

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

Hiermit erkläre ich mich damit einverstanden, dass die Klausurergebnisse per Aushang im Internet der HAW veröffentlicht werden.		
--	--	--

Punkteverteilung:	Aufgabengruppe	Teilaufgaben	Gesamtpunkte	Punkte
	1	10	10	
	2	10 + 15	25	
	3	20 + 20	40	
	4	25 + 20	45	
	Gesamt:		120	
Note 5 Punkte	ab 50 Punkten			Endnote
Note 15 Punkte	ab 100 Punkten			
Erlaubtes Material:	6 Seiten Notizen			
Dauer:	120 Minuten			

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

1 Testen im Lebenszyklus

1.1 Testebenen - Komponententest

10 Punkte

- Geben Sie zwei Testziele des Komponententests an und erläutern Sie sie kurz

Testziel 1	
Testziel 2	

- Nennen Sie kurz zwei wesentlichen Unterschiede zwischen dem Test-First Ansatz und herkömmlichen Testansätzen für den Komponententest

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

2 Statischer Test

2.1 Reviews

10 Punkte

Begründen Sie kurz, warum folgendes Auswahlkriterium für die Review-Arten sinnvoll ist:

Wenn viel Fachwissen über das Prüfobjekt für die Gutachter notwendig ist, sollte zunächst ein Walkthrough durchgeführt werden, danach ev. noch eine Inspektion oder ein technisches Review.

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

2.2 Metriken - Komplexität

15 Punkte

Berechnen Sie die Zyklomatische Komplexität der folgenden Methode:

```

1  public int f (int x; boolean b2){
2      int res = 0;
3      int i;
4      if (x > 10) {
5          if b2 {
6              i = x;
7              b2 = false; }
8          else {
9              i = 0;
10             if b2 {
11                 b2 = false; }
12             else {
13                 b2 = true; }; }; };
14     if ((i < x) || b2) {
15         res = res * x; }
16     return sum;
17 }
```

- a) Bestimmen Sie den Kontrollflussgraphen G zu dem angegebenen Code. (Beachten Sie dabei, dass Kopf der Methode und die letzte schließende Klammer als Anweisungen zählen)

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

- b) Berechnen Sie die Zyklomatische Zahl $cn(G)$ des Kontrollflussgraphen G sowie die McCabe-Metrik $v(G)$ des Codes

	Formel	konkreter Wert
$cn(G)$		
$v(G)$		

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

3 Dynamischer Test: Black-Box und White-Box

3.1 Äquivalenzklassen und Grenzwerte

20 Punkte

Betrachten Sie die Methode `public int H (int x, int y)` mit den Vorbedingungen

V1 x soll echt größer 1 sein

V2 y soll echt kleiner als 10 sein

V3 y soll größer oder gleich 0 sein

Die Funktion H soll den Wert 1 liefern falls $x \leq y$ und sonst 0.

- Bestimmen Sie für diese Funktion die Äquivalenzklassen (Angabe als logische Testfälle)
- für jede Äquivalenzklasse einen Repräsentanten und den zugehörigen Sollwert

Tragen Sie die Ergebnisse in die folgende Tabelle ein:

Äquivalenzklasse					
Repräsentant					
Sollwert					
Äquivalenzklasse					
Repräsentant					
Sollwert					
Äquivalenzklasse					
Repräsentant					
Sollwert					

Anmerkung: es sind u.U. mehr Zellen vorhanden als Klassen benötigt werden.

- Geben Sie 5 weitere Testfälle an (Eingabedaten und Sollwerte), die sich bei einer Grenzwertanalyse zusätzlich ergeben würden.

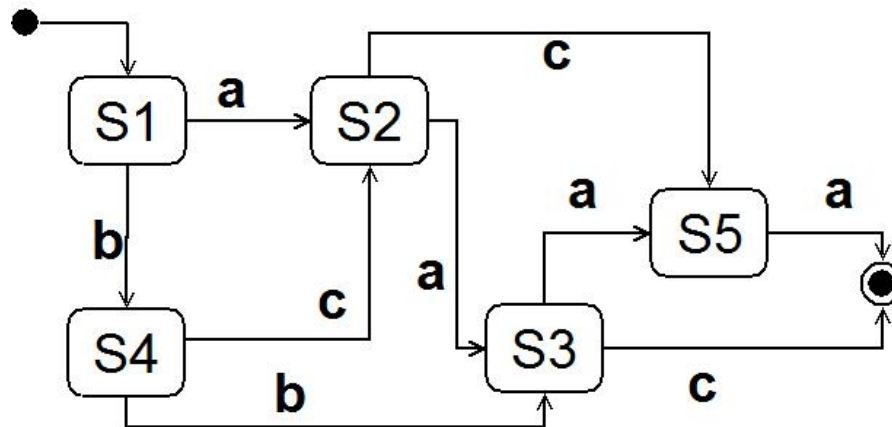
Eingabedaten					
Sollwert					

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

3.2 Zustandsbasierte Testfallerzeugung

20 Punkte

Gegeben sei der folgende Zustandsautomat :



- a) Bestimmen Sie zunächst den Übergangsbaum für den Zustands-Konformanztest (Verwenden Sie den Wurzelknoten S1 und nennen Sie den Endknoten Final)

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

- b) Geben Sie eine möglichst kleine Anzahl von Testfällen an (als Folge von Ereignissen und Zuständen), die für eine 100 %ige Zustandsüberdeckung genügen.

Anmerkung: die Tabelle enthält u.U. mehr Zeilen als notwendig.

TFZ 1	
TFZ 2	
TFZ 3	
TFZ 4	

- c) Geben Sie weitere Testfälle an, die darüberhinaus für eine 100 %ige Zustandsübergangsüberdeckung (Transitionsüberdeckung) beim Konformanztest benötigt werden.

TFT 1	
TFT 2	
TFT 3	
TFT 4	

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

4 Dynamischer Test: White-Box

4.1 Code-Überdeckung

25 Punkte

Gegeben sei der folgende abstrakte Code einer (nicht notwendigerweise sinnvollen) Funktion. Dabei sollen die folgenden Attribute im Rahme der Klasse oder Konstruktoren vordefiniert sein:

```

1  ...
2      public static int grenze = 10;
3      public static bool b1 = false;
4  ...

```

Der Code der Methode sei dann wie folgt gegeben:

```

1  public int g (int x; boolean b2){
2      int res = 10;
3      int i = 0;
4      if (x < grenze) {
5          i = grenze - x;
6          res = 7; };
7      if (res < 10) {
8          i = grenze + x;
9          res = 2 * res;
10         b1 = true; };
11     if ((i < grenze) and b1 and b2) {
12         res = res * res; }
13     else { res = 1; };
14     println('final result:' res)
15     return res;
16 }

```

Betrachten Sie die folgenden Testfälle für die Funktion g:

- TF 1: x == 7; b2 == true (Aufruf von g (7,true)))

a) Bestimmen Sie zunächst die Anweisungsüberdeckung

	Folge der durchlaufenen Zeilen (Zeilennr)
TF 1	
Anzahl Anweisungen gesamt in g	
Anweisungsüberdeckung in % durch TF 1	

Anmerkung: bei den Anweisungen zählen der Methodenkopf (Startknoten) und die schließende Klammer des Rumpfs (Endknoten) auch als je 1 Anweisung.

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

- b) Bestimmen Sie eine möglichst kleine Auswahl von zusätzlichen Testfällen, die eine 100% -ige Zweigüberdeckung sicherstellen

	x	b2	durchlaufene Folge von Zeilen
TFZ 1			
TFZ 2			
TFZ 3			
TFZ 4			
TFZ 5			

Anmerkung: unter Umständen werden auch weniger als die vorgesehenen Testfälle reichen.

- c) Ist Ihrer Meinung nach eine 100 %ige Pfadüberdeckung (gegenüber den Pfaden im Kontrollflussgraphen) für diesen Code erreichbar? Begründen Sie Ihre Aussage.

100 % möglich? (Ja / Nein):
Begründung:

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

4.2 Bedingungsüberdeckung

20 Punkte

- a) Seien A, B, C atomare boolesche Ausdrücke. Bestimmen Sie möglichst wenig generische Testfälle, die für den Ausdruck $(\neg A \wedge B \wedge C)$ eine einfache Bedingungsüberdeckung (EBÜ) und eine minimale Mehrfachbedingungsüberdeckung (MMBÜ) ergeben. Markieren Sie in den letzten Spalten die ausgewählten Testfälle.

A	B	C	$(\neg A \wedge B \wedge C)$	ausgewählt für EBÜ?	ausgewählt für MMBÜ?
0	0	0	0		
0	0	1	0		
0	1	0	0		
0	1	1	1		
1	0	0	0		
1	0	1	0		
1	1	0	0		
1	1	1	0		

- b) Seien A, B, C wie folgt definiert:

$A = x < 10$, $B = y + x < 10$, $C = (z + y == x)$, wobei x, y, z Integervariablen sein sollen.

Geben Sie konkrete Testfälle an, die Ihrer MMBÜ entsprechen.

A	B	C	$(\neg A \wedge B \wedge C)$	x	y	z
0	0	0	0			
0	0	1	0			
0	1	0	0			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	0			

Viel Erfolg!

WP-CT SoSe 10	Klausur	Prof. Dr. B. Buth 16.07.2010
Name		Matrikelnummer

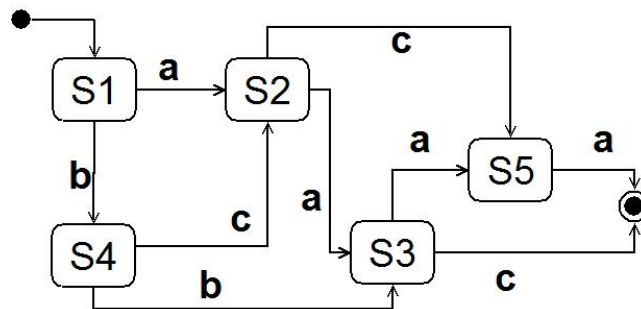
Aufgabe 3.2 Metriken- Wiederholung des Codes

```

1  public int f (int x; boolean b2){
2      int res = 0;
3      int i;
4      if (x > 10) {
5          if b2 {
6              i = x;
7              b2 = false; }
8          else {
9              i = 0;
10             if b2 {
11                 b2 = false; }
12             else {
13                 b2 = true; }; }; };
14      if ((i < x) || b2) {
15          res = res * x; }
16      return sum;
17  }

```

Aufgabe 4.2 Zustandsbasierter Test - Wiederholung Automat



Aufgabe 5.1 Code-Überdeckung - Wiederholung des Codes

```

1  ...
2      public static int grenze = 10;
3      public static bool b1 = false;
4      ...

```

Der Code der Methode sei dann wie folgt gegeben:

```

1  public int g (int x; boolean b2){
2      int res = 10;
3      int i = 0;
4      if (x < grenze) {
5          i = grenze - x;
6          res = 7; };
7      if (res < 10) {
8          i = grenze + x;
9          res = 2 * res;
10         b1 = true; };
11      if ((i < grenze) and b1 and b2) {
12          res = res * res; }
13      else { res = 1; };
14      println("final result:" res)
15      return res;
16  }

```