Name: Matriculation Number:

Midterm Exam
General CS II (320201)

March 31, 2008

You have one hour(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 60 minutes, leaving you 0 minutes for revising your exam.

You can reach 26 points if you solve all problems. You will only need 23 points for a perfect score, i.e. 3 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	1.2	2.1	2.2	3.1	3.2	Sum	grade
total	2	4	4	6	6	4	26	
reached								

Good luck to all students who take this test

1 Graphs

Problem 1.1 (Node Connectivity Relation is an Equivalence Relation) 2pt 6min

Let $G = \langle V, E \rangle$ be a *directed* graph and the relation C be defined as

$$C := \{\langle u, v \rangle \,|\, \text{there is a path from } u \text{ to } v\}$$

Prove or refute that C is an equivalence relation.

Hint: Recall the properties of an equivalence relation!

Solution:

Proof:

P.1.1 reflexivity: From every node, there is a zero-length path to itself. □

P.1.2 symmetry: If there is a path from a node u to a node v, it's not necessary that there is a path from a node v to a node u. So symmetry doesn't hold for this. □

P.1.3 transitivity: If there is a path from u to v and a path from v to w, we can reach w from u via v. □

□

Conjecture 1 *1. Let $G = \langle V, E\rangle$ be a undirected graph. Then $\sum_{i=1}^{\#(V)} deg(v_i)$ is an even number.*

2. The number of vertices with an odd degree is even.

3. If $\#(V) \geq 2$ then $\exists v_1, v_2 \in V.deg(v_1) = deg(v_2)$.

Problem 1.2 (Degrees in an Undirected Graph) 4pt 10min

Prove of refute the conjecture above

Note: For undirected graphs, we introduce the notation deg with $\text{deg}(v) = \text{indeg}(v) = \text{outdeg}(v)$ for each node.

Solution: Here we prove the first item by figuring out what does $\sum_{i=1}^{\#(V)} \text{deg}(v_i)$ equal to.
Proof: by induction over $m = \#(E)$

P.1.1 $m = 0$ **(base case)**: For graphs that only consist of isolated nodes, both assertions hold trivially. □

P.1.2 $m \to m+1$ **(induction step)**:

P.1.2.1 If we remove an arbitrary edge $e \in E$ from G, we obtain $G\backslash\{e\}$

P.1.2.2 $G\backslash\{e\}$ is a directed (or undirected, resp.) graph with m edges.

P.1.2.3.1 **directed graph**: By removing one edge, we have decreased the sum of in-degrees as well as the sum of out-degrees by one. □

P.1.2.3.2 **undirected graph**: By removing one edge $e = \langle u, v\rangle$, we have decreased the degree of u as well as the degree of v by one and thus the sum of degrees by two. □

□

□

---

2 Combinatorial Circuits

Problem 2.1 (Binary Arithmetics) 4pt 12min

Let $A = 586$ and $B = -21$.

1. convert the numbers into a corresponding n-bit TCN system. What is the proper minimal n for converting both A and B?

2. perform the binary operations $B + A$ and $B - A$ and check the result by converting back to the decimal system.

Solution:

Problem 2.2 (Conditional circuit) 6pt 10min

Design a 2-bit conditional circuit that implements the following operation:

if $x \leq y$ then $x + y$ else $x - y$

Solution: It's Ankur's problem.

---

3 Machine Programming

Problem 3.1 (Discrete Integration) 6pt 10min

Given is $n \geq 1$ stored in $P(0)$ and n numbers stored in $P(1) \ldots P(n)$. Write an assembler program that calculates $\sum_{i=1}^{n} -1^{i} D(i)$ and stores the result in $P(1)$. Write comments to each line of your code (like in the example codes from the slides).

Problem 3.2 (Convert Highlevel Code to $\mathcal{L}$(VM) Code) 4pt 12min

Given are natural numbers i, k and s and the following piece of imperative code:

```
while(i > k) do begin
 s := s + (i - k);
 i := i - 1;
 k := k + 1;
end;
```

Suppose i, k and s is loaded on a stack correspondingly (top value being s). Convert the code into $\mathcal{L}$(VM) code, considering that the final value of s should be on the top of a stack.