Review
Static Analysis and Model Checking

SWEN 326: Safety Critical Systems

School of Engineering and Computer Science
Victoria University of Wellington
Static Analysis

```c
int abs(int x) {
    int r;
    if (x < 0) {
        r = -x;
    }
    return r;
}
```

- **Algorithm:**
  - Generate **Control Flow Graph** (CFG)
  - Perform **Traversal** of CFG
  - Maintain **set of assigned variables** at each point
  - **Intersect** assigned variables at join points
But **method invocations** visit instructions more than once!

Therefore, **call** instruction begins **new traversal**.

And, **ret** instructions **terminates traversal**
Indirection is a problem!

We can use Integer Range Analysis to deal compute ranges.
NonNull Analysis

When starting from **scratch** ...

... adding one `@NonNull` Annotation ...

... often leads to another!

Also, `@NonNull` and `@Nullable` are related!
new Rectangle(1,1,5,5).contains(1,1);

Q) How can we determine the control-flow for `main()`?
Model Checking

- Each **state** models 608 bytes of data

- **Execute** from initial `rjmp` instruction

- Memory locations initially given **don’t care** values

- Current state **cloned** at choice points...

- ... and clone executed **after** current one finished.
Model Checking

- Explicit State Model Checking
- Model Checking vs Static Analysis
Acknowledgements

SWEN 326 Safety Critical Systems was originally developed by David J. Pearce, School of Engineering and Computer Science, Victoria University of Wellington