Computer Science Comprehensive Exam—Fall 2013
Compiler Construction

Instructions: Do *not* put your name on the exam, please answer all the questions directly on the exam itself. You may write on the back of the pages. You may need scratch paper to work out the answers before writing them on the exam. Answer **all** the questions. You have 90 minutes. Explain answers as fully as possible, give examples or define terms, if appropriate.

1. C++11 has lambdas, a construction with which one can write anonymous functions which depend on parameters (`n` in the example below) and capture non-local variables (`x` in the example below) In this example, a lambda is used to remove elements less than 5 from a C++ vector.

```cpp
#include <vector>
#include <iostream>
#include <algorithm>
#include <functional>
int main() {
    std::vector<int> c { 1,2,3,4,5,6,7 };
    int x = 5;
    c.erase(std::remove_if(c.begin(), c.end(),
        [x](int n) { return n < x; } ), c.end());
    std::cout << "c: ";
    for (auto i: c) {
        std::cout << i << ' ';
    }
    std::cout << '\n';
}
```

   What implementation challenges does this new construct pose for C++? What approach would you take in its implementation?

2. Give a regular expression for all sequences of 0's and 1's that

   (a) contain exactly three 1's.

   (b) contain no consecutive 0s.

   (c) contain an even number of 0s.

3. Give a (simple) grammar for which the following two LR(1) items appear in the same state of the LR(1) parsing automaton. Or, explain why it cannot happen. A, P, Q, and R are non-terminals; and x is a terminal.

$$A \to P \bullet Q\,R\,, x \qquad A \to P\,Q \bullet R\,, x$$

4. Consider the following grammar with non-terminals $\{S, E, B, L\}$.

$$
\begin{array}{ll}
1 & S \rightarrow \textbf{print}\,(\,E\,)\,; \\
2 & S \rightarrow \textbf{while}\,(\,B\,)\,S \\
3 & S \rightarrow \{\,L\,\} \\
4 & E \rightarrow \textbf{id} \\
5 & E \rightarrow \textbf{num} \\
6 & B \rightarrow E > E \\
7 & L \rightarrow S\,L \\
8 & L \rightarrow \epsilon
\end{array}
$$

(a) Compute nullable, FIRST, and FOLLOW for all nonterminals of the grammar.

	nullable	FIRST	FOLLOW
S			
E			
B			
L			

(b) Fill-in the table below. Fill-in the FIRST of the right-hand side, or the FOLLOW of the left-hand side, as appropriate for computing the LL(1) parsing table.

$N \rightarrow \alpha$	$null(N)$?	FIRST(α)	FOLLOW(N)
1 $S \rightarrow \textbf{print}\,(\,E\,)\,;$			
2 $S \rightarrow \textbf{while}\,(\,B\,)\,S$			
3 $S \rightarrow \{\,L\,\}$			
4 $E \rightarrow \textbf{id}$			
5 $E \rightarrow \textbf{num}$			
6 $B \rightarrow E > E$			
7 $L \rightarrow S\,L$			
8 $L \rightarrow \epsilon$			

(c) Fill-in the partial LL(1) parse table below for the indicated terminals.

	id	**num**	**while**	**print**	{	}
S						
E						
B						
L						

5. For the following augmented grammar:

$$
\begin{array}{clcl}
0 & S' & \rightarrow & S\,\$ \\
1 & S & \rightarrow & a\,A \\
2 & S & \rightarrow & b\,B \\
3 & A & \rightarrow & C\,a \\
4 & A & \rightarrow & D\,b \\
5 & B & \rightarrow & C\,b \\
6 & B & \rightarrow & D\,a \\
7 & C & \rightarrow & E \\
8 & D & \rightarrow & E \\
9 & E & \rightarrow &
\end{array}
$$

(a) Give a diagram of the LR(1) states and transitions.

(b) Give the LR(1) parsing tables.

(c) Is the grammar LR(1)? Explain.

(d) Is the grammar LALR(1)? Explain.