

Testen
WS 2020
Automatisches Testen

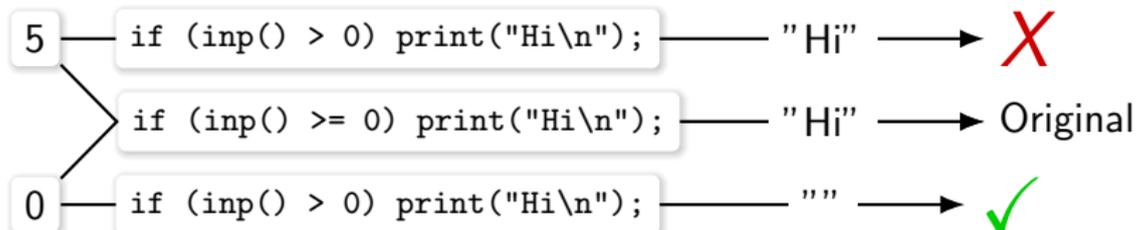
Prof. Dr. Dirk Beyer,
Thomas Lemberger

- Einführung
- Automatische Testerzeugung
 - Blackbox Testing (Random)
 - Greybox Testing (Distanzmetriken)
 - Whitebox Testing (Symbolic Execution)
- Testziel-Spezifikation mit FQL
- Mutations-Testen
- Übungen: PRTest, AFL-fuzz, KLEE, CPAchecker, FShell, SRCIROR

Mutations-Testen

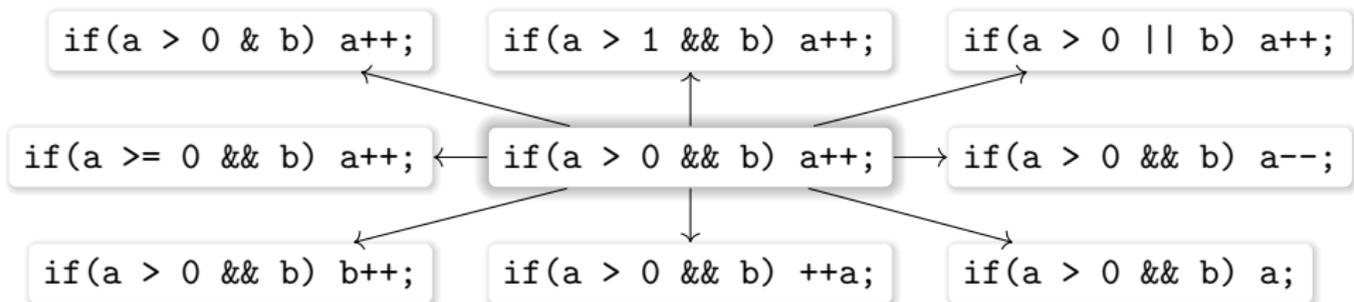
Mutations-Testen

- ▶ Kein Test-Generierungs-Verfahren, sondern Test Goal
- ▶ Test Goal: Decke möglichst viele Mutanten ab
- ▶ Mutant: Verändertes Original-Programm
- ▶ Mutant von Test abgedeckt: Test erkennt, dass Mutant Programmverhalten verändert
- ▶ Ein Test kann beliebig viele Mutanten abdecken



Programm-Mutationen

- ▶ Typische Mutations-Operatoren: Zahlen ersetzen, Operatoren durch andere Operatoren des gleichen Typ ersetzen



- ▶ Extrem viele Mutanten möglich
- ▶ Erstellung schnell möglich, aber Validierung skaliert nicht
- ▶ Offene Fragen:
 - ▶ Wieviele Programm-Stellen verändern?
 - ▶ Wieviele/Welche Mutanten notwendig?

- ▶ Competent Programmer Hypothesis: Programmierer schreiben Programme, die nahe an korrektem Programm sind
 - ▶ Coupling Effect: Tests, die viele kleine Fehler erkennen, können auch komplexere Fehler erkennen
- ⇒ First-Order Mutanten reichen (nur eine Änderung pro Mutant)

Kriterien für Mutanten-Abdeckung

- ▶ Gegeben Mutant P' , ein Test muss folgende Bedingungen erfüllen, um P' abzudecken:
 - R** Reachability: Mutierte Stelle muss erreicht werden
 - I** Infection: Mutierte Stelle muss Programmzustand verursachen, der unterschiedlich von Zustand im Original-Programm ist
 - P** Propagation: Fehlerhafter Programmzustand muss zu beobachtbarem Programmverhalten (z. B. Output) propagiert werden
- ▶ Existiert kein Test, der die Kriterien erfüllt, ist der Mutant *äquivalent*.
 - ▶ Erkennen von äquivalenten Mutanten unentscheidbar
 - ▶ Verschlechtern Test-Maß

- ▶ Java: <http://pitest.org/>
- ▶ C/C++/LLMV:
<https://github.com/mull-project/mull>,
<https://github.com/thierry-tct/mart>

Übungsblock 2: Mutationstesten