Quizzes for General CS II (320102) Spring 2014

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

August 9, 2014

Contents

Quiz 1: Graphs and Trees **2**

Quiz 2: Combinatorial Circuits **3**

Quiz 3: Twos Complement Numbers **4**

Quiz 4: Simple While and L(VM) **5**

Quiz 5: Static procedures **6**

Quiz 6: Turing Machines **7**

Quiz 8: Internet **8**

Quiz 9: Encryption **9**

Quiz 10: Breadth First Search **10**

Quiz 11: Search **11**

Quiz 1: Graphs and Trees
(Given Feb. 9. 2014)

Problem 1.1 (Graph and Trees)

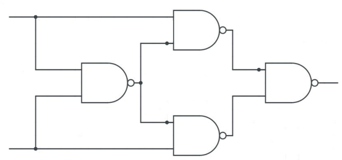
12pt

1. Draw the parse tree of the following expresion: $(\overline{x_1} + x_3) * (x_1 + x_2)$.
2. Give an example of a different graph that is isomorphic to this parse tree and write its mathematical representation.
3. Can this parse tree be isomorphic to a graph with a cycle? Why or why not?

Quiz 2: Combinatorial Circuits
(Given Feb. 17. 2014)

Problem 2.1 (Combinatorial Circuits)

Design a full adder consisting of NAND gates ONLY, based on the definition from the slides. Note that giving the circuit for a full adder that consists of OR, AND and XOR gates will only bring half the points. The XOR gate represented with NAND gates is provided below. 12pt



Quiz 3: Twos Complement Numbers
(Given Feb. 24. 2014)

Problem 3.1: Given following integer numbers in base ten. Convert them to 32-bit 12pt Two's Complement numbers.

1. 3643
2. 5731923
3. -128
4. -24689

Quiz 4: Simple While and L(VM) (Given Mar. 10. 2014)

Problem 4.1 (A sum in $\mathcal{L}(\texttt{VM})$)
Compute the following sum in $\mathcal{L}(\texttt{VM})$ using an **iterative** approach (i.e. you may not compute the result using a formula): 12pt

$$\sum_{i=1}^{N} i \cdot (i-1)$$

You have $\mathcal{S}(0) = N$ and you should "output" the result of the sum in $\mathcal{S}(1)$. For example, for $N = 3$, your program should halt with $\mathcal{S}(1) = 8$.

Also, simulate the execution of your program (including the stack evolution), for $N = 3$.

Note: It is always a good idea to comment your code!

Quiz 5: Static procedures (Given March. 17. 2014)

Problem 5.1 (Towers of Hanoi)
The Towers of Hanoi is a very famous mathematical puzzle. Given three pegs, one with a set of N disks of increasing size, determine the minimum (optimal) number of steps it takes to move all the disks from their initial position to another peg without placing a larger disk on top of a smaller one. 12pt

Write down a static procedure in $\mathcal{L}(\texttt{VMP})$ that computes the solution to the puzzle, given N.

Quiz 6: Turing Machines
(Given Mar. 24. 2014)

Problem 6.1 (TM compare two numbers)
Given the alphabet $\{0, 1, \#\}$, where # symbolizes an empty cell, consider a tape with the input $0^n 1^m$, where n and m are natural numbers, (followed by infinitely many #s). Design a TM that halts in a state "yes" if $n > m$ and in state "no" otherwise. 12pt

Quiz 8: Internet
(Given March. 31. 2014)

Problem 8.1 (Internet Protocol Suite)
Explain very briefly the layered structure of the Internet Protocol Suite. 4pt

Problem 8.2 (Internet Design Principles)
Name and explain three of the Internet design principles. 8pt

Quiz 9: Encryption (Given April. 28. 2014)

Problem 9.1 (Security by Encryption)

Please explain the terms 12pt

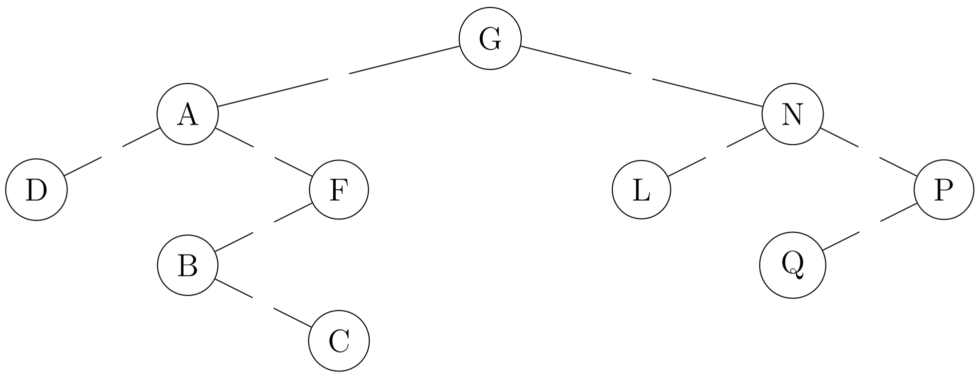
- one-way function
- trapdoor function

In addition, give one **example** of a candidate for a good one-way/trapdoor function, explaining how the function is computed, and how it is inverted.

Quiz 10: Breadth First Search (Given May. 04. 2014)

Problem 10.1 (BFS)

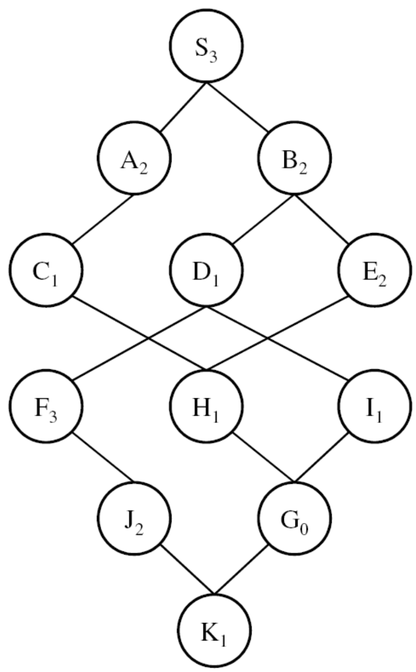
Apply BFS on the following tree (the goal is not node G!): 12pt



Quiz 11: Search
(Given May. 12. 2014)

Problem 11.1 (Moving a Knight)
Consider the problem of moving a knight on a 3x4 board, with start and goal states labeled as S and G in the figure below. The search space can be translated into the following graph. The letter in each node is its name and you do not need to worry about its subscript for now. 9pt

S_3	H_1	D_1	K_1
I_1	J_2	A_2	E_2
C_1	B_2	G_0	F_3



Make the following assumptions:

- The algorithms do not go into infinite loops (i.e. once a node appears on a path, it will not be considered again on this path)
- Nodes are selected in alphabetical order when the algorithm finds a tie.

Write the sequence of nodes in the order visited by the specified methods (until the goal is reached). Note: You may find it useful to draw the search tree corresponding to the graph above.

- DFS
- BFS