Challenge

Analyze **non-interference**

Changing values of higher-level variables does not impact values of lower-level variables.

The technique should be: sound, static, compositional, automatic, practical.

Information Flow Calculus

```
void main() {
    if (z == 1)
      then if (x == 1)
        then y = 1
        else y = 0
    else x = y
}
```

Label command variables as in/out

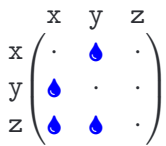
Example C: x = exp

$$\mathrm{In}(\mathtt{C}) = \mathrm{Occ}(\mathtt{exp})$$
$$\mathrm{Out}(\mathtt{C}) = \mathtt{x}$$

Record data flows in a matrix.

Security Flow Matrix

```
void main() {
    if (z == 1)
      then if (x == 1)
        then y = 1
        else y = 0
    else x = y
}
```

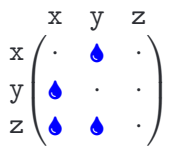


- · no dependency from v_{in} to v_{out}
- 💧 NI violation if $\ell(v_{in}) \not\leq \ell(v_{out})$

Evaluation

1. Matrix of constraints
2. Information flow policy,
   Variable security classes
3. Evaluate satisfiability

SAT means program is non-interfering.



- $\cdot$ no dependency from v_{in} to $\mathrm{v}_{\mathsf{out}}$
- 💧 NI violation if $\ell(\mathrm{v}_{\mathsf{in}}) \not\leq \ell(\mathrm{v}_{\mathsf{out}})$