EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

**Mathematisch-Naturwissenschaftliche Fakultät**

**Programmiersprachen**

**Prof. Klaus Ostermann**

Steven Lolong
steven.lolong@uni-tuebingen.de

# Object Meets Function

Assignment 4 – Winter Semester 23/24

Tübingen, 10. Januar 2024

**Handin** Please submit this homework until Thursday, January 18th 2024 via email to Steven Lolong (steven.lolong@uni-tuebingen.de) before 24:00.

**Email Format** Use this format for the email's subject and file's name: **OmF-W23/24-Assign**[no]**-**[YourName]

## Task 1: Language for Propositional Logic (16 Points)

### 1.1 ADT (6 Points)

Design ADT of programming language for propositional logic with operators And, Or, Not, NotAnd (NAND), and Implication.

### 1.2 Evaluator (4 Points)

Build an interpreter (evaluator) to translate your language to Boolean. Assume your ADT's name is Logic, then your evaluator type:

```
1 def eval(lgc: Logic) : Boolean = { ... }
```

### 1.3 Syntactic Sugar (6 Points)

Syntactic sugar means making something like sugar candy for the programmer. For example (see assignment 3): multiply is syntactic sugar for additional. The additional is syntactic sugar for the successor (succ). In this task, please design a programming language based on subsection 1.1 but eliminate the abstraction for And, Or, Not, and Implication by defining them as syntactic sugar for NAND. An example of syntactic sugar for the And using the NAND:

```
1 def and(lhs: Logic, rhs: Logic) : Logic = NAND(NAND(lhs, rhs), NAND(lhs, rhs
    ))
```

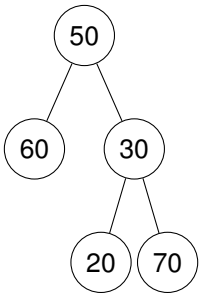
*note:

1. Please use the evaluator from task 1.2 to prove your design is correct.

## Task 2: Binary Tree (10 Points)

### 2.1 Abstracting binary tree (3 Points)

Please design the ADT for a binary tree that contains Integer. The requirement is for every node should have two leaves and one core for Integer.




### 2.2 Operation on Binary Tree (10 Points)

1. Make a function to count the number of node. For example, in the tree above, the total number of members is 5. (2 points)
2. Make a function to count the number of cores. A core should a node and two leaf. For example, in the tree above, the total number of cores is two. The first core is 50, 60, and 30. The second core is 30, 20, and 70. (2 points)
3. Make a function to transform the binary tree expression into a list of integers. (3 points)
4. Make a function to translate from List of Integer into your ADT's binary tree expression (3 points)

## Task 3: Polymorphism (6 Points)

Abstract a class that can take type parameters such as Integer or Boolean (or another type). The class has a property with the type either List of Integer or List of Boolean. The class also has methods: plus and multiply. The type constructor is defined when the class is instantiated. In math, plus is the same as the operator Or, and multiply is the same as the operator And. When the members' type is Boolean, the method plus will fold it using the logic operator Or. But, when its type is Integer, the plus will sum it all using the arithmetic operator +. If the type is other than Integer or Boolean, then you must print alert information about the supporting type. Ex.:

```
1 class Ex [...]{
2   var member : List[...]
3   def mult(...) : ... = { ... }
4   def plus(...) : ... = { ... }
```

> Please make an example for each task, as the evidence of your design is correct!