Homework Assignments
for
General Computer Science II (320102) Spring 2015

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

May 1, 2015

Contents

Assignment 1 (Graph Theory and Circuits) Given Feb. 5., Due Feb. 12.

Conjecture 1.1 *Let $G = \langle V, E \rangle$ be a undirected graph. Then $\sum_{i=1}^{\#(V)} \deg v_i$ is even.*

Proof: by induction on $\#(V)$

P.1.1 $\#(V) = 1$ **(base case)**: The sum is 0 because there are no edges => even □

P.1.2 $\#(V) = n \to \#(V) = n + 1$ **(induction step)**: We know that for $\#(V) = n$, the sum over all nodes of their degrees is even by I.H. To have $n + 1$ nodes, we need to add a node $v_n + 1$ to the graph. This node will be connected to some other node v_j, so the edge uniting them adds 2 to the overall sum of degrees. Thus we have that

$$\sum_{i=1}^{n+1} \deg v_i = 2 + \sum_{i=1}^{n} \deg v_i$$

and since $\sum_{i=1}^{n} \deg v_i$ is even by I.H., the new sum is even as well. □

□

Problem 1.1 (Wrong induction proof)
A lot of students write proofs like this on exams. Can you find what is wrong with it? 30pt

Note: In doing induction proofs, you need to make sure that your proof will cover all possible cases. For this, you need a good induction step that will take you from one case to the next more complicated case without leaving anything in between. Also, the set of base cases are important: starting from this set, you have to be able arrive at any other case by just applying the induction step. Another thing about the base cases is that they should not be unnecessarily many.

Problem 1.2 (k-connected graph)
Let a **k-connected graph** be a graph in which every vertex is an endpoint of exactly k edges. Prove or refute that in every k-connected graph there is a path of exactly k edges such that no edge is included twice. 15pt

Problem 1.3 (Generate Boolean Tree)
Write an SML function `MakeBTree` which takes an odd integer n and returns a balanced binary tree with n nodes of the type `datatype btree = leaf | parent of btree*btree` 25pt

Problem 1.4 (DNF Circuit with Quine McCluskey)
Design a combinational circuit for the following Boolean function: 30pt

X_3	X_2	X_1	$f_3(X)$	$f_2(X)$	$f_1(X)$
0	0	0	1	0	1
0	0	1	1	0	0
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

Assignment 2 (Graph Theory and Circuits) Given Feb. 12., Due Feb. 19.

Problem 2.1 (Converting to decimal in SML)
Write an SML function to_int = fn : string −> int that takes a string in binary, octal or hexadecimal notation and converts it to a decimal integer. If the string represents a binary number, it begins with 'b' (e.g. b1011), if it is an octal number - with '0' (e.g. 075) and if it is a hexadecimal number it begins with '0x' (e.g. 0x3A). 20pt

If the input does not represent an integer in one of these three forms raise the InvalidInput exception.

For example we have to_int("b101010") −> 42

Problem 2.2 Design a two bit multiplier using only NOT, OR, AND and XOR gates. 10pt

Problem 2.3 (6-ary AND gate)
Using only normal AND gates draw the inside of a 6-ary AND gate. What is the minimal depth of such a circuit and why? 5pt

Problem 2.4 (Conditional circuit)
Design a 4-bit conditional circuit that implements the following operation: 20pt

if $x \leq y$ then $x + y$ else $2x$

Problem 2.5 (Periodic Number Representation)
Prove that if the base-b representation of a real number $x = x_0.x_1x_2x_3\ldots_b$ is periodic then x is a rational number. The representation is periodic if there are positive integers n and p such that $x_k = x_{k+p}$ for all integers $k > n$. For example, $10.11_2 = 2\frac{3}{4}$, $0.112_3 = 0.112112112\ldots_3 = \frac{7}{13}$, and $2.012_8 = 2.012222222\ldots_8 = 2\frac{9}{448}$. 35pt

Assignment 3 (Sequential Circuits and RAM) Given Feb. 19., Due Feb. 26.

Problem 3.1 (Coca-Cola Vending Machine)

A vending machine is a machine which dispenses items such as snacks or drinks to customers automatically, after the customer has inserted some form of credit. 25pt

You are asked to design the "brain" for a rudimentary Coca-Cola vending machine. The price of 1 Coca-Cola is 3 coins.

More specifically, your task is to assemble a circuit which receives 2 binary inputs and has 1 binary output.

Here are the input details:

- The 1st input will be connected to the machine's credit detector. The credit detector will signal a short pulse of 1 (like a clock) whenever a coin is inserted in the machine.
- The 2nd input will be connected to the "BUY" button. When the buy button is pressed, a short pulse of 1 will be emitted on the second input.

The output will be used by other parts of the machine to control the insert coins and buy cycle. Here are the details:

- Output is 0, then inserting a coin should change the output to 1 only if it is the 3rd one introduced. Also, pressing the "BUY" button should have no effect.
- Output is 1, then inserting a coin shouldn't change anything, while pressing the "BUY" button should change the output to 0 and "restart" the coin count (so that 3 new coins can be introduced).

Note: Introducing 4 coins consecutively would result in losing 1 coin, but we assume our customers are smart and buy after having introduced 3 coins.

Problem 3.2 (Accumulator Circuit)

To finish off your work on designing circuits, here is an all-encompassing assignment. You have to design an accumulator. 30pt

It should be implemented as a circuit which takes an 8-bit input value, together with a 2-bit control input. When this control input is "off" (i.e. its value is 00), the accumulator should store the 8-bit input value. When the control input is 10, the accumulator should add the 8-bit input value to the value previously stored in it, and save the new result. Similarly, when the control input is 11, the accumulator should subtract the 8-bit value from its old value. You are supposed to use a CSA for this problem.

Note: Since this is a problem that is supposed to summarize all your knowledge on circuits from GenCS, please create your circuit out of gates. You will need building blocks like flip-flops and adders, but please draw their detailed circuits (you can copy from the slides), and then use some short notation for them.

Problem 3.3 (Incremental Adder)

25pt

Design a circuit that recursively adds consecutive numbers, i.e. it first adds 1, then 2, then 3 etc. The starting number is zero and all the numbers are represented with 4 bits. There is no need to deal with overflows.

The outputs should be 1, 3, 6, 10, 15, 5 ... etc.

Problem 3.4 (TCN in SML)

Given the datatype: `type tcn = int list` write the following SML functions 20pt

- `extend = fn : tcn -> int -> tcn` which takes a number in Two's complement representation and an integer n and makes the tcn number n bits wide. Here n should always be bigger or equal to the current width of the number.

- int2tcn = fn : int −> int −> tcn which converts an integer number to a tcn number given the width.
- tcn2int = fn : tcn −> int which converts a tcn number to an integer number.

If in any of the functions a number can't fit into the requested number of bits raise the NumberDoesNotFit exception.

Note: A number's width is simply the number of bits that are used to represent that number.

As an example conisder

```
int2tcn 10 8 −> [0,0,0,0,1,0,1,0]
```

Assignment 4 (REMA) Given Feb. 26., Due Mar. 04.

Problem 4.1 Write a REMA program that calculates the result of $D(1) - D(2) + D(3) \ldots D(n)$ where n is the first index such that $D(n) = 0$. Write the result in $D(0)$ 30pt

Problem 4.2 (Simulating REMA in SML) 40pt

Given the following declarations:

```
datatype register = acc | in1 | in2;
datatype instr = load of int | loadi of int | loadin1 of int | loadin2 of int |
                 store of int | storein1 of int | storein2 of int |
                 add of int | addi of int | sub of int | subi of int |
                 move of register*register| nop of int | stop of int |
                 jump of int | jumpe of int | jumpne of int |
                 jumpl of int | jumple of int | jumpg of int | jumpge of int;
type program = instr list;
type memory = int list;

(* This is the state of the machine. From left to right the values mean:
   PC register; ACC register; IN1 register; IN2 resigter; Memory cells*)
type state = int*int*int*int*(int list);
```

Write two SML functions:

- execute_instr : instr −> state −> state
- run : program −> memory −> memory

The first function takes an ASM instruction and the current state of the REMA as arguments and returns the new state after the instruction is executed. The second function takes a program and the initial configuration of the memory. It then simulates the program until a STOP 0 instruction is reached and returns the memory at that point. In both functions 'memory' is just a list of integers that represent the current state of the memory of the REMA. Once the initial list is supplied, during simulation its length shoudn't change.

Note: For this problem and the next it will be very helpful to use built-in SML functions. Make sure to check the forums for more info.

Problem 4.3 (A Shift Program)

Write an assembler program that shifts the values of only the first n cells to its upper neighbor, where n is the content of the accumulator; i.e. if $D(k) = a_k$ for $k = 1 \ldots n$ is the state before the execution of the program then it must be $D(k+1) = a_k$ for $k = 1 \ldots n$ afterwards and the program must terminate. 20pt

Problem 4.4 Write an ASM program that multiplies numbers at addresses 0 and 1 and stores the product at address 2. Show intermediate states of registers/memory. 10pt

Assignment 5 (Stack Based Languages) Given Mar. 04., Due Mar. 11.

Problem 5.1 (SML simulator for $\mathcal{L}(\texttt{VM})$ with static procedures)

Write an SML function vm_simulator that simulates a program written in Virtual Machine Language. It should take as input: 40pt

- a string list, a list of $\mathcal{L}(\texttt{VM})$ commands including `con` i, `add`, `sub`, `mul`, `leq`, `jp` i, `cjp` i, `peek` i, `poke` i, `proc` i j, `arg` i, `call` i, `return`, `halt` and numbers. Note that, for example, "`con` " and "5" count as 2 instructions.
- an int list describing the initial state of the stack; the first element in the list represents the top of the stack while the last element represents bottom of the stack. Of course it can also be initially empty.

```
val vm_simulator = fn : string list * int list -> int list
```

As there are a few things that could possibly go wrong, make sure to use the following exceptions:

- NumberExpected, if no number follows after `con` , `jp` , `cjp` , `peek` , `poke` .
- InvalidPeekPoke, if the number following `peek` or `poke` is smaller than 0 or bigger than the size of the stack.
- InvalidJump, if the number following `jp` , `cjp` , `call` or `proc` makes the virtual program counter point to an unexisting instruction.
- InvalidStackSize, if the stack is not big enough so that the current instruction can be normally executed. (`add`, `sub`, `leq` etc.)
- UnknownCommand, if the input string list contains strings that are not one of the VML commands described in class.

Signature and examples:

```
val vm_simulator = fn : string list * int list -> int list

vm_simulator(["con","~2","con","~7","mul","con","3","con","4","mul","sub"],[2,9,7]);
val it = [~2,2,9,7] : int list


vm_simulator(["proc","2","9","arg","1","arg","2","add","return",
              "con","1","con","2","con","3","con","4",
              "call","0","call","0","call","0","halt"], []);
val it = [10] : int list
```

Note: You can safely assume that in each test case only 1 version of $\mathcal{L}(\texttt{VM})$ will be used, i.e. either $\mathcal{L}(\texttt{VM})$ with `peek` i and `poke` i or $\mathcal{L}(\texttt{VM})$ with `proc` i j and `call` i.

Problem 5.2 (SW TCN Converter)

Write a Simple While program that converts TCN to decimal. You will be given the TCN number t (consider it a variable set at the beginning of your program). You can consider that the number t is an integer formed only from 1s and 0s and the most significant bit is represented by the first digit; remember that the first bit also tells you whether the number is negative or positive. 25pt

Your program should start, for example, with:

```
var t=11001;
```

That is, $t = 11001$, which corresponds to -7.

Problem 5.3 (Simple While program on Fibonacci)

15pt

Write a Simple While Program that takes a number N and computes the N^{th} Fibonacci number. Then provide the Abstract Syntax for your code.

Show how the $\mathcal{L}(\mathrm{VM})$ version of it looks like by compiling it.

Problem 5.4 (Divisibility check)

Assume the data stack initialized with `con` a and `con` b for some natural numbers a and b. Write a $\mathcal{L}(\texttt{VM})$ program that returns on top of the stack 1 if a is divisible by b, and 0 otherwise. 20pt

Assignment 6 (Stack Based Languages) Given Mar. 11., Due Mar. 18.

Problem 6.1 (Separation TM)
Design a Turing Machine that accepts an input of the form $w \in \{0, 1, *, +\}^*$. Its purpose is to relocate all $*$ on the left side of the tape and all $+$ on the right side of the tape. Initially on the tape we have randomly scattered elements of our alphabet, for example $010 + + * +01 + * + + * 1$. In this case the tape should be $* * *010011 + + + ++$ at the end of the execution. There is no restriction on how you handle non-$+$ and non-$*$ characters in your alphabet. 30pt

Problem 6.2 Given a binary string w, terminated by a hash mark, program a Turing machine that erases its input (i.e. $w\#$) and outputs $\#x$, where $x = w^R$ is the word w reversed. 20pt

Simulate your Turing machine on the input 101#.

Problem 6.3 (Formal languages with Turing Machines)

25pt

Design and implement a Turing Machine that, given a string surrounded by #, determines whether it is of the form $0^n1^{2n}0^n$. It should end in the state "yes" if the string matches the pattern and in the state "no" otherwise. You can assume that $n > 0$ and that the head will be positioned at the first non-# character of the input. Please add comments on how the general idea works.

Problem 6.4 (TM for double word)
How would you design a Turing Machine that accepts the language $\{vv \mid v \in \{0, 1\}^*\}$, halting with "yes" if such input is given and halts with "no" otherwise? Assume a tape infinite in both directions, with the controll unit positioned at the first character of the input. Describe in plain English the details of your algorithm. Think of possible wrong inputs, and show how your TM handles them. 30pt

---

Note: You have to be thorough and specific in all the steps of your algorithm. E.g.:

Traverse the input of the tape from left to right and stop at the last character... - **OK!**

Divide the input in two and and compare the two halves - **NOT OK!** - How do you do that?

Assignment 7 (Internet) Given Mar. 26., Due Apr. 9.

Problem 7.1 (SML HTTP Server)
Your task is to implement a simple HTTP web server in SML. Carefully examine the sample HTTP GET request given in the slides and also experiment with other sites to see some more examples. Try and see what happens if you request a non-existing file. Your server should be able to handle such requests. As there may be some differences between different browsers, make sure your server works well at least with Firefox and/or Chrome. 50pt
Signature:

```
val webserver = fn : string * int -> bool
```

The first argument specifies the root directory of your server and the second one is the port number. If an exception occurs during the execution of this function, then false should be returned, otherwise, it should run forever, waiting to accept new connections.

Note: Your server should deal only with GET requests.

Assignment 9 (Internet) Given Apr. 16., Due Apr. 23.

Problem 9.1 (For Future Generations)

70pt

As one of the last assignments, we would like you to look a bit into the future. Imagine yourselves one year from now. Some of you will definitely be TAs at that time, so it's time to show your creativity and teaching skills. Your task is to basically create an HTML form representing the examination you would give to the freshmen in 2017. It can be any midterm or final for GenCS I or II. There are only a few specifications you must look out for. The rest is fully up to you.

The web form must:

1. Include multiple choice and 'fill in the blanks' questions, enough for an actual exam time of 75 or 120 minutes.
2. Include all of the following: button, radio button, check box, drop down box, text input.
3. The exam must contain figures and sections of code from any of the studied programming languages that you ask questions on.
4. Link your exam to some useful pages. Make it like an 'open book' exam and offer some actual existing resources.
5. The overall style should be professional. Put a bit of effort into appearance and aesthetics.
6. In the end, the scoring system should work. Nothing too fancy, but it should be an operational exam from start to finish.

Assignment 10 (Encryption and Search) Given Apr. 23., Due Apr. 30.

Problem 10.1 (SML Web Crawler)

A web crawler is a program that will store a copy (mirror) of a web site. Generally, crawlers access a given web page and, after retrieving the HTML source, they extract the links and also download those pages (or images or scripts). This will provide the user the possibility to access these pages even when they are not connected to the internet or to perform different measurements on the pages. 30pt

Your task is to write your own SML Web Crawler, following these steps:

1. Make sure that you downloaded and understood the SML sockets example file used in the last assignment. Use the following updated socketReceive function:

```
(* Receives maxbytes bytes from the socket. Returns the string message. *)
fun socketReceive(sock, maxbytes) =
        Byte.bytesToString(
                Socket.recvVecNB(sock, maxbytes)
        );
```

   The problem with this function is that, if the server sends a message longer than maxbytes, all the remaining bytes will be queued on the socket, but not processed. Write your own fullMessage function that overcomes this problem by reading the whole reply from the server (you can use socketReceive, it will return a string of length 0 if the message from the server is finished). Your function should have the following type:

```
val fullMessage= fn : ('a,Socket.active Socket.stream) Socket.sock -> string
```

2. Now, write a method that, given a *host* and *page*, will make a HTTP **GET** request to the server for the given *page* on that *host*, and will return the HTTP response. Your function should have the following signature:

```
val getPage = fn : string * string -> string
```

   For example, you should be able to run getPage("en.wikipedia.org","/wiki/Main_Page") and retrieve the home page of Wikipedia.

3. Now that you have the HTTP response, check it closely and you will discover that it contains the HTML web page, but also some headers. In order to be sure that you will only store the HTML page, write a function extractHTML that scans the string and discards everything that **is not** between <html> and </html>. Of course, your function will have the signature:

```
val extractHTML : string -> string
- extractHTML("Discard me! <html><head><title>Hello!</title></head></html>");
val it = "<html><head><title>Hello!</title></head></html>" : string;
```

4. Write a function extractLinks that will go through your HTML source code and will return all the links that it contains. Feel free to look into the HTML or RegExp library of SML, but making your function only going through the string and extracting sequences like the following will suffice:

```
<a href="extract me!">...
<img src="extract me!"> ...
```

   You are not required to handle links other than the ones found in anchors and images. Your function will have the following signature (get a string and return a list of strings which are the links found):

```
val extractLinks = fn : string -> string list;
```

5. Mind the fact that these links might contain the protocol ("http://"), might be relative to the root of the host ("/img/happy.png"), or might be relative to the current page ("next/index.html"). Your getPage function requires a *host* and a *page* as arguments, and the *page* should be relative to the host root (i.e. absolute path). Write an SML function normalizeLinks that, given a host, page and list of strings, will return a list of pairs (*host*, *page*) that can be used by the getPage:

```
val normalizeLinks : string * string * string list -> (string * string) list;
normalizeLinks("www.example.com", "/en/test.html",
        ["http://www.google.com/something/x", "/img/happy.png", "next/index.html"]
);
val it = [
        ("www.google.com", "/something/x"),
        ("www.example.com", "/img/happy.png"),
        ("www.example.com", "/en/next/index.html")
] : (string * string) list;
```

6. This sub-task will be to write the wrapping crawler function.

   Have a look at the following SML function that writes a string to a file:

```
fun writeToFile(file, content) =
        let
                val os = TextIO.openOut(file)
                val vc = String.toString(content) (* we need an SML vector *)
                val _ = TextIO.output(os, vc)
                val _ = TextIO.flushOut(os)
        in
                TextIO.closeOut(os)
        end;
```

   This function will be used in storing the HTML page to disk. Your crawler will have the following signature:

```
val crawler : string * string * int -> unit;
```

   The first two parameters are the host and the starting page (i.e. "`www.example.com`" and "`/test/index.html`"). The third parameter is an integer representing the maximum depth you should go into. You will follow the following steps:

   (a) use getPage to retrieve the HTTP response
   (b) use extractHTML to extract only the HTML part of the response
   (c) write the HTML part to a file (see the note below!)
   (d) use extractLinks and normalizeLinks to get the list of links to follow further
   (e) recursively call the crawler method; remember to decrease the depth and not proceed with a negative depth!

Note: There might be problems with storing images. We will not grade this problem based on the output, but rather on how well you managed to follow the instructions and on your intermediary results. Please think about what the problem with images is and write a short comment at the end of your sml file!

Problem 10.2 (Transport Layer Secutrity)

15pt

- What is Transport-Layer Security (TLS)? Specify the steps which are executed by a client and a server in a TLS handshake.
- Explain the difference between public and private key encryption?
- What is an asymmetric key?
- Name 4 current commonly used encryption algorithms and explain one in detail.

Assignment 11 (eXtensible Markup Format) Given Apr. 30., Due May. 7.

Problem 11.1 (Parsing XML)
We are going to consider XML files with a special, more restricted format: 50pt

- Each line contains a Tag or a Content String.
- A Tag line is either of form “<NAME>” (for opening tags) or “</NAME>” (for closing tags). The lines between the opening and the closing tags can contain a Content String (in which case the Tag is called “leaf”) or several other tags (and then the Tag is named “node”).
- The Content String is a valid SML string.
- The file always starts with a Tag and ends with the corresponding closing Tag (this tag is called the “root” of the document).
- A Tag “B” that is contained in a node “A” is called “son” of “A”. Conversely, “A” is the “father” of “B”. “B” can have multiple “brothers” (which are Tags that also have “A” as a father).

An example of a valid XML file:

```
<first>
 <a2>
  <b3>
   text
  </b3>
  <b1>
   text3
  </b1>
 </a2>
</first>
```

The root is “first”, we have node “a2”, with leaves “b3” and “b1”.

We can “normalize” an XML file by alphabetically sorting the children of every node after the tag name. In case there are several children with the same name, we want them to keep the relative order they had in the input file (in other words, the sort is stable).

You are required to write an SML function sortTags that takes two arguments (the input file name and the output file name), reads the input file and then writes the normalized XML file to the output.

Example:

File “input.xml”:	SML:	File “output.xml”:
<r>		<r>
<a3>		<a1>
<b2>		<ff>
Father		BBB
</b2>		</ff>
<b3>		<ffx>
Mother		AAA
</b3>		</ffx>
</a3>	sortTags(‘‘input.xml’’, ‘‘output.xml’’);	</a1>
<a1>		<a3>
<ffx>		<b2>
AAA		Father
</ffx>		</b2>
<ff>		<b3>
BBB		Mother
</ff>		</b3>
</a1>		</a3>
</r>		</r>

Note: The whitespace at the beginning of each line is ignored, but it should be preserved to the output file. Consider the input file to be well formatted (no closing tag / opening tag without match inside their parent tag).