

Strukturorientierte Verfahren

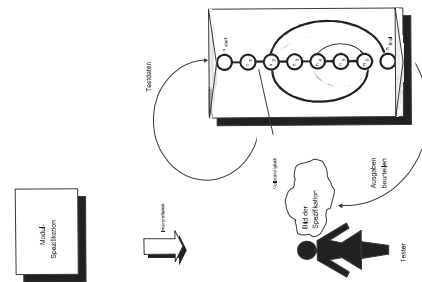


- Arbeitsweise: Beurteilung der Adäquanz und der Vollständigkeit des Tests und evtl. Herleitung der Testdaten anhand der Modulstruktur. Beurteilung der Ausgaben anhand der Modulspezifikation.
 - Vorteil: Implementationsstruktur wird beachtet (Anweisungen, Zweige, Datenzugriffe, etc.)
 - Nachteil: Nicht realisierte, aber spezifizierte Funktionen können nicht erkannt werden
- Kontrollflussorientierte Verfahren
- Datenflussorientierte Verfahren

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 1

Strukturorientierte Verfahren Arbeitsweise



Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 2

Strukturorientierte, kontrollflussorientierte Testverfahren



- Anweisungsüberdeckungstest
- Zweigüberdeckungstest
- Bedingungsüberdeckungstest
 - einfacher
 - minimaler mehrfacher
 - mehrfacher
- LCSAJ-basierter Test
- boundary interior-Pfadtest
- Strukturierter Pfadtest
- Pfadtest

Die **kontrollflussorientierten Testverfahren** basieren auf der Kontrollstruktur bzw. dem Kontrollfluss. Die Basis bildet der Kontrollflussgraph.

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 3

Strukturorientierte, kontrollfluss orientierte Testverfahren



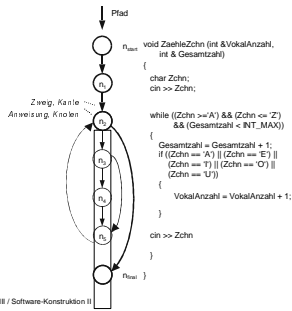
Beispiel:

```
void ZaehleZchn(int &VokalAnzahl, int &Gesamtzahl)
// Vorbedingung: VokalAnzahl <= Gesamtzahl
{
    char Zchn;
    cin >> Zchn;
    while ((Zchn >= 'A') && (Zchn <= 'Z') &&
           (Gesamtzahl < INT_MAX))
    {
        Gesamtzahl = Gesamtzahl + 1;
        if ((Zchn == 'A') || (Zchn == 'E') || (Zchn == 'I') ||
            (Zchn == 'O') || (Zchn == 'U'))
        {
            VokalAnzahl = VokalAnzahl + 1;
        }
        cin >> Zchn;
    } //end while
}
```

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 4

Strukturorientierte, kontrollflussorientierte Testverfahren Kontrollflussdiagramm zur Operation ZaehleZchn



```

n_start void ZaehleZchn (int &VokalAnzahl,
int &Gesamtzahl)
{
char Zchn;
cin >> Zchn;

while ((Zchn >='A') && (Zchn <='Z')
&& (Gesamtzahl < INT_MAX))
{
Gesamtzahl = Gesamtzahl + 1;
if ((Zchn == 'A') || (Zchn == 'E') ||
(Zchn == 'I') || (Zchn == 'O') ||
(Zchn == 'U'))
{
VokalAnzahl = VokalAnzahl + 1;
}
}
cin >> Zchn;
}
n_final
    
```

Strukturorientierte, kontrollfluss orientierte Testverfahren Anweisungsüberdeckungstest



- Der Anweisungsüberdeckungstest ist die einfachste kontrollflussorientierte Testtechnik. Man bezeichnet ihn abkürzend auch als C_0 -Test. Im Englischen heißt er *statement coverage test*.
- Das Ziel der Anweisungsüberdeckung ist die mindestens einmalige Ausführung aller Anweisungen des zu testenden Programms, also die Ausführung aller Knoten (Kreise) des Kontrollflussgraphen.
- Als Testmaß wird der erreichte Anweisungsüberdeckungsgrad definiert. Er ist das Verhältnis der ausgeführten Anweisungen zu der Gesamtzahl der im Prüfling vorhandenen Anweisungen.

$$C_{\text{Anweisung}} = \frac{\text{Anzahl der ausgeführten Anweisungen}}{\text{Anzahl der Anweisungen}}$$

- Sind alle Anweisungen des zu testenden Moduls durch die eingegebenen Testdaten mindestens einmal ausgeführt worden, so ist eine vollständige Anweisungsüberdeckung erreicht.

Strukturorientierte, kontrollflussorientierte Testverfahren Anweisungsüberdeckungstest



- Anweisungsüberdeckungstest verlangt die Ausführung aller Knoten des Kontrollflussgraphen. Von den Testfällen wird verlangt, dass die entsprechenden Programmpfade alle Knoten des Kontrollflussgraphen enthalten.
- Testfall:
Aufruf von ZaehleZchn mit: Gesamtzahl = 0
Eingelesene Zeichen: 'A', '1'
Durchlaufener Pfad: (nstart, n1, n2, n3, n4, n5, n2, nfinal)
- Der Testpfad enthält alle Knoten. Er enthält aber nicht alle Kanten des Kontrollflussgraphen. Die Kante (n3,n5) ist nicht enthalten.

Strukturorientierte, kontrollfluss orientierte Testverfahren Anweisungsüberdeckungstest



- Der Anweisungsüberdeckungstest gilt als zu schwaches Kriterium für eine sinnvolle Testdurchführung. Er besitzt eine untergeordnete praktische Bedeutung.
- Der Standard RTCA DO-178B für Softwareanwendungen in der Luftfahrt fordert den Anweisungsüberdeckungstest für Software ab Stufe C. Derartige Software kann im Falle eines Fehlverhaltens einen bedeutenden Ausfall (*major failure condition*) verursachen.

Strukturorientierte, kontrollflussorientierte Testverfahren Zweigüberdeckungstest



- Das Ziel des Zweigüberdeckungstests ist die Ausführung aller Zweige des zu testenden Programms. Das verlangt den Durchlauf durch alle Kanten des Kontrollflussgraphen. Man bezeichnet ihn auch abkürzend als C_1 -Test. Im Englischen sagt man *branch coverage test*.
- Der Zweigüberdeckungstest ist eine strengere Testtechnik als der Anweisungsüberdeckungstest. Der Anweisungsüberdeckungstest ist im Zweigüberdeckungstest vollständig enthalten. Man sagt auch: Der Zweigüberdeckungstest **subsumiert** den Anweisungsüberdeckungstest.
- Der Zweigüberdeckungstest gilt allgemein als das Minimalkriterium im Bereich des kontrollflussorientierten Testens.
- Der Standard RTCA DO-178B für Softwareanwendungen im Bereich der Luftfahrt schreibt einen Zweigüberdeckungstest für Software ab Stufe B vor.

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 9

Strukturorientierte, kontrollflussorientierte Testverfahren Zweigüberdeckungstest



- **Beispiel**
Der Zweigüberdeckungstest fordert die Überdeckung aller Zweige eines Kontrollflussgraphen. Dies wird erreicht, falls jede Entscheidung des zu testenden Moduls mindestens einmal den Wahrheitswert falsch und wahr besessen hat.
- **Testfall:**
Aufruf von ZaehleZchn mit: Gesamtzahl = 0
Eingelesene Zeichen: „A“, „B“, „1“
Durchlaufener Pfad: $(n_{start}, n_1, n_2, n_3, n_4, n_5, n_2, n_3, n_5, n_2, n_{end})$
- Der Testpfad enthält alle Kanten. Er enthält insbesondere die Kante (n_3, n_5) , die durch den Anweisungsüberdeckungstest nicht notwendig ausgeführt wird. Der Zweigüberdeckungstest subsumiert den Anweisungsüberdeckungstest.

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 10

Strukturorientierte, kontrollflussorientierte Testverfahren Zweigüberdeckungstest



Frage: Ist der Zweigüberdeckungstest adäquat für den Test von komplizierten Entscheidungen?

- **Beispiele:**
 - Einfache Entscheidung: $\text{if } (x > 5) \dots$
 - Die Entscheidung $(x > 5)$ kann als ausreichend getestet betrachtet werden, falls beide Wahrheitswerte beim Test aufgetreten sind. Die Entscheidung unterteilt die möglichen Testdaten in zwei Klassen und verlangt, dass mindestens ein Testdatum aus jeder Klasse gewählt wird.
 - Komplexe Entscheidung: $\text{if } (((u == 0) \parallel (x > 5)) \&\& ((y < 6) \parallel (z == 0))) \dots$
 - Ein Test der Entscheidung $((u == 0) \parallel (x > 5)) \&\& ((y < 6) \parallel (z == 0))$ gegen beide Wahrheitswerte kann nicht als hinreichend betrachtet werden, da die Struktur der Entscheidung nicht geeignet beachtet wird.
 - Ein vollständiger Zweigüberdeckungstest kann z. B. mit den folgenden Testfällen erreicht werden:
Testfall 1: $u = 1, x = 4, y = 5, z = 0$
Testfall 2: $u = 0, x = 6, y = 5, z = 0$

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 11

Strukturorientierte, kontrollflussorientierte Testverfahren Bedingungsüberdeckungstest



- **Annahme:** Zusammengesetzte Entscheidungen werden von links nach rechts geprüft. Die Prüfung wird beendet, wenn der Wahrheitswert der Gesamtentscheidung bekannt ist. Man bezeichnet das als unvollständige Evaluation von Entscheidungen:
- **Testfall 1** führt dann zu folgender Situation:
 - Der Wert 1 der Variable u ergibt für die erste Teilentscheidung der ODER-Verknüpfung den Wahrheitswert falsch. Daher bestimmt die zweite Teilentscheidung der ODER-Verknüpfung den Wahrheitswert der ODER-Verknüpfung. Die Wahl des Wertes 4 für die Variable x ergibt eingesetzt in die erste Teilentscheidung $(x > 5)$ ebenfalls den Wahrheitswert falsch. Die Verknüpfung der ersten beiden Entscheidungen besitzt ebenfalls den Wahrheitswert falsch. Aufgrund der darauf folgenden UND-Verknüpfung ist unabhängig von den Wahrheitswerten der anderen Teilentscheidungen zu diesem Zeitpunkt bereits bekannt, dass die Gesamtentscheidung den Wahrheitswert falsch besitzen wird.
 - In vielen Fällen werden die Wahrheitswerte der verbleibenden Teilentscheidungen nicht mehr geprüft. Unabhängig davon, ob sie geprüft werden, maskiert der Wahrheitswert falsch für die erste Teilentscheidung in einer UND-Verknüpfung die Wahrheitswerte aller weiteren Teilentscheidungen. Dieser Testfall ist daher „blind“ gegenüber Fehlern in den verbleibenden Teilentscheidungen.

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 12

**Strukturorientierte, kontrollflussorientierte Testverfahren
Bedingungsüberdeckungstest**



- Testfall 2 verursacht die folgende Situation:
 - Die Wahl des Wertes 0 für die Variable u führt dazu, dass die erste Teilentscheidung ($u == 0$) den Wahrheitswert wahr besitzt. Aufgrund der ODER-Verknüpfung der ersten beiden Teilentscheidungen ist sichergestellt, dass das Ergebnis der ersten ODER-Verknüpfung wahr ist. Die zweite Teilentscheidung muss nicht geprüft werden. Die Prüfung kann direkt mit der ersten Teilentscheidung der zweiten ODER-Verknüpfung fortgesetzt werden. Der Wahrheitswert der zweiten ODER-Verknüpfung bestimmt das Gesamtergebnis. Der Wert 5 der Variable y verursacht den Wahrheitswert wahr für die Teilentscheidung ($y < 6$). Diese Teilentscheidung ist mit der vierten Teilentscheidung ODER-verknüpft. Daher ist zu diesem Zeitpunkt sichergestellt, dass das Gesamtergebnis unabhängig vom Wahrheitswert der vierten Teilentscheidung wahr sein wird. Daher ist dieser Testfall „blind“ gegenüber Fehlern in der vierten Teilentscheidung.
- Die Testfälle 1 und 2 bewirken eine vollständige Zweigüberdeckung. Keiner der beiden Testfälle prüft die vierte Teilentscheidung. Grundsätzlich gilt bei einer Evaluation von Entscheidungen von links nach rechts, dass Teilentscheidungen um so schlechter geprüft werden, je weiter rechts sie in einer zusammengesetzten Entscheidung stehen.

**Strukturorientierte, kontrollflussorientierte Testverfahren
Bedingungsüberdeckungstest**



- Die Entscheidung ($((u == 0) \parallel (x > 5)) \&\& ((y < 6) \parallel (z == 0))$) soll abkürzend als $((A \parallel B) \&\& (C \parallel D))$ geschrieben werden. Wir wollen annehmen, dass zwischen den Werten der Variablen u, x, y und z keine Abhängigkeiten existieren. Dann können die Teilentscheidungen A, B, C und D unabhängig voneinander wahr und falsch werden. Bei einer vollständigen Evaluation von Entscheidungen sind 16 Wahrheitswertkombinationen möglich.

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

**Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest**



- Der einfache Bedingungsüberdeckungstest fordert den Test aller atomarer Teilentscheidungen gegen *wahr* und *falsch*.
- Vorteile: Einfach, wenig Testaufwand
- Nachteile:
 - Wenig gründlich
 - Im allgemeinen Fall (bei vollständiger Evaluation der Entscheidungen) kann nicht garantiert werden, dass die einfache Bedingungsüberdeckung die Zweigüberdeckung subsumiert

**Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest**



- Ein einfacher Bedingungsüberdeckungstest kann z. B. mit den zwei Testfällen 6 und 11 erreicht werden. Die vier Teilentscheidungen A bis D werden jeweils gegen wahr und falsch geprüft.
- Die Teilentscheidungen $(A \parallel B)$ und $(C \parallel D)$ und die Entscheidung $((A \parallel B) \&\& (C \parallel D))$ sind jedoch in beiden Fällen wahr.
- Eine Zweigüberdeckung erreichen diese Testfälle **nicht**.

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

Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest



- Wären für den Test die Fälle 1 und 16 ausgewählt worden, so hätte man eine vollständige Zweigüberdeckung erreicht.
- Wie das Beispiel zeigt, gibt es aber Testfälle, die den einfachen Bedingungsüberdeckungstest erfüllen ohne eine Zweigüberdeckung zu gewährleisten.
- Der einfache Bedingungsüberdeckungstest stellt bei einer vollständigen Evaluation der Entscheidungen die Zweigüberdeckung **nicht** sicher.

	A	B	C	D	A B	C D	(A B)&&(C 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	w	f
6	f	w	f	w	f	w	f
7	f	w	w	f	f	w	f
8	f	w	w	w	f	w	f
9	w	f	f	f	w	f	f
10	w	f	f	w	w	f	f
11	w	f	w	f	w	f	f
12	w	f	w	w	w	f	f
13	w	w	f	f	w	w	f
14	w	w	f	w	w	w	f
15	w	w	w	f	w	w	f
16	w	w	w	w	w	w	w

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeier, 17

Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest



- Bei einer unvollständigen Evaluation von Entscheidungen existieren statt der 16 Wahrheitswertkombinationen nur 7 Kombinationsmöglichkeiten.
- Die Testfälle 6 und 11, die vollständiger Evaluation von Entscheidungen eine einfache Bedingungsüberdeckung ergeben, werden auf die Zeilen III und VII abgebildet.
- Im Unterschied zur vollständigen Evaluation von Entscheidungen bewirken die beiden Testfälle **keinen** vollständigen einfachen Bedingungsüberdeckungstest. Um das zu erreichen, müssen weitere Testfälle durchgeführt werden.

Testfälle	A	B	C	D	A B	C D	(A B)&&(C D)
I	1,2,3,4	f	f	-	f	-	f
II	5	f	w	f	f	w	f
III	6	f	w	f	w	w	w
IV	7,8	f	w	-	w	w	w
V	9,13	w	-	f	w	f	f
VI	10,14	w	-	f	w	w	w
VII	11,12,15,16	w	-	w	w	w	w

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeier, 18

Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest



- Teilentscheidung B kann nur durch Wahl des Testfalls I gegen falsch geprüft werden.
- Um D gegen falsch zu prüfen, muss entweder Testfall II oder V ausgeführt werden.
- Eine einfache Bedingungsüberdeckung ist z.B. mit den Testfälle I, II, III und VII möglich.
- Diese Testfälle stellen darüber hinaus eine vollständige Zweigüberdeckung sicher.
- Dieser Zusammenhang gilt grundsätzlich: **Bei einer unvollständigen Evaluation von Entscheidung subsumiert der einfache Bedingungsüberdeckungstest den Zweigüberdeckungstest.**

	A	B	C	D	A B	C D	(A B)&&(C D)
I	f	f	-	f	-	f	f
II	f	w	f	w	f	f	f
III	f	w	f	w	w	w	w
IV	f	w	w	-	w	w	w
V	w	-	f	f	w	f	f
VI	w	-	f	w	w	w	w
VII	w	-	w	-	w	w	w

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeier, 19

Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest



- Die Funktion ZaehleZchn enthält zwei Entscheidungen:
- a) ((Zchn >= "A") && (Zchn <= "Z") && (Gesamtzahl < INT_MAX)) und
 - b) ((Zchn == "A") || (Zchn == "E") || (Zchn == "I") || (Zchn == "O") || (Zchn == "U"))
- Entscheidung a) enthält drei atomare Entscheidungen.
 - Entscheidung b) enthält fünf atomare Entscheidungen.

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeier, 20

Strukturorientierte, kontrollflussorientierte Testverfahren
Einfacher Bedingungsüberdeckungstest



- ZaehleZchn: Vollständige Evaluation von Entscheidungen

Testfälle	Variablenwerte							
	1	2	3	4	5			
Variable 1								
Gesamtzahl	0	1	2	3	4	5	0	MAX INT
Zohn	w	w	w	w	w	f	w	w
Zohn == 'A'	w	w	w	w	w	f	w	w
Zohn <= 'Z'	w	w	w	w	w	f	w	w
Gesamtzahl < MAX INT	w	w	w	w	w	f	w	w
atomare Entscheidungen								
Zohn == 'A'	w	f	f	f	f	-	-	-
Zohn == 'E'	f	w	f	f	f	-	-	-
Zohn == 'I'	f	f	w	f	f	-	-	-
Zohn == 'O'	f	f	f	w	f	-	-	-
Zohn == 'F'	f	f	f	f	w	-	-	-

Strukturorientierte, kontrollflussorientierte Testverfahren
Bedingungs- / Entscheidungsüberdeckungstest



- Der Bedingungs-/ Entscheidungsüberdeckungstest (*condition/decision coverage*) garantiert ergänzend zu einer einfachen Bedingungsüberdeckung einen vollständigen Zweigüberdeckungstest
- Er verlangt, dass die Zweigüberdeckung explizit zusätzlich zur Bedingungsüberdeckung hergestellt wird.
- Da dies bei einer unvollständigen Evaluation von Entscheidungen bereits der einfache Bedingungsüberdeckungstest sicherstellt, ist diese Technik nur für den Fall der vollständigen Evaluation von Entscheidungen von Bedeutung
- Vorteile: Einfach, wenig Testaufwand, Zweigüberdeckungstest sichergestellt
- Nachteile:
 - Wenig gründlich
 - Struktur von Entscheidungen wird nicht wirklich beachtet

Strukturorientierte, kontrollflussorientierte Testverfahren
Bedingungs- / Entscheidungsüberdeckungstest



- Die Ausführung der Testfälle 5 und 12 ergibt eine vollständige Bedingungs-/ Entscheidungsüberdeckung, da die Teilentscheidungen A, B, C und D und die Gesamtentscheidung jeweils zu *wahr* und *falsch* evaluiert werden
- Dies ist möglich, ohne dass die zusammengesetzten Teilentscheidungen (A || B) und (C || D) gegen beide Wahrheitswerte geprüft werden.
- Der Bedingungs-/ Entscheidungsüberdeckungstest prüft atomare Teilentscheidungen und Gesamtentscheidungen.
- Er ignoriert aber weitgehend die logische Gliederung kompliziert aufgebauter Entscheidungen in mehreren Ebenen.

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

Strukturorientierte, kontrollflussorientierte Testverfahren
Minimaler Mehrfach-Bedingungsüberdeckungstest



- Der minimale Mehrfach-Bedingungsüberdeckungstest (*minimal multiple condition coverage*) verlangt, dass neben den atomaren Teilentscheidungen und der Gesamtentscheidung auch alle zusammengesetzten Teilentscheidungen gegen *wahr* und *falsch* geprüft werden.
- Da Entscheidungen hierarchisch strukturiert sein können, ist es sinnvoll, diese Struktur beim Testen zu beachten.
- Diese Form der Bedingungsüberdeckung berücksichtigt die Struktur von Entscheidungen besser als die oben dargestellten Techniken, da alle Schachtelungsebenen einer komplizierten Entscheidung gleichermaßen beachtet werden.

Strukturorientierte, kontrollflussorientierte Testverfahren
Minimaler Mehrfach-Bedingungsüberdeckungstest



- Bei einer vollständigen Evaluation von Entscheidungen ergibt die Ausführung der Testfälle 1 und 16 eine vollständige minimale Mehrfach-Bedingungsüberdeckung (Alle Teilentcheidungen A, B, C, D, (A || B) und (C || D) und die Entscheidung ((A || B) && (C || D)) sind gegen beide Wahrheitswerte getestet)
- Bei näherer Betrachtung erkennt man jedoch, dass diese beiden Testfälle die logische Struktur der Entscheidung nicht wirklich sinnvoll prüfen: Falls die Entscheidung fehlerhafterweise ((A && B) || (C && D)) lauten würde, so hätte das keiner der beiden Testfälle erkannt, obwohl alle Operatoren fehlerhaft wären. Es hätten sich für alle Teilentcheidungen und die Gesamtentscheidung identische Wahrheitswerte eingestellt. Die Testfälle sind „blind“ gegenüber diesem Fehler.

	A	B	C	D	A B	C D	((A B) && (C 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	f	f
7	f	w	w	f	f	f	f
8	f	w	w	w	f	f	f
9	w	f	f	f	f	f	f
10	w	f	f	w	f	f	f
11	w	f	w	f	f	f	f
12	w	f	w	w	f	f	f
13	w	w	f	f	f	f	f
14	w	w	f	w	f	f	f
15	w	w	w	f	f	f	f
16	w	w	w	w	f	f	f

Strukturorientierte, kontrollflussorientierte Testverfahren
Minimaler Mehrfach-Bedingungsüberdeckungstest



- Bei einer unvollständigen Evaluation von Entscheidungen sind z. B. die vier Testfälle I, II, VI und VII erforderlich:
 - Mehr Testaufwand
 - Bessere Ergebnisse
- Falls die Entscheidung fehlerhafterweise ((A && B) || (C && D)) lauten würde, so wäre dieser Testfall anders abgelaufen. Die Teilentcheidungen A, C, (A && B) und (C && D) wären zu *falsch* evaluiert worden. Die Teilentcheidungen B, D und wären nicht evaluiert worden. Die Gesamtentscheidung ist *falsch*. Man erhält das gleiche Ergebnis, aber auf eine andere Weise. Die Evaluation der Entscheidung wird an anderen Stellen abgebrochen, was eine Chance zur Erkennung der Fehler bietet.

	A	B	C	D	A B	C D	((A B) && (C D))
I	f	f	-	-	f	-	f
II	f	w	f	f	w	f	f
III	f	w	w	w	w	w	w
IV	f	w	-	-	w	-	w
V	w	-	f	f	w	f	f
VI	w	-	w	w	w	w	w
VII	w	-	-	-	w	-	w

Strukturorientierte, kontrollflussorientierte Testverfahren
Minimaler Mehrfach-Bedingungsüberdeckungstest



- Zählzahn: Vollständige Evaluation von Entscheidungen

Testfälle →	Variablenwerte				
	1	2	3	4	5
Variablen	0	1	2	3	4
Gesamtzahl	0	1	2	3	4
Zahn	A	E	T	O	U
Entscheidung					
Zahn == 'A'	w	w	w	w	w
Zahn == 'E'	w	w	w	w	w
Zahn == 'T'	w	w	w	w	w
Zahn == 'O'	w	w	w	w	w
Zahn == 'U'	w	w	w	w	w
Gesamtzahl < MAX INT	w	w	w	w	w
Entscheidung a1	w	w	w	w	w
Zahn == 'A'	w	f	f	f	f
Zahn == 'E'	f	w	f	f	f
Zahn == 'T'	f	f	w	f	f
Zahn == 'O'	f	f	f	w	f
Zahn == 'U'	f	f	f	f	w
Entscheidung b1	w	w	w	w	w

Zusätzliches Testdatum zur vollständigen Überdeckung von Entscheidung b1 bei vollständiger Entscheidungsevaluation

Strukturorientierte, kontrollflussorientierte Testverfahren
Minimaler Mehrfach-Bedingungsüberdeckungstest



- Zählzahn: Unvollständige Evaluation von Entscheidungen

Testfälle →	Variablenwerte				
	1	2	3	4	5
Variablen	0	1	2	3	4
Gesamtzahl	0	1	2	3	4
Zahn	A	E	T	O	U
Entscheidung					
Zahn == 'A'	w	w	w	w	w
Zahn == 'E'	w	w	w	w	w
Zahn == 'T'	w	w	w	w	w
Zahn == 'O'	w	w	w	w	w
Zahn == 'U'	w	w	w	w	w
Gesamtzahl < MAX INT	w	w	w	w	w
Entscheidung a1	w	w	w	w	w
Zahn == 'A'	w	f	f	f	f
Zahn == 'E'	-	w	f	f	f
Zahn == 'T'	-	-	w	f	f
Zahn == 'O'	-	-	-	w	f
Zahn == 'U'	-	-	-	-	w
Entscheidung b1	w	w	w	w	w

Zusätzliches Testdatum zur vollständigen Überdeckung von Entscheidung b1 bei unvollständiger Entscheidungsevaluation

**Strukturorientierte, kontrollflussorientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



- Der modifizierte Bedingungs-/Entscheidungsüberdeckungstest (*modified condition/decision coverage*) verlangt Testfälle, die demonstrieren, dass jede atomare Teilentscheidung den Wahrheitswert der Gesamtentscheidung unabhängig von den anderen Teilentscheidungen beeinflussen kann.
- Die Anwendung dieser Technik wird vom Standard RTCA DO-178 B für flugkritische Software (Level A) in der Avionik gefordert.
- Grundsätzlich zielt die Technik auf einen möglichst umfassenden Test der Logik von zusammengesetzten Entscheidungen mit einem vertretbaren Testaufwand.
 - Der Zusammenhang zwischen der Anzahl der atomaren Teilentscheidungen einer Entscheidung und der Anzahl der erforderlichen Testfälle ist linear.
 - Für den Test einer Entscheidung mit n Teilentscheidungen sind mindestens $n + 1$ Testfälle erforderlich.

**Strukturorientierte, kontrollfluss orientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



- Test der atomaren Teilentscheidung B mit den Testfällen 2 und 6:
 - Weisen **identische** Wahrheitswerte für die Teilentscheidungen A, C und D auf
 - **Unterscheiden** sich in den Wahrheitswerten der Teilentscheidung B. In Testfall 2 besitzt B den Wahrheitswert *falsch*. In Testfall 6 ist B *wahr*.
 - **Unterscheiden** sich im Gesamtergebnis (Testfall 2 liefert das Gesamtergebnis *falsch*, während in Testfall 6 die Gesamtentscheidung den Wert *wahr* besitzt)
- Damit ist nachgewiesen, dass die atomare Teilentscheidung B **unabhängig** von den anderen atomaren Teilentscheidungen den Wahrheitswert der Gesamtentscheidung beeinflusst.
- Eine entsprechende Situation liegt für die Testfälle 2 und 10 bezogen auf A, 9 und 10 bezogen auf D und 9 und 11 bezogen auf C vor

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

**Strukturorientierte, kontrollflussorientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



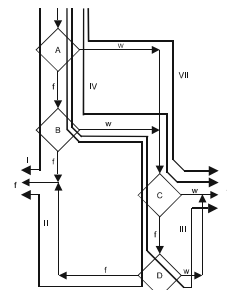
- Bei einer unvollständigen Evaluation von Entscheidungen ist es notwendig, die Anforderung abzuschwächen, die Wahrheitswerte der jeweils nicht getesteten atomaren Teilentscheidungen beizubehalten, während sich die Wahrheitswerte der jeweils getesteten atomaren Teilentscheidung und der Gesamtentscheidung verändern.
- Nun ist pro atomarer Teilentscheidung die Existenz eines Testfallpaares gefordert, das
 - bezüglich dieser Teilentscheidung beide Wahrheitswerte abdeckt und
 - bezüglich der Gesamtentscheidung beide Wahrheitswerte abdeckt und
 - für alle anderen atomaren Teilentscheidungen identische Wahrheitswerte aufweist oder an dieser Stelle nicht evaluiert wurde.
- Beispiel: Testfälle I und VII testen die atomare Teilentscheidung A. Sie weisen für die Teilentscheidung A und die Gesamtentscheidung unterschiedliche Wahrheitswerte auf und besitzen bei den verbleibenden atomaren Teilentscheidungen nur Wahrheitswerte, falls diese bei dem jeweils anderen Testfall nicht evaluiert wurden.

	A	B	C	D	A&B	C&D	(A&B)&(C&D)
I	f	f	-	f	-	-	f
II	w	f	f	w	f	f	f
III	w	w	f	w	w	f	w
IV	w	w	w	f	w	w	w
V	w	w	w	w	w	w	w
VI	w	f	w	w	w	w	w
VII	w	w	-	w	-	-	w

**Strukturorientierte, kontrollfluss orientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



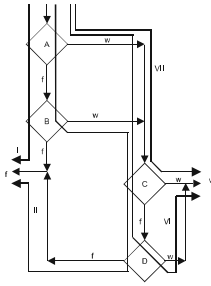
- Ein vollständig durchgeführter modifizierter Bedingungs-/Entscheidungsüberdeckungstest bewirkt einen Zweigüberdeckungstest auf der Objektcodeebene. Auf der Objektcodeebene sind ausschließlich einfache – d. h. atomare – Entscheidungen vorhanden.



**Strukturorientierte, kontrollflussorientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



- Aber: Nicht jeder Zweigüberdeckungstest auf der Objektcodeebene bewirkt einen vollständigen modifizierten Bedingungs-/Entscheidungsüberdeckungstest



**Strukturorientierte, kontrollflussorientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



- ZaehleZchn: Vollständige Evaluation von Entscheidungen mit gekoppelten Teilentscheidungen (Jede Teilentscheidung kann unabhängig von den anderen falsch werden, aber nicht wahr)

	1) Zchn == 'A'	2) Zchn == 'E'	3) Zchn == 'T'	4) Zchn == 'O'	5) Zchn == 'U'	(Zchn == 'A') (Zchn == 'E') (Zchn == 'T') (Zchn == 'O') (Zchn == 'U')
1	w	f	f	f	f	w
2	f	w	f	f	f	w
3	f	f	w	f	f	w
4	f	f	f	w	f	w
5	f	f	f	f	w	w
	f	f	f	f	f	f

**Strukturorientierte, kontrollflussorientierte Testverfahren
Modifizierter Bedingungs-/Entscheidungsüberdeckungstest**



- ZaehleZchn: Unvollständige Evaluation von Entscheidungen mit gekoppelten Teilentscheidungen

	1) Zchn == 'A'	2) Zchn == 'E'	3) Zchn == 'T'	4) Zchn == 'O'	5) Zchn == 'U'	(Zchn == 'A') (Zchn == 'E') (Zchn == 'T') (Zchn == 'O') (Zchn == 'U')
1	w	-	-	-	-	w
2	f	w	-	-	-	w
3	f	f	w	-	-	w
4	f	f	f	w	-	w
5	f	f	f	f	w	w
	f	f	f	f	f	f

**Strukturorientierte, kontrollflussorientierte Testverfahren
Mehrfach-Bedingungsüberdeckungstest**



- Der Mehrfach-Bedingungsüberdeckungstest (*multiple condition coverage*) fordert den Test aller Wahrheitswertkombinationen der atomaren Teilentscheidungen.

- Vorteile:
 - sehr umfassender Test
 - subsumiert den Zweigüberdeckungstest und alle anderen Bedingungsüberdeckungstesttechniken.
- Nachteile:
 - hoher Testaufwand (2ⁿ Testfälle bei einer Entscheidung, die aus n Teilentscheidungen zusammengesetzt ist)
 - Manchmal können nicht alle Kombinationen hergestellt werden (z.B. wg. unvollständiger Evaluation von Entscheidungen oder Abhängigkeiten zwischen atomaren Entscheidungen)

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

Strukturorientierte, kontrollflussorientierte Testverfahren Mehrfach-Bedingungsüberdeckungstest



- Abhängigkeiten zwischen atomaren Teilentscheidungen der Operation $Z\text{aehle}Z\text{chn}$: Von den $32 (2^5)$ Wahrheitswertkombinationen der Entscheidung $((Z\text{chn} == 'A') \vee (Z\text{chn} == 'E') \vee (Z\text{chn} == 'T') \vee (Z\text{chn} == 'O') \vee (Z\text{chn} == 'U'))$ der Operation $Z\text{aehle}Z\text{chn}$ sind nur 6 herstellbar

Zchn	'A'	'E'	'T'	'O'	'U'	alle anderen Zeichen	?	?	?
Zchn == 'A'	w	f	f	f	f	f	w	...	w
Zchn == 'E'	f	w	f	f	f	f	w	...	w
Zchn == 'T'	f	f	w	f	f	f	f	...	w
Zchn == 'O'	f	f	f	w	f	f	f	...	w
Zchn == 'U'	f	f	f	f	w	f	f	...	w
Entscheidungs b)	w	w	w	w	w	f	w	...	w

6 Möglichkeiten (herstellbar) 26 Möglichkeiten (nicht herstellbar)
32 Möglichkeiten

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 37

Strukturorientierte, kontrollflussorientierte Testverfahren Bedingungsüberdeckungstest: Bewertung



- Sinnvoll im Falle von komplizierter Verarbeitungslogik, die zu kompliziert aufgebauten Entscheidungen führt

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 38

Strukturorientierte, kontrollflussorientierte Testverfahren Pfadüberdeckungstest



- Eine Programmausführung verursacht einen Durchlauf durch einen Programmpfad, der mehrere Zweige und Anweisungen enthält.
- Frage: Wie kann dies in ein Testverfahren eingebracht werden?

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 39

Strukturorientierte, kontrollflussorientierte Testverfahren Pfadüberdeckungstest



- Eine vollständige Pfadüberdeckung fordert die Ausführung aller unterschiedlichen Pfade des zu testenden Programms.
 - Ein Pfad p ist eine Sequenz von Knoten (i, n_1, \dots, n_m, j) des Kontrollflussgraphen mit dem Startknoten i und dem Endknoten j .
- Nachteile:
 - Der Pfadüberdeckungstest ist für reale Programme in der Regel nicht durchführbar, da sie eine unendlich hohe Anzahl von Pfaden besitzen können. Unter der Annahme, dass der größte Wert einer Variablen vom Typ INTEGER 32767 beträgt, erhält man für die Operation $Z\text{aehle}Z\text{chn}$ die unvorstellbar hohe Anzahl von $2^{32767} - 1$ Testpfaden (etwa $1,41 \cdot 10^{9864}$ Pfade). Die erforderliche Testdauer bei einem Tag und Nacht pausenlos durchlaufenden Test und einer zugrundegelegten Testintensität von 1000 Pfaden pro Sekunde würde $4,5 \cdot 10^{9843}$ Jahre dauern. Zum Vergleich: Das Alter der Erde wird mit etwas mehr als $4,5 \cdot 10^9$ Jahren angegeben. Ein vollständiger Pfadüberdeckungstest der Operation $Z\text{aehle}Z\text{chn}$ ist daher absolut ausgeschlossen.
 - Oft ist ein Bruchteil der anhand des Kontrollflussgraphen konstruierbaren Pfade nicht ausführbar.
- Frage: Wie kann der Pfadüberdeckungstest modifiziert werden, so dass er durchführbar wird?

Software-Basistechnologie III / Software-Konstruktion II

© Prof. Dr. Lippemeyer, 40

Strukturorientierte, kontrollflussorientierte Testverfahren Beispiel



```
BEGIN
  IF (A=5) AND (B=0) THEN
    Anweisung1;
  END;
  IF (X=0) THEN
    Anweisung2;
  END;
END;
```

Strukturorientierte, kontrollflussorientierte Testverfahren Strukturierter Pfadtest und boundary interior-Pfadtest



- Grundsätzlich werden beim strukturierten Pfadtest nur Pfade bis zur k -ten Ausführung des Schleifenrumpfs unterschieden (vermeidet die Explosion der Pfadanzahl durch Schleifen)
- Wir geben mit k nicht die Anzahl der Schleifenwiederholungen sondern die Ausführungsanzahl des Schleifenrumpfs an.
- Der strukturierte Pfadtest mit $k=2$ wird als Grenze-Inneres-Überdeckung (*boundary interior coverage*) bezeichnet.
- Die *boundary interior coverage* unterscheidet die drei Fälle *keine Schleifenausführung*, *eine Schleifenausführung* und *mindestens zwei Schleifenausführungen*. Das ist aufgrund der möglichen Abhängigkeiten zwischen Variablen vor, in und hinter der Schleife besonders sinnvoll.

Strukturorientierte, kontrollflussorientierte Testverfahren Strukturierter Pfadtest und boundary interior-Pfadtest



□ Beispiel

Im Folgenden sind die für den *boundary interior*-Test der Operation ZaehleZchn benötigten Testfälle angegeben.

- 1. Testfall für den Pfad außerhalb der Schleife:
Die Ausführung mit *Gesamtzahl* = *INT_MAX* bewirkt die Nichtausführung des Schleifenrumpfes.
Testpfad: $n_{\text{start}}, n_1, n_2, n_{\text{final}}$

Strukturorientierte, kontrollflussorientierte Testverfahren Strukturierter Pfadtest und boundary interior-Pfadtest



- 2. Boundary-Testfälle:
 - a. Die Ausführung mit *Gesamtzahl* = 0 und Eingabe der Zeichenfolge A1 bewirkt das Betreten des Schleifenrumpfes, die Ausführung des *wahr*-Zweiges der Selektion und anschließend einen Schleifenabbruch: Testpfad: $n_{\text{start}}, n_1, n_2, n_3, n_4, n_5, n_2, n_{\text{final}}$
 - b. Die Ausführung mit *Gesamtzahl* = 0 und Eingabe der Zeichenfolge B1 bewirkt das Betreten des Schleifenrumpfes, die Ausführung des *falsch*-Zweiges der Selektion und anschließend einen Schleifenabbruch.
Testpfad: $n_{\text{start}}, n_1, n_2, n_3, n_5, n_2, n_{\text{final}}$

Strukturorientierte, kontrollflussorientierte Testverfahren
Pfadüberdeckungstest:
Strukturierter Pfadtest und boundary interior-Pfadtest



3. Interior-Testfälle:

- a. Die Ausführung mit *Gesamtzahl* = 0 und Eingabe der Zeichenfolge *EIN!* bewirkt eine dreimalige Ausführung des Schleifenrumpfes. Bei den ersten zwei Ausführungen wird der *wahr*-Zweig der Selektion durchlaufen. Der dritte Durchlauf ist ohne Bedeutung für den Test
 Testpfad: $n_{\text{start}}, n_1, n_2, n_3, n_4, n_5, n_2, n_3, n_4, n_5, n_2, n_3, n_4, n_5, n_2, n_{\text{final}}$
- b. Die Ausführung mit *Gesamtzahl* = 0 und Eingabe der Zeichenfolge *AH!* bewirkt eine zweimalige Ausführung des Schleifenrumpfes. Bei der ersten Ausführung wird der *wahr*-Zweig der Selektion durchlaufen. Beim zweiten Durchlauf wird der *falsch*-Zweig der Selektion ausgeführt. Das Rufzeichen bricht den Wiederholungsvorgang ab, was für *interior*-Tests nach der zweiten Ausführung des Schleifenrumpfes gestattet ist.
 Testpfad: $n_{\text{start}}, n_1, n_2, n_3, n_4, n_5, n_2, n_3, n_5, n_2, n_{\text{final}}$

Strukturorientierte, kontrollflussorientierte Testverfahren
Pfadüberdeckungstest:
Strukturierter Pfadtest und boundary interior-Pfadtest



- c. Die Ausführung mit *Gesamtzahl* = 0 und Eingabe der Zeichenfolge *HA!* bewirkt eine zweimalige Ausführung des Schleifenrumpfes. Bei der ersten Ausführung wird der *falsch*-Zweig der Selektion durchlaufen. Beim zweiten Durchlauf wird der *wahr*-Zweig der Selektion ausgeführt. Das Rufzeichen bricht den Wiederholungsvorgang ab.
 Testpfad: $n_{\text{start}}, n_1, n_2, n_3, n_5, n_2, n_3, n_4, n_5, n_2, n_{\text{final}}$
 - d. Die Ausführung mit *Gesamtzahl* = 0 und Eingabe der Zeichenfolge *HH!* bewirkt eine zweimalige Ausführung des Schleifenrumpfes. Bei beiden Ausführungen wird der *falsch*-Zweig der Selektion durchlaufen. Das Rufzeichen bricht den Wiederholungsvorgang ab.
 Testpfad: $n_{\text{start}}, n_1, n_2, n_3, n_5, n_2, n_3, n_5, n_2, n_{\text{final}}$
- Die aufgeführten sieben Testfälle sind hinreichend für den umfassenden Test der Schleife nach dem *boundary interior*-Kriterium.

Strukturorientierte, kontrollflussorientierte Testverfahren
Subsumptionsrelationen der kontrollfluss orientierten Tests

