Name: Matriculation Number:

Midterm Exam
General CS II (320201)
April 4. 2006

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

The estimated time for solving this exam is 54 minutes, leaving you 6 minutes for revising your exam.

You can reach 29 points if you solve all problems. You will only need 27 points for a perfect score, i.e. 2 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	2.2	3.1	3.2	3.3	Sum	grade
total	5	5	3	6	6	4	29	
reached								

Good luck to all students who take this test

1 Resolution Calculus

Problem 1.1 (Resolution Calculus with Nand Connective) 5pt 10min

Develop a variant $PropCNFCalcNAND$ of the CNF transformation calculus presented in class that transforms propositional formulae expressed with $NAND$ (denoted by $\uparrow$) as the only logical connective. To do so just complete the scheme of inference rules given here:

$$\frac{\mathbf{C} \lor \mathbf{A} \uparrow \mathbf{B}^{\mathsf{T}}}{?} \quad \frac{\mathbf{C} \lor \mathbf{A} \uparrow \mathbf{B}^{\mathsf{F}}}{?}$$

With this variant $\mathcal{CNF}^{\uparrow}$ together with the usual inference rule from resolution calculus conduct a resolution proof to verify the formula $(A \uparrow A) \uparrow ((A \uparrow B) \uparrow (A \uparrow B))$

Solution:

$$\frac{\mathbf{C} \lor \mathbf{A} \uparrow \mathbf{B}^{\mathsf{T}}}{\mathbf{C} \lor \mathbf{A}^{\mathsf{F}} \lor \mathbf{B}^{\mathsf{F}}} \quad \frac{\mathbf{C} \lor \mathbf{A} \uparrow \mathbf{B}^{\mathsf{F}}}{\mathbf{C} \lor \mathbf{A}^{\mathsf{T}}; \mathbf{C} \lor \mathbf{B}^{\mathsf{T}}}$$

---

2 Combinatorial Circuits

Problem 2.1 (Combinational Circuit for Shift) 5pt 10min

Design an explicit 3-bit shifter (using only NOT, AND and OR gates) that shifts its input to the left by one, iff the control bit is 1 and leaves it as it is otherwise. Formally, design a circuit that corresponds to $f_{\text{shift}} \colon \mathbb{B}^3 \times \mathbb{B} \to \mathbb{B}^3$ with

$$f_{\text{shift}}(\langle a2, a1, a0\rangle, c) = \begin{cases} \langle a1, a0, 0\rangle & \text{if } c = 1 \\ \langle a2, a1, a0\rangle & \text{if } c = 0 \end{cases}$$

Hint: Think of a variant of multiplexer.

Solution:

Problem 2.2 (TCN Substraction) 3pt 6min

Let $A = 576$ and $B = 9$.

1. convert the numbers into an n-bit TCN system. What is the minimal n in order to encode A as well as B?
2. perform a binary subtraction $A - B$ and check the result by converting back to the decimal system.

Solution: n should be 11. The 11-bit TCN representations are:

$$\langle\!\langle B(576) \rangle\!\rangle = 01001000000$$

$$\langle\!\langle B(9) \rangle\!\rangle = 00000001001$$

The subtraction $A - B$ in TCN representation is

$$\langle\!\langle B(576) \rangle\!\rangle - \langle\!\langle B(9) \rangle\!\rangle = \langle\!\langle B(576) \rangle\!\rangle + \langle\!\langle \overline{B(9)} \rangle\!\rangle + 1$$

This corresponds to

$$01001000000 - 00000001001 = 01001000000 + 11111110110 + 1 = 01000110111$$

In fact: $\langle\!\langle B(576 - 9) \rangle\!\rangle = \langle\!\langle B(565) \rangle\!\rangle = 01000110111$

---

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 performs a sum of the array and stores the result finally in $P(1)$. Write comments to each line of your code (like in the example codes from the slides).

Solution:

P	instruction	comment
0	`LOAD` 0	$ACC{:}=P(0)=N$
1	`SUBI` 1	$ACC{:}=ACC-1$
2	`JUMP` 8	if $n=1$ then stop
3	`MOVE` ACC $IN1$	$IN1{:}=ACC=n-1$
4	`LOADIN` 1 1	$ACC{:}=P(ACC)+1=P(n)$
5	`ADD` 1	$ACC{:}=ACC+P(1)$
6	`STORE` 1	$P(1){:}=ACC=P(1)+P(n)$
7	`MOVE` $IN1$ ACC	$ACC{:}=IN1=N-1$
8	`SUBI` 1	$ACC{:}=ACC-1=N-1$
9	`JUMP` -7	if more numbers then continue
10	`STOP` 1	stop

Problem 3.2 (While Loop in $\mathcal{L}$(VM)) 6pt 10min

Write a program in the Simple While language that takes two numbers A and B, given at the memory addresses 1 and 2, and returns $(A+B)^{42}$. Show how the compiled version of it looks like in the Virtual Machine Language $\mathcal{L}$(VM) (concrete, not abstract syntax).

Solution:

`var n := 1; var a := A; var b := B;`	`con` 1 `peek` 1 `peek` 2
`var c := a+b;`	`add` `con` 1
`while n >= 42 do (`	`peek` 0 `con` 43 `leq` `cjp` *halt*
`  p := c * p;`	`peek` 1 `peek` 2 `mul` `poke` 2
`  n := n + 1;`	`peek` 0 `con` 1 `add` `poke` 0
`)`	`jp` back
`return p;`	`halt`

Problem 3.3 (A Turing Machine that stops for twins) 4pt 8min

Given a tape arbitrarily filled with ones and zeros. Define a transition table such that the machine reads the tape from left to right as long as the entries are alternating between ones and zeros and terminates otherwise.

Hint: The Turing machine terminates when there is no action in the transition table applicable.

Solution: Since the write operation is irrelevant for this solution it is omitted in the transition table

Old	Read	Move	New
s_1	0	right	s_2
s_1	1	right	s_3
s_2	0	right	s_1
s_3	1	right	s_1