More description is needed for how everything is stored, how two tree joins are handled

What happens if the lattices have the same fields, how do you combine those?

What happens when you find a new field?

Also improve the scoping section

If a variable is declared and used within a loop, the reuse is still possible.

Say a variable is used in a loop, always assign infinite to it

When i have a variable that comes from out of scope and it is used within something that has a backedge, make it infinite.

Add the bottom element to the lattice

Looking at variable initialization analysis could help

Say something like: “an element of the altice has this form, it is a dictionary...”

Also improve description for what happens within functions

Read only variables for globals can be used

Consider two cases, when initializing the class and when using the class. During initialization you can “change” the variable of vals.

“top level variables”

If i’m calling a function, everything you know about the fields is invalidated, because you don’t know what happens to them.

Also have a bunch of representative examples that show the usage analysis works.