

Technische Universität
 Clausthal Institut für Informatik
 Prof. G. Kemnitz

8. Februar 2017

Test und Verlässlichkeit: Aufgabenblatt 1

Hinweise: Schreiben Sie die Lösungen, so weit es möglich ist, auf die Aufgabenblätter. Tragen Sie Namen, Matrikelnummer und Studiengang in die nachfolgende Tabelle ein und schreiben Sie auf jedes zusätzlich abgegebene Blatt ihre Matrikelnummer.

Name	Matrikelnummer	Studiengang	Punkte von 16

Aufgabe 1.1: Kontrollfragen:

- Unter welchen Bedingungen lässt sich ein System mit Gedächtnis wie ein System ohne Gedächtnis testen? 1P
- Was sind Fehler und was sind Fehlfunktionen? 1P
- Wie wie lässt sich kontrollieren, ob ein System deterministisch arbeitet? 1P
- Worin unterscheidet sich die Definition des Begriff »Modellfehler« vom Begriff »potenzieller Fehler«? 1P

Aufgabe 1.2: Eine Service-Leistung sei definiert durch

- das Eingabeformat: uint16_t (16 Bit, vorzeichenfrei),
- das Ausgabeformat: uint8_t (8 Bit, vorzeichenfrei),
- die Sollfunktion: Rückgabe der ganzzahligen Anteils der Wurzel. Hinreichende Kontrollbedingung

$$(y^2 \leq x) \wedge ((y + 1)^2 > x)$$

(x – 16-Bit-Eingabe; y – 8-Bit-Ergebnis).

Die nachfolgende Implementierung als C-Funktion ist fehlerhaft:

```
uint8_t wurzel(uint16_t x){
    uint8_t w=0;
    uint16_t sum=0;
    while (sum<x){sum += (w<<1)+1;
    w++;}
    return w;
}
```

Testen Sie das Unterprogramm in einer Schleife mit allen Eingabemöglichkeiten. Kontrollieren Sie zuerst, ob es für alle Eingabemöglichkeiten terminiert. Falls es zum Absturz kommt, suchen Sie eine Eingabe, bei der das passiert, untersuchen Sie dann nach der printf-Methode oder im Schrittbetrieb, warum es zum Absturz kommt. Beseitigen Sie die Ursache und wiederholen Sie den Test. Wenn das Testobjekt soweit repariert ist, dass es für alle Eingaben ein Ergebnis berechnet, ergänzen Sie im Testrahmen die vorgebebe und als ausreichend definierte Kontrollfunktion

$$(y^2 \leq x) \wedge ((y - 1)^2 > x)$$

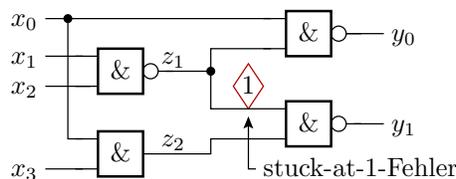
so, dass das Programm bei der ersten unzulässigen Ausgabe z.B. mit einem Unterbrechungspunkt gestoppt wird und die Eingabe sowie die fehlerhafte Ausgabe angezeigt werden. Daraus oder mit Zusatzuntersuchungen ist die Ursache der Fehlfunktion abzuschätzen, ein Fehlerbeseitigungsversuch zu unternehmen und der Reparaturerefolg durch Testwiederholung zu kontrollieren. Zum Schluss muss der Test komplett durchlaufen, d.h. für alle 2^{16} Eingabemöglichkeiten muss das Testobjekt ein Ergebnis liefern, dass die Kontrollbedingung erfüllt.

Abzugebende Ergebnisse:

- a) Eingabewerte, für die das Originalprogramm terminiert das richtige Ergebnis liefert? 2P
- b) Korrigiertes Programm, dass es für alle Eingabewerte richtige Ergebnis zurück gibt. 2P
- c) Der in Aufgabenteil b verwendete Testrahmen¹. 2P
- d) Eine Beschreibung der beseitigten Fehler. 2P

Aufgabe 1.3: Bestimmen Sie für den in der nachfolgenden Gatterschaltung eingezeichneten Haftfehler die Menge der binären Eingabevektoren \mathbf{x} 4P

- a) M_A mit denen der Fehler angeregt wird,
- b) M_B mit denen der Fehler beobachtbar ist und
- c) M_N mit denen der Fehler nachweisbar ist.



x_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1
x_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1
M_A														
M_B														
M_N														

Hinweis: Es genügt, wenn Sie jeweils die Spalten der Eingabevektoren in den Mengenzeilen ankreuzen, die zur jeweiligen Menge gehören.

¹Das Programm »main()«, das den Test ausgeführt hat.