Assignment #5a - Static Analysis
Datastructure
Discrete Mathematics

Anders Kalhauge and Tobias Grundtvig

Spring 2021

The `VSSL` programming language

To experiment with static analysis, we will use a **V**ery **S**imple **S**tructured **L**anguage.

The key features of the language is:

- The only data types are booleans (`BOOLEAN`) and integers (`INTEGER`).
- It supports functions and procedures, but we will not use that in this example.
- It supports blocks of code, curly braces around statements ended with semicolons.
- It supports selections with the `IF (...)` *`code block`* and `IF (...)` *`code block`* `ELSE code block` statements
- It supports iterations with the `WHILE (...)` *`code block`* statement.
- Code blocks, selections, and iterations can be nested.

The computer state

To do the static analysis we need a data structure to register the possible states of the computer after each statement of the program has been executed.

The computer state should:

- Keep track of each variable in the program and which possible values it can have in the state. In rare cases we know the exact value of a variable, but mostly, we only know a span of possible values or maybe only the fact that the variable is initialized.
- Be partially ordered.
- Support local minimum (intersection) and local maximum (union) between states (will be explained in class).

1 Create a computer state data class

Create a class that supports the features described above. The class must implement a set of known variables each being a set of possible values. Remember, that the minimum and maximum functions must operate on the variable sets as well.

Hand in

A link to the github repository. In groups on Peergrade by Friday April 9^{th}