Exam in INF3110 Programming Languages
Day of exam: November 29, 2017
Exam hours: 14:30 – 18:30
This examination paper consists of XXXX pages.
Appendices: No
Permitted materials: All printed and written, including the textbook

Make sure that your copy of this examination paper is complete before answering.

This exam consists of 3 questions that may be answered independently. If you think the text of the questions is unclear, make your own assumptions/interpretations, but be sure to write these down as part of the answers.

Good luck!
Question 1. Runtime-systems, scoping, types (weight 40%)

Exercise 1:

Consider the program below, which is written in some new, hitherto unknown language (i.e., it is made up for this exam). The language supports interfaces and structs as the main units of decomposition, as well as variables and functions ("methods") within such structs.

Assume that the program is correct in the given language, that is, it contains no syntax errors or type errors, and its execution does not result in a runtime error.

Furthermore, assume that the program starts executing the Main function, and that print(<expr>) will print the string representation of the expression (akin to toString() in Java) to the console/terminal.

```java
interface I {
    int function1();
}

struct A {
    int function1() {
        int x = 21;
        int y = x + function2();
        return y;
    }

    int function2() {
        return x;
    }
}
```
struct B {
    int thisOneGoesTo = 11;

    int function1() {
        return thisOneGoesTo;
    }
}

void Main() {
    I myI = new A();
    int result = getValue(myI);

    myI = new B();
    result = result + getValue(myI);

    printResult();
}

int getValue(I i) {
    return i.function1();
}

void printResult() {
    print("The result is: ");
    print(result); // HERE!
}

1a)

With a reasonable semantics, and a corresponding implementation of the language, what will be printed by the statement labeled "HERE!" in the program text above? Explain your reasoning briefly.
1b)

It is reasonable to assume that this language has some properties that differ a bit from most "mainstream" languages such as e.g. Java and C#. Discuss briefly (and informally) your interpretation of the semantics of the language, focusing on topics discussed in class.

1c)

Draw the runtime stack at the point labeled "HERE!" in the program text above. That is, right after the call to print(result). Utilize your interpretation of the language's semantics from question 1b. Include objects and closures in the drawing as necessary.

1d)

Consider the following Java program snippet:

```java
public class Program {
    public static void main(String[] args) {
        Object[] myArgs = args;
        myArgs[0] = 42;
    }
}
```

Explain why this piece of code is unsafe (there might be more than one reason!). Explain briefly what, from a language designer standpoint, can be done to amend the situation, should you be the creator of Java from scratch.

1e)
Assume now, that our made-up language should support generic function definitions (parametric polymorphism). Propose a syntax for this. You should do this by providing an EBNF notation for function signatures. (You do not need to include a definition of the function body.) The definition should support generic parameters with both covariant and contravariant constraints. Explain how this is solved with your suggestion.

You can assume that standard definitions of identifiers, strings, parameter lists, etc are provided in the grammar for you.

Use the following notational convention when writing your EBNF grammar:

<non-terminal>
“terminal”
[ optional ]
alternative1 | alternative2
( grouping )
zero-or-more-repetitions*
one-or-more-repetitions+

In this question, we will allow structs to be generic, and use this to define function parameters. We will assume that there is a predefined struct called Func which has two generic parameters, representing a function’s (single) formal parameter type (U), and return type (T), respectively. Example:

struct Func of T and U { … }

If you will, you can imagine “of T and U” replaced with your own syntax for generic parameters from the previous exercise. If you did not answer the previous question, just use the provided example syntax.

We will in this question use this struct to define formal function parameters to other
functions.

With this in mind, consider the following program:

```java
int f((Func of int and int) g) {
    int x = g(42);
    return x;
}

int dummy(int val) { return val; }

void Main() {
    f(dummy);
}
```

Draw the runtime stack for the program above, right before the line “return val” is executed in the function dummy.

**Question 2. ML (weight 40%)**

**2a**

Evaluate the following ML expressions:

a) `(fn x => fn y => x @ y) (List.filter (fn x => x > 2) [1,2,3]) (List.filter (fn x => x < 2) [1,2,3])`

b) `map (fn x => x 1) (map (fn x => fn y => x + y) [1,2,3])`

**2b**

Assume the standard definitions of `foldl` and `foldr`:

```ml
fun foldl(f: 'a*'b->'b)(acc: 'b)(l: 'a list): 'b =
    case l of
```
\[
\begin{align*}
\text{[] } &= \text{ acc} \\
| x::xs &= \text{ foldl } f (f(x, \text{acc})) \hspace{1mm} xs
\end{align*}
\]

\textbf{fun foldr} (f: 'a*'b ->'b) (acc: 'b) (l: 'a list): 'b = 
\textbf{case} \hspace{1mm} l \hspace{1mm} \textbf{of} \\
|[] &= \text{ acc} \\
|x::xs &= f (x, \text{ (foldr f acc xs)})
\]

Describe based on the example of \texttt{foldr/foldl (op @) [] [1,2]} the difference in stack usage during evaluation of both expressions.
2c

1) Define a type 'coord' to represent a position in three-dimensional space as tuple of three real numbers.
2) Give two example expression that define different values of that type.
3) The Pythagorean distance between two points \((x_0,y_0,z_0)\) and \((x_1,y_1,z_1)\) in this three-dimensional space is calculated as

\[ d = \sqrt{(x_1-x_0)^2 + (y_1-y_0)^2 + (z_1-z_0)^2} \]

Define the function \(\text{distance} = \text{fn} : \text{coord} \times \text{coord} \rightarrow \text{real}\) that calculates that distance, where \(x_0\) is the value on the x-dimension of the first argument and \(x_1\) the value on the x-dimension of the second, etc.

You may use the predefined functions

\[ \text{sqrt} : \text{real} \rightarrow \text{real} \]
\[ \text{pow} : \text{real} \times \text{real} \rightarrow \text{real} \]

The latter returns the first argument raised to the power of the second.

2d

In a café, menu items are represented by the following datatype:

\[
\text{datatype} \text{ item} \\
\quad = \text{Vgn} \text{ of string} \times \text{real} \\
\quad \mid \text{Vgt} \text{ of string} \times \text{real} \\
\quad \mid \text{Omn} \text{ of string} \times \text{real};
\]

The constructors are:

\(\text{Vgn} == \text{vegan} == \text{no animal produce}\)
\(\text{Vgt} == \text{vegetarian} == \text{no meat}\)
\(\text{Omn} == \text{omnivorous} == \text{may contain animal produce}\)

The string is the item’s name and the real is its price. A menu is represented as a list of items, for example:

\([\text{Vgn}("\text{tofu}", 3.6), \text{Vgt}("\text{quiche}", 2.7), \text{Omn}("\text{haggis}", 1.9)]\);

Give the type-signatures for all your definitions in the answers a)-e) below!

1) Write a function to find all vegan items on a list.

2) Write a function which, given an order tuple and a menu, returns the total cost of the order. An order is a tuple of a food item and a quantity, for example:
3) Write a function which, given a list of order tuples and a menu, returns the total cost of all the orders.

2e
Calculate the type for the following expression according to the ML type inference algorithm:

\[
\text{fn x => fn y => (y x) (x (y x));}
\]

Derive the corresponding equations through help of a parse graph, and solve the resulting equation system to obtain the type of the root node.

**Question 3. Prolog (weight 20%)**

3a
1) Define the predicate \( \text{and/3} \) to model the boolean function "and", using the constants 0 and 1 to mean “false” and “true” respectively.

2) Assume the following definition of logical disjunction \( \text{or/3} \):

\[
\text{or(0,0,0).} \\
\text{or(_,_,1).}
\]

We can now ask the following query, obtaining an obviously undesirable second solution:

\[
?\text{- or(0,0,Result).} \\
\text{Result = 0 ;} \\
\text{Result = 1.}
\]

Explain why we get two solutions!

3) Give an alternative definition of \( \text{or/3} \) with only two clauses that exhibits the correct behaviour of logical disjunction.

3b
Natural numbers can be represented in Prolog through the Peano encoding using e.g. 0 (for the constant zero), and \( \text{s(X)} \) for the successor of \( X \) (in other words \( X + 1 \) in standard arithmetic).

Example: \( 2 = \text{s(s(0)).} \)

1) Write a ternary predicate \( \text{add(X,Y,Z)} \) that is true if \( Z \) is the sum of \( X \) and \( Y \), where the arguments are expected to be Peano-encoded natural numbers.
That is, e.g. add(s(0),s(0),s(s(0))) is true.

2) Recall that the Fibonacci numbers are recursively defined as a function:
\[ f(0) = 0 ; f(1) = 1 ; f(n+2) = f(n+1) + f(n) . \]

Give a Prolog predicate \( \text{fib}(A,B) \), such that it is true if \( B \) is \( f(A) \).

Use the Peano encoding of natural numbers, instead of Prolog's built-in math, and refer to \( \text{add}/3 \) from (a) if necessary.

3) Define a predicate \( \text{fibAcc}/4 \) that uses an accumulator to calculate the Fibonacci numbers with a single recursive call instead of two.

That is, \( \text{fibAcc}(N,A,B,X) \) should be true, if \( X \) is the \( N \)th Fibonacci number, using the accumulators \( A \) and \( B \).

Also give a wrapper function \( \text{fibA}/2 \) that calls \( \text{fibAcc}/4 \) and correctly initialises the two accumulator-parameters.