General Computer Science II (320201) Spring 2007

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

April 8, 2013

Contents

Assignment 1: Graphs and Trees
(Given Feb. 12.)

Conjecture 1 *Given a tree $G = \langle V, E\rangle$ with root node v_r. For any node $v \in V$ except the root v_r, there is exactly one path $p \in \Pi(G)$ with start$(p) = v_r$ and end$(p) = v$.*

20pt

Problem 1.1 (Unique Paths in Trees)
Prove the conjecture above by induction on the number of nodes or refute it by a counterexample.

Solution:
Proof: by induction over $n = \#(V)$

P.1.1 $n = 1$ **(base case)**: $V\backslash\{v_r\} = \emptyset$, i.e. there is no node except the root node, and thus we have nothing to prove. □

P.1.2 $n \to n + 1$ **(induction step)**: Let $v \in V$ be a non-root node. From the definition of a tree, it follows that indeg$(v) = 1$, and thus there is exactly one $u \in V$ with $\langle u, v\rangle \in E$. As $\#(V\backslash\{v\}) = n$ we can infer from the inductive hypothesis that there is exactly one path $p \in \Pi(G)$ with start$(p) = v_r$ and end$(p) = u$. If we append the edge $\langle u, v\rangle$ to that path, we obtain a unique path $q = \langle v_r, \ldots, u, v\rangle$ with start$(q) = v_r$ and end$(q) = v$ □

□

Assignment 2: Positional Number Systems
(Given Feb. 19.)

12pt

Problem 2.2: Write down your date of birth in DDMM format (e.g. 3009 for September 30th) and (considering it as decimal number) represent it as binary, octal, and hexadecimal number.

Assignment 3: Two's Complement Numbers
(Given Feb. 26.)

12pt

Problem 3.3 (Binary Number Systems)

- Write down the definition of $\langle\!\langle\cdot\rangle\!\rangle$, $(\langle\!\langle\cdot\rangle\!\rangle^{-})$, and $\langle\!\langle n\rangle\!\rangle^{2s}_{\cdot}$.
- Given the binary number $a = 10110$ compute $\langle\!\langle a\rangle\!\rangle$, $(\langle\!\langle a\rangle\!\rangle^{-})$, and $\langle\!\langle a\rangle\!\rangle^{2s}_{n}$.

Assignment 4: Memory
(Given Mar 5.)

12pt

Problem 4.4 (RS Flipflop from NAND gates)
Construct the RS flipflop from NAND gates. You may use other elementary gates (AND, OR, NOT) as appropriate. Draw the sequential logic circuit as a graph, as well as a state table.

Solution: The solution looks exactly like the NOR implementation, with the NOR's replaced by NAND's, but the inputs need to be inverted using NOT's first.

See `http://www.play-hookey.com/digital/rs_nand_latch.html`

Assignment 5: Virtual Machine
(Given March. 12.)

12pt

Problem 5.5 (Multiplication and Modulo)
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 $a \cdot b \, mod \, 3$. Furthermore draw the evolution of the stack during the execution of your program for $a = 2$ and $b = 4$.

Assignment 6: Virtual Machine
(Given March. 19.)

12pt

Problem 6.6 (Reasons for Virtual Machines)
Thinking back to the lectures about $\mathcal{L}(\mathtt{VM})$ and SW, sum up the benefits of compiling programs in high-level languages to the language of a virtual machine instead of directly compiling them to an assembler language ASM.

Assignment 7: Turing Machines
(Given April 15)

12pt

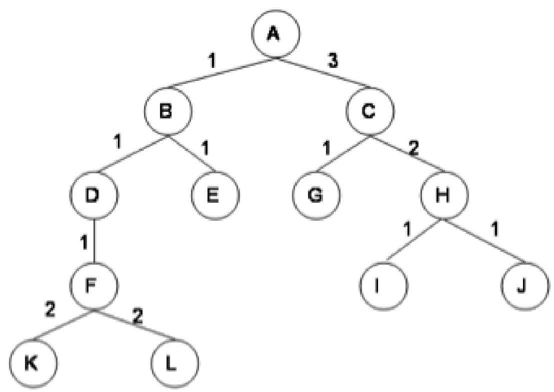
Problem 7.7: Give the action table of a Turing machine that flips all bits but the first and last one of a given binary number n. Assume that the number is delimited by hash marks (i.e. the overall alphabet is $\{0, 1, \#\}$) and that, initially, the head is on the leftmost bit of n. What does your machine do if there is only one bit of input?

Simulate your Turing machine on the input #0101#.

Assignment 8: Tree Search
(Given April 23)

12pt

Problem 8.8 (Search Strategy Comparison on Tree Search)
Consider the tree shown below. The numbers on the arcs are the arc lengths.



Assume that the nodes are expanded in alphabetical order when no other order is specified by the search, and that the goal is state G. No visited or expanded lists are used. What order would the states be expanded by each type of search? Stop when you expand G. Write only the sequence of states expanded by each search.

Search Type	Sequence of States
Breadth First	
Depth First	
Iterative Deepening (step size 1)	
Uniform Cost	

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 May 7)

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).
```
