



Lösungsvorschlag 13. Übung Softwaretechnik

Testverfahren

1 Kontrollflussorientiertes Testen (18 Punkte)

- a) Kontrollflussgraph mit Start-und Endknoten:
b) Testfälle für Anweisungsüberdeckung: Diese zwei Testfälle stellen Anwei-

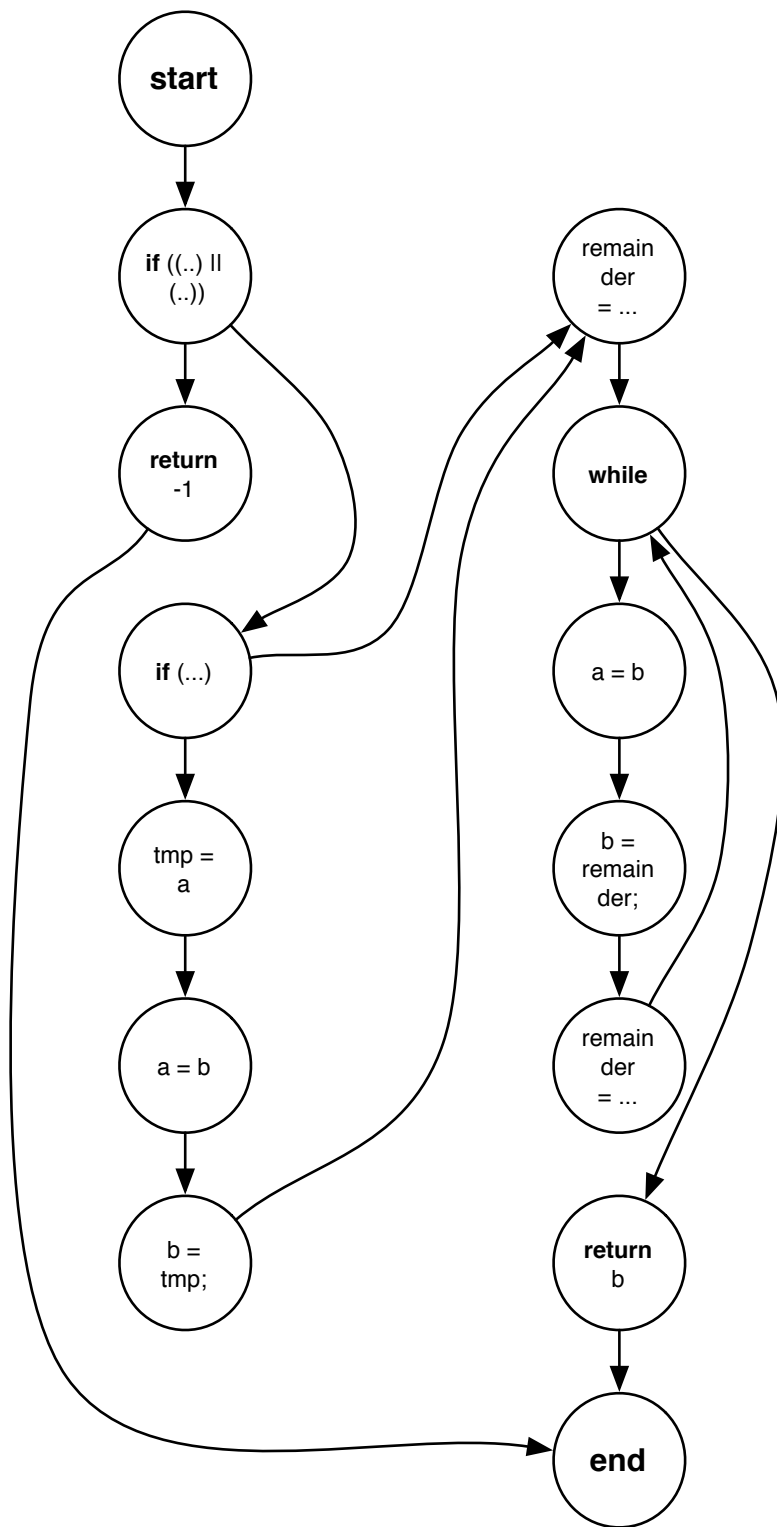
a	b	Kontrollfluss
0	0	Der erste <code>if</code> -Block wird ausgeführt und <code>-1</code> zurückgegeben.
7	8	Der erste <code>if</code> -Block wird nicht ausgeführt, der zweite schon, die <code>while</code> -Schleife wird einmal durchlaufen

sungsüberdeckung her.

- c) Zweigüberdeckung fordert, dass jede Kante im Kontrollflussgraph überdeckt wird, wohingegen Anweisungsüberdeckung nur fordert, dass jeder Knoten besucht wird. Die beiden Testfälle aus b) stellen keine Zweigüberdeckung sicher, da die Kante vom zweiten `if` zu `remainder=a%b` nicht besucht wurde.

a	b	Kontrollfluss
9	8	Der erste <code>if</code> -Block wird nicht ausgeführt, der zweite auch nicht, die <code>while</code> -Schleife wird einmal durchlaufen

- d) Durch `while`-Schleifen gibt es potentiell unendlich viele Pfade im Kontrollflussgraphen, so dass es nicht möglich ist, alle zu testen.
e) Beim Boundary-Interior-Pfadtest muss jede `while`-Schleife nur zwei oder mehrmals, jedoch nicht unendlich oft durchlaufen werden. Die Testfälle aus b) und c) genügen noch nicht, da die `while`-Schleife nur einmal durchlaufen wird.



a	b	Kontrollfluss
16	25	Der erste <code>if</code> -Block wird nicht ausgeführt, der zweite schon, die <code>while</code> -Schleife wird 4 mal durchlaufen

2 Minimale Mehrfach-Bedingungsüberdeckung (6 Punkte)

- a) Bei verknüpften bool'schen Bedingungen kann es vorkommen, dass sich atomare Bedingungen gegenseitig ausschließen (z.B. `if (employee.isProf() || employee.isHiwi())`), und somit nicht mehr alle Kombinationen der Prädikate getestet werden können.
- b) Testplan mit 2 Tests:

<code>!isAlreadyLoggedIn()</code>	<code>user.hasNeededCash()</code>	<code>user.isSuperUser()</code>	if(...)
T	T	T	T
F	F	F	F

3 Mutationstesten (6 Punkte)

- a) Mutationstesten kann eingesetzt werden, um die Güte einer Testsuite zu bestimmen (sh. Aufgabe b)) und die Zahl der Restfehler im Programm zu schätzen (sh. Aufgabe c)).
- b) Die Testsuite deckt nur die Hälfte der Mutationen auf, was viel zu wenig ist. Die Testsuite muss überarbeitet und erweitert werden.
- c) Berechnung gemäß der Formel im Skript.

$$\begin{aligned}
 E &= (M - X) \cdot \frac{N}{X} \\
 &= 8 \cdot \frac{2000}{1950} \\
 &\approx 8.21
 \end{aligned}$$