

Thema, Ziele: Verifikation und Validation (V&V)

Aufgabe 1: Codeabdeckung (code coverage)

a)

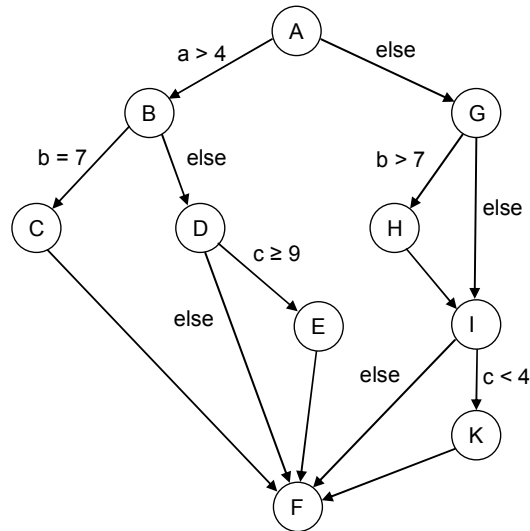


Abbildung 1: Verzweigungsbaum

- b) siehe Tabelle 1: Instruction coverage
- c) siehe Tabelle 2: Branch coverage
- d) siehe Tabelle 3: Path coverage

Tabelle 1: Instruction coverage

Pfad	a	b	c
ABCF	> 4	7	x
ABDEF	> 4	≠ 7	≥ 9
AGHIKF	≤ 4	> 7	< 4

Tabelle 2: Branch coverage

Pfad	a	b	c
ABCF	> 4	7	x
ABDEF	> 4	≠ 7	≥ 9
ABDF	> 4	≠ 7	< 9
AGHIF	≤ 4	> 7	≥ 4
AGIKF	≤ 4	≤ 7	< 4

Tabelle 3: Path coverage

Pfad	a	b	c
ABCF	> 4	7	x
ABDEF	> 4	≠ 7	≥ 9
ABDF	> 4	≠ 7	< 9
AGHIF	≤ 4	> 7	≥ 4
AGHIKF	≤ 4	> 7	< 4
AGIF	≤ 4	≤ 7	≥ 4
AGIKF	≤ 4	≤ 7	< 4

Es ist jeweils ein konkreter Testvektor zu bestimmen, der die entsprechende Bedingung erfüllt.

Aufgabe 2: Primzahlen bestimmen (Prüfungsaufgabe im FS 2011)

- a) Im schlimmsten Fall wird erst bei $(b*b \geq cand)$ abgebrochen. Die Schleife wird dann b mal durchlaufen, d.h. $O(b)$, bzw. $O(\sqrt{cand})$.
- b) Die folgenden Äquivalenzklassen könnten genommen werden:
- I. Keine Primzahl (Grenzwert 1)
 - II. Kleinste Primzahl
 - III. Kleinste Nicht-Primzahl ausser 1
 - IV. "Grosse" Primzahl
 - V. Keine Primzahl
- c) Man beachte, dass ein Testfall immer aus den Inputdaten und dem erwarteten Output besteht. Mögliche Testfälle sind pro Äquivalenzklasse:

#	Input (cand)	Erwarteter Output
(1)	cand = 1	false
(2)	cand = 2	true
(3)	cand = 4	false
(4)	cand = 31	true
(5)	cand = 15	false

- d) Der wirkliche Output ist in der Spalte "Resultat" aufgelistet.

#	Input (cand)	Erwarteter Output	Resultat	Ok
(1)	cand = 1	false	true	↔
(2)	cand = 2	true	true	➡
(3)	cand = 4	false	true	↔
(4)	cand = 31	true	true	➡
(5)	cand = 15	false	false	➡

- e) Wie zu sehen ist, funktioniert die Ausgabe nicht richtig. Die Zahl 1 müsste abgefangen werden. Alle Quadratzahlen einer Primzahl ergeben ebenfalls ein falsches Ergebnis.

Aufgabe 3: Implementation des Primzahlenalgorithmus

- a) Siehe Eclipse-Projekt in ./Loesung/Wrong
- b) Siehe Eclipse-Projekt in ./Loesung/Correct