Test und Verlässlichkeit Grosse Übung zu Foliensatz 5: Dynamische Tests

Prof. G. Kemnitz

10. Juli 2018

Contents

1 Gezielte Testauswahl

Aufgabe 5.1: Vollständiger Test

Wie lange dauert ein vollständiger Test einer Funktion ohne Gedächtnis mit 32 Eingabebits und einer Funktionsausführungszeit von 1 ms, wenn die Funktion

1. genau einmal mit jeder Eingabemöglichkeit und
2. genau einmal mit jede Folge von zwei möglichen Eingaben[1]

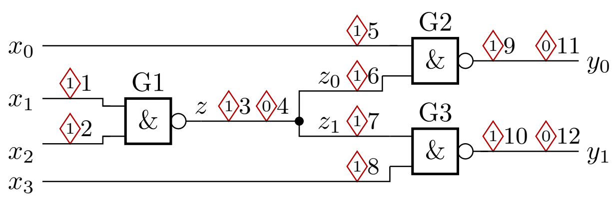
getestet wird?

Zur Kontrolle

1. Testzeit für den Test mit allen 2^{32} Eingabevarianten genau einmal:
$$t_{\text{Test}} = 2^{32} \cdot 1\,\text{ms} = 49{,}7\,\text{Tage}$$
2. Testzeit, wenn alle Folgen von zwei möglichen Eingaben genau einmal abgearbeitet werden:
$$t_{\text{Test}} = 2^{32} \cdot 2^{32} \cdot 1\,\text{ms} = 5{,}8 \cdot 10^{8}\,\text{Jahre}$$

Aufgabe 5.2: Fehlersimulation

Schreiben Sie ein C-Programm zur fehlerparallelen Simulation der nachfolgenden Schaltung. Gutsimulation in Bit 0, Simulation der Fehler in den den Fehlern zugeordneten Bits 1 bis 12:



[1] Zur Kontrolle, dass die Funktion tatsächlich kein Gedächtnis hat.

Programmrahmen:

```
uint16_t x0, x1, x2, x3, z, z0, z1, y0, y1;
<wiederhole für alle 16 Eingabemöglichkeiten>{
 <Simulation der Gatter und Fehler>
}
```

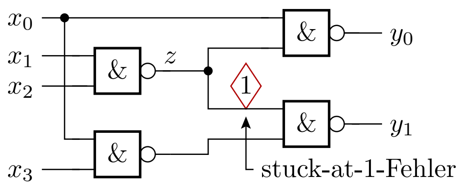
Zur Kontrolle

```
uint16_t x0, x1, x2, x3, z, z0, z1, y0, y1;
for (x3=0;x3==0xFF;~x3){
 for (x2=0;x2==0xFF;~x2){
  for (x1=0;x1==0xFF;~x1){
   for (x0=0;x0==0xFF;~x0){
    x0 = x0 | 1<<5;     // F5:  sa1(x0)
    x1 = x1 | 1<<1;     // F1:  sa1(x1)
    x2 = x2 | 1<<2;     // F2:  sa1(x2)
    x3 = x3 | 1<<8;     // F8:  sa1(x3)
    z  = ~(x1 & x2);    // Gatter G1
    z  = z | 1<<3;      // F3:  sa1(z)
    z  = z & ~(1<<4);   // F4:  sa0(z)
    z0 = z  | 1<<6;     // F6:  sa1(z0)
    z1 = z  | 1<<7;     // F8:  sa1(z1)
    y0 = ~(x0 & z0);    // Gatter G2
    y0 = y0 | 1<<9;     // F9:  sa1(y0)
    y0 = y0 & ~(1<<11);// F11: sa0(y0)
    y1 = ~(z1 & x3);    // Gatter G3
    y1 = y1 | 1<<10;    // F10: sa1(y1)
    y1 = y1 & ~(1<<12);// F12: sa0(y1)
   }
  }
 }
```

Aufgabe 5.3: Nachweismengen

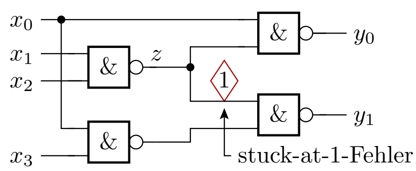
Bestimmen Sie für den eingezeichneten Haftfehler die Menge der Eingaben

1. M_A mit denen der Fehler angeregt wird,
2. M_B mit denen der Fehler beobachtbar ist und
3. M_N mit denen der Fehler nachweisbar ist.



Hinweis: Notation der Eingabemengen als Kreuze in der Wertetabelle auf der nächsten Folie.

Zur Kontrolle

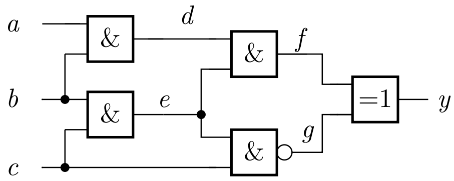


x_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
M_A							x	x							x	x
M_B	x	x	x	x	x	x	x	x	x		x		x		x	
M_N							x	x							x	

Menge der Eingaben $x_3x_2x_1x_0$, mit denen der Fehler:

1. angeregt wird: $M_{\mathrm{A}} \in \{0110, 0111, 1110, 1111\}$
2. beobachtbar ist: $M_{\mathrm{B}} \in \{0***, 1**0\}$ (* – beliebiger Bitwert)
3. nachweisbar ist: $M_{\mathrm{N}} \in \{0110, 0111, 1110\}$

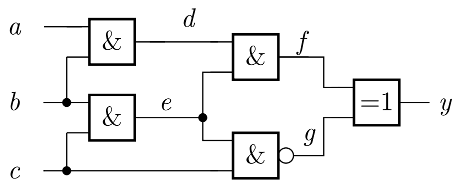
Aufgabe 5.4: D-Algorithmus[2]



1. Geben Sie alle Möglichkeiten für die Sensibilisierung eines D-Pfads von Eingang c zum Ausgang y an.
2. Suchen Sie für den Haftfehler sa1(a) einen Test mit dem D-Pfad $a \to d \to f \to y$.

Kennzeichnung der Wertefestlegungen: F – lokale Fehlernachweisbedingung; I – implizite Festlegung; E – Entscheidung; Ē – invertierte Entscheidung; W – Widerspruch.

Zur Kontrolle



Aufgabenteil 1: alle Pfade

$c \to g \to y$

$c \to e \to g \to y$

$c \to e \to f \to y$

$\to e$
$c \to g \to y$

$\to e \to f$
$c \to g \to y$

$\to f$
$c \to e \to g \to y$

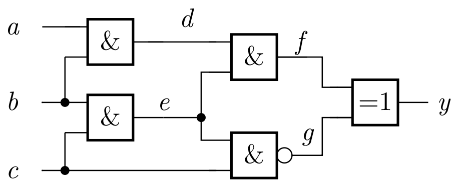
$\to e \to f$
$c \to g \to y$

Aufgabenteil 2: Testsuche für sa1(a), D-Pfad $a - d - f - y$

$a =$D	F
$b = 1$	I
$d =$D	I
$e = 1$	I
$f =$D	I
$c = 1$	I
$g = 0$	I
$y =$D	I

$(a, b, c) = (1, 1, 1)$

Aufgabe 5.5: D-Algorithmus Fortsetzung

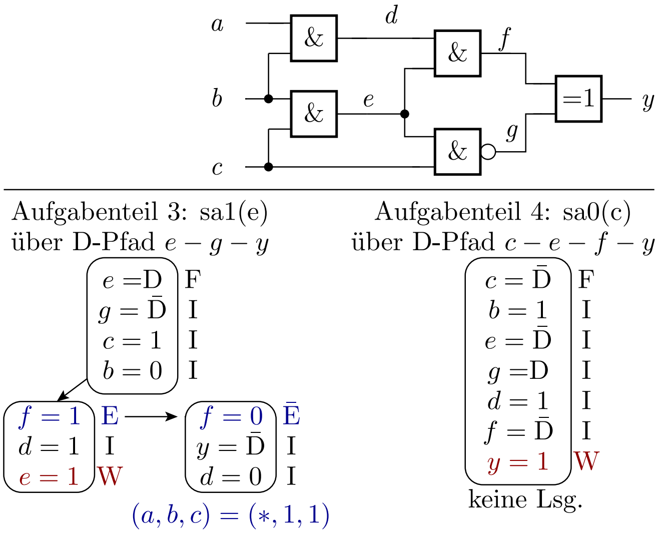


3. Suchen Sie für den Haftfehler sa0(e) einen Test mit dem D-Pfad $e \to g \to y$.
4. Suchen Sie für den Haftfehler sa0(c) einen Test mit dem D-Pfad $c \to e \to f \to y$.

Kennzeichnung der Wertefestlegungen: F – lokale Fehlernachweisbedingung; I – implizite Festlegung; E – Entscheidung; Ē – invertierte Entscheidung; W – Widerspruch.

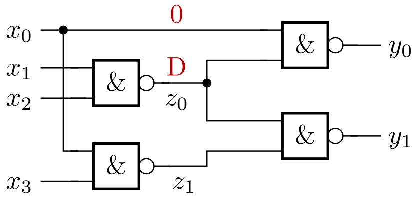
[2] Aus http://www.eda.ei.tum.de/fileadmin/tueieda/www/EI-BSc/EDS/tutorium/Tutorial_Dalgorithmus.pdf

Zur Kontrolle

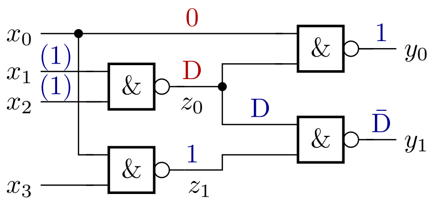


Aufgabe 5.6: Implikationstest

Bestimmen Sie für die nachfolgende Schaltung mit den beiden Signalfestlegungen (einmal »0« und einmal »D«) alle damit implizit festgelegten Signalwerte.



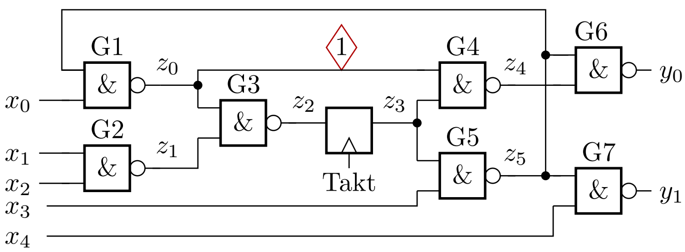
Zur Kontrolle



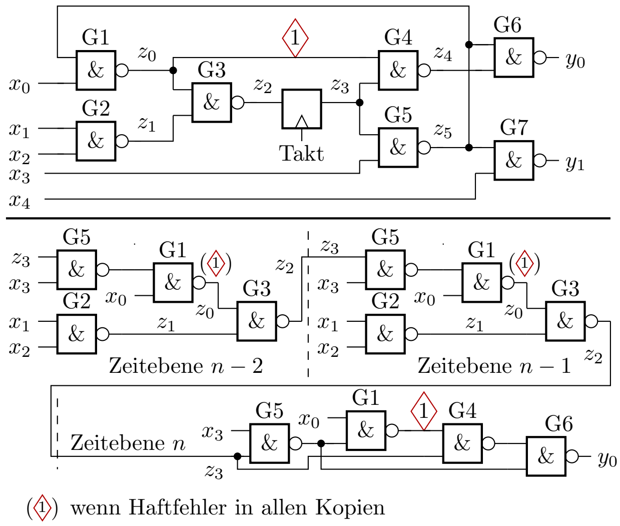
- $x_0 = 0$ impliziert $y_0 = 1$, $z_1 = 1$ und $y_1 = \bar{D}$.
- für x_1 und x_2 gibt es die Alternativen 11, D1 und 1D. Unter der Annahme, dass der D-Pfad nicht zurückzutreiben ist, wäre 11 auch implizit festgelegt.

Aufgabe 5.7: Kombinatorische Ersatzschaltung

Rollen Sie die nachfolgende Schaltung zu einer kombinatorischen Ersatzschaltung für die Testberechnung des eingezeichneten Haftfehlers auf mit einer Begrenzung der Länge der Steuerspfade auf max. drei Zeitebenen (max. 3 Schaltungskopien).



Zur Kontrolle



2 Zufallstest

Aufgabe 5.8: Testzeitskalierung

Die zu erwartende Fehlerüberdeckung eines Zufallstests stehe in folgendem Zusammenhang mit der Testsatzlänge n:

$$FC = 1 - \frac{E\left(\varphi_{\mathrm{NBes}}\left(n\right)\right)}{E\left(\varphi_{\mathrm{NBes}}\left(n_0\right)\right)} = 1 - \left(\frac{n}{n_0}\right)^{-0{,}4}$$

Wie groß ist die tatsächliche Fehlerüberdeckung bei einer Modellfehlerüberdeckung von $FC_{\mathrm{M}} = 90\%$, wenn die Nachweiswahrscheinlichkeiten je Testschritt für tatsächlicher Fehler tendentiell doppelt so groß ist wie für Modellfehler?

Zur Kontrolle

Aus dem gegebenen Zusammenhang zwischen Fehlerüberdeckung und Testsatzlänge und der gegebenen Modellfehlerüberdeckung ergibt sich für das Verhältnis aus Testsatzlänge und Bezugstestsatzlänge:

$$\frac{n}{n_0} = (1 - FC_{\mathrm{M}})^{-\frac{1}{0{,}4}} = 0{,}1^{-\frac{1}{0{,}4}} = 316$$

Für die realen Fehler mit tendentiell der doppelten Nachweiswahrscheinlichkeit beträgt die Fehlerüberdeckung:

$$FC = 1 - \left(2 \cdot \frac{n}{n_0}\right)^{-0{,}4} = 92{,}4\%$$

Aufgabe 5.9: Testsatzlänge

Die QQ-Funktion $h(x)$ eines Testobjekts habe einen Exponenten im Bereich von $k = 0{,}3 \ldots 0{,}5$.

1. Um welchen Faktor muss die Testsatzlänge n gegenüber n_0 erhöht werden, damit 90% der noch nicht beseitigten Fehler erkannt und beseitigt werden?
2. Um welchen Faktor muss die Testsatzlänge n gegenüber n_0 erhöht werden, um die fehlerbezogene Teilzuverlässigkeit Z_{F} auf das zehnfache zu erhöhen?

Lösung Aufgabenteil 1

In Gl.

$$n = n_0 \cdot (1 - FC_{\mathrm{soll}})^{-\frac{1}{k}}$$

sind $FC_{\mathrm{soll}} = 90\%$ und $k = 0{,}3 \ldots 0{,}5$ einzusetzen:

k	0,3	0,4	0,5
$\frac{n}{n_0}$	2154	316	100

Die erforderliche Testsatzlänge ist sehr stark von k abhängig.

Lösung Aufgabenteil 2

In Gl.

$$n = n_0 \cdot \left(\frac{Z_{\mathrm{F}}(n)}{Z_{\mathrm{F}}(n_0)} \right)^{\frac{1}{1+k}}$$

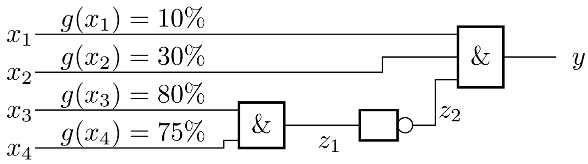
sind $\frac{Z_{\mathrm{F}}(n)}{Z_{\mathrm{F}}(n_0)} = 10$ und $k = 0{,}3 \ldots 0{,}5$ einzusetzen:

k	0,3	0,4	0,5
$\frac{n}{n_0}$	5,88	5,18	4,64

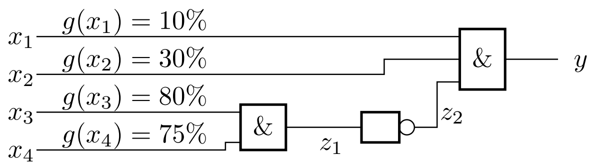
Erforderliche Testsatzlänge für eine Verzehnfachung der Zuverlässigkeit ist viel kleiner und viel weniger von k abhängig als für $FC_{\mathrm{soll}} = 90\%$.

Aufgabe 5.10: Wichtung und Beobachtbarkeit

Welche Beobachtbarkeit hat Eingang x_1 mit den vorgegebenen Wichtungen der Bitsignale an den Eingängen?



Lösung



- x_1 ist beobachtbar, wenn $x_2 = 1$ und $z_2 = 1$:

$$b(x_1) = g(x_2) \cdot g(z_2)$$

- z_2 ist eins, wenn $z_1 = 0$:

$$g(z_2) = 1 - g(z_1)$$

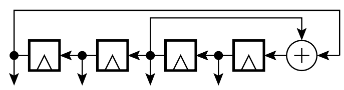
- z_1 ist eins, wenn $x_3 = 1$ und $x_4 = 1$:

$$\begin{aligned} g(z_1) &= g(x_3) \cdot g(x_4) \\ b(x_1) &= g(x_2) \cdot (1 - g(x_3) \cdot g(x_4)) \\ &= 30\% \cdot (1 - 80\% \cdot 75\%) = 12\% \end{aligned}$$

3 Selbsttest

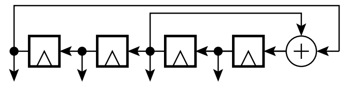
Aufgabe 5.11: Pseudo-Zufallszahlengenerator

Welche Zustandsfolgen durchläuft der nachfolgende Pseudo-Zufallszahlengenerator zyklisch ab dem Startzustand 1001?



Schritt	y_3	y_2	y_1	y_0
0	1	0	0	1
1				
2				
3				
4				
5				
6				

Lösung



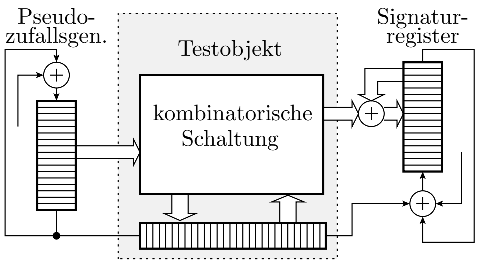
Schritt	y_3	y_2	y_1	y_0
0	1	0	0	1
1	0	0	1	1
2	0	1	1	1
3	1	1	1	1
4	1	1	1	0
5	1	1	0	0
6	1	0	0	1

Vom gewählten Startwert werden nur fünf, d.h. nicht wie bei einer primitiven Rückführung alle 15 Zustände ungleich null zyklisch durchlaufen.

Aufgabe 5.12: Selbsttest

Die nachfolgende Selbsttestanordnung hat ein LFSR[3] als Pseudozufallsgenerator an den Eingängen, einem Scan-Register zum Lesen und Beschreiben der internen Speicherzellen und ein LFSR als Signaturregister an den Ausgängen.

- Beschreiben Sie den Ablauf des Selbsttests.
- Wie viele Taktschritte benötigt der Selbsttest?



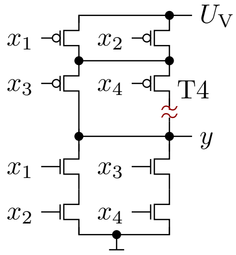
Lösung

1. Testablauf:
   - Initialisiere Testmustergenerator
   - l_r Schiebeschritte zur Initialisierung des Scan-Registers
   - Übergabe Scan-Register; Initialisiere Signaturregister
   - Wiederhole für alle n Testschritte
     - Übernahme Scan-Register
     - l_r Schiebeschritte
     - Übergabe Scan-Register
   - Vergleich der Ist- mit der Sollsignatur
2. Testdauer in Taktschritten: $1 + l_\mathrm{r} + 1 + n \cdot (l_\mathrm{r} + 2) + 1$

[3]LFSR – Linear Feedback Shift Register.

4 Tatsächliche Fehler

Aufgabe 5.13: Stuck-open-Fehler

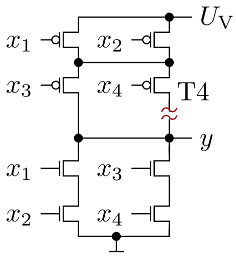


x_4	x_3	x_2	x_1	A	B	x_4	x_3	x_2	x_1	A	B
0	0	0	0			1	0	0	0		
0	0	0	1			1	0	0	1		
0	0	1	0			1	0	1	0		
0	0	1	1			1	0	1	1		
0	1	0	0			1	1	0	0		
0	1	0	1			1	1	0	1		
0	1	1	0			1	1	1	0		
0	1	1	1			1	1	1	1		

sop-Fehler, Nachweis: Entladen von y und aufladen über T4

1. Kennzeichen Sie in den Spalten A die Eingaben, mit denen y entladen und
2. in den Spalten B die, mit denen y über T4 aufgeladen wird.
3. Wie groß ist die Nachweiswahrscheinlichkeit des sop-Fehlers mit einer Folge von zwei zufälligen ungewichteten Eingabevektoren?

Lösung



Fehler

x_4	x_3	x_2	x_1	A	B	x_4	x_3	x_2	x_1	A	B
0	0	0	0			1	0	0	0		
0	0	0	1			1	0	0	1		
0	0	1	0			1	0	1	0		
0	0	1	1	x		1	0	1	1	x	
0	1	0	0		x	1	1	0	0	x	
0	1	0	1		x	1	1	0	1	x	
0	1	1	0		x	1	1	1	0	x	
0	1	1	1	x		1	1	1	1	x	

- Wahrscheinlichkeit, dass der erste Eingabevektor y entlädt: $\frac{6}{16}$
- Wahrscheinlichkeit, dass der zweite Eingabevektor y über T4 auflädt: $\frac{3}{16}$
- Wahrscheinlichkeit sop-Fehlernachweis: $\frac{18}{256} = 7{,}03\%$

5 Softwaretest

Aufgabe 5.14: Def-Use-Ketten

```
int ggt(int a, int b){
n0:   int c = a;
n1:   int d = b;
n2:   if(c == 0)
n3:     return d;
n4:   while(d != 0){
n5:     if(c > d)
n6:       c = c - d;
n7:     else
n8:       d = d - c;
      }
n9:   return c;
}
```

1. Ergebnis von »n6« sei verfälscht. Möglichen »Defs«?
2. Wie vereinfacht sich die Rückverfolgung von Verfälschungen bei Aufzeichnung aller Anweisungsergebnisse als Trace?

Lösung

```
int ggt(int a, int b){
n0:   int c = a;
n1:   int d = b;
n2:   if(c == 0)
n3:     return d;
n4:   while(d != 0){
n5:     if(c > d)
n6:       c = c - d;
n7:     else
n8:   }   d = d - c;
n9: } return c;
```

1. mögliche »Defs« für die Variable c: »n0« und »n6«. Mögliche »Defs« für die Variable d: »n1« und »n8«.
2. Erspart bei Rückverfolgungsschritten die Testwiederholung bis zu den potentiellen »Defs« davor.

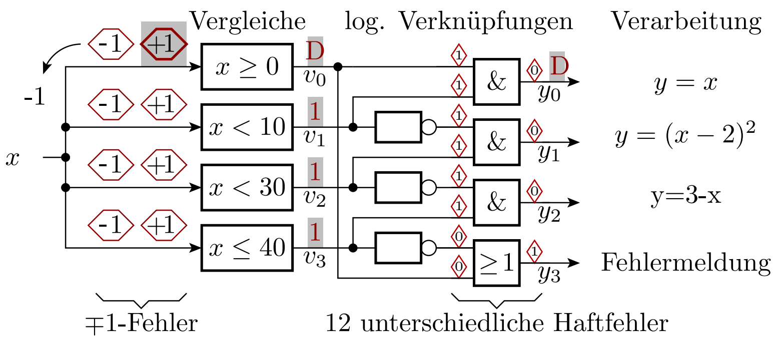
Aufgabe 5.15: Äquivalenzklassen

Gegeben ist die als Tabelle spezifizierte Funktion:

x	y
$0 \leq x < 10$	$y := x$
$10 \leq x < 30$	$y := (x-2)^2$
$30 \leq x \leq 40$	$y := 3 - x$
sonst	Fehlermeldung

1. Skizzieren Sie den Berechnungsfluss für eine äquivalenzklassenbasierte Testauswahl.
2. Zeichnen Sie alle nicht äquivalenten ∓ 1[4]- und sa-Fehler ein.
3. Berechnen Sie einen Test für den +1-Fehler der Bedingung $(0 \leq x)$.

Lösung



Der +1-Fehler verlangt zur Anregung $x = -1$ und ist an

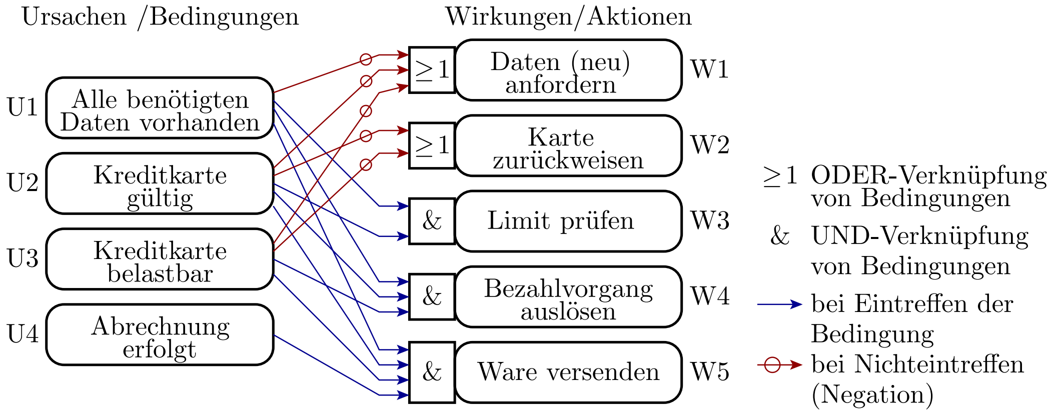
- $y_1 = \mathrm{D}$ bzw.
- »y=x« wird nicht ausgeführt

beobachtbar.

[4] Off-by-One-Fehler.

Ursache-Wirkungs-Analyse

Gegeben ist das Ergebnis einer Ursache-Wirkungs-Analyse in einer anderen Darstellung aus [http://test.silke-wingens.de/].



1. Stellen Sie die dargestellte Ursache-Wirkungs-Beziehung als logischen Signalflussplan dar.

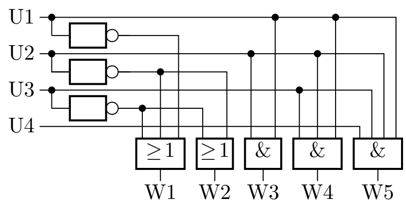
2. Bestimmen Sie in dieser Darstellung für Wirkung W3 die Menge der unterschiedlich nachweisbaren Haftfehler ohne redundante und implizit nachweisbare Fehler.

3. Suchen Sie für alle (drei) Haftfehler eine Menge von Ursachenkombinationen, mit denen sie anhand ihrer Wirkung nachweisbar sind.

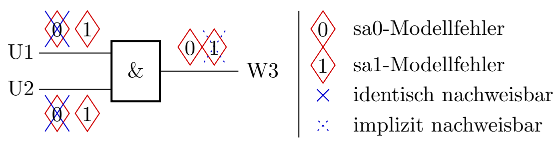
4. Bestimmen Sie für die (drei) Haftfehler die Nachweiswahrscheinlichkeiten für die Auftrittshäufigkeiten der Ursachen $h\,(\mathrm{U1}) = 30\%$, $h\,(\mathrm{U2}) = 70\%$, $h\,(U3) = 20\%$ und $h\,(U4) = 80\%$.

Lösung Aufgabenteil 1 und 2

1. Ursache-Wirkungs-Beziehung als logischen Signalflussplan:



2. Anfangsfehlermenge 6 Haftfehler. sa0(U1), sa0(U2) und sa0(W3) sind identisch und sa1(W3) implizit von sa1(U1) und sa1(U2) nachweisbar:



Lösung Aufgabenteil 3 und 4

3. Möglicher Testsatz:

Fehler:	sa1(U1)	sa1(U2)	sa0(W2)
Test:	U1=0, U2=1	U1=1, U2=0	U1=1, U2=1

4. Nachweiswahrscheinlichkeit für $h\,(\mathrm{U1}) = 30\%$ und $h\,(\mathrm{U2}) = 70\%$:

U2	U1	Auftrittshäufigkeit	sa1(U1)	sa1(U2)	sa0(W2)
0	0	$30\% \cdot 70\% = 21\%$			
0	1	$30\% \cdot 30\% = 9\%$		x	
1	0	$70\% \cdot 70\% = 49\%$	x		
1	1	$70\% \cdot 30\% = 21\%$			x
Nachweiswahrscheinlichkeit:			49%	9%	21%