Possible values of usage lattice

A diagram of a structure

Description automatically generated

VariableDeclerationNode:

|  |  |
| --- | --- |
| Variable a | Domain R |
| VariableDeclarationNode a | Add a to analysis domain R with value 0 |
| QalifiedAccessNode a | If a has value 0, set to 1. If 1, set to ∞ |
|  |  |
|  |  |

Does x.y count as just a use for (x.y), or also for x?

In the default version: no

fun exampleFieldCount()  
{  
 val a = blub()  
 val y = *something*(a) // usage of a is on one  
 val z = a.x + 3 // a.x has usage one, but a's usage does not increase  
}

In this example if a.x does not count as a usage for x, a would have a use count of one. Because of this, the reference of a could get reused to store y, causing the next reference of a to be pointing to the wrong address, resulting in undefined behaviour.

While this does mean that using a field of a variable also incurs a use of the variable, counting a field access in combination with the variable it is from is still useful, because if a field is only used once, the memory can be safely reused, even if the rest of the object would be used elsewhere.

Does passing something as an argument to a function count as a use?

Yes. For example, if you pass variable x as an argument, and x is returned as a result, this would cause issues.

What initial value would function arguments have?

Ideally function arguments would retain their usage value from when they were called in analysis. If a function was called with different usage values, each unique combination could be considered a unique function.

One problem is potential aliasing within arguments that could not be clear from the get go, so aliasing will have to be solved.