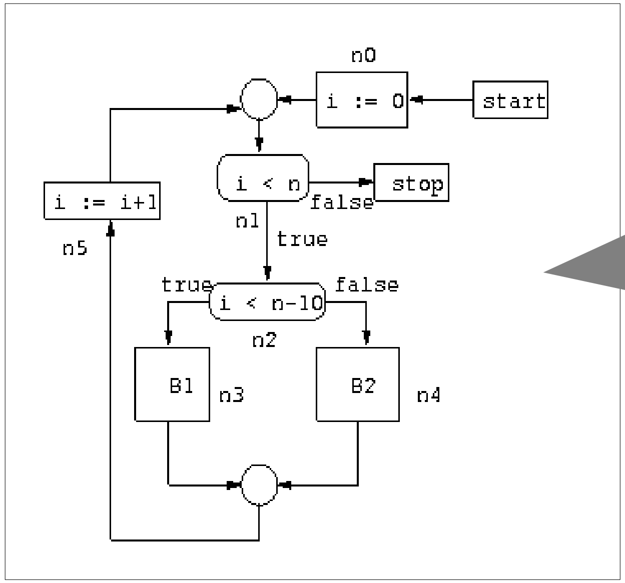
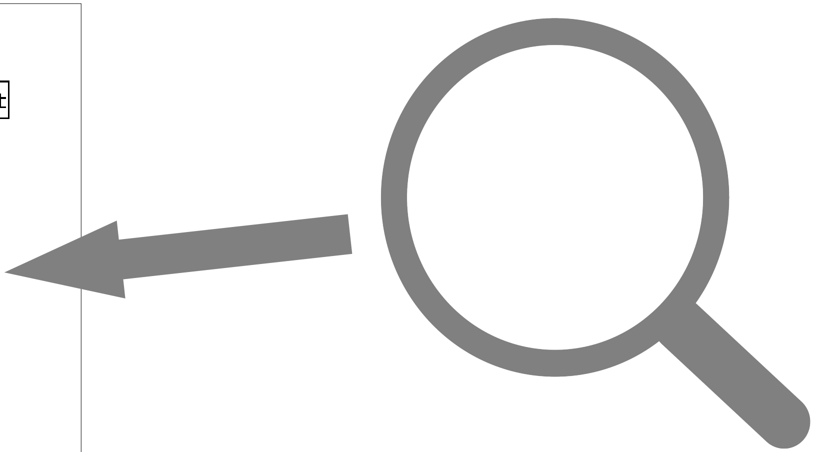
Static Program Analysis
Lecture 2: Safety, and Relation to Testing





Safety of Static Program Analysis

Static program analyses are typically designed to be *safe*

If the analysis says “no error” *then this can be trusted*

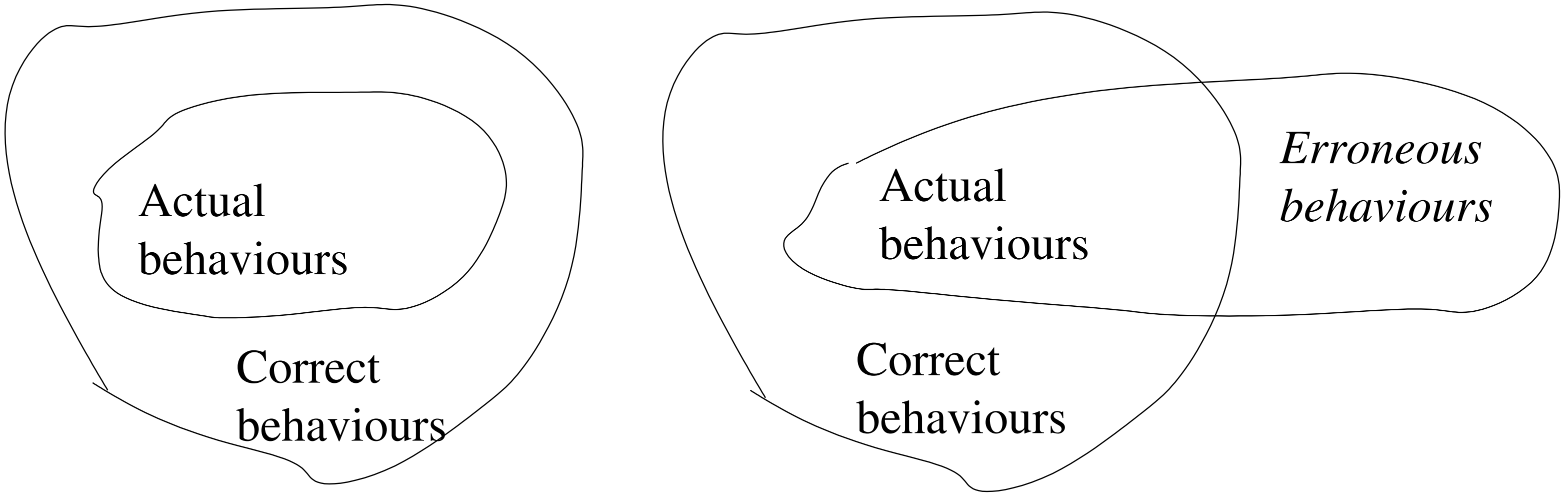
What about the reverse direction? If the analysis says “error”, is that always true?

The answer is no in general!

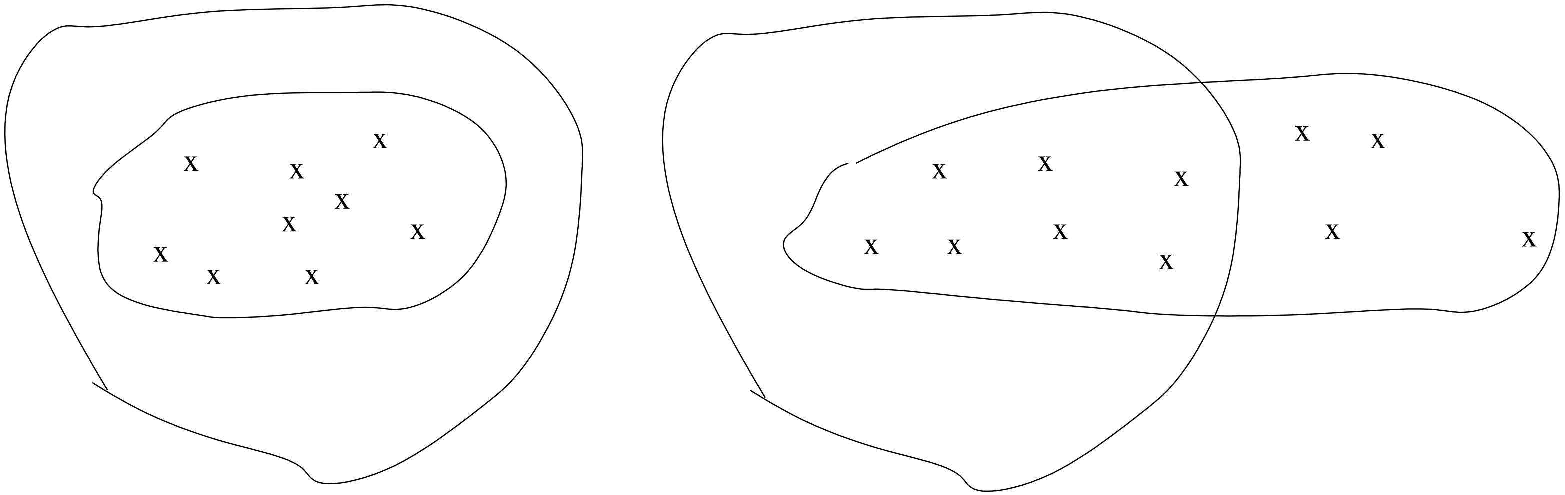
This is since many interesting properties to check are *undecidable*

Thus we’ll have to make do with weaker methods, saying either “no error” or “possibly an error”

Relation to Testing: A Correct and a Faulty System

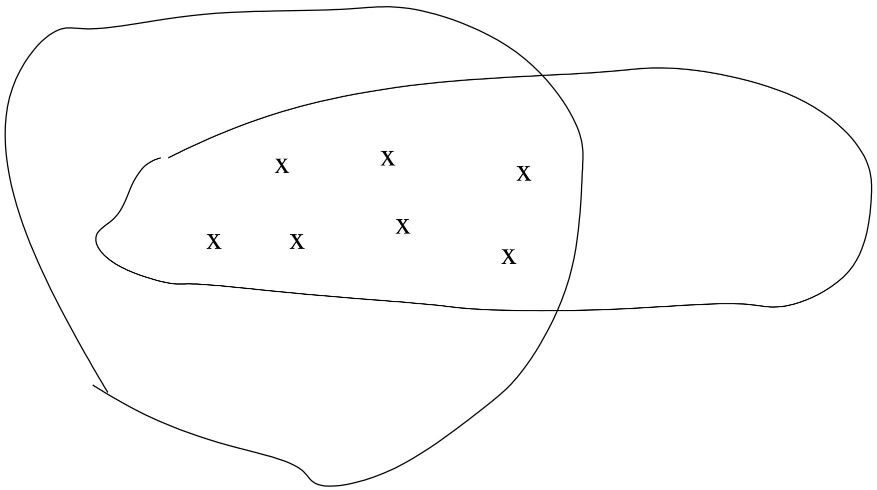


Testing



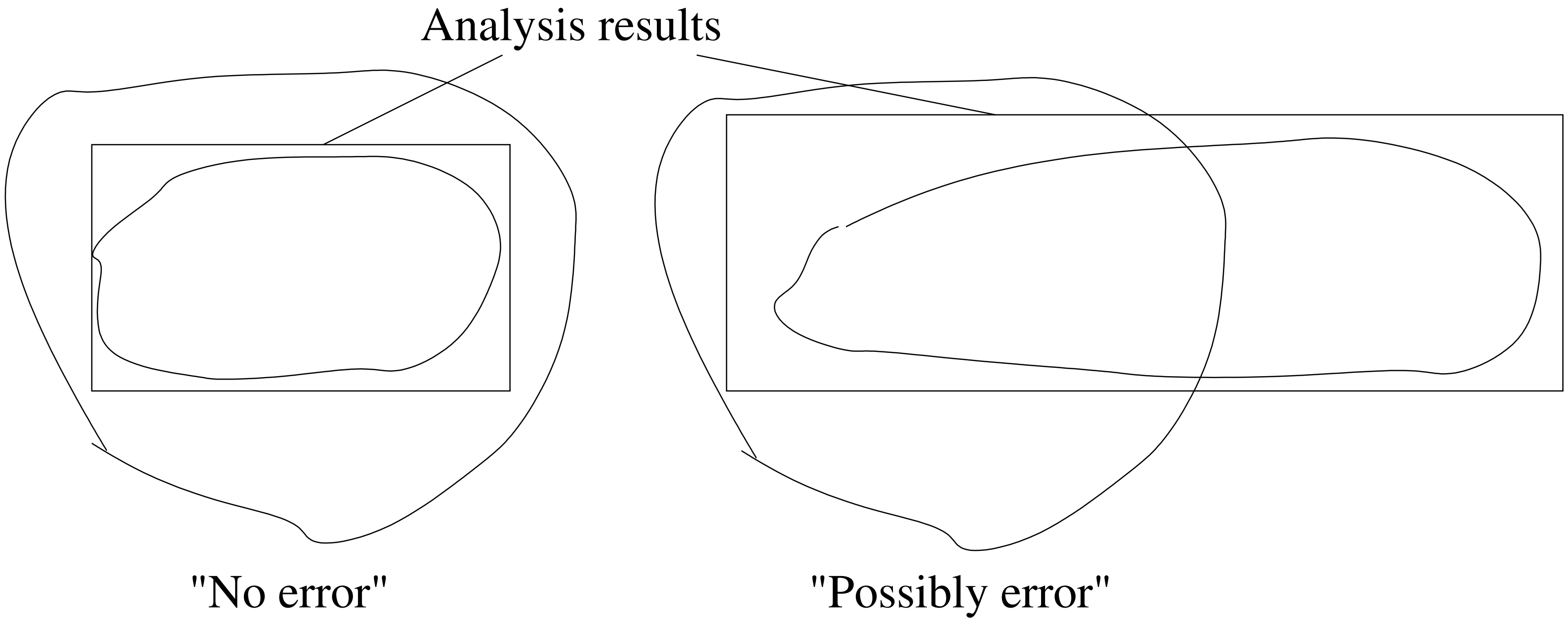
Only actually possible behaviours can be tested

Testing – the Problematic Case



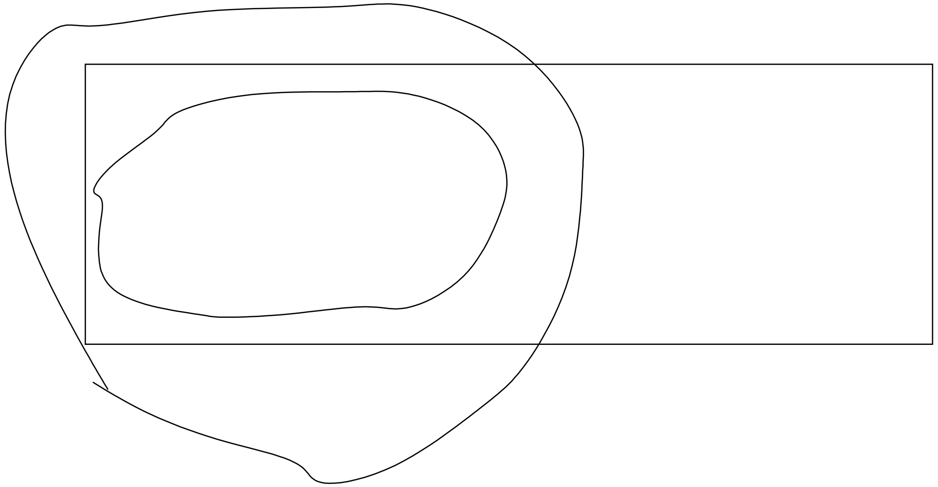
May miss errors unless exhaustive

Static Analysis



The set of actual behaviours is typically overestimated by the analysis

Static Program Analysis – the Problematic Case



A “false positive”

Too many false positives make the analysis less useful

Duality of Static Program Analysis and Testing

Static program analysis and testing are complementary:

- An error found by testing is real
- Testing can not guarantee absence of errors (unless exhaustive)
- An error found by static program analysis may be a false positive
- Static analysis can guarantee absence of errors

Both have their place in the SW engineer's toolbox!