Mutation Testing

Jan Tretmans

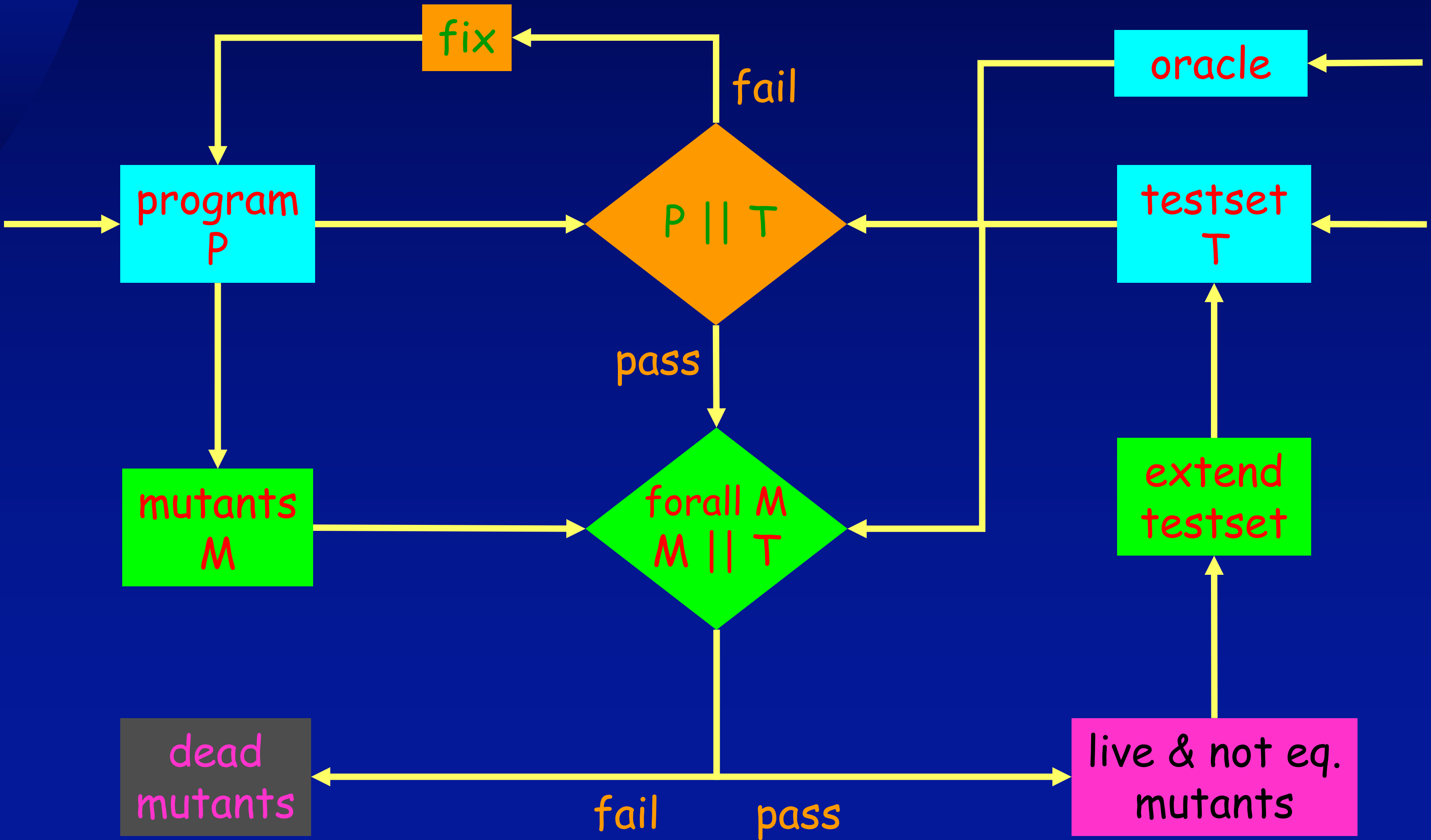
ESI
Eindhoven, NL

Radboud University
Nijmegen, NL

Mutation Testing

- Evaluation of test suite : What is good test data ?
- Measure/comparison of test data quality/test methods
- Improve/increment a test set
- Evaluation by large number of mutants :
  small modifications applied to IUT
- Try to make test suite that detects all mutants
- If test suite eliminates all mutants
  then IUT is likely correct (empirical evidence)
  else extend test suite to eliminate more mutants
- Iterative process until (almost) all mutants have been killed

Mutation Testing



Mutation Testing

- Assumptions :
  - Competent programmer hypothesis:
    programmers write programs that are (almost) correct
  - Coupling effect :
    if small faults are detected then also complex faults are
  - Mutant operators :
    small faults can be described as small modifications
    of the program by a set of predefined mutant operators
    - $x \to y$, $\geq \to >$, $< \to <>$, $+ \to -$
  - Test oracle :
    criterion to check correctness of output

Mutation Testing

- ☞ Complexity
  - ♦ number of mutants ≈ $O(loc^2)$
- ☞ Detection of equivalent mutants (testing equivalence on program-level).
- ☞ Useful for test suite quality determination, test selection (remove redundant tests), and experimental comparison of methods
- ☞ Tool support required
- ☞ Optimizations possible:
  - ♦ systematic test generation from living mutants
  - ♦ symbolic methods : mutation templates
  - ♦ approximations