General Computer Science II (320201) Spring 2008

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

April 8, 2013

Contents

Assignment 1: Graphs and Trees
(Given Feb. 11.)

4pt

Problem 1.1 (Tree representation)

- Provide an example of a tree which contains 6 edges. Give a mathematical representation.
- For a tree $G = \langle V, E \rangle$ prove that $\#(E) = \#(V) - 1$.

Solution: The solution is obvious

Assignment 2: Combinatorial Circuits
(Given Feb. 18.)

12pt

Problem 2.2 (Combinational Circuit)
Consider the following Boolean function

$$f\colon \{0,1\}^3 \to \{0,1\}^2; \langle i_1, i_2, i_3\rangle \mapsto \langle \overline{i_1} * i_2 + i_2 * \overline{i_3}, \overline{i_1 + i_2} * i_3\rangle$$

Draw the corresponding combinational circuit and write down its labeled graph $G = \langle V, E, f_g\rangle$ in explicit math notation.

Solution: The solution is obvious. All we need is to just draw graph and write some line about math representation.

Assignment 3: Number Conversion
(Given Feb. 25.)

12pt

Problem 3.3: Write down the last 4 digits of your matricle number and, considering it as *hexadecimal* number, represent it as binary, octal, and decimal number.

Assignment 4: Two's Complement Numbers
(Given Mar 3.)

12pt

Problem 4.4: Given following integer numbers in base ten. Convert them to 32-bit Two's Complement numbers.

1. 4574
2. 5163632
3. -256
4. -37419

Solution: First we need the numbers' normal binary representation, therefore

1. $4574 = \varphi_{1000111011110}(2)$
2. $5163632 = \varphi_{10011101100101001110000}(2)$
3. $-256 = -\varphi_{100000000}(2)$
4. $-37419 = -\varphi_{1001001000101011}(2)$

To align a positive number to 32 bits we just fill in the space to the left with its sign bit (0), therefore To convert a negative number we need to complement it and add 1 to the resulted number. Afterwards we will fill in the space on its left with its sign bit (1).

Assignment 5: Virtual Machine
(Given March 10.)

12pt

Problem 5.5 (Multiplication and Evenness)
Assume the data stack initialized with `con` a and `con` b for some natural number a and b. Write a $\mathcal{L}$(`VM`) program that returns on top of the stack 1 if $a \cdot b$ is even, and 0 otherwise. Also draw the evolution of the stack during the execution of your program for $a = 3$ and $b = 2$.

Assignment 6: SW Language
(Given March 24.)

12pt

Problem 6.6 (SW program)
Write a program in Simple While Language which is equivalent to the following program in abstract syntax:

```
var i := 0; var a := 1; var b := 1; var f := 0;
while (i <= 10) do begin
  f := a + b;
  a := b;
  b := f;
  i = i + 1;
end
return (a + b) ∗ f;
```

Assignment 7: Turing Machines
(Given April 7)

12pt

Problem 7.7: Give a detailed description of a Turing Machine with all of its components. Why do we need a notion of a Turing Machine?

Solution: tbd

Assignment 8: Search
(Given April 14)

12pt

Problem 8.8: Given the vector $\langle 5, 3, 42, 12, 0, 2, 1, 4, 19, 2, 7 \rangle$. Initial state is 1 and goal state is 42, define operators to traverse through it and construct the BFS tree until the solution is found.

What are the branching factor and depth of your tree?

What does it mean that a search algorithm is complete? Is BFS complete?

Assignment 9: Greedy Search
(Given April 30)

12pt

Problem 9.9 (A looping greedy search)
Draw a graph and give a heuristic so that a greedy search for a path from a node A to a node B gets stuck in a loop. Draw the development of the search tree, starting from A, until one node is visited for the second time.

Indicate, in one or two sentences, how the search algorithm could be modified or changed in order to solve the problem without getting stuck in a loop.

Solution:

1. The example from the lecture, i. e. traveling through Romania.

2. Use A^*, or remember which nodes have been visited before and don't visit them again.

Assignment 10: Local Search
(Given April 28)

12pt

Problem 10.10 (Local Search)
What is a *local* search algorithm?

1. What does the "fringe" known from generic search algorithms look like in a local search algorithm?
2. What is the space complexity of local search?
3. Name two practical applications for local search.
4. Name a simple algorithm for local search. Give a brief overview of its advantages and disadvantages.

Assignment 11: Prolog
(Given May 14)

12pt

Problem 11.11 **(A ProLog warm-up)**
Given as a ProLog fragment the clauses

```
nat(zero).
nat(s(X)):−nat(X).
```

Write unary predicates even and odd with the obvious meaning as well as a binary predicate leq meaning less or equal.

Solution:

```
nat(zero).
nat(s(X)) :− nat(X).

even(zero).
even(s(X)) :− odd(X).
odd(s(X)) :− even(X).

leq(zero, X).
leq(s(X), s(Y)) :− leq(X,Y).
```
