

Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2017 / Frühjahr

## Thema 2 / Teilaufgabe 2 / Aufgabe 1

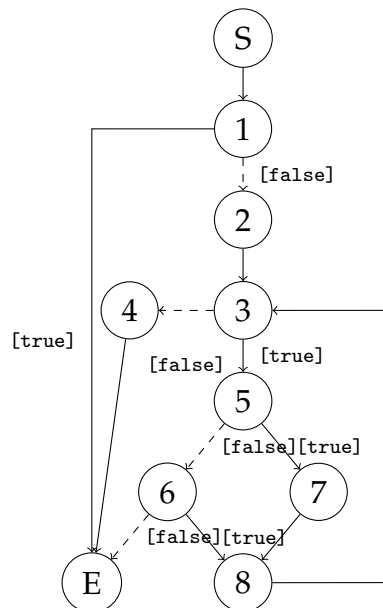
(Methode „binToInt()“ und

Kontrollflussgraph)

**Stichwörter:** Kontrollflussgraph, Zyklomatische Komplexität nach McCabe, Datenfluss-annotierter Kontrollflussgraph

Gegeben Sei folgende Methode und ihr Kontrollflussgraph:

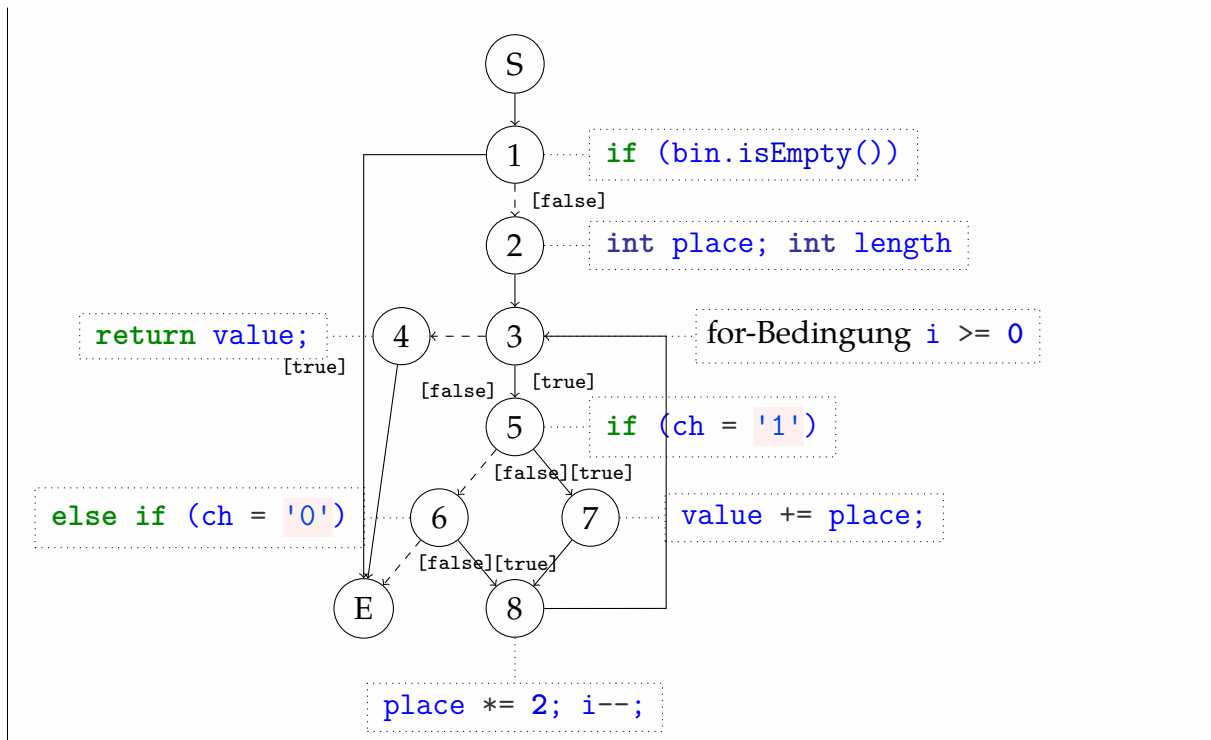
```
int binToInt(String bin) {
    if (bin.isEmpty())
        return -1;
    int place = 1, value = 0;
    int length = bin.length() -
    ↪ 1;
    for (int i = length; i >= 0;
    ↪ --i) {
        char ch = bin.charAt(i);
        if (ch == '1') {
            value += place;
        } else if (ch == '0') {
            // do nothing
        } else {
            return -1;
        }
        place *= 2;
    }
    ↪ return value;
}
```



Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/sosy/ab\\_7/Aufgabe5.java](https://github.com/bschlangaul/aufgaben/sosy/ab_7/Aufgabe5.java)

- (a) Geben Sie je einen Repräsentanten aller Pfadklassen im Kontrollflussgraphen an, die zum Erzielen einer vollständigen

Lösungsvorschlag



(i) Verzweigungsüberdeckung

Lösungsvorschlag

p1 (Pfad 1) S 1 E  
 p2 S 1 2 3 4 E  
 p3 S 1 2 3 5 7 8 3 5 6 8 3 5 6 E

(ii) Schleife-Inneres-Überdeckung

Lösungsvorschlag

**Äußere Pfade (äußere Pfade):** (ohne Ausführung der Wiederholung)

p1 S 1 E  
 p2 S 1 2 3 4 E

**Grenzpfade (boundary test)** (alle Pfade, die die Wiederholung betreten, aber nicht wiederholen; innerhalb des Schleifenrumpfes alle Pfade!)

p4 S 1 2 3 5 6 E

**Innere Pfade (interior test)** (alle Pfade mit einer Wiederholung des Schleifenrumpfes; innerhalb des Schleifenrumpfes wieder alle Pfade!)

p5 S 1 2 3 5 7 8 3 5 7 8 3 4 E  
 p6 S 1 2 3 5 7 8 3 5 6 8 3 4 E  
 p7 S 1 2 3 5 6 8 3 5 6 8 3 4 E  
 p8 S 1 2 3 5 6 8 3 5 7 8 3 4 E  
 p9 S 1 2 3 5 7 8 3 5 7 8 3 5 6 E  
 p10 = p3 S 1 2 3 5 7 8 3 5 6 8 3 5 6 E

**p11** S 1 2 3 5 6 8 3 5 6 8 3 5 6 E

**p12** S 1 2 3 5 6 8 3 5 7 8 3 5 6 E

mit minimaler Testfallanzahl genügen würden.

- (b) Welche der vorangehend ermittelten Pfade sind mittels Testfälle tatsächlich überdeckbar („feasible“)? Falls der Pfad ausführbar ist, geben Sie bitte den Testfall an, andernfalls begründen Sie kurz, weshalb der Pfad nicht überdeckbar ist.

Erweitere Methode, die die Knotennamen ausgibt:

```
public static final String RESET = "\u001B[0m";
public static final String ROT = "\u001B[31m";
public static final String GRÜN = "\u001B[32m";

static int binToIntLog(String bin) {
    System.out.println("\nInput: " + bin);
    System.out.print("S");
    System.out.print(1);
    if (bin.isEmpty()) {
        System.out.print("E");
        System.out.println("\nOutput: " + -1);
        return -1;
    }
    System.out.print(2);
    int place = 1, value = 0;
    int length = bin.length() - 1;
    System.out.print(3);
    for (int i = length; i >= 0; --i) {
        char ch = bin.charAt(i);
        System.out.print(5);
        if (ch == '1') {
            System.out.print(ROT + 7 + RESET);
            value += place;
        } else {
            System.out.print(GRÜN + 6 + RESET);
            if (ch == '0') {
                // do nothing
            } else {
                System.out.print("E");
                System.out.println("\nOutput: " + -1);
                return -1;
            }
        }
    }
    System.out.print(8);
    place *= 2;
    System.out.print(3);
}
System.out.print(4);
System.out.print("E");
System.out.println("\nOutput: " + value);
return value;
}
```

```

public static void main(String[] args) {
    binToIntLog(""); // p1
    binToIntLog("??"); // p2 not feasible
    binToIntLog("x01"); // p3
    binToIntLog("x"); // p4

    binToIntLog("11"); // p5
    binToIntLog("01"); // p6
    binToIntLog("00"); // p7
    binToIntLog("10"); // p8

    binToIntLog("x11"); // p9
    binToIntLog("x01"); // p10
    binToIntLog("x00"); // p11
    binToIntLog("x10"); // p12
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/sosy/ab\\_7/Aufgabe5.java](https://github.com/src/main/java/org/bschlangaul/aufgaben/sosy/ab_7/Aufgabe5.java)

Alle mit Ausnahme von p2.

p2 ist nicht überdeckbar. Passiert ein Wert der Variable `bin` die erste if-Verzweigung, dann hat der Wert eine Länge größer 0 und betritt deshalb die Wiederholung mit fester Anzahl.

p1	S 1 E	<code>binToInt("");</code>
p2	S 1 2 3 4 E	not feasible
p3	S 1 2 3 5 7 8 3 5 6 8 3 5 6 E	<code>binToInt("x01");</code>
p4	S 1 2 3 5 6 E	<code>binToInt("x");</code>
p5	S 1 2 3 5 7 8 3 5 7 8 3 4 E	<code>binToInt("11");</code>
p6	S 1 2 3 5 7 8 3 5 6 8 3 4 E	<code>binToInt("01");</code>
p7	S 1 2 3 5 6 8 3 5 6 8 3 4 E	<code>binToInt("00");</code>
p8	S 1 2 3 5 6 8 3 5 7 8 3 4 E	<code>binToInt("10");</code>
p9	S 1 2 3 5 7 8 3 5 7 8 3 5 6 E	<code>binToInt("x11");</code>
p10 = p3	S 1 2 3 5 7 8 3 5 6 8 3 5 6 E	<code>binToInt("x01");</code>
p11	S 1 2 3 5 6 8 3 5 6 8 3 5 6 E	<code>binToInt("x00");</code>
p12	S 1 2 3 5 6 8 3 5 7 8 3 5 6 E	<code>binToInt("x10");</code>

- (c) Bestimmen Sie anhand des Kontrollflussgraphen die maximale Anzahl linear unabhängiger Programmpfade, also die zyklomatische Komplexität nach Mc-Cabe.

Lösungsvorschlag

**Binärverzweigungen** 4

**Knoten** 10

**Kanten** 13

Anhand der Binärverzweigungen:

$$\begin{aligned}
 M &= b + p \\
 &= 4 + 1 \\
 &= 5
 \end{aligned}$$

oder durch Anzahl Kanten  $e$  und Knoten  $n$

$$\begin{aligned}
 M &= e - n + 2p \\
 &= 13 - 10 + 2 \cdot 1 \\
 &= 5
 \end{aligned}$$

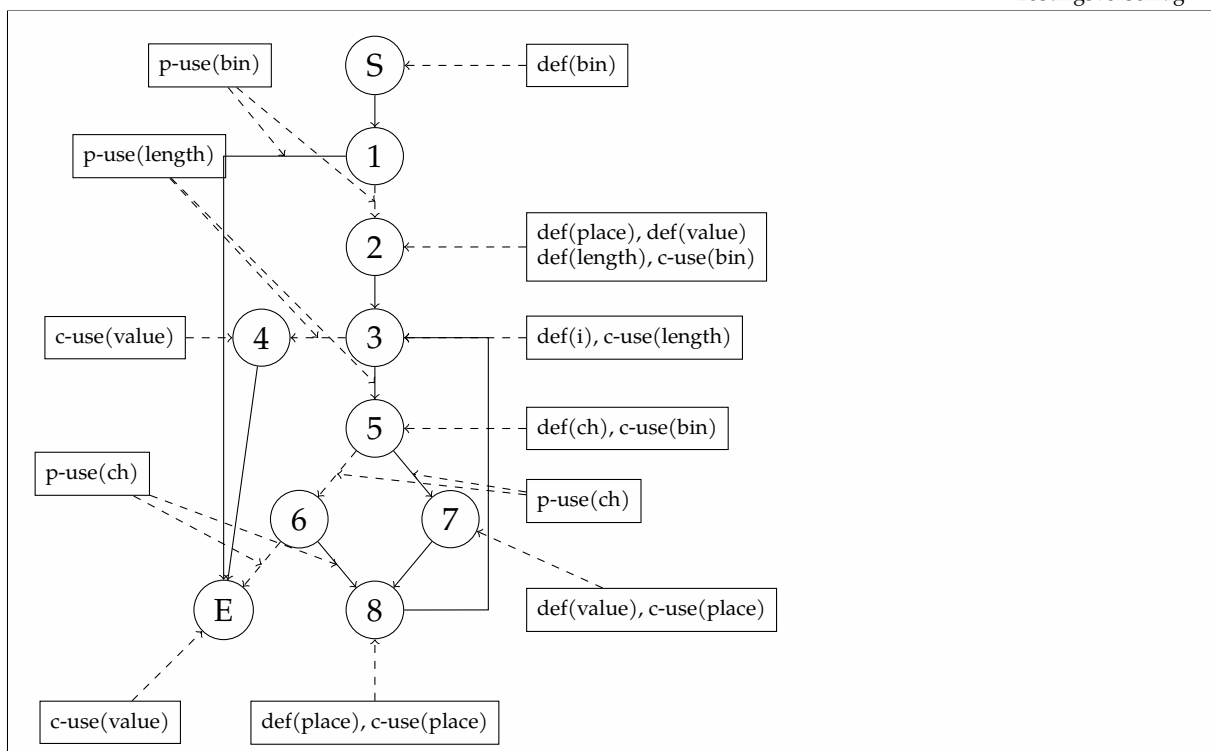
- (d) Kann für dieses Modul eine 100%-ige Pfadüberdeckung erzielt werden? Begründen Sie kurz Ihre Antwort.

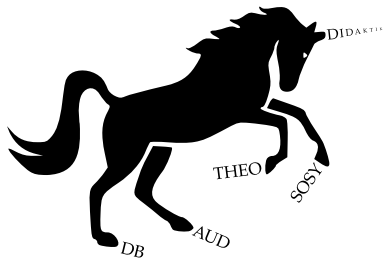
Lösungsvorschlag

Nein, da **p2** nicht überdeckbar ist.

- (e) Geben Sie zu jedem Knoten die jeweilige Datenfluss-Annotation (defs bzw. uses) für jede betroffene Variable in der zeitlichen Reihenfolge ihres Auftretens zur Laufzeit an.

Lösungsvorschlag





## Die Bsclangaul-Sammlung

### Hermine Bsclangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bsclangaul@gmx.net](mailto:hermine.bsclangaul@gmx.net). Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bsclangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/66116/2017/03/Thema-2/Teilaufgabe-2/Aufgabe-1.tex>