University of Cape Town ~ Department of Computer Science

Computer Science 3003S ~ 2009

November Exam

Marks : 100

Time : 180 minutes

Instructions:

a) Answer ALL questions in Sections A and C.

b) Answer 3 questions in Section B and 3 questions in Section D.

c) Show all calculations where applicable.

Section A : COMPILERS [Answer questions 1 and 2 – both compulsory]

Question 1 : You must answer this question. [10]

a) Explain the difference between a grammar and the language it generates. [2]

b) When is a context free grammar ambiguous? [2]

c) Describe in detail how a lexical analyser can be automatically constructed from a list of regular expressions and used to match tokens in an input stream. Your description should mention how to disambiguate rules and how the longest matching substring can be found. [6]

Question 2: You must answer this question. [10]

Compilers that generate code for mobile devices must pay particular attention to the restrictions of mobile devices. Answer the following questions in the context of a mobile device with very little memory to store both instructions and data.

a) What is the purpose of the compiler? [1]

b) Describe one advantage and one disadvantage of separating the compiler front-end from the compiler back-end. [2]

c) Which peephole optimisations from the following list will not typically result in larger code size?

Constant propagation; Constant folding; Inlining; Common subexpression elimination; Strength reduction; Loop unrolling [3]

d) In the dynamic programming instruction selection algorithm, how would you ensure that the generated code is as small as possible? [2]

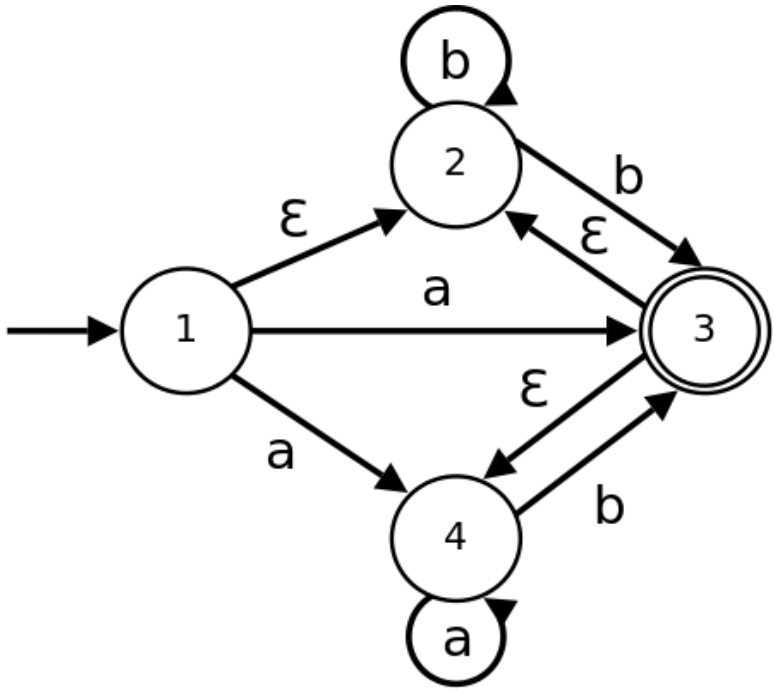
e) When allocating registers, MOVE-related nodes potentially result in smaller code. How does this happen? [2]

Section B COMPILERS [Answer any 3 questions ONLY]

Question 3 – Lexical Analysis [10]

a) For the alphabet {a,b}, find a regular expression that matches all strings of even length, starting with "a". [2]

b) Consider the Non-Deterministic Finite Automaton (NFA) below:



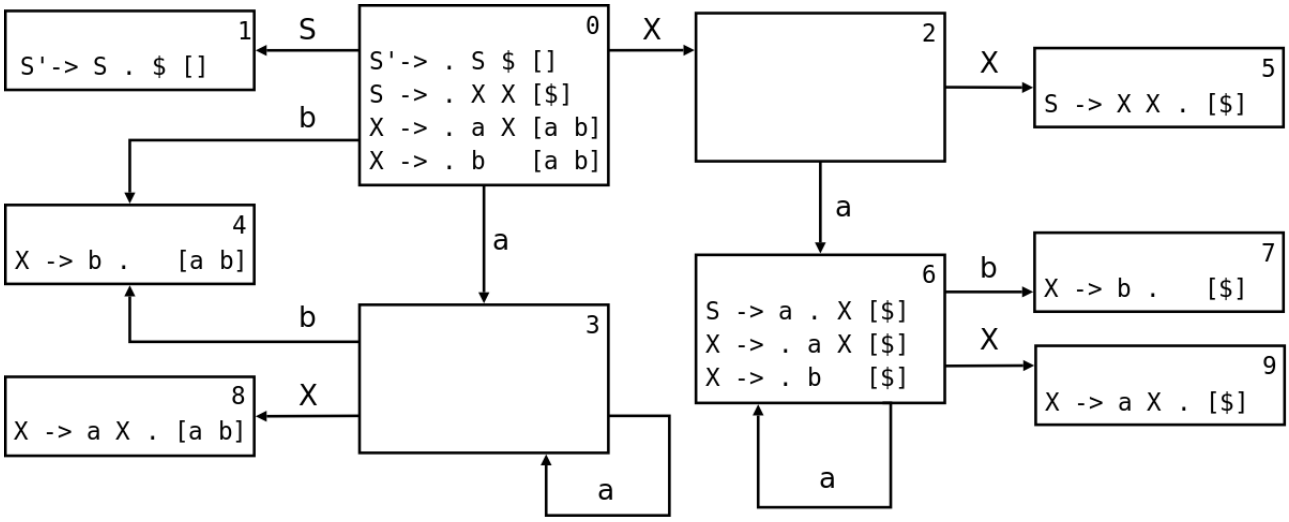
i. Convert it to a Deterministic Finite Automaton (DFA). Remember to indicate the start and final state(s). [6]

ii. Two strings of length 3 that are clearly acceptable are "aab" and "bbb". Find all other strings of length 3 that are also acceptable. [2]

Question 4 – Parsing [10]

Consider the following grammar and LR(1) automaton and answer the questions below:

```
0: S' → S $
1: S → X X
2: X → a X
3: X → b
```



a) Complete State 2 of the LR(1) Automaton. [3]

b) Complete State 3 of the LR(1) Automaton. [3]

c) Is the automaton an LALR(1) automaton? Motivate your answer. [2]

d) Complete the LR(1) parsing table entries for State 0 and State 8. Use the template provided below when answering this question. [2]

	a	b	$	S	X
0					
8					

Question 5: Basic Blocks and Canonicalisation [10]

a) You have been given the following list of canonicalisation rules:

```
ESEQ(s1, ESEQ(s2, e))    => ESEQ(SEQ(s1,s2), e)
BINOP(op,ESEQ(s,e1),e2) => ESEQ(s, BINOP(op, e1, e2))
MEM(ESEQ(s, e1))         => ESEQ(s, MEM(e1))
JUMP(ESEQ(s, e1))        => SEQ(s, JUMP(e1))
MOVE(ESEQ(s, e1), e2)    => SEQ(s, MOVE(e1, e2))
MOVE(e1, ESEQ(s, e2))    => SEQ(MOVE(TEMP t, e1), SEQ(s, MOVE (TEMP t, e2)))
```

Transform this IR tree as much as possible, using the above rules. Show each rule being applied as a separate step.

MOVE (ESEQ, (s1, ESEQ (s2, e1)), MEM (ESEQ (s3, e2))) [4]

b) What is a basic block? [2]

c) Perform basic block and trace analysis on the following code. Clearly show (i) the initial set of basic blocks; (ii) the set of traces; and (iii) the final optimised code. [4]

```
Start: ----------
       jump E
B:     ----------
       jump F
C:     ----------
       jump B
D:     ----------
       jump C
E:     ----------
       jump D
F:     ----------
```

Assume the -------s refer to sequences of statements.

Question 6: Register Allocation [10]

a) What is the purpose of liveness analysis? [2]

b) What is the purpose of the register allocation algorithm? [2]

c) Suppose that liveness analysis yields the following live-in/out sets:

```
AX, BX, BCY, BC
```

where X and Y are pre-coloured nodes and A and C are MOVE-related.

Based on just this information, draw a corresponding interference graph. [4]

d) Explain how actual spills are dealt with by the register allocation algorithm. [2]

Section C Theory of Algorithms [Answer questions 7 and 8 – both compulsory]

Question 7 : You must answer this question. [10]

a) Approximation algorithms are one way of tackling NP-hard problems. Using the discrete knapsack problem as an example, explain the approximation approach. [2]

b) Two very competitive friends have asked you to be a a judge of their programming contest. After a month of work, they reveal their code. Anna has developed a matrix multiplication algorithm of time complexity $O(n^2)$ and Thandi has revealed a graph colouring algorithm of time complexity $O(n^5)$.

i. What are the brute force time complexities for these two algorithms? [2]

ii. Should you be impressed with Anna's work? Explain your answer. [2]

iii. Should you be impressed with Thandi's work? Explain your answer.[2]

iv. Who wins the prize? Explain your decision. [2]

Question 8: You must answer this question. [10]

a) Match **any TWO** of the three problem-solving techniques below with a problem from the list that follows. In each case, define the problem clearly and state briefly how you would apply the associated technique.

i. decrease-and-conquer

ii. transform-and-conquer

iii. dynamic programming

List of problems:

Binary coefficients; Fake coin; GCD (Greatest Common Divisor); Interpolation search; LCM (Lowest Common Multiple); Optimal binary search tree; River-crossing; Transitive closure of directed graphs.

[6]

b) Given a (very large) data file of 2-digit codes in random order, and a query file consisting of code ranges, you must find how many times the codes in each range occur in the data file. Example:

Data file: 61 67 61 67 61 64
Query file:
32 43
61 67
61 64
Output: 0 6 4
(because 32-43 occur 0 times in the data file; 61-67 occur 6 times; and 61-64 occur 4 times).

i. Briefly describe a brute force algorithm for solving this problem.

ii. What is the time complexity of the brute force algorithm if there are M codes in the data file and N ranges in the query file?

iii. Briefly describe a more efficient algorithm for solving this problem.

[4]

Section D THEORY OF ALGORITHMS [Answer any 3 questions ONLY]

Question 9: Greedy algorithms [10]

a) Consider the problem of scheduling *n* jobs of known durations $t_1,t_2,t_3,....,t_n$ for execution by a single processor. The jobs can be excuted in any order, one at a time. You want to find a schedule that minimizes the total time spent by all the jobs in the system. (The time spent by one job in the system is the sum of the time spent by this job in waiting plus the time spent on its execution.)

i. Write down, in pseudocode, a greedy algorithm for this problem. [3]

ii. What is the time complexity of this greedy algorithm? Justify your answer. [2]

iii. What would the time complexity be of a brute-force algorithm that performed an exhaustive seach of all possible schedules? [1]

iv. Prove that the greedy algorithm always yields an optimal solution for this scheduling problem. [4]

Question 10 : Lower bounds and backtracking[10]

a) Consider the problem of finding the median of a three-element set {a,b,c}.

   i. What is the information-theoretic lower bound for a comparison-based algorithm solving this problem? Show your working. [1]

   ii. Draw a decision tree for an algorithm solving this problem. [3]

b) Prove, with an adversary argument, that any comparison-based algorithm for finding the largest amoung *n* given numbers must make *n-1* comparisons in the worst case. [3]

c) Apply backtracking to solve the following instance of the subset-sum problem: S={1,2,5,6,8} and *d =9.* [3]

Question 11 : Pattern matching [10]

Consider the problem of **searching for the pattern** ED1C0DED in a hex (hexadecimal) string.

a) How many character comparisons would the brute force algorithm take in searching for ED1C0DED in the string 1212ED1C0DED ?

b) Give an example of an 8-character pattern and a 12 character string that would give the worst case performance when brute force is used.

c) Give the values of elements D, E and F of the bad-symbol shift table as used by the Horspool and Boyer-Moore algorithms when searching for ED1C0DED

d) Show the good-suffix table of the Boyer-Moore algorithm used in searching for ED1C0DED

e) How many character comparisons would the Horspool or the Boyer-Moore algorithm take in searching for ED1C0DED in the string 1212ED1C0DED ?

f) Describe/characterise the search patterns for which Boyer-Moore and Horspool's algorithms perform exactly the same

g) Another common problem besides pattern matching is finding the best path between 2 nodes in a graph.

   i. What type of problem-solving technique is used in Breadth First Search?

ii. When would you use Breadth First Search and when would you use Floyd's algorithm (i.e. how do you know which to use when)?

[10]

Question 12 : Divide-and-Conquer and Dynamic Programming [10]

a) Briefly explain the divide-and-conquer problem solving technique, using any example problem to illustrate your answer.

[2]

b) A (huge) data file gives the costs of N items, with the cost of item 1 first, then the cost of item 2, …, and the cost of item N last. A query file consists of several integers Mj. For each query Mj, you must find the minimum cost of buying any Mj consecutive items. Example:

Data file: 4 6 9 11 7
Query file: 3 4
Output: 19 30

(4+6+9 = 19; 6+9+11 = 26; 9+11+7 = 27 so 19 is the minimum of consecutive threesomes;
4+6+9+11=30; 6+9+11+7 = 33 so 30 is the minimum of consecutive foursomes)

i. Briefly describe a brute force algorithm for solving this problem.

ii. Briefly describe a more efficient algorithm for solving this problem.

[4]

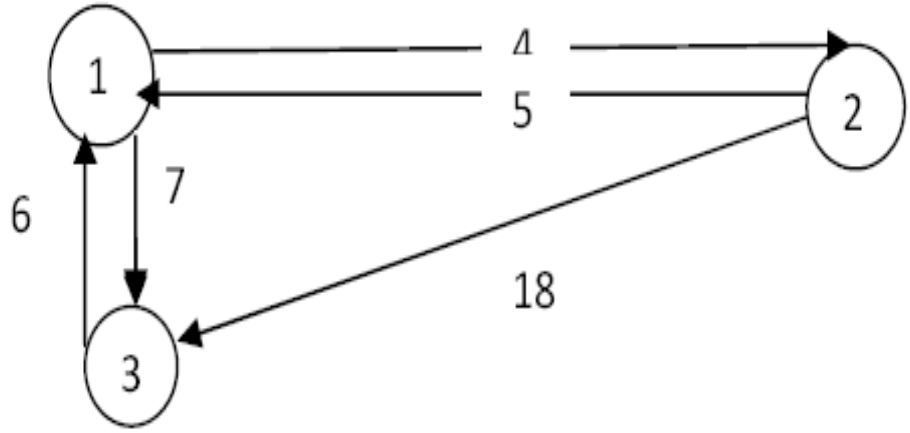
c) Show the **array contents** when dynamic programming is used to solve

EITHER (1) the **knapsack** problem below
OR (2) the all-pairs shortest path problem below using **Floyd**'s algorithm

i. Item 1 weighs 2kg and is worth R6; item 2 weighs 1kg and is worth R3; item 3 weighs 2kg and is worth R8. The knapsack can hold at most 3kg. What is the greatest Rand value it can hold?

OR

ii. Show **only the first two arrays** in applying Floyd's algorithm to this graph:



[4]