Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Compilers

Date: Friday 31st May 2013
Time: 14:00 - 16:00

Please answer any THREE Questions from the FIVE Questions provided

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]
1. a) For each item in column one, choose the best match for column two. Each item in column two should be used only once.

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Abstract Syntax Tree</td>
<td>a. Code optimisation</td>
</tr>
<tr>
<td>2. Activation Record</td>
<td>b. DFA minimisation</td>
</tr>
<tr>
<td>4. Bison</td>
<td>d. Instruction scheduling</td>
</tr>
<tr>
<td>5. Chaitin’s heuristic</td>
<td>e. Memory management</td>
</tr>
<tr>
<td>6. Constant Folding</td>
<td>f. Parsing</td>
</tr>
<tr>
<td>7. Hopcroft’s algorithm</td>
<td>g. Parsing</td>
</tr>
<tr>
<td>8. List scheduling</td>
<td>h. Register Allocation</td>
</tr>
<tr>
<td>9. LR(1) item</td>
<td>i. Register Allocation</td>
</tr>
</tbody>
</table>

(5 marks)

b) Assume that you have developed a basic compiler for a subset of the C language that generates code for certain Intel processors. Identify the components of the compiler that would need to be modified or enhanced to provide the following capabilities:

i) allow C++ (or Java) like comments, that is, //
ii) add the type complex to describe complex numbers
iii) support code generation for an Intel quad-core processor
iv) include a repeat … until construct. (This construct will repeat the execution of a block of statements until a condition becomes true.)

Justify your answers.

(8 marks)

c) Crafting a successful compiler requires attention to many details. Since most algorithms in a compiler manipulate sets, maps, tables, and graphs, many of the performance issues that arise revolve around the implementation of these data structures. Give two examples of data structures used by a compiler where design choices may make an impact on either the time or the space needed for compilation. Justify your answer.

(7 marks)
2. Consider the following automaton. S0 is the start state and S3 is the final state.

![Automaton Diagram]

a) Explain why this is a Non-Deterministic Finite Automaton (NFA) (2 marks)

b) A string of length 3 that is clearly acceptable is aab. Find all strings of length 3 that are acceptable. (3 marks)

c) Apply the subset construction algorithm to convert this NFA to a DFA. Show the DFA. (7 marks)

d) Apply Hopcroft’s algorithm to produce a minimised DFA. Show the DFA. (4 marks)

e) Use the DFA produced to write a regular expression that describes the grammar represented by the automaton above. Describe what this regular expression represents in plain English. (4 marks)
3. a) Consider the following grammar for boolean expressions.

\[
E \rightarrow E \text{ or } E \\
E \rightarrow E \text{ and } E \\
E \rightarrow \text{ not } E \\
E \rightarrow (E) \\
E \rightarrow \text{ true} \\
E \rightarrow \text{ false} \\
E \rightarrow \text{id}
\]

(i) Show that this grammar is ambiguous. (4 marks)

(ii) Rewrite the grammar to remove the ambiguity by introducing new non-terminal symbols. Your revised grammar must accept the same language as the original. (8 marks)

b) Consider the grammar and the Action and Goto tables below:

<table>
<thead>
<tr>
<th>STATE</th>
<th>ACTION</th>
<th>GOTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>S3</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>S4</td>
<td>S5</td>
</tr>
<tr>
<td>2</td>
<td>S3</td>
<td>S2</td>
</tr>
<tr>
<td>3</td>
<td>R4</td>
<td>R4</td>
</tr>
<tr>
<td>4</td>
<td>S3</td>
<td>S2</td>
</tr>
<tr>
<td>5</td>
<td>S3</td>
<td>S2</td>
</tr>
<tr>
<td>6</td>
<td>S4</td>
<td>S5</td>
</tr>
<tr>
<td>7</td>
<td>R1</td>
<td>S5</td>
</tr>
<tr>
<td>8</td>
<td>R2</td>
<td>R2</td>
</tr>
<tr>
<td>9</td>
<td>R3</td>
<td>R3</td>
</tr>
</tbody>
</table>

Show, in full detail, the steps that an LR parser, using the tables above, would follow to parse the string \((x+y)\) * z. Once you produce all the steps, you need to draw the resulting parse tree. For each step your answer should show the contents of the stack, what the next input is and the action that is taken. (8 marks)
4. a) Draw the control flow graph of the following code fragment.

```plaintext
m=0; n=1000; s=0
L1: if (m>n) goto L4
    r=a[m]
L2: if (r<m) goto L3
    n=n+1
    goto L1
L3: x=a[r]
    s=s+x
    if (s<m) goto L4
    m=s
    r=r+1
    goto L2
L4: return
```

(4 marks)

b) Method inlining is an optimization, often used in object-oriented languages, such as Java, which replaces a method invocation by the actual code to be executed. Provide two reasons that one would want to do this. Explain why it may not always be possible in a language like Java to apply this optimisation.

(5 marks)

c) Provide generic code that could be used to unroll the body of a loop, such as the one below, s times. You should assume that n can be any positive integer and s is an integer, greater than or equal to 1 and less than or equal to n.

```plaintext
for (i=0;i<n; i++)
    { loop_body; }
```

(6 marks)

d) Provide a general expression that a compiler might use to access element [I][J] of an array of the form [0…N][0…M] as an offset from the address used to store the first element, that is [0][0], of the array, when:

i) the array is stored in row-major order;
ii) the array is stored in column-major order.

(5 marks)
5. a) Consider the following basic block.

1. load r1, @x
2. load r2, @y
3. add r3, r1, r2
4. mult r4, r1, r2
5. add r5, r3, 1
6. add r6, r4, r3
7. sub r7, r6, r4
8. mult r8, r5, r7

(i) Explain why a processor with only 4 registers would be sufficient to execute this block without the need to do any spilling.
(4 marks)

(ii) Explain how Best’s algorithm can be used for register allocation in the above basic block. Show what the basic block would look like after applying the algorithm.
(4 marks)

b) (i) Describe an efficient list scheduling algorithm for instruction scheduling on a hypothetical system with two functional units, if the cost of executing an instruction (delay latency of the instruction) on the second functional unit is one cycle more than the cost of executing the same instruction on the first functional unit.
(6 marks)

(ii) Apply your algorithm above to the basic block below and show the schedule.

1. load r1, @x
2. load r2, [r1+4]
3. add r3, r3, 1
4. mult r6, r6, 17
5. store r6
6. div r5, r5, 29
7. add r4, r2, r5
8. mult r5, r2, r4
9. store r4

You should assume that, on the fastest functional unit, the cost of a load and a store is 3 cycles, the cost of a mult is 2 cycles and the cost of all remaining operations is 1 cycle.
(6 marks)