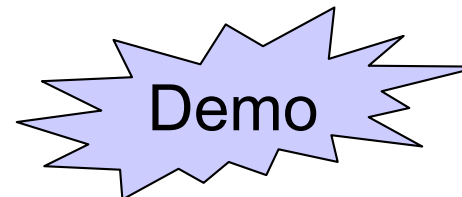


AK Test eingebetteter Systeme, 11.10.2002

Strukturtestkriterien und deren Automatisierung

Joachim Wegener, André Baresel, Torsten Pehl
DaimlerChrysler AG, Research and Technology
Joachim.Wegener@DaimlerChrysler.com

- Code-basierte Strukturtestkriterien
- Evolutionäre Strukturtests
- Automatisierung



Strukturtests

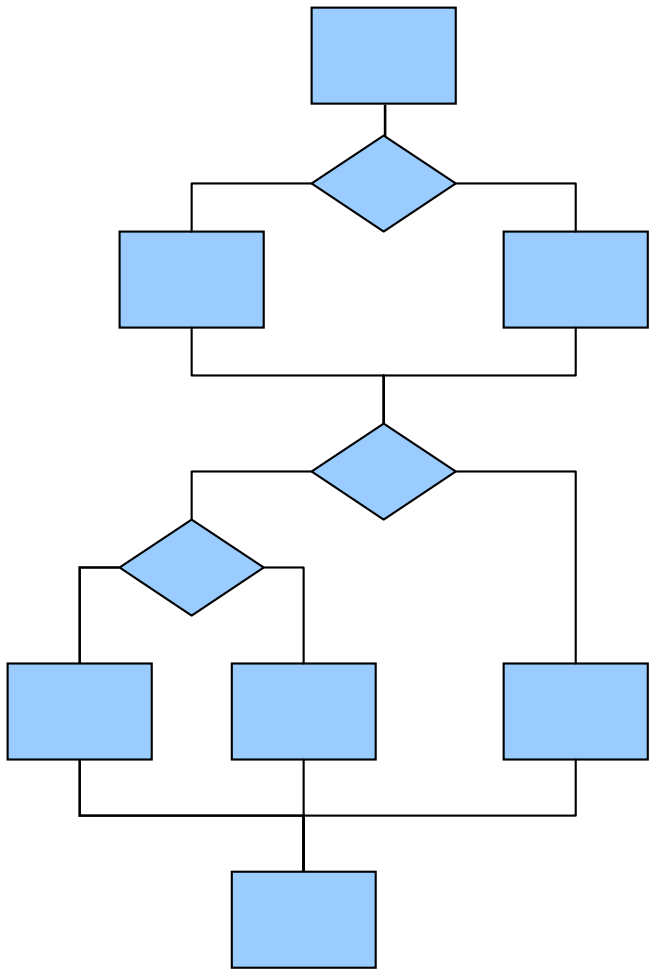
- Gängige Anwendungsgebiete
 - Testfallermittlung (anhand interner Programmstrukturen)
 - Beurteilung der Testqualität anhand von Überdeckungsmessungen
- Die Testkriterien definieren die Testziele, z.B.
 - Anweisungstest Ausführung aller Anweisungen
 - Zweigttest Ausführung aller Zweige
 - Bedingungstest Ausführung aller atomaren Bedingungen mit True und False
 -

Kontrollflussgraph

```

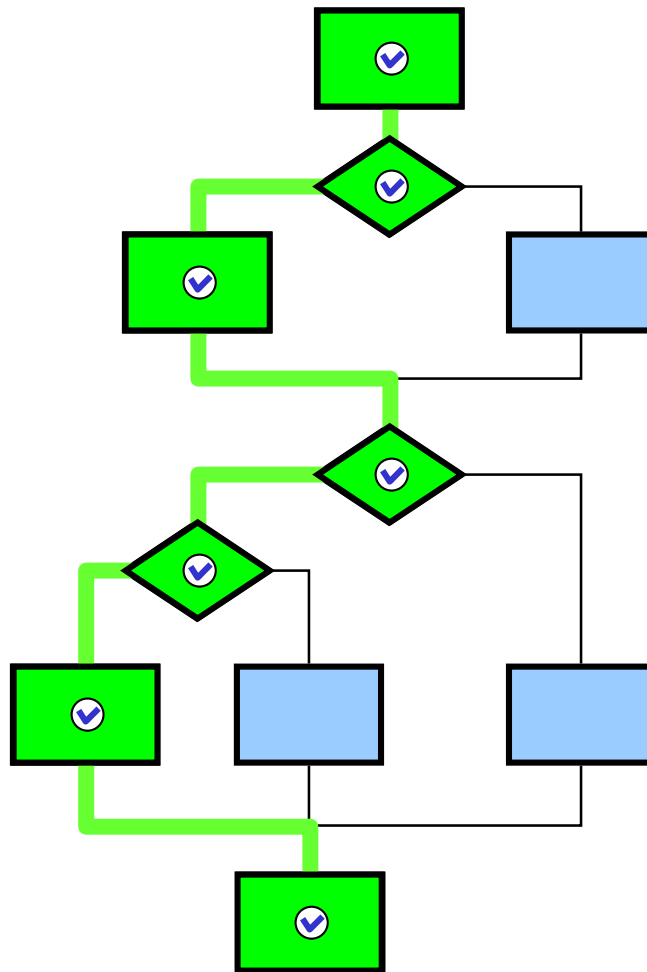
int TestFunction( int x, int y, int z )
{
    int a = 0;
    int b = 10;
    int c = 100;
    int result;

    if ( y > b || x < b || z == 100 ){
        a = 6;
        b = x;
        result = a * x;
    }else{
        b = 7;
        a = x;
        result = a * y;
    }
    if ( a == 6 && b == 7 && c == 42 ){
        if ( x == y || z == 23 ){
            b = b + 1;
            c = result + b;
        }else{
            a = a + 1;
            c = result * b;
        }
    }else{
        result = x * y * z;
    }
    return result;
}
    
```



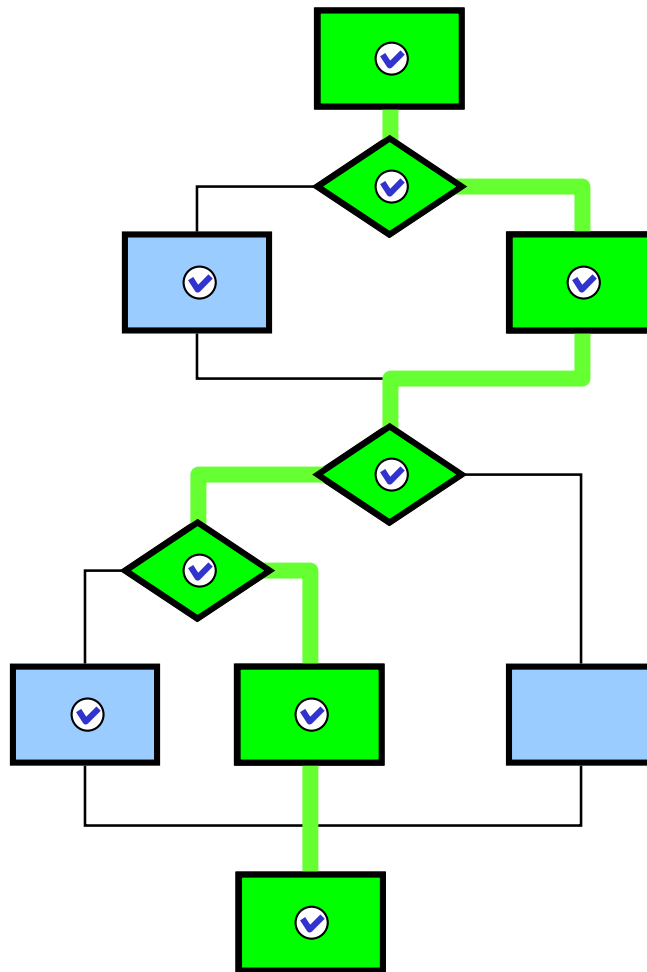
- Strukturtestkriterien beziehen sich üblicherweise auf den Kontrollflussgraphen des zu testenden Programms (Ebene Programmcode)
- die Knoten eines Kontrollflussgraphen bilden Anweisungsblöcke

Anweisungsüberdeckung - Testziele



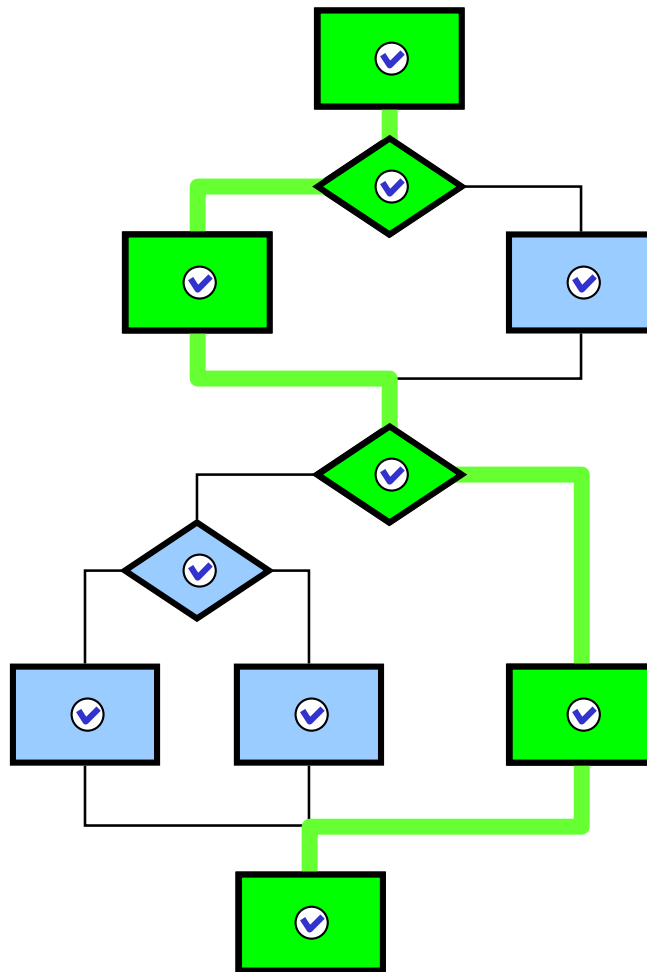
- Alle Knoten (Anweisungsblöcke) des Graphen sind Testziele

Anweisungsüberdeckung - Testziele



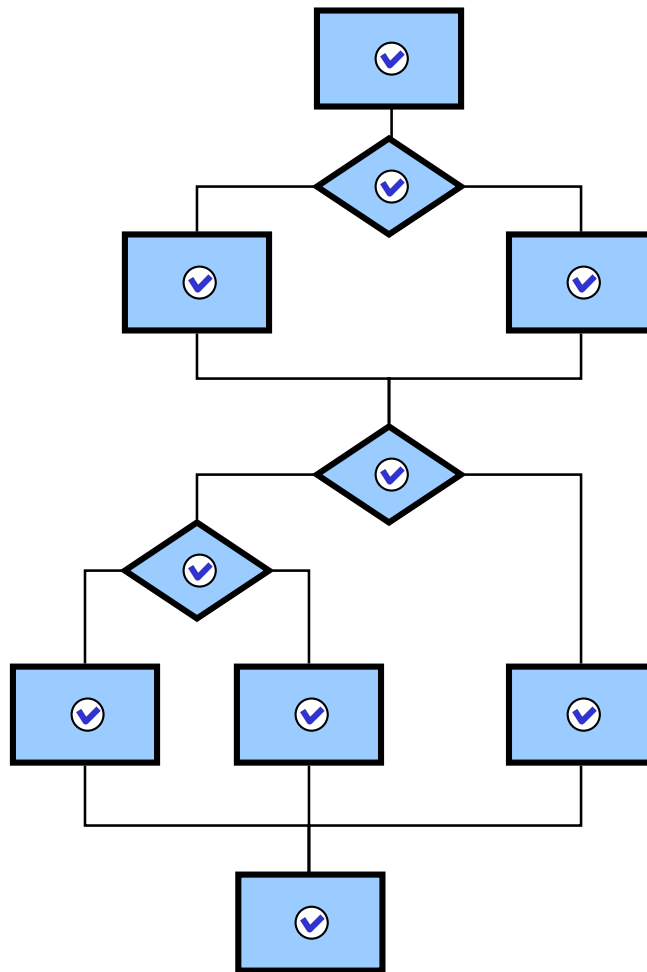
- Alle Knoten (Anweisungsblöcke) des Graphen sind Testziele

Anweisungsüberdeckung - Testziele



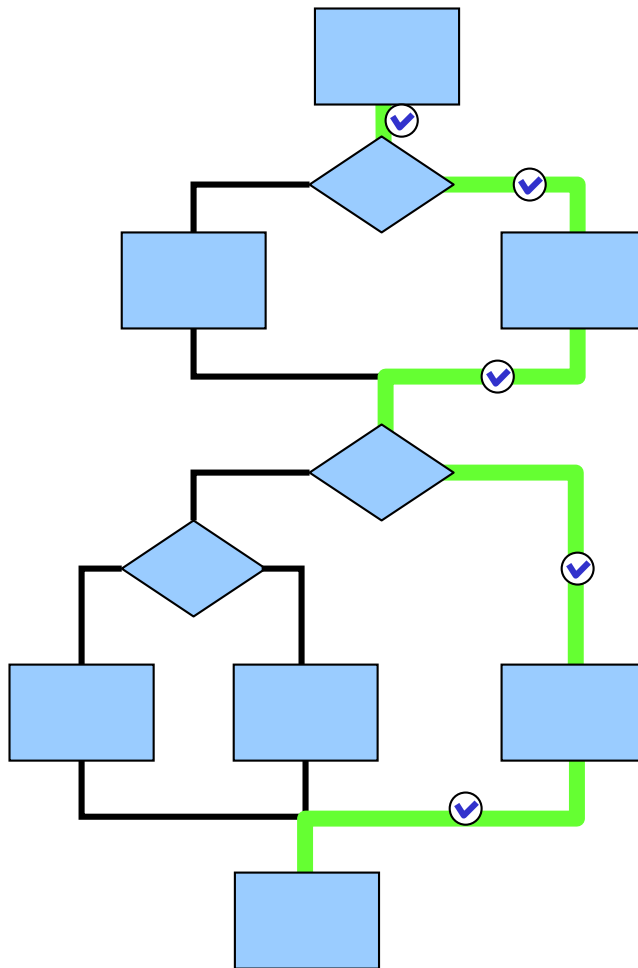
- Alle Knoten (Anweisungsblöcke) des Graphen sind Testziele

Anweisungsüberdeckung - Testziele



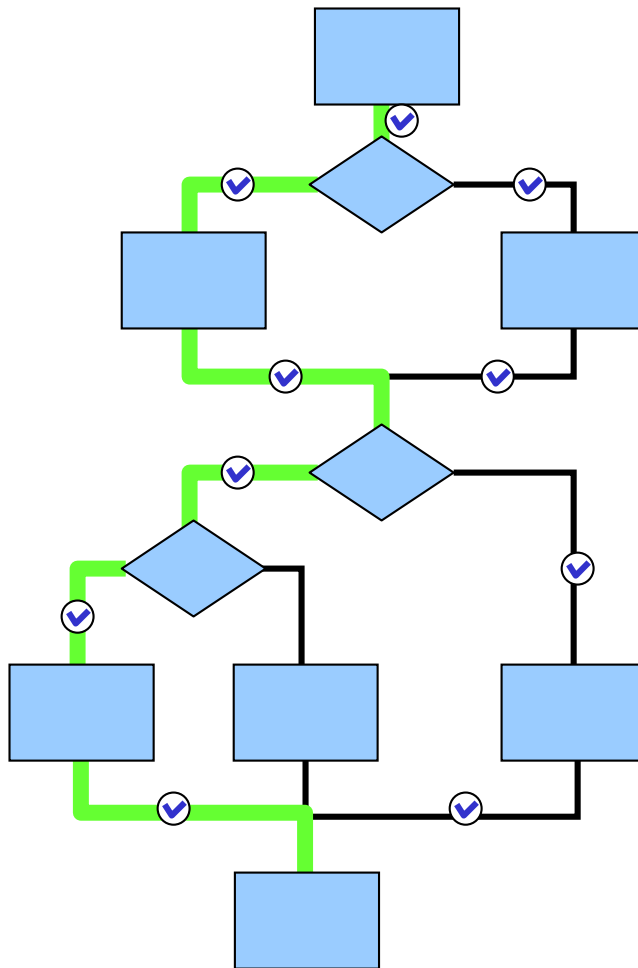
- Alle Knoten (Anweisungsblöcke) des Graphen sind Testziele

Zweigüberdeckung - Testziele



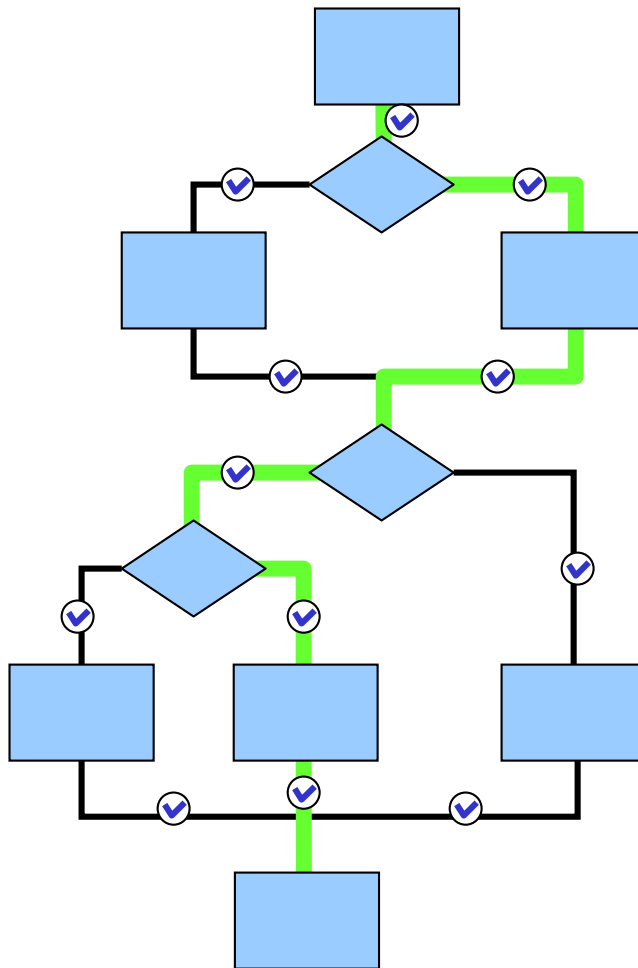
- Alle Zweige (Übergänge zwischen Anweisungsblöcken) des Graphen sind Testziele

Zweigüberdeckung - Testziele



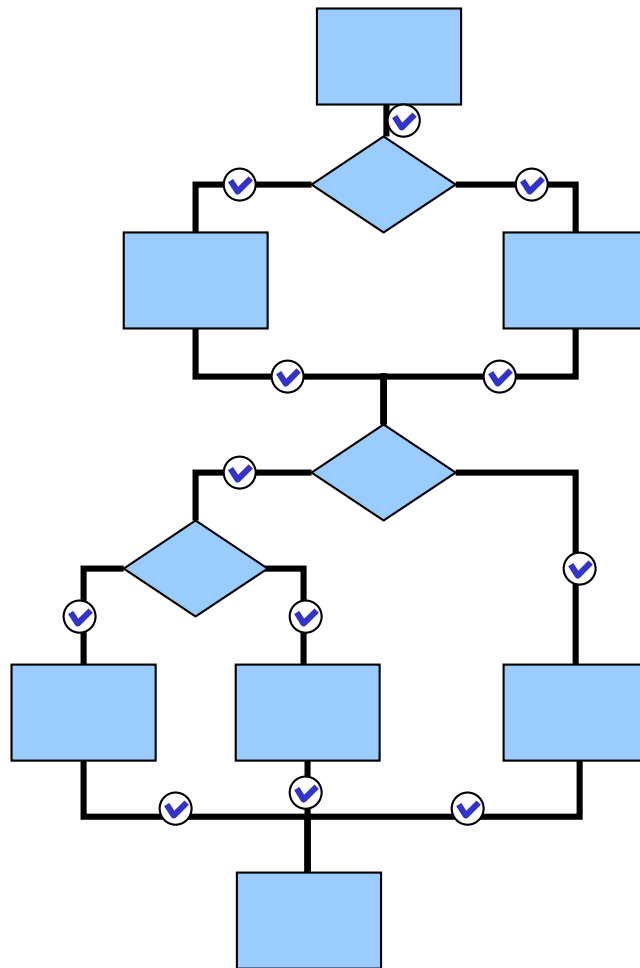
- Alle Zweige (Übergänge zwischen Anweisungsblöcken) des Graphen sind Testziele

Zweigüberdeckung - Testziele



- Alle Zweige (Übergänge zwischen Anweisungsblöcken) des Graphen sind Testziele

Zweigüberdeckung - Testziele



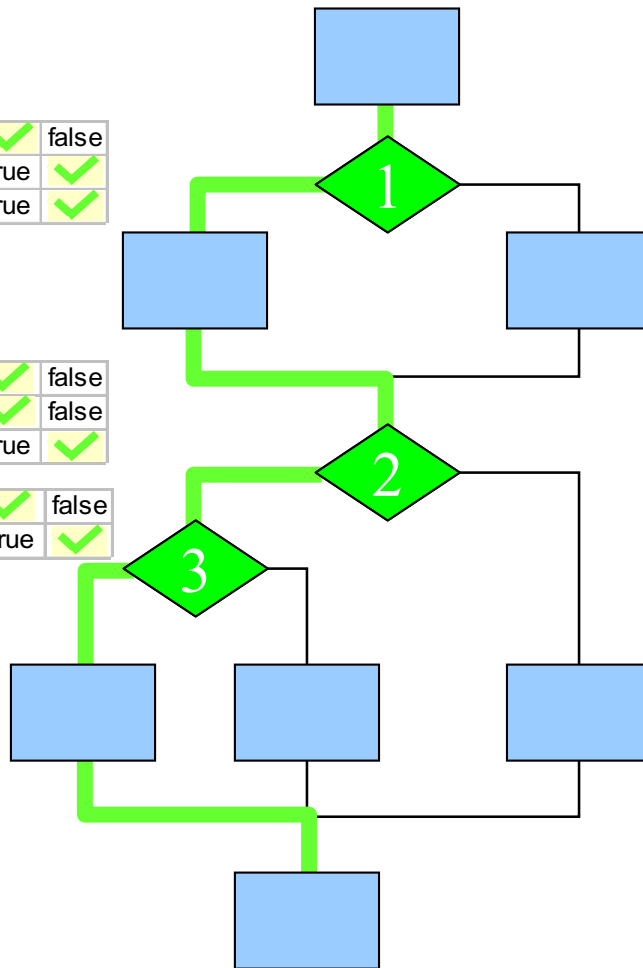
- Alle Zweige (Übergänge zwischen Anweisungsblöcken) des Graphen sind Testziele

Einfache Bedingungsüberdeckung - Testziele

y > b	✓	false
x < b	true	✓
z == 100	true	✓

a == 6	✓	false
b == 7	✓	false
c == 42	true	✓

x == y	✓	false
z == 23	true	✓



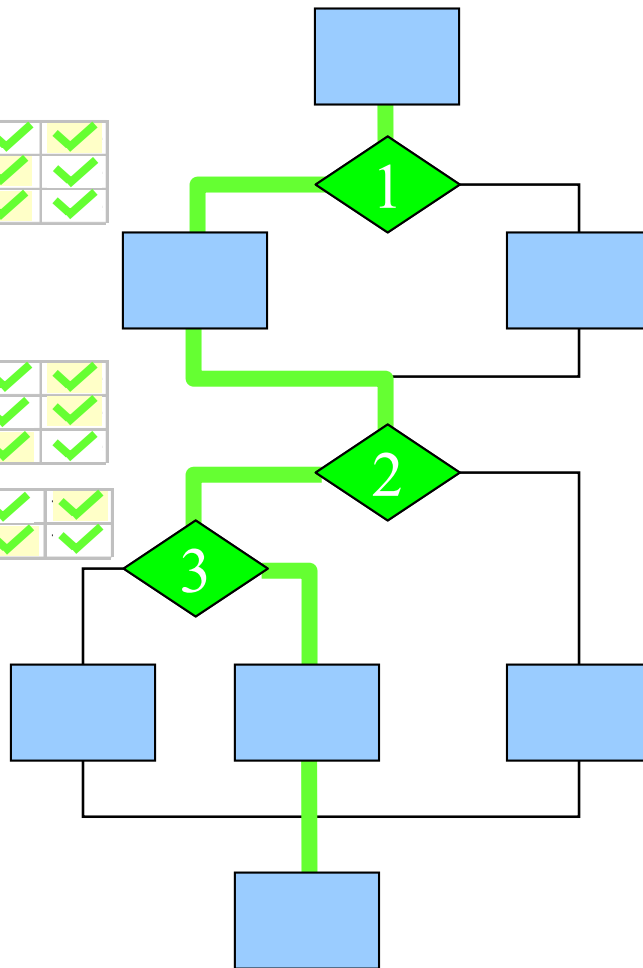
- Alle atomare Bedingungen sind Testziele und müssen wenigstens einmal mit True und einmal mit False ausgewertet werden

Einfache Bedingungsüberdeckung - Testziele

y > b	✓	✓
x < b	✓	✓
z == 100	✓	✓

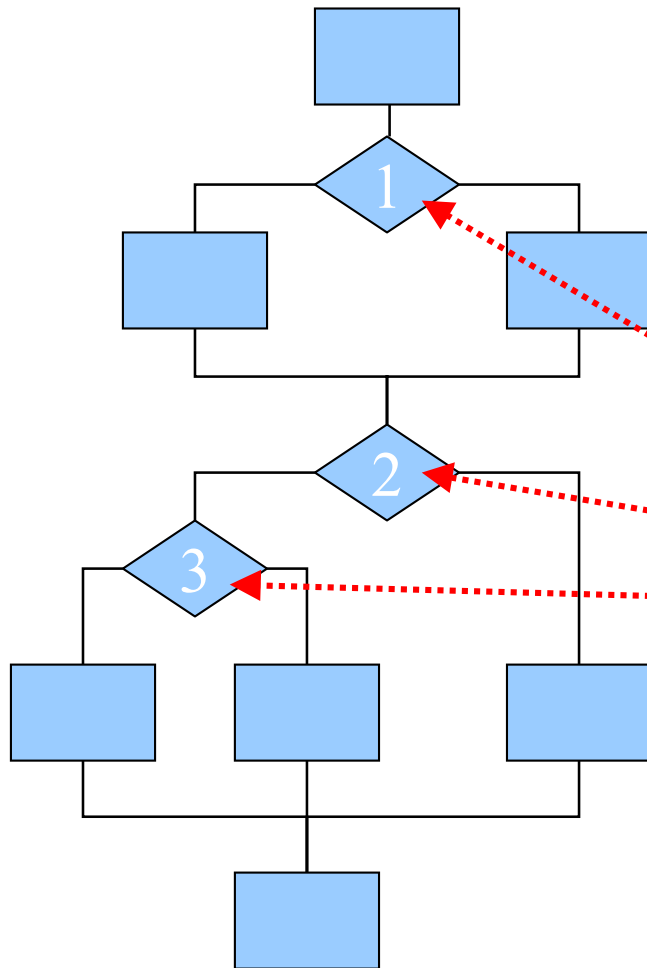
a == 6	✓	✓
b == 7	✓	✓
c == 42	✓	✓

x == y	✓	✓
z == 23	✓	✓



- Alle atomare Bedingungen sind Testziele und müssen wenigstens einmal mit True und einmal mit False ausgewertet werden

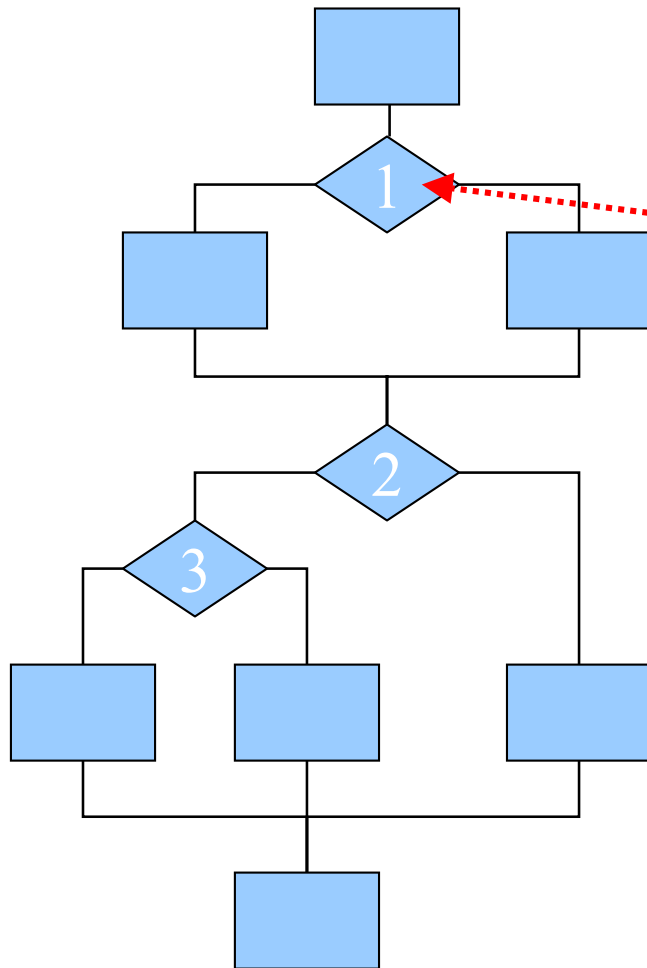
Mehrfachbedingungsüberdeckung - Testziele



- Alle Wertekombinationen der atomaren Bedingungen sind Testziele
 - jede Wertekombination entspricht einem Testziel
 - Testziele:

Testziel	y > b	x < b	z == 100
9	false	false	false
10	true	false	true
11	false	true	false
12	true	true	true
13	false	false	false
14	true	false	true
15	true	true	false
16	true	true	true
17	false	false	false
18	false	false	true
19	false	true	false
20	false	true	true

MCDC (Modified Condition Decision Coverage) - Testziele

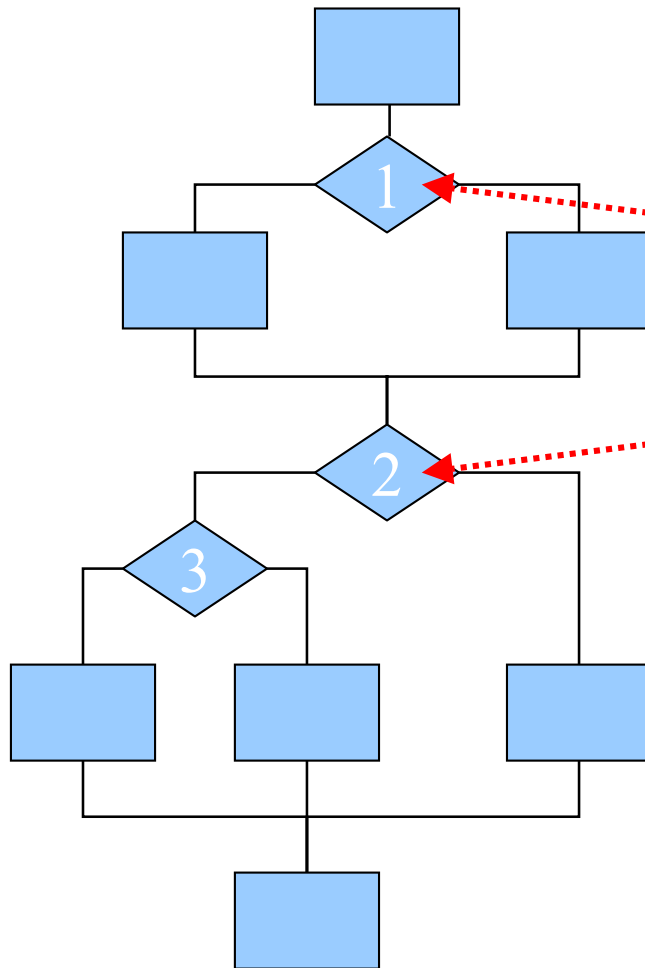


- Untermenge der Mehrfachbedingungsüberdeckung

Testziel	y > b	x < b	z == 100
1	false	false	false
2	false	false	true
3	false	true	false
4	false	true	true
5	true	false	false
6	true	false	true
7	true	true	false
8	true	true	true

- umfaßt alle Wertekombinationen, für die sich aus der Veränderung eines Wertes eine Änderung des Wertes der Gesamtbedingung ergibt

MCDC (Modified Condition Decision Coverage) - Testziele

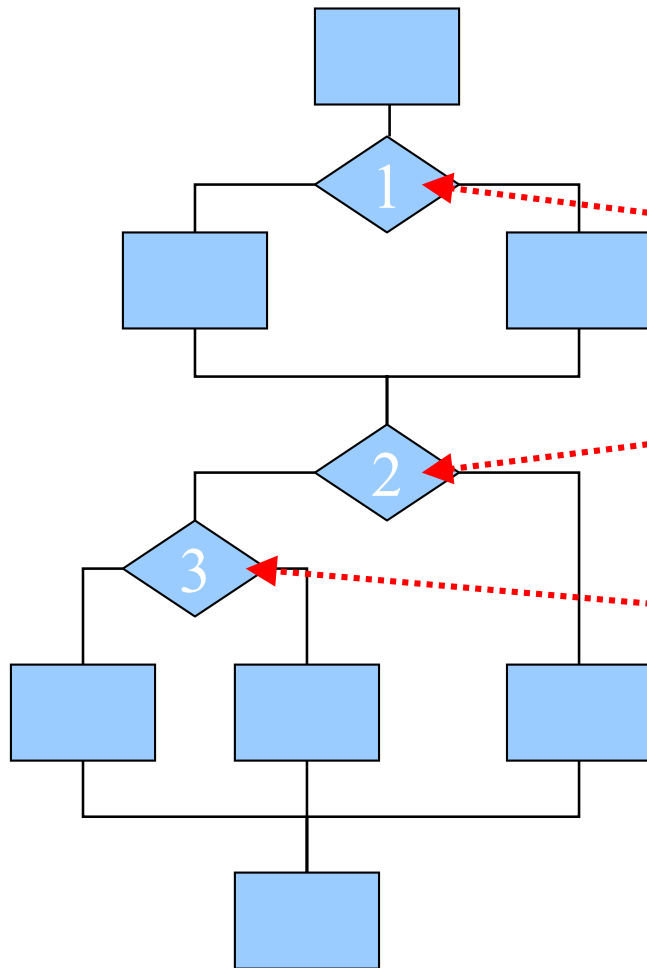


- Untermenge der Mehrfachbedingungsüberdeckung

Testziel	y > b	x < b	z == 100
1	false	false	false
2	false	false	true
3	false	true	false
4	true	false	false

Testziel	a==6	b==7	c==42
0	false	false	false
10	false	false	true
11	false	true	false
12	false	true	true
13	true	false	false
14	true	false	true
15	true	true	false
16	true	true	true

MCDC (Modified Condition Decision Coverage) - Testziele



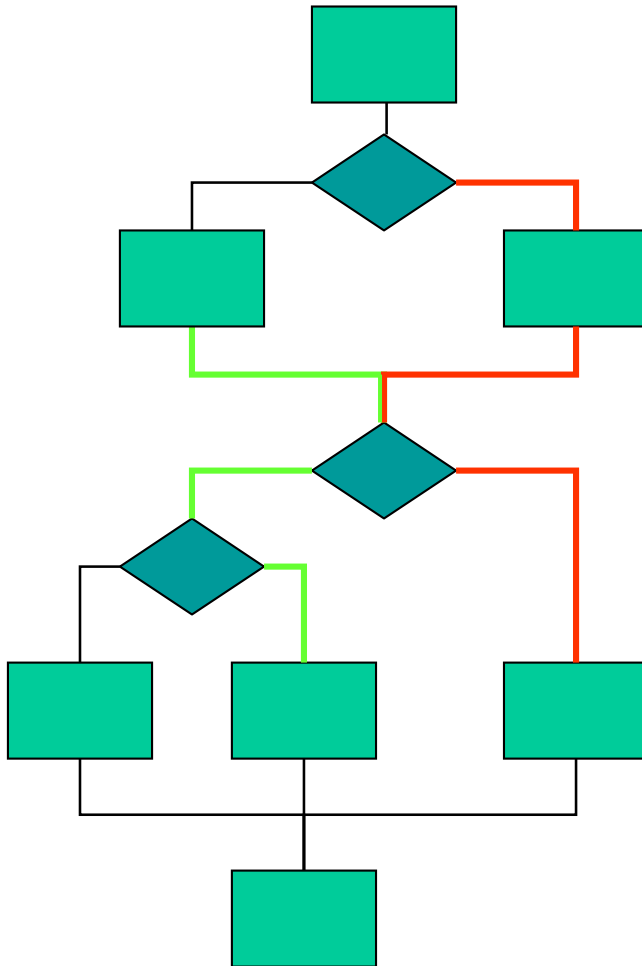
- Untermenge der Mehrfachbedingungsüberdeckung

Testziel	y > b	x < b	z == 100
1	false	false	false
2	false	false	true
3	false	true	false
4	true	false	false

Testziel	a==6	b==7	c==42
5	false	false	false
6	false	false	true
7	false	true	false
8	true	false	false

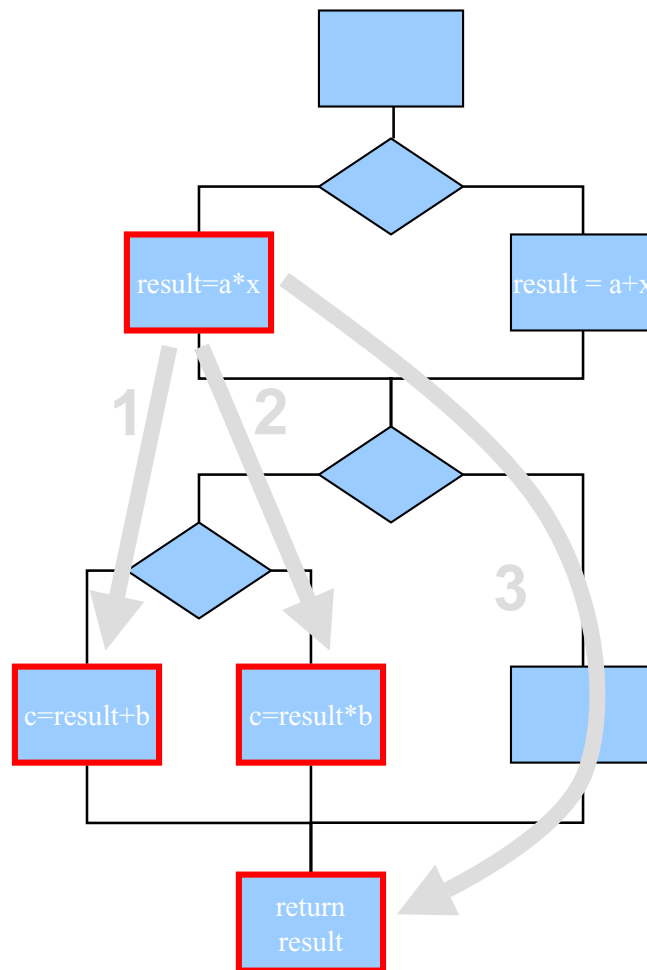
Testziel	x==y	z==23
9	false	false
10	true	false
11	false	true
12	true	true

Segment-Pair-Überdeckung Länge 3 - Testziele



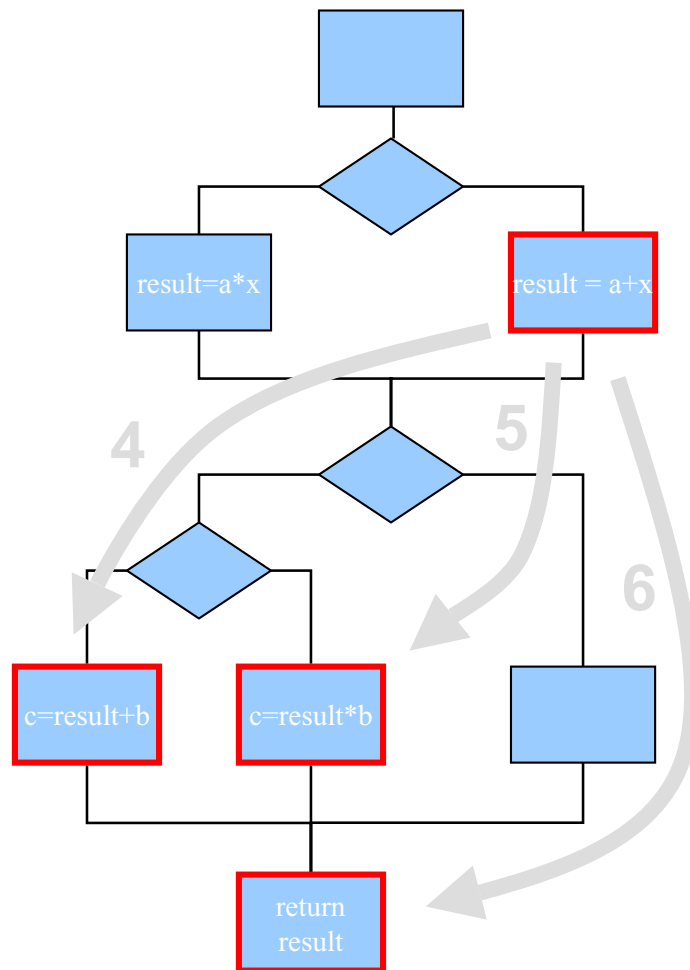
- Alle Pfade der Länge 3 (3 aufeinander folgende Zweige) im Graphen sind Testziele
- Definition analog zu voriger Folie: ??? Noch mal prüfen ???
- 2 Testziele ???

DefUse-Überdeckung - Testziele



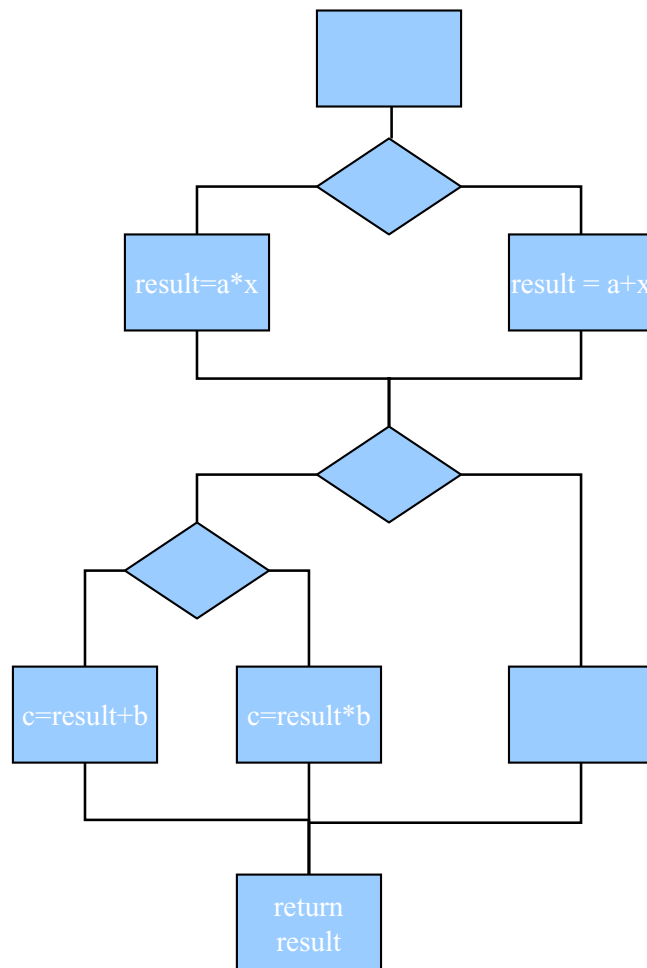
- Alle Kombinationen von Schreib- (Def) und Lesezugriffen (Use) auf Variablen sind Testziele

DefUse-Überdeckung - Testziele



- Alle Kombinationen von Schreib- (Def) und Lesezugriffen (Use) auf Variablen sind Testziele

DefUse-Überdeckung - Testziele



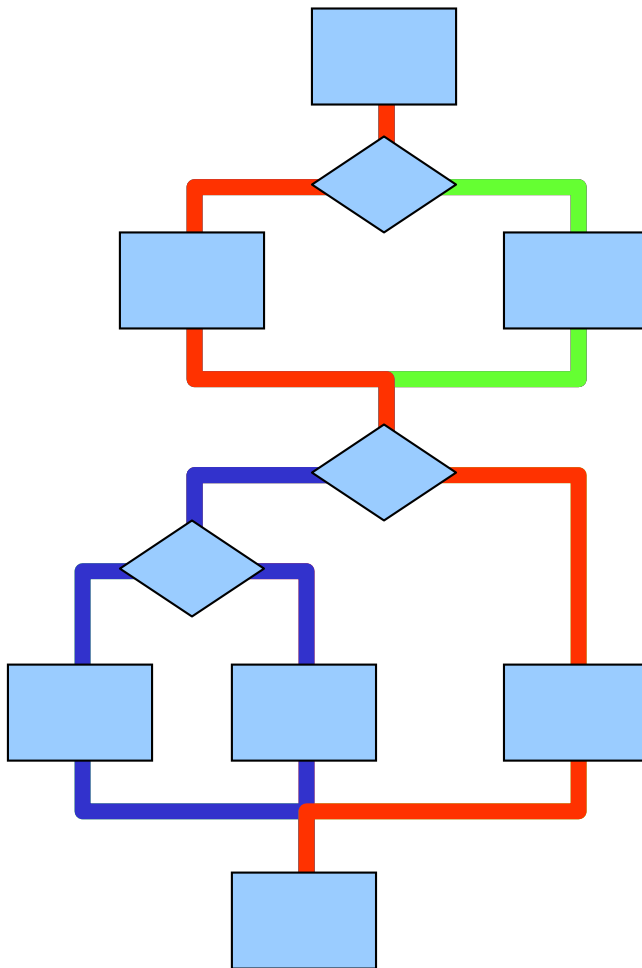
- Alle Kombinationen von Schreib- (Def) und Lesezugriffen (Use) auf Variablen sind Testziele

- Es gibt drei Varianten:

- DefUse
- All-Defs
- All-Uses

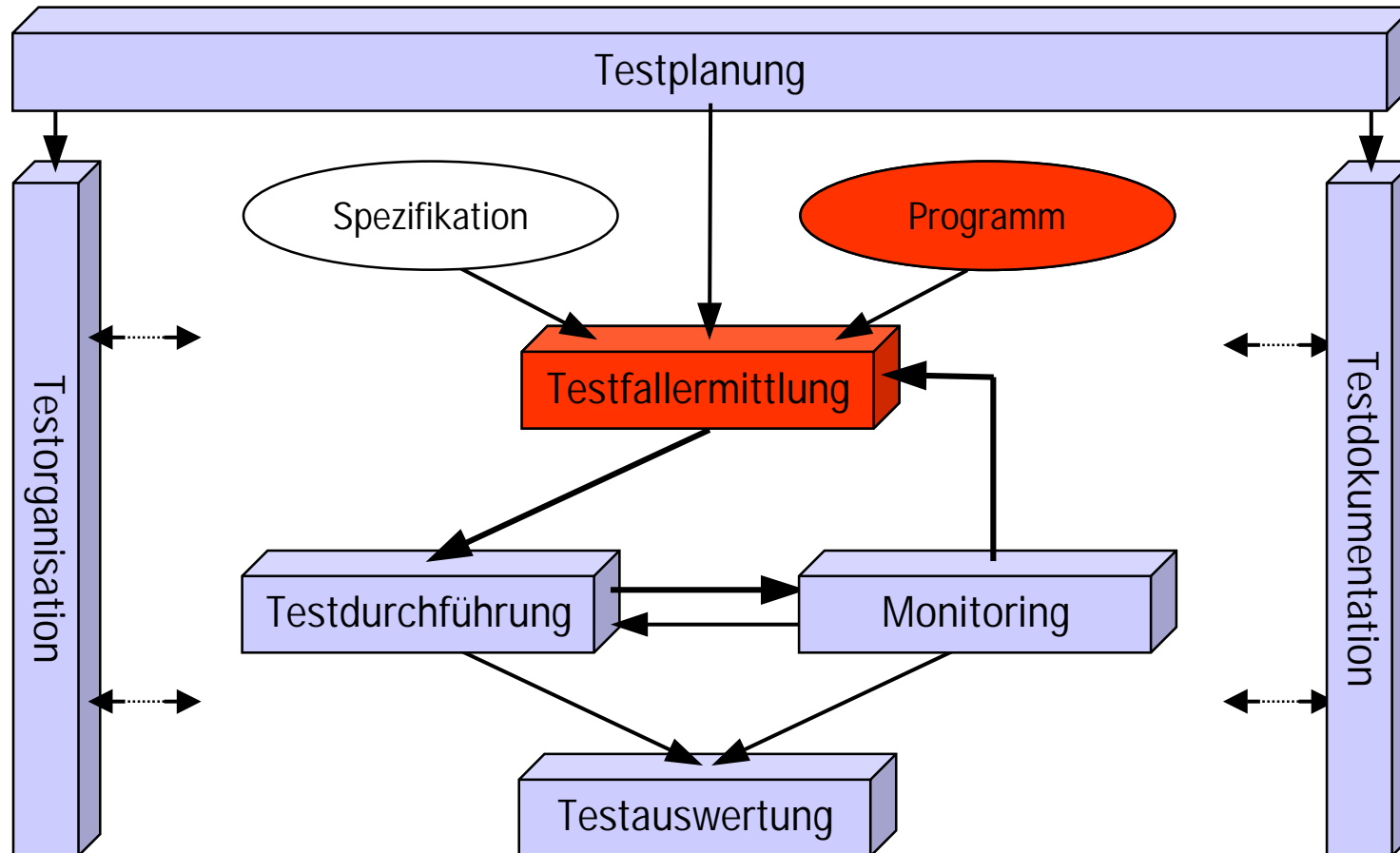
All-Defs und All-Uses sind ein Subset von DefUse

Pfadüberdeckung - Testziele

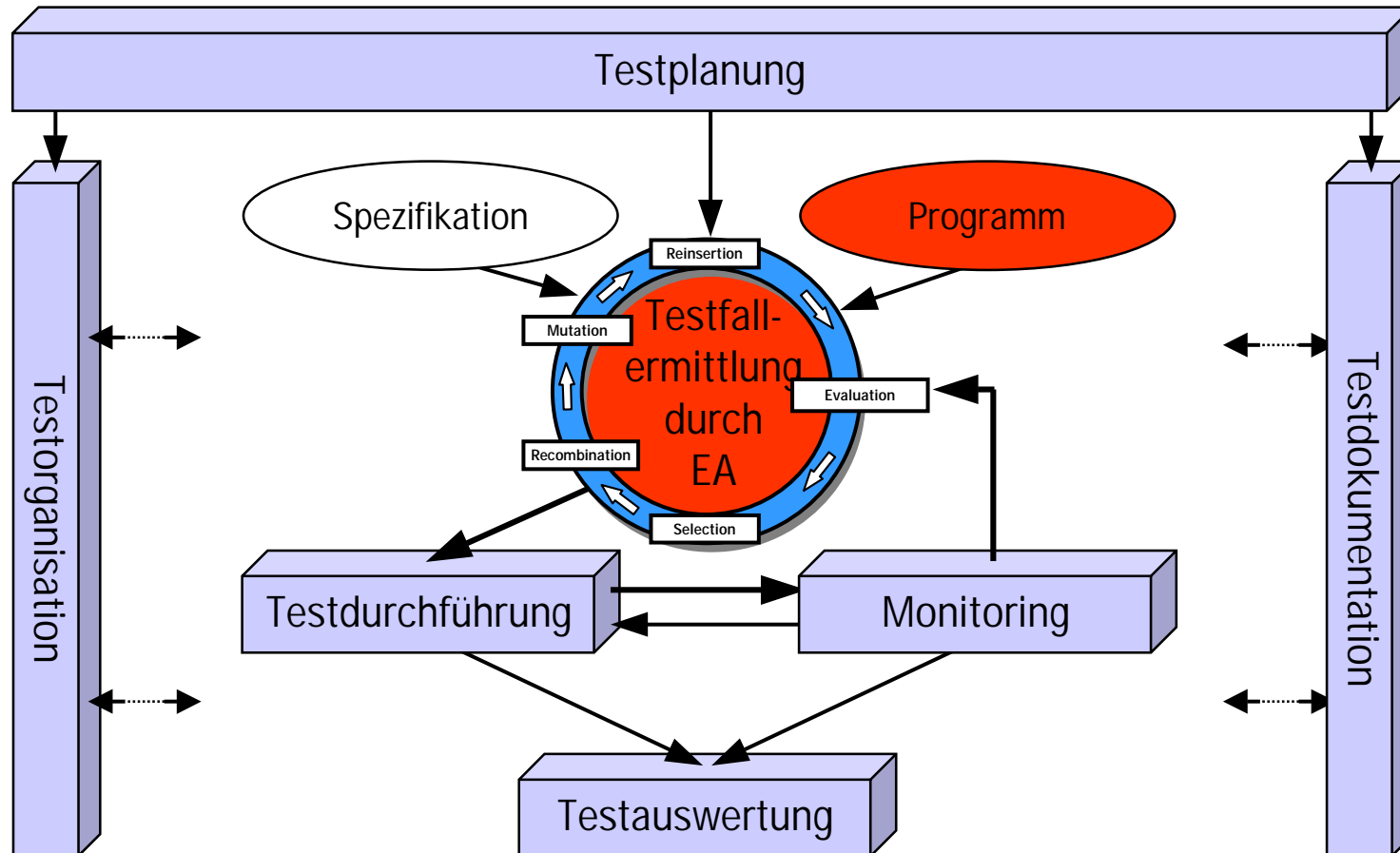


- Alle Pfade vom Beginn bis Ende des Graphen sind Testziele

Evolutionärer Strukturtest

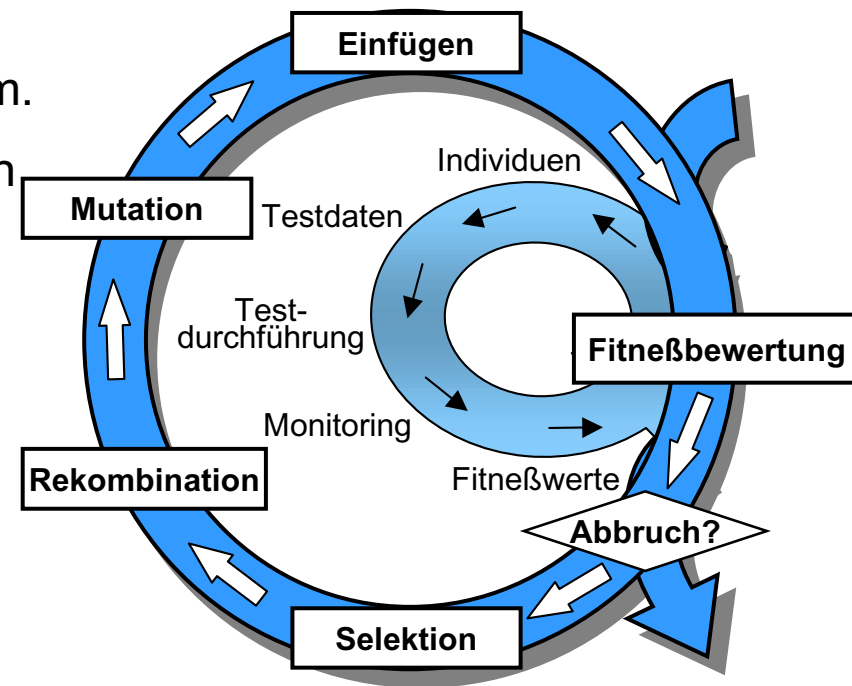


Evolutionärer Strukturtest

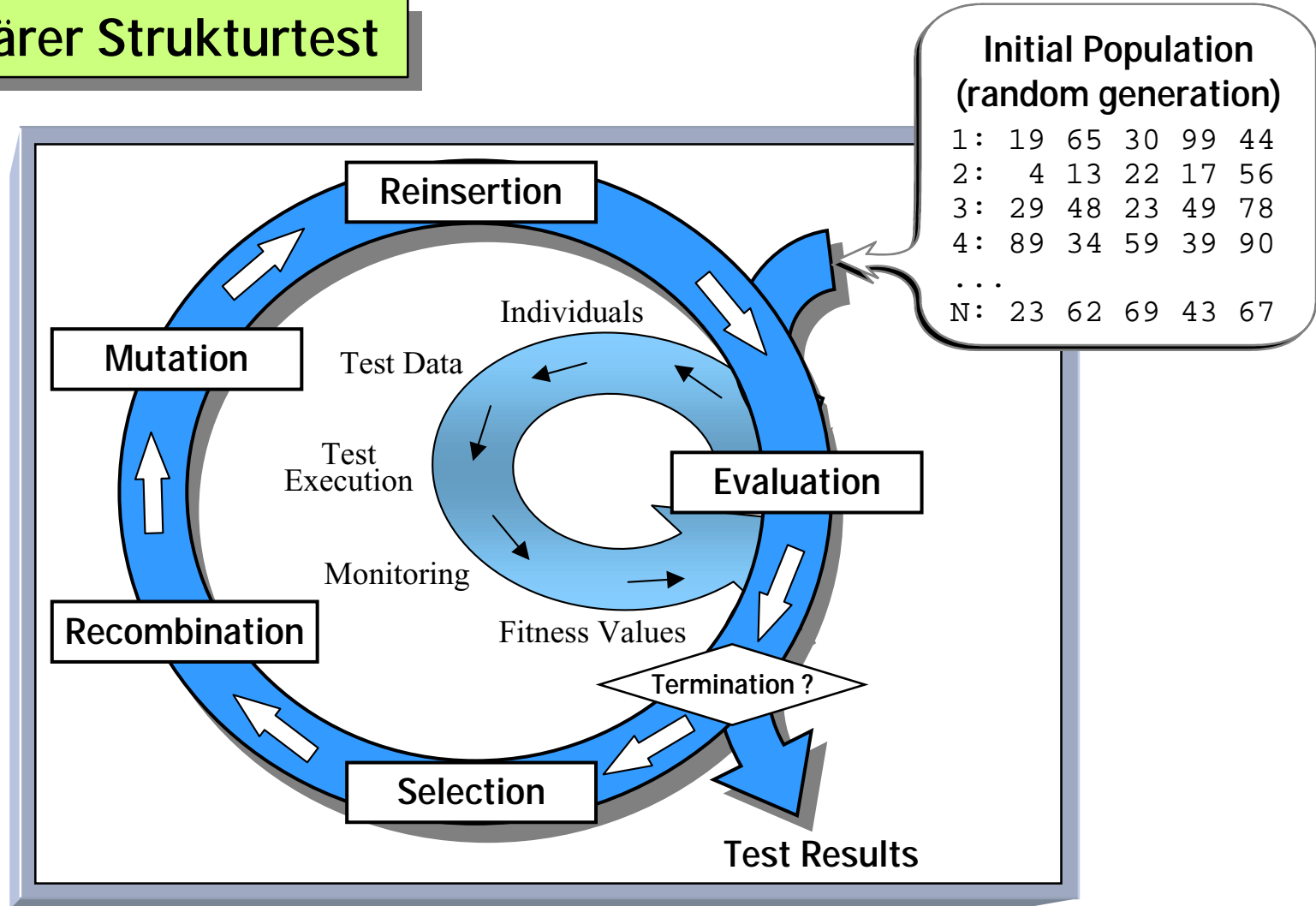


Evolutionärer Strukturtest

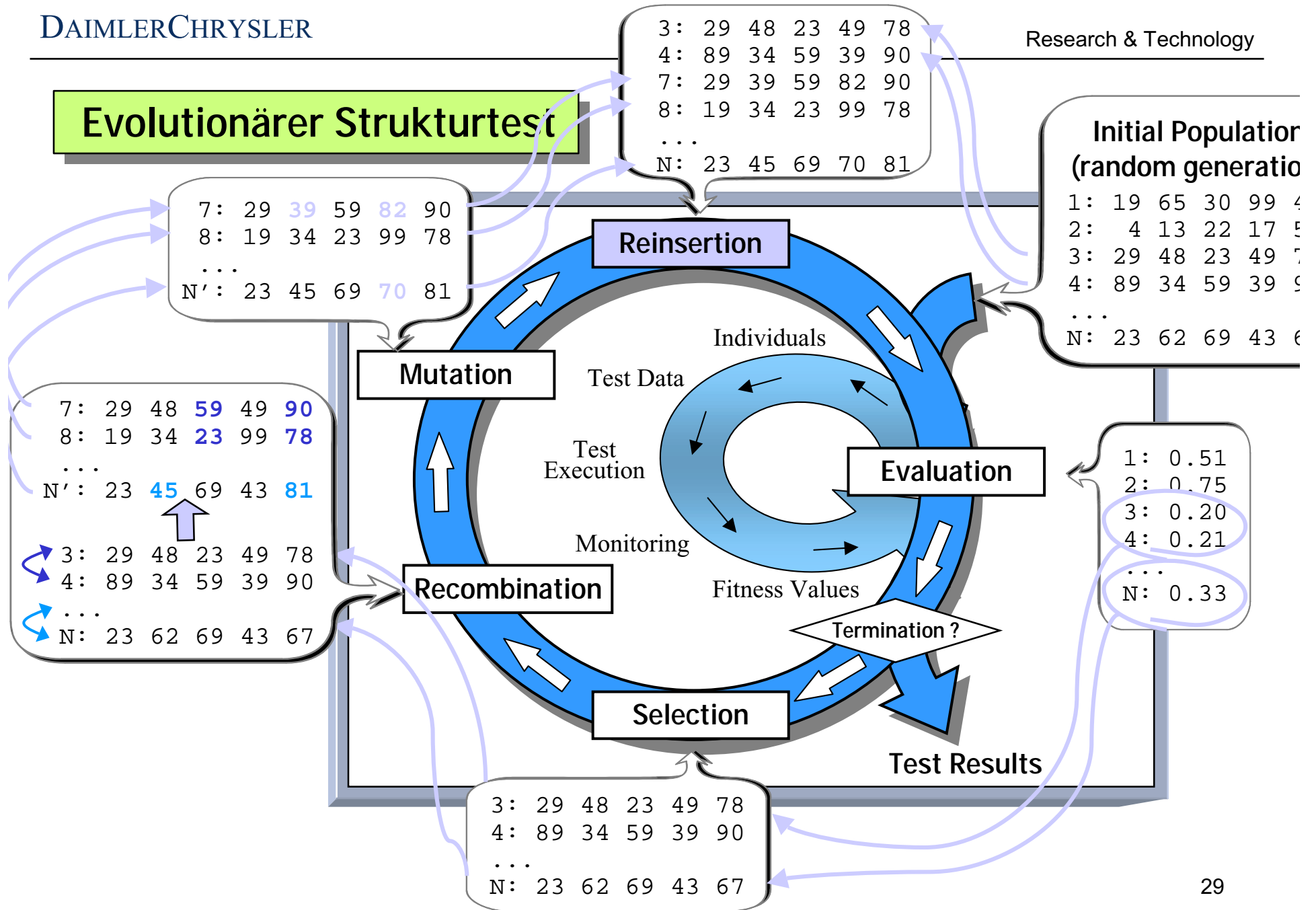
- Neuartiges Testverfahren für systematische und vollautomatische Generierung von Testfällen und Testdaten durch leistungsfähige Suchverfahren, z.B. evolutionäre Algorithmen.
- Testziel wird numerisch formuliert und in Optimierungsproblem übersetzt (Fitneßfunktion).
- Eingabedatenraum bildet den Suchraum.
- Eingangsparameter bilden die Variablen für die Optimierung.
- Generierte Eingabesituationen werden anhand Ablaufüberwachung (Monitoring) hinsichtlich Testziel beurteilt. Vielversprechende Eingabesituationen werden kombiniert.
- Iterativer Prozeß



Evolutionärer Strukturtest



Evolutionärer Strukturtest



Evolutionärer Strukturtest

Komponenten

Testautomatisierung:

- GUI,
- Parser,
- Instrumentierer,
- Treibergenerator,
- Teststeuerung,
- Toolbox evolutionärer Algorithmen



	AMT_Ernorg	AMT_Ernorg
True	✓	0.000540450
False	✓	✓

```

if ( ErrEtr_Mode ) {
    Sa24_Switch_R = 0;
} else {
    Sa24_Switch_R = Sa24_Switch_C;
}

/*
Switch: pana20/pana20_ErrEntry/Err
# combined # Saturations: pana20/pa
pana20/pana20_ErrEntry/ErrEntry3/1
pana20/pana20_ErrEntry/ErrEntry3/1
if ( Sa24_Switch_R >= 1 | (
    Sa20_Switch_2 = 2;
) else {
    Sa20_Switch_2 = 0;
}

/*
Switch: pana20/pana20_ErrEntry/Err
Omitted comparison with constant.
if ( !AMT_Mode ) {

/* # combined # Unit delay: pana20/
X_Sa31_Memory_C = 0;
} else {

/*
# combined # Switch: pana20/pa
Omitted comparison with constant
if ( AMT_Mode == Sa31_BelOp_C ) {

/*
# combined # Unit delay: pana20/

```