

# Übung Signalflussdiagramm Teil 4

Im vierten Teil der Aufgabenstellung wollen wir nun die Berechnung einbauen. Die Implementation der Regel nach Mason geschieht in der Methode *berechnung()* der Klasse Modell. Dazu wurden zuvor im Teil 3 die zum Signalflussdiagramm zugehörigen Daten mittels folgender Methoden in entsprechenden Attributen abgelegt:

```
public void setNetlist(String[] netlist) {
    trace.methodeCall();
    for (int k = 0; k < netlist.length; k++) {
        String[] item = netlist[k].split("[{}]+");
        String branchID = item[0].split("[\\t ]+")[0];
        branchMap.put(branchID, new RationalFunction(new Polynom(item[1]),
            new Polynom(item[2])));
    }
}

public void setSignalflow(String[] signalflow) {
    trace.methodeCall();
    fwdPath = new String[(signalflow.length - 1) / 2];
    kofaktor = new String[(signalflow.length - 1) / 2];
    for (int k = 0; k < signalflow.length - 1; k += 2) {
        fwdPath[k] = signalflow[k];
        kofaktor[k] = signalflow[k + 1];
    }
    determinante = signalflow[signalflow.length - 1];
}

public void setLoops(String[] loops) {
    trace.methodeCall();
    this.loops = loops;
    System.out.println(loops);
}
```

Die Attribute *fwdPath*, *Kofaktor* und *determinante* sind Zeichenketten mit den zugehörigen Daten. *branchMap* ist eine HashMap mit Schlüssel *branchID* und zugehöriger gebrochenrationaler Funktion.

## Regel von Mason

$$H(s) = \frac{U_{out}(s)}{U_{in}(s)} = \frac{\sum_{k=1}^N P_k(s) \cdot \Delta_k(s)}{\Delta(s)}$$

Wobei alle Funktionen gebrochenrationale Funktionen in *s* sind. Für die Berechnung gilt folgendes:

- $P_k(s)$  ist der  $k^{te}$  Vorwärtspfad und ergibt sich aus dem Produkt aller Zweige  $b_i$  entlang des Pfades  $P_k$ .
- $\Delta(s)$  ist die Determinante des Graphen und gegeben durch:

$$\Delta(s) = 1 - \sum L_i + \sum L_i L_j - \sum L_i L_j L_k + \dots$$

wobei nur Kreise, die sich gegenseitig nicht berühren, zu berücksichtigen sind.

- $\Delta_k(s)$  ist der Kofaktor des  $k^{ten}$  Vorwärtspfades und wird gleich wie die Determinante berechnet, wobei nur Kreise, die den  $k^{ten}$  Vorwärtspfad nicht berühren, zu berücksichtigen sind.

### Aufgabe 1: Berechnung der Loops

Die Berechnung der Loops geschieht mit folgendem Code:

```
for (int k = 0; k < loops.length; k++) {
    RationalFunction loopFun = new RationalFunction(new Polynom(1.0), new Polynom(1.0));
    String[] items = loops[k].split("[-\\t ]+");
    String loopID = items[0];
    for (int i = 1; i < items.length - 1; i += 2) {
        loopFun = loopFun.multiply(branchMap.get(items[i + 1]));
    }
    loopMap.put(loopID, loopFun);
}
```

- a) Analysieren Sie den Code mittels des Debuggers. Welche Werte/Attribute werden verwendet resp. stellen sich ein?

### Aufgabe 2: Berechnung der Vorwärtspfade

Die Vorwärtspfade sind gegeben durch:

$P_k(s)$  ist der  $k^{te}$  Vorwärtspfad und ergibt sich aus dem Produkt aller Zweige  $b_i$  entlang des Pfades  $P_k$ .

- a) Berechnen Sie sinngemäss zu den Loops die Vorwärtspfade.  
b) Analysieren Sie wiederum den Code mittels Debugger. Welche Werte/Attribute werden verwendet resp. stellen sich ein?

### Aufgabe 3: Berechnung der Determinante

Die Determinante ist gegeben durch:

- $P_k(s)$  ist der  $k^{te}$  Vorwärtspfad und ergibt sich aus dem Produkt aller Zweige  $b_i$  entlang des Pfades  $P_k$ .
- a) Berechnen Sie sinngemäss Determinante.  
b) Analysieren Sie wiederum den Code mittels Debugger. Welche Werte/Attribute werden verwendet resp. stellen sich ein?

### Aufgabe 4: Berechnung der Kofaktoren

Die Kofaktoren sind gegeben durch:

- $\Delta_k(s)$  ist der Kofaktor des  $k^{ten}$  Vorwärtspfades und wird gleich wie die Determinante berechnet, wobei nur Kreise, die den  $k^{ten}$  Vorwärtspfad nicht berühren, zu berücksichtigen sind.

$$\Delta(s) = 1 - \sum L_i + \sum L_i L_j - \sum L_i L_j L_k + \dots$$

wobei nur Kreise, die sich gegenseitig nicht berühren, zu berücksichtigen sind.

- a) Berechnen Sie sinngemäss die Kofaktoren.  
b) Analysieren Sie wiederum den Code mittels Debugger. Welche Werte/Attribute werden verwendet resp. stellen sich ein?

### **Aufgabe 5: Berechnung der Übertragungsfunktion**

Die Übertragungsfunktion ist gegeben durch:

$$H(s) = \frac{U_{out}(s)}{U_{in}(s)} = \frac{\sum_{k=1}^N P_k(s) \cdot \Delta_k(s)}{\Delta(s)}$$

- a) Berechnen Sie die Übertragungsfunktion.
- b) Analysieren Sie wiederum den Code mittels Debugger. Welche Werte/Attribute werden verwendet resp. stellen sich ein?

Bye, bye Java – Übungen ...