Sent an email to Wouter that the proposal is done.

The first step is usage analysis.

TO define one of these analyises, in need to figure out for every variable for how often it is used.

Begin with never used, used once and used more than once lattice. This lattice has three elements, [0,0], [1,1], [1 – infinity]

This lattice will need a join.

Then for every CFG fragment, I will need to figure out the effect on the lattice.

Defining a variable will add an element to the set.

Aliasing will not be dealt with yet at this point.

In an if else block, at the end, any variables introduced there will be discarded.

You can assume variable parameters are initialized.

It is good to separate checks from computations.

Separate errors from detecting how often a variable is used, this should be done in different stages.

The structure in the lecture Alejandro will send me should be good.

FIRTree plugin for IntelliJ shows the exact control flow graph for Kotlin code.

[Granule](https://granule-project.github.io/) are doing a type system approach.