Program Correctness in the 1970s

- Informal: testing
- Formal: hand proofs using Hoare logic:
  “15-page papers about correctness of 1/2-page programs”

⇒ existing formal methods did not scale.

(Plus: Hoare logic initially only for terminating programs.)
New Requirements

Automation:

to cope with complexity and lack of ingenuity
(e.g.: algorithm to find a loop invariant?)

Reactive systems:
ongoing behavior, infinite computations
(e.g.: operating systems, flight controllers)

Non-deterministic systems:
difficult to test (errors hard to reproduce)
What is Model Checking?

≈ “clever search through all possible states”
- can be automated (search algorithms)
- can handle non-determinism (is exhaustive)

Exhaustive search suffices for simple
“Is there a path to an error state?”
properties, but we need more . . .
What Else is Model Checking?

Advanced Properties:

- is statement $s$ infinitely often executed?
- is every request followed by a grant?
- is there a point after which never $x < 0$?

Counter Examples: if feasible, print out error trace as proof of violation (for credibility and debugging)

Contrast these with testing, theorem proving.