limited-time, I have taken the liberty of including more questions.

1. In the lectures and in the reading, we discussed many events. Some of those occur at compile-time (while the compiler is running) and some that occur at run-time (while the compiled code is running).

For each of the following events, does it occur at compile time or at runtime?

   a. Spill code is created
   b. Access link is traversed
   c. Activation record is instantiated
   d. Display is referenced
   e. Static coordinate is created
   f. Dependence graph is created
   g. False zero is computed (A_0 is the “false zero” of A)
   h. Dope vector is used
   i. New lexical level is added to the symbol table
   j. Tree pattern is matched

2. In our discussion of peephole-based instruction selectors, we described a peephole selector as consisting of three distinct phases: a code expander, a simplifier, and a matcher.

   a. What is the role of the simplifier in this process?
   b. How does the simplifier work?
   c. What transformations does it apply?

3. One technique for eliminating redundancies in a basic block is called Value Numbering.

   a. State the algorithm for value numbering a single basic block. Be sure to state any assumptions that you make about whether or not values are live before and after the block.
   b. What is the complexity of your algorithm? On what does it depend?

Further refinements to this algorithm have been proposed. In particular:

   c. Explain how the basic algorithm can be extended to discover constant-valued expressions and to evaluate those expressions at compile time.
   d. Explain how the basic algorithm can be extended to capitalize on algebraic identities. What additional costs are entailed in handling algebraic identities?
4. In an Object-Oriented Language, method invocations (function calls) that are resolved at runtime (e.g., virtual functions in C++) incur different kinds of costs than method invocations that are resolved at compile time. This difference is sometimes characterized as static dispatch versus dynamic dispatch. Explain the difference between these two and the impact that the difference has on the cost of a method invocation.

5. In trace scheduling, the compiler must first construct a set of traces through the control-flow graph.

   Given the example control-flow graph below, with the estimated edge frequencies shown, apply the trace construction algorithm and record, below, the traces that you found. For each trace that you construct, show the sequence of edges added to the trace and the sequence of nodes that make up the trace.

   ![Control-Flow Graph](image)

6. One of the strengths of a Chaitin-Briggs register allocator is that it contains a powerful mechanism for coalescing copies.
   
   a. Explain how copy coalescing works in a Chaitin-Briggs allocator.
   
   b. How does the desire to perform coalescing interact with the algorithm for building the interference graph?
   
   c. Under what conditions can the allocator combine two live ranges that are connected by a copy?