Applied Static Analysis

An Introduction to Points-to and Alias Analysis

Software Technology Group
Department of Computer Science
Technische Universität Darmstadt
Dr. Michael Eichberg

> If you find any issues, please directly report them: GitHub

Some of the images on the following slides are inspired by slides created by Eric Bodden.

Points-to analysis vs. alias analysis

- Points-to analysis computes for each variable the allocation sites whose objects the variable *may*/*must* point to: $\textbf{points-to}(v) = \{a1, a2, \ldots\}$
- Alias analysis determines which variables *may* or *must* alias, i.e., point to the same objects:
  - $\textbf{may-alias}(v1, v2) = true/false$
  - $\textbf{must-alias}(v1, v2) = true/false$

In case of a *may* analysis $\boldsymbol{true}$ means **maybe**. I.e., if two variables may alias then they may point to the same object, but they don't have to. If the answer is $\boldsymbol{false}$, they definitively never alias.
In case of a *must* analysis $\boldsymbol{false}$ (only) means *maybe not*.

May vs. must alias analysis

```
a = new A();
if(..) {
    b = a;
}
c = new C();
d = c;
```

$\text{may-alias}(a, b) = true$

$\text{must-alias}(a, b) = false$

$\text{may-alias}(a, c) = false$

$\text{must-alias}(c, d) = true$

References