

# Computer Science 320: Final Examination

May 5, 2011

You have up to 3 hours to answer the following questions. This final exam is closed book. For partial credit, show all work. State all assumptions. Write out and sign the Honor Code pledge before turning in the test. Using more than 3 hours to take this exam is considered a violation of the Honor Code.

*“I pledge my honor that I have not violated the Honor Code during this examination.”*

## 1 Path-convergence Criterion

The path-convergence criterion states that there should be a  $\phi$ -function for variable  $a$  at node  $z$  of the flow graph exactly when all of the following are true:

1. There is a block  $x$  containing a definition of  $a$ ,
2. There is a block  $y$  ( $y \neq x$ ) containing a definition of  $a$ ,
3. There is a nonempty path  $P_{xz}$  of edges from  $x$  to  $z$ ,
4. There is a nonempty path  $P_{yz}$  of edges from  $y$  to  $z$ ,
5. Paths  $P_{xz}$  and  $P_{yz}$  do not have any node in common other than  $z$ , and
6. The node  $z$  does not appear within *both*  $P_{xz}$  and  $P_{yz}$  prior to the end, though it may appear in one or the other.

Show that the phrase “though it may appear in one or the other” in the last statement is necessary.

## 2 Grammars

1. Which one of the following is true?

- $LL(1) \subset SLR$
- $LL(1) \supset SLR$
- $LL(1) \cap SLR = \emptyset$
- none of the above

2. Prove it in an organized manner.

### 3 Control Flow Analysis

```
        START
L1:
1      r1 = MEM[r2]
2      branch r1 < 10, L4
      L2:
3      r5 = r1 + r3
4      branch r5 < 10, L4
5      r6 = r2 + r3
6      r7 = MEM[r4]
      L4:
7      r6 = r2 + 1
8      branch r6 < r7, L2
9      r3 = r2 + r3
10     branch r5 < r6, L1
        STOP
```

Do the following:

1. Draw the control flow graph.
2. Is the flow graph reducible? If not, transform it into a reducible flow graph.
3. Draw the dominator tree of the reducible flow graph.
4. Compute the dominance frontier for each node in the reducible flow graph.
5. Show the code in this reducible flow graph in SSA form.

## 4 Optimization:

Optimize the following code segment. Full credit will be given for an optimized code schedule which executes to completion in less than 250 cycles on a machine with these characteristics:

- 3 instructions per cycle, no other resource constraints
- Memory and Multiply instructions have a fixed latency of 2 cycles
- All other instructions have a fixed latency of 1 cycle

```
r2 = 0
LOOP:
  r1 = MEM[r2 + &FOO]
  r1 = r1 + 1
  r2 = r2 + 1
  r3 = r2 * 4
  MEM[r3 + &BAR] = r1
  BRANCH r2 < 100, LOOP
```

## 5 Optimization: Phase Ordering Problem

Optimizations interact with each other in many ways. As a result, the optimization application order in a compiler can influence final code quality. Demonstrate this. With any two optimizations denoted *A* and *B*, show that application order *ABAB* yields better code than application order *BABA*. Clearly define your chosen optimizations and your measure of code quality.

## 6 Scheduling

Find a schedule of minimum length for this code:

```
1  r1 = MEM1[r2]
2  r3 = MEM4[r4]
3  r5 = MEM1[r3]
4  r8 = MEM4[r7]
5  r6 = r8 + r1
6  r9 = r1 + r6
```

The resources used by each instruction type can be described with the following regular expressions:

```
F = F1 | F2      /* Instruction Fetch */
D = D1 | D2      /* Decode */
X = X1 | X2      /* Execute */
```

```
MEM1: F; D; X; M1;
MEM4: F; D; X; ((M1; M1) | (M2; M2));
ADD: F; D; X;
```

Each stage takes 1 cycle. This means, for example, that a MEM4 instruction can consume the following resources in time:

```
Cycle  Resource
1      F1
2      D2
3      X1
4      M2
5      M2 /* M2 is reserved for two cycles */
```

In this machine, an instruction cannot execute (X) until instructions it is dependent upon have completed all execution (X) and memory access (M) stages.

## 7 Types

Explain why type systems for languages such as FUN benefit from having a notion of subtyping by describing a situation where some code wouldn't typecheck otherwise. Give the subtyping rule for the typing judgement (i.e. hypothesis and conclusion should have the form  $\Gamma \vdash e : \tau$  where  $\Gamma$  is a context,  $e$  an expression and  $\tau$  a Fun-type). Also, give the rules that establish the subtype relationship for produces and functions, by writing down the hypotheses and/or side conditions of rules with conclusion  $(\tau_0, \dots, \tau_n) <: (\sigma_0, \dots, \sigma_m)$  and  $(\tau_1 \rightarrow \tau_2) <: (\sigma_1 \rightarrow \sigma_2)$ .

## 8 Garbage Collection

Briefly describe the major characteristics of and differences between garbage collection using reference counting, garbage collection according to Cheney, and garbage collection according to Baker. Mention the organization of the heap and of heap objects, and the data structures required to implement the approaches, name the basic way of operation of each algorithm, and discuss their relative advantages in terms of runtime overhead.