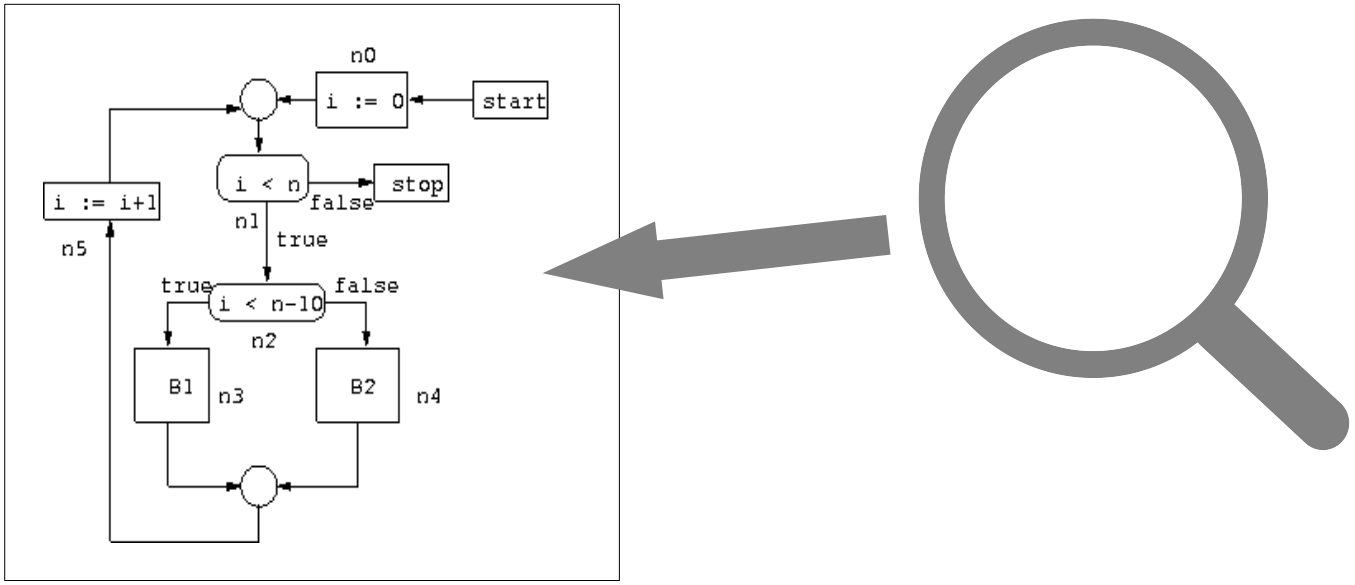
# Dynamic Program Analysis
# Lecture 8: Introduction



# Dynamic Program Analysis

- What?
- How?
- Why?

# What is Dynamic Program Analysis?

To analyze a program by running it and observing its behaviour

Basically a form of testing. But rather than testing the functionality in general, the goal is to find out about some particular property of the program

Examples:

- Illegal memory references (array index out of range, dereferencing null pointers)
- Memory leaks (dynamic memory being allocated, but never de-allocated)
- Performance (profiling)
- Program parts that can potentially be executed in parallel
- . . .

# Why Dynamic Program Analysis?

Some uses:

- Program verification (find possible bugs)
- Find out about other program properties, like performance (profiling)
- Help understanding the program's behaviour (program comprehension)
- Detecting security vulnerabilities

# Dynamic vs. Static Program Analysis

Considerable overlap in their uses

Pros and cons:

| **Property** | **static** | **dynamic** |
|---|---|---|
| Can guarantee absence of bugs | yes | no |
| False positives | yes | (no) |
| Requires advanced theory | yes | no |
| Must be able to run the code | no | yes |
| Requires good test data | no | yes |
| Time-consuming | sometimes | yes |

They can complement each other. Some bugs can be ruled out by static analysis, possible bugs left can be tracked down by dynamic analysis