Name: Matriculation Number:

General CS II (320201) Final Exam
May 23 2008

You have two hours(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 14 minutes, leaving you 106 minutes for revising your exam.

You can reach 14 points if you solve all problems. You will only need 100 points for a perfect score, i.e. -86 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

	To be used for grading, do not write here			
prob.	1.1	2.1	Sum	grade
total	4	10	14	
reached				

Good luck to all students who take this test

1 An Old GenCS Favourite

4pt

Problem 1.1 (Function Definition) 4min

Let A and B be sets. State the definition of the concept of a partial function with domain A and codomain B. Also state the definition of a total function with domain A and codomain B.

Solution: Let A and B be sets, then a relation $R \subseteq AB$ is called a **partial/total function**, iff for each $a \in A$, there is at most/exactly one $b \in B$, such that $\langle a, b \rangle \in R$.

2 Graphs

10pt
10min

Problem 2.1 (Depth of a Fully Balanced Binary Tree)

Prove or refute that in a fully balanced binary tree with $n \geq 1$ nodes, the depth is $\log_2(n)$.

Solution:

Proof: by induction over n

P.1.1 $n = 1$ **(base case)**: $\log_2(1) = 0$, the depth □

P.1.2 $n \to n+1$ **(induction step)**: $n+1$ leaves are added (because fully balanced), so we have $\log_2(2n+1)$.

$\log_2(2n) = (\log_2(n)) + 1$.

Since $2n+1$ is not a power of 2, $\log_2(2n+1) = \log_2(2n) = 1 + (\log_2(n))$, which by I.H is depth+1 . □

□

3 Binary numbers

Problem 3.1 (Binary Numbers) 15pt 15min

- Prove that $\langle\langle \overline{a} \rangle\rangle_n^{2s} = -\langle\langle a \rangle\rangle_n^{2s} - 1$.

Hint: You can use the following without proving: $\overline{a_i} = 1 - a_i$ and $\sum_{i=0}^{n-1} 2^i = 2^n - 1$

- Convert -42 to 12-bit two's complement.
- Convert the hexadecimal number $BEEF$ to decimal. You do not need to carry out the computations, just write down a the answer without simplifications.

Solution:

- Proof on the slides
- $42 = 0101010$, flip all bits $\rightarrow 1010101$, add $1 \rightarrow 1010110$, duplicate sign bit until 12 bits $\rightarrow 111111010110$
- $11 * 16^3 + 14 * 16^2 + 14 * 16 + 15 * 1$

---

4 Combinatorial Circuits

Problem 4.1 (Alarm System) 15pt 15min

You have to devise an alarm system that signals if the image recorded by a camera changes. The camera is preprogrammed with a static image, divided into 8 regions. Whenever an observed region is different from the preprogrammed one, the corresponding input bit $\langle r_0, ..., r_7 \rangle$ is set to 1. The image is sampled at discrete time periods. The value of an input (clk) changes between 0 and 1 on every time interval.

Design a circuit with one output which is set to 1 if two or more regions (the inputs $\langle r_0, ..., r_7 \rangle$) are different from the preprogrammed image for two consecutive intervals. We do not care if different sets of regions are marked as different between the consecutive intervals. We also don't care what happens once the output is set to one.

You may use all elementary gates and all circuit blocks studied in class.

Hint:

- First make a circuit that determines how many of the regions are different.
- Make a circuit that outputs 1 if two or more regions are different in 2 consecutive intervals.

---

5 Virtual Machines

Problem 5.1 (Static Procedure for Binomial Coefficients) 10pt 10min

Write a $\mathcal{L}$(VM) static procedure that computes the value of the binomial $C(n,k)$. Use the recursion formula:

$$C(n+1,k+1) = C(n,k+1) + C(n,k)$$

$$C(n,0) = C(0,0) = 1$$

$$C(0,n) = 0$$

Solution:

```
; C(n,k)
proc 2 46

; if k == 0 return 1
con 0 arg 2 leq cjp 5
con 1 return

; if n == 1 return 0
con 0 arg 1 leq cjp 5
con 0 return

; C(n-1,k)
arg 2
con 1 arg 1 sub
call 0

; C(n-1,k-1)
con 1 arg 2 sub
con 1 arg 1 sub
call 0

; return C(n-1,k) + C(n-1,k-1)
add return
```

---

6 SML

14pt
14min

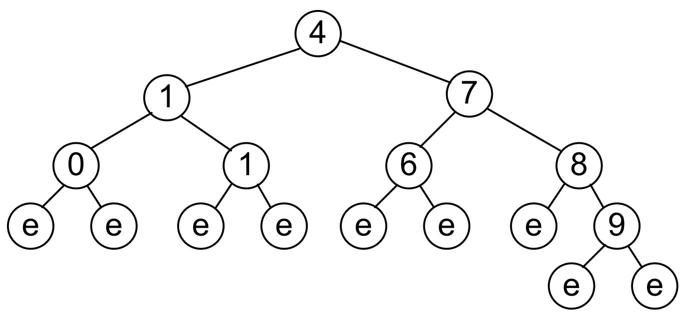
Problem 6.1 (Treesort Function)

Your task is to write a treesort function in SML that sorts a list of integers by first creating a binary search tree from the list and then loading the tree (in a sorted order) back into a list.

Use the following definition of a binary search tree:

- All leaves are empty nodes.
- All internal nodes carry a value and a left and a right subtree.
- The values of all nodes in a node's left subtree are smaller than the node's value and all nodes in its right subtree are greater or equal to the node's value.

The following tree is an example of a binary search tree:



Given the following datatype:

```
datatype searchtree = empty | node of searchtree*searchtree*int;
```

The tree above would be represented as follows:

```
node(node(node(empty,empty,0),node(empty,empty,1),1),
node(node(empty,empty,6), node(empty,node(empty,empty,9),8),7) , 4);
```

Write the functions using the searchtree datatype. The function sort should be of the following type:

```
fn treesort: int list -> int list
```

Solution:

```
datatype searchtree = empty | node of searchtree*searchtree*int;

(* insert a new value into a binary search tree *)
fun insert empty new = node(empty,empty,new)
 | insert (node(left,right,n)) new = if (n > new)
   then node((insert left new),right,n)
   else node(left,(insert right new),n);

(* create a binary search tree from a list *)
fun maketree list = foldl (fn (a,b) => insert b a) empty list;
```

```
(* loads a binary search tree into a list in a sorted manner *)
fun load empty = []
 | load (node(left,right,n)) = (load left)@[n]@(load right);

(* first makes a tree from the original values and them loads them
  back in a sorted order *)
fun treesort list = load (maketree list);
```

7 Turing Machines

Problem 7.1 (Turing Machine) 6pt 6min

Construct a TM that adds 1 to a binary number. Your input is preceded and followed by ##. The head points to the first # in the beginning. E.g. : ##0100## → ##0101##

Note: You will receive 6 $points - the\ number\ of\ states\ above$ 4 (excluding the halting state, if you choose to use one).

Solution:

```
q0: # q1 > #

q1: 0,1 q1 > 0,1
q1: # q2 < # //go back one

q2: 0 halt < 1 //replace 0 with 1 and halt
q2: 1 q2 < 0 ////replace 1 with 0 and go left
q2: # halt > 1 //if the bits are not enough add 1 in the beginning
```

8 Problem Solving and Search

Problem 8.1 (Power Source Search) 20pt 20min

A robot is on the 5x5 map shown below. It wants to reach a power source, but its sensors only allow it to detect the source once it is in the same cell with it. Find a problem formulation in the quadruple format presented in the lecture such that depth first search will find a solution after expanding exactly 6 nodes.

Assume that the `next` function of the DFS algorithm used returns the `(action, state)` tuples in the order in which the corresponding operators are defined. For example, if your operators are `jump` and `sing`, then the next function called on state i would return a list `[(jump, state j), (sing, state k)]` and not the other way around. (this is just an example, these operators will not do a very good job ... :))

Define a path cost for this problem. What is the cost of this solution? Is the solution optimal?

How many node expansions would BFS make considering the same `next` function?

			R	
P				P

R represents the robot and P a power source.

Solution:

```
S = {1,2,...25}
I = 9 //initial state
G = 20 //goal state
O = {down, right, up, left} // this is why the
//order in which they are specified matters, like this you get 5 expansions
```

```
DFS will go
9 -> 14 (down) -> 19 (down) -> 24 (down) now there is no down available so ->
25 (right) -> no down, no right, so up takes u to 19, goal state
```

So it expands 6 nodes.

With cost 1 per move, the cost of the solution is 5 and it is not optimal

BFS would find the solution at depth 3, so for this it will make all the expansions until depth 2, so $1 + 4 + 16$, then on the last level, the solution would be the second node expanded because the correct path is down down right. The first node will correspond to down down down, but the next will be the right solution. So $1 + 4 + 16 + 2 = 23$.

Problem 8.2 (Greedy Search and Hill Climbing) 10pt 10min

Name *relevant* similarities and differences between greedy search and hill climbing. Describe a real-life problem where you would use greedy and one where you would use hill climbing.

Note: A search tree or the queen's problem are not real-life problems.

Solution:

Similarities:

- both pick the best next state
- both search methods
- neither is optimal

Differences:

- one local, the other informed search
- one does not keep track of the path
- greedy get stuck, local search not

---

9 Prolog

8pt
8min

Problem 9.1: Generate all strings with N ones and M zeros.
Sample input:

```
gen(3,2,X).
X = [1, 1, 1, 0, 0] ;
X = [1, 1, 0, 1, 0] ;
X = [1, 1, 0, 0, 1] ;
X = [1, 0, 1, 1, 0] ;
X = [1, 0, 1, 0, 1] ;
X = [1, 0, 0, 1, 1] ;
X = [0, 1, 1, 1, 0] ;
X = [0, 1, 1, 0, 1] ;
X = [0, 1, 0, 1, 1] ;
X = [0, 0, 1, 1, 1] ;
```

Hint: The solution should not be more than 5 clauses.

Solution:

```
gen(0,0,[]).
gen(M,N,[1|L]):-M>0,X is M-1,gen(X,N,L).
gen(M,N,[0|L]):-N>0,X is N-1,gen(M,X,L).
```