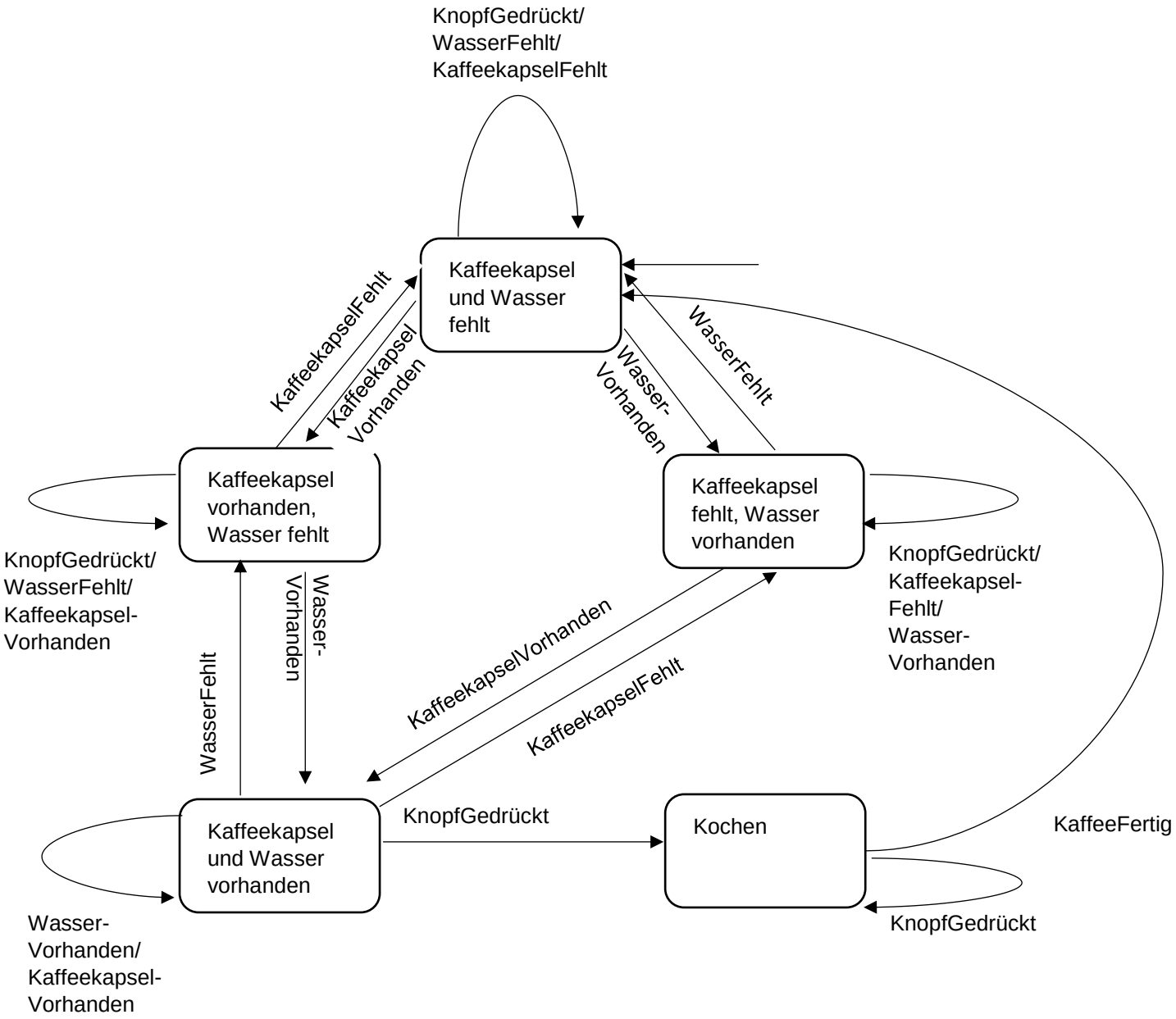
2965826, 3220544, 3245875, 2965839
Jonas Allali, Timo Hüttner, Heinrich Pauli, Jena Satkunarajan

Software Engineering
Übungsblatt 3

31. Mai 2018

1
a)


KnopfGedrückt/
WasserFehlt/
KaffeekapselFehlt
Kaffeekapsel
und Wasser
fehlt
KaffeekapselFehlt
Kaffeekapsel
Vorhanden
Wasser-
Vorhanden
WasserFehlt
Kaffeekapsel
vorhanden,
Wasser fehlt
Kaffeekapsel
fehlt, Wasser
vorhanden
KnopfGedrückt/
WasserFehlt/
Kaffeekapsel-
Vorhanden
KnopfGedrückt/
Kaffeekapsel-
Fehlt/
Wasser-
Vorhanden
WasserFehlt
Wasser-
Vorhanden
KaffeekapselVorhanden
KaffeekapselFehlt
Kaffeekapsel
und Wasser
vorhanden
KnopfGedrückt
Kochen
KaffeeFertig
Wasser-
Vorhanden/
Kaffeekapsel-
Vorhanden
KnopfGedrückt

b)

	Kaffeekapsel und Wasser fehlt	Kaffeekapsel vorhanden, Wasser fehlt	Kaffeekapsel fehlt, Wasser vorhanden	Kaffeekapsel und Wasser vorhanden	Kochen
Kaffeekapsel-Fehlt	*Kaffeekapsel und Wasser fehlt*	*Kaffeekapsel und Wasser fehlt*	*Kaffeekapsel fehlt, Wasser vorhanden*	*Kaffeekapsel fehlt, Wasser vorhanden*	
Kaffeekapsel-Vorhanden	*Kaffeekapsel vorhanden, Wasser fehlt*	*Kaffeekapsel vorhanden, Wasser fehlt*	*Kaffeekapsel und Wasser vorhanden*	*Kaffeekapsel und Wasser vorhanden*	
WasserFehlt	*Kaffeekapsel und Wasser fehlt*	*Kaffeekapsel vorhanden, Wasser fehlt*	*Kaffeekapsel und Wasser fehlt*	*Kaffeekapsel vorhanden, Wasser fehlt*	
Wasser-Vorhanden	*Kaffeekapsel fehlt, Wasser vorhanden*	*Kaffeekapsel und Wasser vorhanden*	*Kaffeekapsel fehlt, Wasser vorhanden*	*Kaffeekapsel und Wasser vorhanden*	
KnopfGedrückt	*Kaffeekapsel und Wasser fehlt*	*Kaffeekapsel vorhanden, Wasser fehlt*	*Kaffeekapsel fehlt, Wasser vorhanden*	*Kochen*	
KaffeeFertig					*Kaffeekapsel und Wasser fehlt*

c)

Zustand	Ereignis	Nächster Zustand
Kaffeekapsel und Wasser fehlt	WasserFehlt	Kaffeekapsel und Wasser fehlt
Kaffeekapsel und Wasser fehlt	WasserVorhanden	Kaffeekapsel vorhanden, Wasser fehlt
Kaffeekapsel und Wasser fehlt	KaffeekapselVorhanden	Kaffeekapsel fehlt, Wasser vorhanden
Kaffeekapsel vorhanden, Wasser fehlt	WasserVorhanden	Kaffeekapsel und Wasser vorhanden
Kaffeekapsel und Wasser vorhanden	KnopfDrücken	Kochen

d)

Zustand	Ereignis	Nächster Zustand
Kaffeekapsel und Wasser fehlt	WasserFehlt	Kaffeekapsel und Wasser fehlt
Kaffeekapsel und Wasser fehlt	KaffeekapselFehlt	Kaffeekapsel und Wasser fehlt
Kaffeekapsel und Wasser fehlt	KnopfGedrückt	Kaffeekapsel und Wasser fehlt
Kaffeekapsel und Wasser fehlt	KaffeekapselVorhanden	Kaffeekapsel vorhanden, Wasser fehlt
Kaffeekapsel und Wasser fehlt	WasserVorhanden	Kaffeekapsel fehlt, Wasser vorhanden
Kaffeekapsel vorhanden, Wasser fehlt	KaffeekapselFehlt	Kaffeekapsel und Wasser fehlt
Kaffeekapsel vorhanden, Wasser fehlt	KnopfGedrückt	Kaffeekapsel vorhanden, Wasser fehlt
Kaffeekapsel vorhanden, Wasser fehlt	WasserFehlt	Kaffeekapsel vorhanden, Wasser fehlt
Kaffeekapsel vorhanden, Wasser fehlt	KaffeekapselVorhanden	Kaffeekapsel vorhanden, Wasser fehlt
Kaffeekapsel vorhanden, Wasser fehlt	WasserVorhanden	Kaffeekapsel und Wasser vorhanden
Kaffeekapsel fehlt, Wasser vorhanden	WasserVorhanden	Kaffeekapsel fehlt, Wasser vorhanden
Kaffeekapsel fehlt, Wasser vorhanden	KaffeekapselFehlt	Kaffeekapsel fehlt, Wasser vorhanden
Kaffeekapsel fehlt, Wasser vorhanden	KnopfGedrückt	Kaffeekapsel fehlt, Wasser vorhanden
Kaffeekapsel fehlt, Wasser vorhanden	WasserFehlt	Kaffeekapsel und Wasser fehlt
Kaffeekapsel fehlt, Wasser vorhanden	KaffeekapselVorhanden	Kaffeekapsel und Wasser vorhanden
Kaffeekapsel und Wasser vorhanden	WasserVorhanden	Kaffeekapsel und Wasser vorhanden
Kaffeekapsel und Wasser vorhanden	KaffeekapselVorhanden	Kaffeekapsel und Wasser vorhanden
Kaffeekapsel und Wasser vorhanden	WasserFehlt	Kaffeekapsel vorhanden, Wasser fehlt
Kaffeekapsel und Wasser vorhanden	KaffeekapselFehlt	Kaffeekapsel fehlt, Wasser vorhanden
Kaffeekapsel und Wasser vorhanden	KnopfGedrückt	Kochen
Kochen	KnopfGedrückt	Kochen
Kochen	KaffeeFertig	Kaffeekapsel und Wasser fehlt

e)

Um vollständige Ereignisüberdeckung zu erhalten, muss jeder Zustand alle Ereignisse, die ihm zur Verfügung stehen einmal durchgehen. Dies ist in d) zu sehen. da d) vollständige Ereignisüberdeckung hat, folgt daraus automatisch vollständige Zustandsüberdeckung und vollständige Zustandsübergangs-überdeckung. Also hätte man die Tabelle von d) auch für c) verwenden können. Es sollte jedoch beachtet werden, dass Fehlerbehandlungen nicht getestet werden konnten, da dazu keine näheren Informationen vorliegen.

Aufgabe 2

a)

	A	B	C	D	B && C	B && D	A || (B&&C) || (B&&D)
1	f	f	f	f	f	f	f
2	f	f	f	w	f	f	f
3	f	f	w	f	f	f	f
4	f	f	w	w	f	f	f
5	f	w	f	f	f	f	f
6	f	w	f	w	f	w	w
7	f	w	w	f	w	f	w
8	f	w	w	w	w	w	w
9	w	f	f	f	f	f	w
10	w	f	f	w	f	f	w
11	w	f	w	f	f	f	w
12	w	f	w	w	f	f	w
13	w	w	f	f	f	f	w
14	w	w	f	w	f	w	w
15	w	w	w	f	w	f	w
16	w	w	w	w	w	w	w

b)

Ein Tabelleneintrag repräsentiert ein Zeilenpaar der Wahrheitstabelle aus der a).
Der Eintrag ist die Variable, deren Änderung sich auf den Gesamtausdruck fortsetzt.

Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-
2	-	-	-	-	-	B	-	-	-	A	-	-	-	-	-	-
3	-	-	-	-	-	-	B	-	-	-	A	-	-	-	-	-
4	-	-	-	-	-	-	-	B	-	-	-	A	-	-	-	-
5	-	-	-	-	-	D	C	-	-	-	-	-	A	-	-	-

c)

	A	B	C	D	B && C	B && D	A || (B&&C) || (B&&D)
1	f	f	f	f	f	f	f
2	f	f	f	w	f	f	f
3	f	f	w	f	f	f	f
4	f	f	w	w	f	f	f
5	f	w	f	f	f	f	f
6	f	w	f	w	f	w	w
7	f	w	w	-	w	-	w
8	f	w	w	-	w	-	w
9	w	-	-	-	-	-	w
10	w	-	-	-	-	-	w
11	w	-	-	-	-	-	w
12	w	-	-	-	-	-	w
13	w	-	-	-	-	-	w
14	w	-	-	-	-	-	w
15	w	-	-	-	-	-	w
16	w	-	-	-	-	-	w

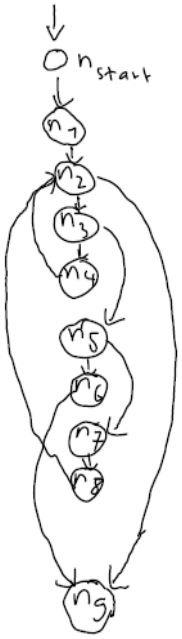
d)

Ein Tabelleneintrag repräsentiert ein Zeilenpaar der Wahrheitstabelle aus der c).
Der Eintrag ist die Variable, deren Änderung sich auf den Gesamtausdruck fortsetzt.

Row	1	2	3	4	5	6	7 und 8	9 bis 16
1	-	-	-	-	-	-	-	A
2	-	-	-	-	-	B	-	A
3	-	-	-	-	-	-	B	A
4	-	-	-	-	-	-	B	A
5	-	-	-	-	-	D	C	A

Aufgabe 3

a)




```
int fct(List int)
{
int diff=0;
for(Integer number: list) {
   if (number < 0){
      diff--; }
      else if(number ==0){
            break;}
      else {
          diff++;}
}
return diff;
```

b)

Schleifendurchlauf k	Eingabewert (List)	Rückgabewert	Pfad
k = 0	{/}	0	nStart, n1, n2, n9
k = 1	{-1}	-1	nStart, n1, n2, n3, n4, n2, n9
k = 1	{0}	0	nStart, n1, n2, n3, n5, n6, n9
k = 1	{1}	1	nStart, n1, n2, n3, n5, n7, n8, n2, n9
k = 2	{-1, -1}	-2	nStart, n1, n2, n3, n4, n2, n3, n4, n2, n9
k = 2	{-1, 0}	-1	nStart, n1, n2, n3, n4, n2, n3, n5, n6, n9
k = 2	{-1, 1}	0	nStart, n1, n2, n3, n4, n2, n3, n5, n7, n8, n2, n9
k = 2	{1, -1}	0	nStart, n1, n2, n3, n5, n7, n8, n2, n3, n4, n2, n9
k = 2	{1 , 0}	1	nStart, n1, n2, n3, n5, n7, n8, n2, n3, n5, n6, n9
k = 2	{1, 1}	2	nStart, n1, n2, n3, n5, n7, n8, n2, n3, n5, n7, n8, n2, n9

c)

```
public int fct(List <Integer > list) {
 int diff = 0;

 if (list == null) {
      list = new List<Integer>();
 }

 if (list.isEmpty == true) {
      list.add(0);
 }


 for (Integer number : list) {
 if (number < 0) {
 diff --;
 } else if (number == 0) {
 break;
 } else {
 diff ++;
 }
 }
 return diff;
 }
```

d)

Die Schleife wird in diesem Fall mindestens 1x ausgeführt. Der Fall des BIK, dass die Schleife **kein** Mal durchgeführt wird, kann hier nicht umgesetzt werden.

Lösungen

1. Baue aus der do-while-Schleife eine while-Schleife (ist möglich)
2. Kopiere den Teil der do-while-Schleife, der mindestens 1x ausgeführt werden muss, vor die Schleife. Füge die do-while-Schleife nun in ein if-else ein.