Bachelorarbeit

# Automatische Programmanalyse

Thomas Leichtle

Universität Ulm
Institut für theoretische Informatik
89069 Ulm, Deutschland
Thomas.Leichtle@uni-ulm.de

Betreuer:

Prof. Dr. Schöning

4. Februar 2012

# Inhaltsverzeichnis

# Kapitel 1

# Einleitung

In dieser Bachelorarbeit geht es darum, mit Hilfe einer Maske, einfache Programme zu entwerfen, die mit einer beschränkten Menge an Operationen auskommen. Diese Programme sollen dann automatisch analysiert werden. Das bedeutet, dass für eine Teilmenge (oder alle) der benutzten Register geschlossene Formeln für ihre Werte nach dem n-ten Schleifendurchlauf gefunden werden sollen.
Hiermit sollen für Zahlenfolgen, wie z.B. die Fibonacci-Zahlen, geschlossene Formeln gefunden werden. Um einen überschaubaren Rahmen einzuhalten und dass einfachere Lösungswege ausreichen, beschränken wir uns auf folgende Operationen:

1. Addition/Subtraktion konstanter Faktoren
2. Multiplikation mit konstantem Faktor
3. Addition/Subtraktion zweier Register

Das Vorgehen wird sein, dass zuerst für alle Register die korrespondierenden Gleichungen im z-Bereich aufgestellt werden. Anschließend werden diese aufgelöst und mit Hilfe einer Korrespondenztabelle zurück in den Zeitbereich transformiert.

## 1.1 Maske

```
/* INITIALISIERUNGSBLOCK */
while (true) {
    /* OPERATIONSBLOCK */
}
```

Abbildung 1.1: vorgegebene Eingabemaske

Die Maske (Abb. 1.1) besteht aus drei Teilen: Einem Initialisierungsblock, einer while-Schleife und einem Operationsblock im Schleifenkörper.

Im Initialisierungsblock sind ausschließlich Initialisierungen erlaubt, d.h. die Zuweisung einer festen Zahl zu einem Register. Das mehrfache Setzen eines Startwerts für ein Register ist nicht sinnvoll und kann durch die Letzte der Initialisierungen ersetzt werden. Jedes Register, das an beliebiger Stelle im Programm verwendet wird, muss hier gesetzt werden.
Der zweite Teil der Maske ist die while-Schleife, welche ein fester Bestandteil ist und nicht manipuliert werden darf. Die Abbruchbedingung ist rein symbolisch gedacht. Somit kann die Schleife als Endlosschleife interpretiert werden, dessen Registerwerte wir nach dem n-ten Durchlauf bestimmen wollen. Äquivalent wäre eine for-Schleife bis n vorstellbar, deren Registerwerte wir nach der Terminierung wissen wollen.
Als zweiter veränderbarer Bereich ist der Operationsblock im Schleifenkörper vorhanden. In diesem können die in der Einleitung aufgeführten Operationen auf den Registern benutzt werden.

## 1.2 Grundlagen

### 1.2.1 Z-Transformation

> Die Z-Transformation wandelt ein zeitdiskretes Signal im Zeitbereich, also eine zeitliche Abfolge von im Allgemeinen komplexen Zahlen, in ein komplexes diskretes Signal im Frequenzbereich um. [Wikc]

Da nur kausale Systeme betrachtet werden, verwenden wir die Unilaterale Z-Transformation:

$$F(z) = Z\{f[n]\} = \sum_{n=0}^{\infty} f[n] z^{-n}$$

Es gelten folgende Eigenschaften (vgl. [Mee]):

- Linearität: $Z\{a_1 x_1[n] + a_2 x_2[n]\} = a_1 Z\{x_1[n]\} + a_2 Z\{x_2[n]\}$
- Verschiebung: $Z\{x[n-k]\} = z^{-k} Z\{x[n]\}$ für $k \geq 0$
- Verschiebung: $Z\{x[n+k]\} = z^k Z\{x[n]\} - \sum_{i=0}^{k-1} z^{k-i} x[i]$ für $k \geq 0$
- Lineare Gewichtung: $k \cdot x[k] = -z \frac{d}{dz} X(z)$
- Anfangswertsatz: $x[0] = \lim_{z \to \infty} X(z)$
- Endwertsatz: $\lim_{k \to \infty} x[k] = \lim_{z \to 1+} (z-1) X(z)$

### 1.2.2 Partialbruchzerlegung (PBZ)

Wie aus der Analysis bekannt ist, lässt sich jede rationale Funktion $R'(x) = \frac{Z'(x)}{N'(x)}$ als endliche Summe von sog. Partialbrüchen (und einem Polynom P, falls Grad(Z') $\geq$ Grad(N') ist) schreiben:

$$R'(x) = \sum_i p_i x^i + \sum_i \sum_{j=1}^{n_i} \frac{r_{i,j}}{(x-\alpha_i)^j}$$

mit $\alpha_i$ Nullstelle von $N(x)$ und $n_i$ die zugehörige Vielfachheit der Nullstelle. Diese Aussage trifft jedoch nur über dem Körper der komplexen Zahlen zu. Möchte man komplexe Zahlen vermeiden, so kann man sich zunutze machen, dass zu jeder komplexen Nullstelle jeweils auch die konjungiert komplexe Nullstelle enthalten sein muss, unter der Voraussetzung, dass die Koeffizienten des Nennerpolynoms reell sind. Man erhält dann die Form:

$$R'(x) = \sum_i g_i x^i + \sum_i \sum_{j=1}^{n_i} \frac{r_{i,j}}{(x-\alpha_i)^j} + \sum_i \sum_{j=1}^{\tilde{n}_i} \frac{s_{i,j}kx + t_{i,j}}{(x^2+\beta_{i,j}x+\gamma_{i,j})^j}$$

Will man nun eine Partialbruchzerlegung durchführen, dann geht man folgenderweise vor (vgl. [Bosb]):

1. Falls Grad(Z') $\geq$ Grad(N'): Abspalten eines Polynoms P', z.B. mittels Polynomdivision. Damit folgt: $R'(x) = \frac{Z'(x)}{N'(x)} = P'(x) + \frac{Z(x)}{N(x)} = P'(x) + R(x)$, mit Grad(Z) < Grad(N).

2. Faktorisierung von Z(x) und N(x) in Produkte der Nullstellen und Kürzen gemeinsamer Faktoren. Dies ist nicht zwingend erforderlich, d.h. bei schwer zu zerlegendem Zählerpolynom evtl. diesen Schritt überspringen.

3. Ansatz der Partialbruchsumme:
   Für jede $n$-fache Nullstelle des Nennerpolynoms $\alpha_i$: $\sum_{j=1}^{n_i} \frac{A_{i,j}}{(x-\alpha_i)^j}$

4. Bestimmen der Konstanten: $A_{i,j}$. Dazu gibt es zwei Möglichkeiten:

   a) Bringt man die ganze Gleichung auf den Hauptnenner $N(x)$ und führt einen Koeffizientenvergleich durch, so erhält man ein Gleichungssystem für die Konstanten.

   b) Für einfache Nullstellen funktioniert auch oftmals die *Zuhaltemethode*. Dabei multipliziert man für eine einfache Nullstelle $\alpha_i$ die gesamte Gleichung mit $(x-\alpha_i)$, kürzt soweit möglich und setzt anschließend $x = \alpha_i$ ein. Es fallen alle Konstanten bis auf diejenige für diese Stelle weg und man erhält die Lösung für diese.
   Formal: $A_i = (x-\alpha_i)\frac{Z(x)}{N(x)}\Big|_{x=\alpha_i}$
   Für $n$-fache Nullstelle $\alpha_i$ gilt: $A_{i,j} = \frac{1}{(n-j)!}\left(\frac{d^{n-j}}{dx^{n-j}}(x-\alpha_i)^n \frac{Z(x)}{N(x)}\right)\Big|_{x=\alpha_i}$

## 1.3 Hilfsmittel

Für jedes Register benötigen wir eine zugehörige Funktion $F(z)$ im z-Bereich. Die zu $R_i$ gehörende Funktion nennen wir $F_i(z)$. Jede der Funktionen kann dabei potentiell von allen Funktionen mit beliebiger Verzögerung abhängen. Wir werden zu Beginn feststellen, dass nach dem 1. Analyseschritt eine maximale Verzögerung von einem Schleifendurchlauf auftritt, jedoch kann diese sich in der weiteren Analyse erhöhen.

### 1.3.1 Registermatrix

Um das Aufstellen der Gleichungen im z-Bereich zu erleichtern, wird für jede eine Matrix verwendet. Diese wird im weiteren als zugehörige Funktions- oder Registermatrix bezeichnet.

$$\left(\begin{array}{cccc|cc} x_{0,0} & x_{0,1} & \dots & x_{0,n-2} & x_{0,n-1} & x_{0,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_{m,0} & x_{m,1} & \dots & x_{m,n-2} & x_{m,n-1} & x_{m,n} \end{array}\right)$$

mit $n-2 =$ Anzahl der verwendeten Register und $m =$ höchster relativer Verzögerungszeit. Auf $m$ wird später noch genauer eingegangen. Der Wert $x_{i,j}$ für $j \leq n-2$ entspricht dabei dem Vorkommen der Funktion $F_j(z)$ mit einer Verzögerung von $i$, was im z-Bereich wiederum mit einem zugehörigen Faktor von $z^{-i}$ in der Gleichung korreliert. Für $j = n-1$ entspricht $x_{i,j}$, $i \geq 1$ dem Faktor vor $\frac{z}{z-1}z^{-i}$ und für $j = n$ dem Faktor zu $z^{-i}$.
Anmerkung: Die erste Zeile (ohne Verzögerung) ist nur vom Initialisierungsblock abhängig.

Möchte man die Rekursionsgleichung (bereits aufgelöst nach $F_k$) aus der Matrix ablesen, so kann man folgende Formel verwenden:

$$\begin{aligned} F_k(z) &= \sum_{i=0}^{m}\Big(\sum_{\substack{j=0 \\ j\neq k}}^{n-2}\Big(\frac{x_{i,j}F_j(z)z^{m-i}}{\sum_{l=0}^{m} -x_{l,k}z^{m-l}}\Big)\Big) + \frac{\sum_{i=0}^{m}\frac{zx_{i,n-1}}{z-1}z^{m-i}}{\sum_{l=0}^{m} -x_{l,k}z^{m-l}} + \frac{\sum_{i=0}^{m} x_{i,n}z^{m-i}}{\sum_{l=0}^{m} -x_{l,k}z^{m-l}} \\ \Leftrightarrow F_k(z) &= \frac{1}{\sum_{l=0}^{m} -x_{l,k}z^{m-l}}\Big(\sum_{i=0}^{m}\sum_{\substack{j=0 \\ j\neq k}}^{n-2} x_{i,j}F_j(z)z^{m-i} + \sum_{i=0}^{m}\frac{x_{i,n-1}}{z-1}z^{m-i+1} + \sum_{i=0}^{m} x_{i,n}z^{m-i}\end{aligned}$$

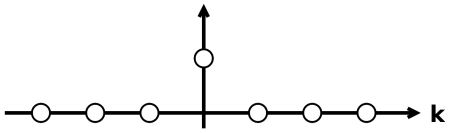
### 1.3.2 Korrespondenztabelle

Für die Rücktransformation aus dem z-Bereich in den Zeitbereich benutzen wir Korrespondenzen. Diese sind zusammen mit den Eigenschaften der z-Transformation sehr mächtige und effektive Hilfsmittel.

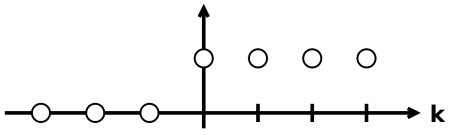
| # | $X(z)$ | $x(k)$ |
|---|---|---|
| (1) | $1$ | $\delta(k)$ |
| (2) | $\frac{z}{z-1}$ | $\epsilon(k)$ |
| (3) | $\frac{z}{(z-1)^2}$ | $k\epsilon(k)$ |
| (4) | $\frac{z}{z-a}$ | $a^k\epsilon(k)$ |
| (5) | $\frac{az}{(z-a)^2}$ | $ka^k\epsilon(k)$ |
| (6) | $\frac{g}{(z-a)^t}$ | $ga^{k-t}q_{k-t}\epsilon(k-t)$ mit $q_n = (1+\frac{n}{1})(1+\frac{n}{2})\dots(1+\frac{n}{t-1}) = \binom{k-1}{t-1}$ |
| (7) | $\frac{gz^n}{z^n-a}$ | $(ga^0, 0, 0, \dots, ga^1, 0, 0, \dots)$ mit je $n-1$ Nullen dazwischen |

Abbildung 1.2: Korrespondenztabelle [Bosa] [I. 03] [Bra]

Diracimpuls: $\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & \text{sonst} \end{cases}$




Sprungfunktion: $\epsilon(k) = \begin{cases} 1 & k \geq 0 \\ 0 & \text{sonst} \end{cases}$




**Beweis der Korrespondenz (6) - Partialbruchregel** (siehe [Bra])

**Schritt 1:** Zunächst geht man von einer Klasse L von Folgen aus, die die Form haben:

$$f = (\underbrace{0, 0, \dots,}_{t} c_0a^0, c_1a^1, c_2a^2, \dots)$$

Die explizite Darstellung dieser Folgen lautet: $f_n = \begin{pmatrix} c_{n-t}a^{n-t} & n \geq t \\ 0 & \text{sonst} \end{pmatrix}$

Die Z-Transformierte einer solchen Folge lautet daher $F(z) = \sum_{k=t}^{\infty} c_{k-t}a^{k-t}z^{-k}$

Daraus kann man nun die Ableitungen der Z-Transformierten bestimmen.

Die ersten Ableitungen lauten:

$$\begin{aligned}
\frac{d^1}{dz^1}F(z) &= \sum_{k=t}^{\infty} c_{k-t}a^{k-t}(-k)z^{-k-1} \\
\frac{d^2}{dz^2}F(z) &= \sum_{k=t}^{\infty} c_{k-t}a^{k-t}(-k)(-k-1)z^{-k-2} \\
\frac{d^3}{dz^3}F(z) &= \sum_{k=t}^{\infty} c_{k-t}a^{k-t}(-k)(-k-1)(-k-2)z^{-k-3}
\end{aligned}$$

Für die $p$-te Ableitung erhält man allgemeiner:

$$\frac{d^p}{dz^p}F(z) = \sum_{k=t}^{\infty} c_{k-t}a^{k-t}(-k)(-k-1)\dots(-k-p+1)z^{-k-p}$$

Aus diesem Ausdruck kann man nun alle Faktoren $(-1)$ jedes Summanden ausklammern und man erhält dann:

$$\frac{d^p}{dz^p}F(z) = (-1)^p \sum_{k=t}^{\infty} c_{k-t}a^{k-t}(k)(k+1)\dots(k+p-1)z^{-(k+p)}$$

Um die so hergeleitete Summe wieder in die Normalform einer Z-Transformierten zu bringen, verschiebt man jetzt die Laufvariable. Das ergibt:

$$\frac{d^p}{dz^p}F(z) = (-1)^p \sum_{k=t+p}^{\infty} c_{k-p-t}a^{k-p-t}(k-p)(k-p+1)\dots(k-1)z^{-k}$$

Das heißt aber, dass die $p$-te Ableitung der Z-Transformierten einer Folge der Klasse L wiederum die Z-Transformierte einer leicht bestimmbaren Folge ist. Denn aus der vorhergehenden Gleichung geht hervor, dass

$$\frac{1}{(-1)^p}\left(\frac{d^p}{dz^p}F(z)\right) = \sum_{k=0}^{\infty}\frac{g_k}{z^k}$$

$$\text{mit } g_n = \begin{pmatrix} c_{n-p-t}a^{n-p-t}(n-p)(n-p+1)\dots(n-1) & n \geq (t+p) \\ 0 & \text{sonst} \end{pmatrix}$$

Auch diese Folge ist eine Folge der Klasse L.

**Schritt 2:** Nun geht man von einer ganz speziellen Folge der Klasse L aus. Man setzt $t = 1$ und alle $c_n = 1$. Die Z-Transformierte dieser Folge lautet dann: $F_u(z) = \sum\limits_{k=1}^{\infty}\frac{a^{k-1}}{z^k}$. Daraus erhält man aber:

$$F_u(z) = \frac{1}{z}\sum_{k=0}^{\infty}\frac{a^k}{z^k} = \frac{1}{z}\left(\frac{1}{1-\frac{a}{z}}\right) = \frac{1}{z-a}$$

Diese Z-Transformierte kann man leicht differenzieren. Für die $p$-te Ableitung erhält man:

$$\frac{d^p}{dz^p}F_u(z) = (-1)^p p!(z-a)^{-(p+1)}$$

Es wurde ja aber bereits gezeigt, wie man die Z-Transformierte einer Folge der Klasse L verallgemeinert differenziert. Da $F_u$ die Z-Transformierte einer Folge der Klasse L ist, lässt sich die in Schritt 1 gewonnene Formel auch hier anwenden und die beiden Formeln können gleichgesetzt werden:

$$\frac{d^p}{dz^p}F_u(z) = (-1)^p \sum_{k=0}^{\infty} \frac{g_k}{z^k},\ g_n = \begin{pmatrix} c_{n-p-t}a^{n-p-t}(n-p)(n-p+1)\dots(n-1) & n \geq (t+p) \\ 0 & \text{sonst} \end{pmatrix}$$

$$\Leftrightarrow (-1)^p p!(z-a)^{-(p+1)} = (-1)^p \sum_{k=p+1}^{\infty} a^{k-(p+1)}(k-p)(k-p+1)\dots(k-1)z^{-k}$$

$$\Leftrightarrow (z-a)^{-(p+1)} = \frac{1}{p!} \sum_{k=p+1}^{\infty} \frac{a^{k-(p+1)}(k-p)(k-p+1)\dots(k-1)}{z^k}$$

Jetzt setzt man $s = p+1$. Dann ergibt sich:

$$(z-a)^{-s} = \frac{1}{(s-1)!} \sum_{k=s}^{\infty} \frac{a^{k-s}(k-s+1)(k-s+2)\dots(k-s+s-1)}{z^k}$$

Dafür kann man aber auch

$$(z-a)^{-s} = \sum_{k=s}^{\infty} \frac{a^{k-s}q_{k-s}}{z^k},\ q_{k-s} = \frac{(k-s+1)(k-s+2)\dots(k-1)}{(s-1)!}$$

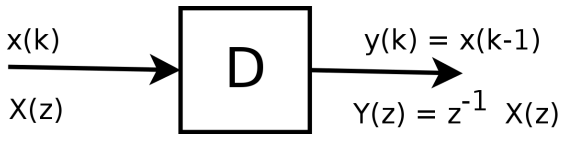
schreiben. Damit ist die Z-Transformierte der Form $(z-a)^{-s}$ aber grundsätzlich mit einer bestimmten Folge der Klasse L lösbar. Dabei ist $t = s$ und die Koeffizienten $c \mathrel{\widehat{=}} q$. Hiermit ist die Rücktransformation bewiesen. Mit der Linearitätsregel kann man nun auch den im Beweis vernachlässigten Koeffizienten $g$ hinzuargumentieren.

Die Korrespondenzen (1) bis (5) können davon abgeleitet werden, indem man die entsprechenden Werte für $g$, $a$, $t$ einsetzt und die Eigenschaft für die Verschiebung verwendet. Zudem gilt: (2) $\Rightarrow$ (3) und (4) $\Rightarrow$ (5) mit der Eigenschaft für lineare Gewichtung.

### 1.3.3 Elemente der Systemtheorie

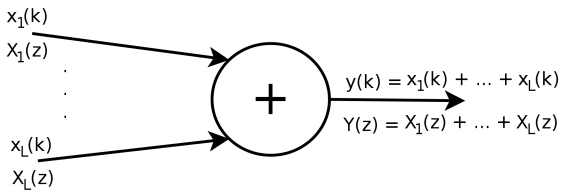
Um die Bezüge zur Systemtheorie herzustellen, in welcher die z-Transformation üblicherweise überwiegend verwendet wird, werden verschiedene Beispiele als Systeme dargestellt. Die wichtigsten Grundelemente dafür sind:

**Verzögerungselement:**



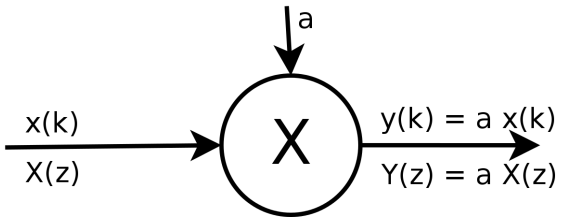

Das Signal wird dabei um einen Taktschritt verzögert, was analog in einem Programm einem Schleifendurchlauf entspricht.

**Addierer:**

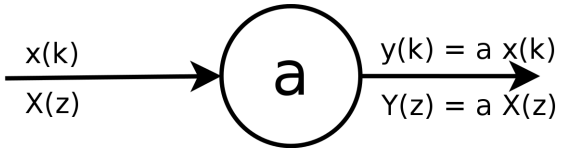



Der Addierer bildet die Summe über beliebig viele Eingangsgrößen. Wegen der Linearität trifft dies sowohl im Zeitbereich als auch im z-Bereich zu.

**Multiplizierer:**

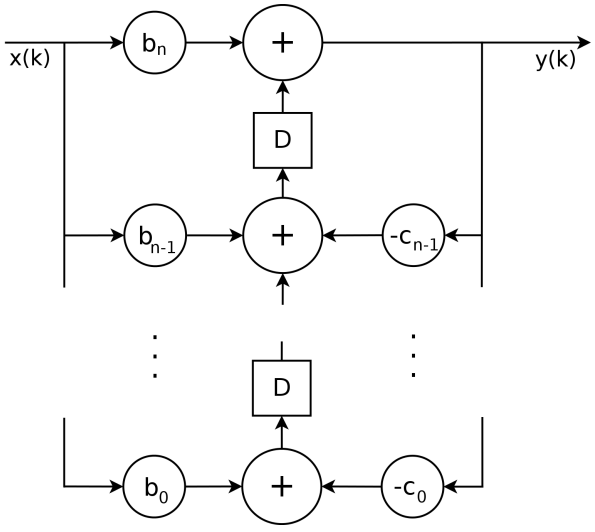



oder auch kürzer:




Multiplizierer werden benötigt, um ein Signal mit einem konstanten Faktor zu multiplizieren. Es wäre auch denkbar, zwei Signale miteinander zu multiplizieren, was hier jedoch aus Linearitätsgründen ausgeschlossen wird.

**IIR (Infinite Impulse Response) - System**




Gegeben sei eine Übertragungsfunktion der Form:

$$H(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0}{z^n + c_{n-1} z^{n-1} + \cdots + c_1 z + c_0} = \frac{\sum_{i=0}^{n} b_i z^i}{z^n + \sum_{i=0}^{n-1} c_i z^i}$$

so kann diese als ein rückgekoppeltes System in der hier gezeigten Form dargestellt werden.
(Bei allen Systemen in dieser Form, in diesem Papier, ist dabei das Eingangssignal $x(k) = \delta(k)$.)

# Kapitel 2

# Analyse

Je nach Quellcode im Operationsblock treten verschiedene Fälle mit unterschiedlich hohem Analyseaufwand auf.

## 2.1 Fall: NOP = keine Operation

Der einfachste, triviale Fall ist ein leerer Operationsblock. Sollte keine Aktion im Schleifenkörper auf dem Register ausgeführt werden, so verbleiben die Werte aus dem Initialisierungsblock als Konstanten in den Registern. Dennoch werden wir diesen Fall normal behandeln!

```
R0 = a;                          1
while (true) {                   2
    // leer                      3
}                                4
```

Abbildung 2.1: Quellcode: keine Operation

Aus Zeile 1 (Initialisierung):
$\rightarrow 0 = -F_0(z) + a \Leftrightarrow \left(\begin{smallmatrix} -1 & | & 0 & a \end{smallmatrix}\right)$
Aus Zeile 2 (Speichern):
$\rightarrow F_0(z) = z^{-1}F_0(z) \Leftrightarrow \left(\begin{smallmatrix} 1 & | & 0 & 0 \end{smallmatrix}\right)$

Abbildung 2.2: Korrespondierende Gleichungen

Aus dem Initialisierungsblock erhalten wir die erste Zeile der Matrix $\left(\begin{smallmatrix} -1 & | & 0 & a \\ 0 & | & 0 & 0 \end{smallmatrix}\right)$. Da die Werte in den Registern gespeichert und nicht nach jedem Schleifendurchlauf verworfen werden

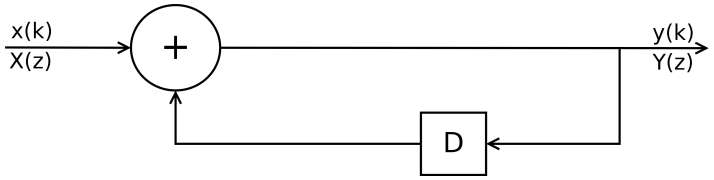



Abbildung 2.3: Systemblockschaltbild zu Quellcode Abb. 2.1 [mit $x(k) = a\ \delta(k)$]

sollen, muss ein fiktives $R0 = R0$ zu Beginn des Operationsblocks eingefügt werden. Dieses wird ab jetzt für alle Register und für alle Fälle als vorhanden angenommen. Somit haben wir nach dem Betrachten des gesamten Codes die Matrix $\left(\begin{smallmatrix}-1 & | & 0 & a\\ 1 & | & 0 & 0\end{smallmatrix}\right)$. Als Gleichung: $-F_0(z) + a + z^{-1}F_0(z) = 0$ oder $F_0(z) = z^{-1}F_0(z) + a$. Löst man diese nach $F_0(z)$ auf, erhält man $F_0(z) = \frac{az}{z-1}$. Diese kann mit Hilfe der Korrespondenztabelle (Korrespondenz (2)) und der Linearitätseigenschaft in den Zeitbereich zurücktransformiert werden und man erhält: $f_0(k) = a\epsilon(k)$. Dies entspricht einem Wert a im Register ab dem 0. Schleifendurchlauf, was mit unserer vorherigen, intuitiven Lösung übereinstimmt.

## 2.2 Fall: Inkrementierung um konstanten Wert

Der nächste Fall ist die Inkrementierung eines Registers um einen konstanten Wert. Intuitiv rechnet man mit einer Lösung der Form $f_0(k) = kb + a$, da zum Startwert $a$ $k$-Mal der Wert $b$ addiert wird. Ebenfalls die Betrachtung über Registermatrix und z-Transformation:

```
R0 = a;                 1
while (true) {          2
    R0 = R0 + b;        3
}                       4
```

Abbildung 2.4: Quellcode: Inkrementierung

Aus Zeile 1 (Initialisierung):
$\rightarrow 0 = -F_0(z) + a \Leftrightarrow \left(\begin{smallmatrix}-1 & | & 0 & a\end{smallmatrix}\right)$
Aus Zeile 2 (Speichern):
$\rightarrow F_0(z) = z^{-1}F_0(z) \Leftrightarrow \left(\begin{smallmatrix}1 & | & 0 & 0\end{smallmatrix}\right)$
Aus Zeile 3 (Addition):
$\rightarrow F_0(z) = z^{-1}F_0(z) + b \Leftrightarrow \left(\begin{smallmatrix}1 & | & b & 0\end{smallmatrix}\right)$

Abbildung 2.5: Korrespondierende Gleichungen

Zeilen 1 & 2 werden wie in Fall 2.1 behandelt. In Zeile 3 wird in jedem Schleifendurchlauf der Wert $b$ addiert. Um dies zu repräsentieren, benutzen wir die $\epsilon$-Funktion, deren Transformierte genau der $n-1$. Spalte der Registermatrix entspricht. Die Registermatrix $\left(\begin{smallmatrix}-1 & | & 0 & a\\ 1 & | & b & 0\end{smallmatrix}\right)$ entspricht der Gleichung $F_0(z) = z^{-1}(F_0(z) + \frac{bz}{z-1}) + a$. Aufgelöst nach $F_0(z)$ erhält man

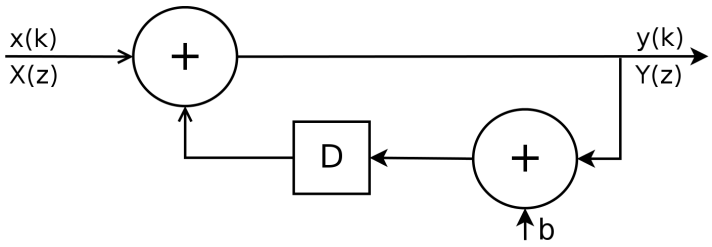



Abbildung 2.6: Systemblockschaltbild zu Quellcode Abb. 2.4 [mit $x(k) = a\ \delta(k)$]

$F_0(z) = \frac{bz}{(z-1)^2} + \frac{az}{z-1}$. Der zweite Teil ist äquivalent zu Fall 2.1. Der erste Teil kann mit der Korresp. (3) zurücktransformiert werden. Gesamt erhält man somit: $f_0(k) = bk\epsilon(k) + a\epsilon(k)$, was wiederum mit dem erwarteten Ergebnis übereinstimmt.

## 2.3 Fall: Multiplikation mit konstantem Faktor

Wird ein Register in der Schleife mit einem konstanten Wert b multipliziert, was auch durch mehrfache Addition desselben Registers erreichbar wäre, so muss man den nächsten Fall betrachten.

```
R0 = a;
while (true) {
    R0 = R0 * b;
}
```

Abbildung 2.7: Quellcode: konst. Multiplikation

1
2 Aus Zeile 1 (Initialisierung):
3 $\rightarrow 0 = -F_0(z) + a \Leftrightarrow \left(\begin{smallmatrix} -1 & | & 0 & a \end{smallmatrix}\right)$
4 Aus Zeilen 2/3 (Speichern & Mult.):
$\rightarrow F_0(z) = z^{-1}F_0(z) \Leftrightarrow \left(\begin{smallmatrix} b & | & 0 & 0 \end{smallmatrix}\right)$

Abbildung 2.8: Korrespondierende Gleichungen

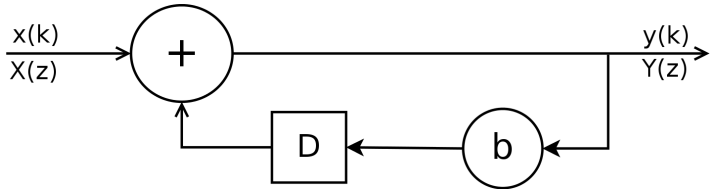



Abbildung 2.9: Systemblockschaltbild zu Quellcode Abb. 2.7 [mit $x(k) = a\ \delta(k)$]

Nach Zeile 1 hat man: $\left(\begin{smallmatrix} -1 & | & 0 & a \\ 0 & | & 0 & 0 \end{smallmatrix}\right)$, nach Zeile 2 folgende Matrix: $\left(\begin{smallmatrix} -1 & | & 0 & a \\ 1 & | & 0 & 0 \end{smallmatrix}\right)$ und nach Zeile 3 die endgültige Matrix: $\left(\begin{smallmatrix} -1 & | & 0 & a \\ b & | & 0 & 0 \end{smallmatrix}\right)$. Das Auflösen ergibt $F_0(z) = \frac{az}{z-b}$. Zurücktransformiert mit Korrespondenz (4) erhält man: $f_0(k) = ab^k\epsilon(k)$.
Etwas komplexer wird das Ganze, wenn man Fall 2.2 hinzunimmt.

Somit gilt es $F_0(x) = z^{-1}(bF_0(z) + \frac{cz}{z-1}) + a$ zu lösen. Es resultiert: $F_0(z) = \frac{cz}{(z-b)(z-1)} + \frac{az}{z-b}$. Um den ersten Teil mit Hilfe der Korrespondenztabelle in den Zeitbereich transformieren zu können, muss zuerst eine Partialbruchzerlegung durchgeführt werden. Als Ansatz verwenden wir:

$$\frac{A}{z-b} + \frac{B}{z-1} = \frac{cz}{(z-b)(z-1)}.$$
$$\Leftrightarrow A(z-1) + B(z-b) = cz$$
$$\Leftrightarrow z(A+B) + (-A - Bb) = cz$$

```
R0 = a;                          1
while (true) {                   2
    R0 = R0 * b;                 3
    R0 = R0 + c;                 4
}                                5
```

Abbildung 2.10: Quellcode: konstante Multiplikation & Addition

Zeile 1 (Initialisierung): $\rightarrow \left(\begin{smallmatrix} -1 & | & 0 & a \end{smallmatrix}\right)$
Zeile 2 (Speichern): $\rightarrow \left(\begin{smallmatrix} 1 & | & 0 & 0 \end{smallmatrix}\right)$
Zeile 3 (Multiplikation):
$\rightarrow \left(\begin{smallmatrix} 1 & | & 0 & 0 \end{smallmatrix}\right) * b = \left(\begin{smallmatrix} b & | & 0 & 0 \end{smallmatrix}\right)$
Zeile 4 (Addition):
$\rightarrow \left(\begin{smallmatrix} b & | & 0 & 0 \end{smallmatrix}\right) + \left(\begin{smallmatrix} 0 & | & c & 0 \end{smallmatrix}\right) = \left(\begin{smallmatrix} b & | & c & 0 \end{smallmatrix}\right)$
$\Rightarrow$ fertige Registermatrix: $\left(\begin{smallmatrix} -1 & | & 0 & a \\ b & | & c & 0 \end{smallmatrix}\right)$

Abbildung 2.11: Registermatrix

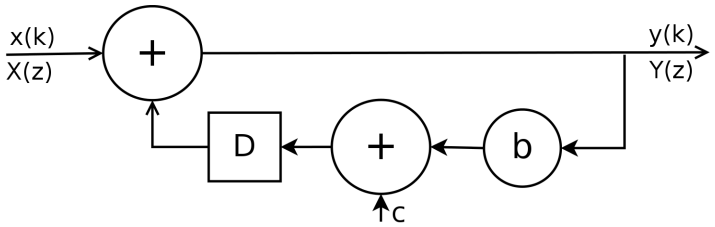



Abbildung 2.12: Systemblockschaltbild zu Quellcode Abb. 2.10 [mit $x(k) = a\ \delta(k)$]

$$\Leftrightarrow\ A + B = c \quad \wedge \quad -A - Bb = 0$$

$$A = c - B\ \Rightarrow\ -(c - B) - Bb = 0 \quad \Leftrightarrow \quad B - Bb = c$$

$$B = \frac{c}{1-b}\ \Rightarrow\ A = c - \frac{c}{1-b} \quad \Leftrightarrow \quad A = \frac{c(1-b) - c}{1-b} \quad \Leftrightarrow \quad A = \frac{-bc}{1-b}$$

$$\Rightarrow F_0(z) = \frac{\frac{-bc}{1-b}}{z-b} + \frac{\frac{c}{1-b}}{z-1} + \frac{az}{z-b}$$

Die ersten beiden Terme ergänzen wir mit $zz^{-1}$. Zusätzlich setzen wir $\alpha = \frac{-bc}{1-b}$, $\beta = \frac{c}{1-b}$, $\gamma = a$ und erhalten:

$$F_0(z) = \alpha\frac{z}{z-b}z^{-1} + \beta\frac{z}{z-1}z^{-1} + \gamma\frac{z}{z-b}$$

Mit den Korrespondenzen (2), (4) aus der Tabelle und der Verschiebungseigenschaft der z-Transformation ($F(z)z^{-1} \triangleq f(k-1)$) ergibt sich:

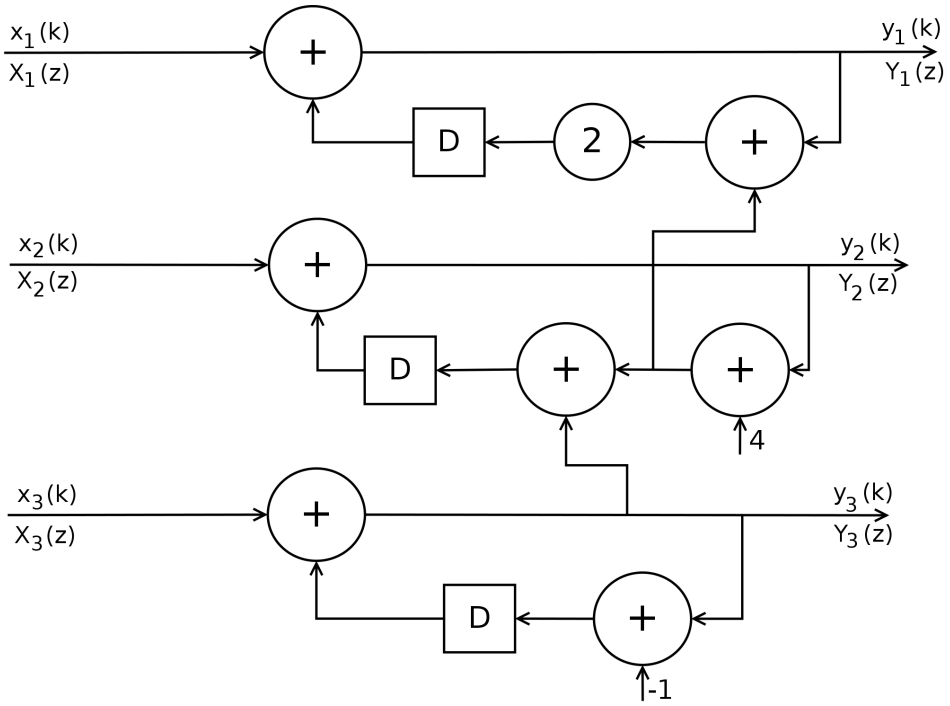
$$f_0(k) = \alpha b^{k-1}\epsilon(k-1) + \beta\epsilon(k-1) + \gamma b^k\epsilon(k)$$

Wieder haben wir eine geschlossene Formel für das Register gefunden.

## 2.4 Fall: Azyklisch

### 2.4.1 Beispiel 1

```
R1 = 1;                                  1
R2 = 2;                                  2
R3 = 3;                                  3
while (true) {                           4
    R2 = R2 + 4;                         5
    R1 = R1 + R2;                        6
    R2 = R2 + R3;                        7
    R3 = R3 − 1;                         8
    R1 = R1 ∗ 2;                         9
}                                        10
```




Im Schaltbild des Systems sind die Eingaben $x_1(k) = \delta(k)$, $x_2(k) = 2\,\delta(k)$ und $x_3(k) = 3\,\delta(k)$, was jeweils einem Impuls zum Zeitpunkt $k = 0$ mit der Höhe des jeweiligen Initialisierungswertes des Registers entspricht. $y_i(k)$ ist ein Signal, das an der Stelle $k$ dem Registerwert des Registers $i$ zum Zeitpunkt $k$ entspricht.
Am Schaltbild ist gut zu erkennen, dass das dritte System nur einen Eingang hat, der wie beschrieben nur mit einem kurzen Impuls angesteuert wird, wohingegen das erste und zweite System noch einen zweiten Eingang haben, welcher von einem anderen Register generiert wird.

Zwischenstand nach Zeile 4 (nach Init. & Speichern):

$$R1 = \left(\begin{array}{ccc|cc} -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{array}\right), R2 = \left(\begin{array}{ccc|cc} 0 & -1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right), R3 = \left(\begin{array}{ccc|cc} 0 & 0 & -1 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

Zeile 5: $R2' = \left(\begin{array}{ccc|cc} 0 & 1 & 0 & 0 & 0 \end{array}\right) + 4\epsilon(k) = \left(\begin{array}{ccc|cc} 0 & 0 & 0 & 4 & 0 \end{array}\right) \Rightarrow R2' = \left(\begin{array}{ccc|cc} 0 & 1 & 0 & \mathbf{4} & 0 \end{array}\right)$

Zeile 6: $R1' = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \end{array}\right) + R2' = \left(\begin{array}{ccc|cc} 0 & 1 & 0 & 4 & 0 \end{array}\right) \Rightarrow R1' = \left(\begin{array}{ccc|cc} 1 & \mathbf{1} & 0 & \mathbf{4} & 0 \end{array}\right)$

Zeile 7: $R2' = \left(\begin{array}{ccc|cc} 0 & 1 & 0 & 4 & 0 \end{array}\right) + R3' = \left(\begin{array}{ccc|cc} 0 & 0 & 1 & 0 & 0 \end{array}\right) \Rightarrow R2' = \left(\begin{array}{ccc|cc} 0 & 1 & \mathbf{1} & 4 & 0 \end{array}\right)$

Zeile 8: $R3' = \left(\begin{array}{ccc|cc} 0 & 0 & 1 & 0 & 0 \end{array}\right) - \epsilon(k) = \left(\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \end{array}\right) \Rightarrow R3' = \left(\begin{array}{ccc|cc} 0 & 0 & 1 & \mathbf{-1} & 0 \end{array}\right)$

Zeile 9: $R1' = \left(\begin{array}{ccc|cc} 1 & 1 & 0 & 4 & 0 \end{array}\right) * 2 \Rightarrow R1' = \left(\begin{array}{ccc|cc} \mathbf{2} & \mathbf{2} & 0 & \mathbf{8} & 0 \end{array}\right)$

Nach Zeile 10 (vollständiges Aufstellen):

$$R1 = \left(\begin{array}{ccc|cc} -1 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 8 & 0 \end{array}\right), R2 = \left(\begin{array}{ccc|cc} 0 & -1 & 0 & 0 & 2 \\ 0 & 1 & 1 & 4 & 0 \end{array}\right), R3 = \left(\begin{array}{ccc|cc} 0 & 0 & -1 & 0 & 3 \\ 0 & 0 & 1 & -1 & 0 \end{array}\right)$$

Betrachtet man dies und sucht eine geschlossene Form für R1, so stellt man fest, dass R1 zusätzlich noch von R2 abhängt (2. Spalte nicht überall 0). R2 hängt wiederum von R3 ab (3. Spalte nicht überall 0). R3 jedoch nur von seinen eigenen Vorgängerwerten (i. Spalte überall 0 für $1 \leq i < 3$), d.h. wir können für R3 eine geschlossene Form finden. Dazu liest man die Gleichung direkt aus der Registermatrix, wie in 1.3.1 beschrieben, ab und man erhält:

$$F_3(z) = \frac{1}{z-1}(\frac{-z}{z-1} + 3z)$$

$$F_2(z) = \frac{1}{z-1}(F_3(z) + \frac{4z}{z-1} + 2z)$$

$$F_1(z) = \frac{1}{z-2}(2F_2(z) + \frac{8z}{z-1} + z)$$

Ja nachdem, was man nun bestimmen möchte, kann man die Gleichungen ineinander einsetzen und umformen. Hier:

$$F_3(z) = \frac{-z}{(z-1)^2} + \frac{3z}{z-1}$$

$$\Rightarrow F_2(z) = \frac{-z}{(z-1)^3} + \frac{7z}{(z-1)^2} + \frac{2z}{z-1}$$

$$\Rightarrow F_1(z) = \frac{-2z}{(z-1)^3(z-2)} + \frac{14z}{(z-1)^2(z-2)} + \frac{12z}{(z-1)(z-2)} + \frac{z}{z-2}$$

Für die Rücktransformation müssen für $F_1(z)$ drei Partialbruchzerlegungen durchgeführt werden. Ansätze:

$$\frac{A_1}{(z-1)^3} + \frac{A_2}{(z-1)^2} + \frac{A_3}{z-1} + \frac{A_4}{z-2} = \frac{-2z}{(z-1)^3(z-2)}$$

$$\frac{B_1}{(z-1)^2} + \frac{B_2}{z-1} + \frac{B_3}{z-2} = \frac{14z}{(z-1)^2(z-2)}$$

$$\frac{C_1}{z-1} + \frac{C_2}{z-2} = \frac{12z}{(z-1)(z-2)}$$

Gelöst erhält man: $F_1(z) = \frac{2}{(z-1)^3} - \frac{10}{(z-1)^1}zz^{-1} - \frac{36}{z-1}zz^{-1} + \frac{48}{z-2}zz^{-1} + \frac{z}{z-2}$. Nun kann man wieder alle Gleichungen mit Hilfe der Korrespondenzentabelle und Eigenschaften in den Zeitbereich transformieren:

$$f_3(k) = -k\epsilon(k) + 3\epsilon(k)$$

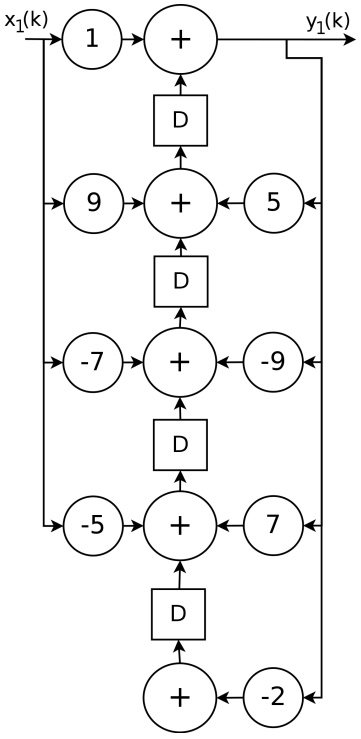
$$f_2(k) = -\frac{(k-1)k}{2}\epsilon(k-2) + 7k\epsilon(k) + 2\epsilon(k)$$

$$f_1(k) = 2(k-2)\frac{k-1}{2}\epsilon(k-3) - 10(k-1)\epsilon(k-1) - 36\epsilon(k-1) + 48 * 2^{k-1}\epsilon(k-1) + 2^k\epsilon(k)$$

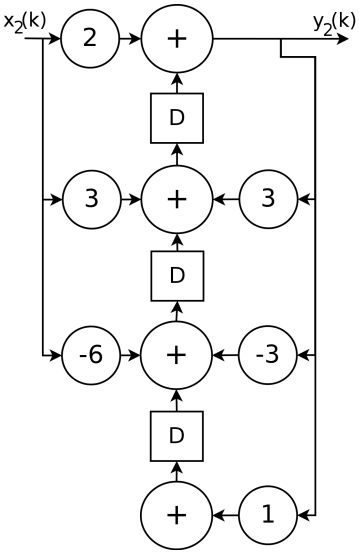
Überprüfung ergibt die gleiche Folge wie das Eingabeprogamm:
$(R1, R2, R3)_k = (1, 2, 3)_0, (14, 9, 2)_1, (54, 15, 1)_2, (146, 20, 0)_3, (340, 24, -1)_4, (736, 27, -2)_5, \ldots$

Möchte man noch zum Vergleich das Systemmodell nach 1.3.3 aufzeichnen, müssen die Gleichungen auf die passende Form gebracht werden und man erhält:

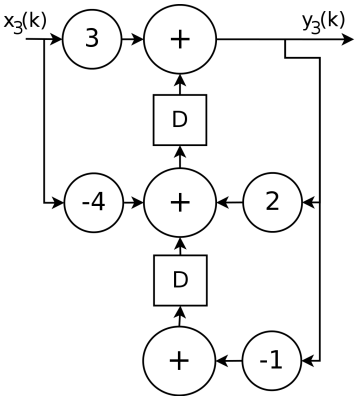
$$F_1(z) = \frac{z^4 + 9z^3 - 7z^2 - 5z}{z^4 - 5z^3 + 9z^2 - 7z + 2} \quad F_2(z) = \frac{2z^3 + 3z^2 - 6z}{z^3 - 3z^2 + 3z - 1} \quad F_3(z) = \frac{3z^2 - 4z}{z^2 - 2z + 1}$$




(a) System f. Reg 1




(b) System f. Reg 2




(c) System f. Reg 3

Diese Systeme sind nur noch abhängig von dem jeweiligen Eingangssignal, was dem jeweiligen Impuls zum Zeitpunkt $k = 0$ entspricht. Man könnte sie seperat, ohne große Änderungen, in Hardware, implementieren um die Zahlenfolge zu generieren, was ein weiterer Vorteil solcher Systeme ist.

### 2.4.2 Allgemein

kann man sagen, dass Eingaben dieser Form immer auf diese Art und Weise transformiert werden können.

**Beweis: Induktion nach m (= Anzahl Abhängigkeiten)**
Sei $M^{(i)}$ die Registermatrix zu Register $i$ direkt nach dem Aufstellen.

$$M^{(i)} = \left(\begin{array}{cccccccc|cc} 0 & 0 & \dots & 0 & -1 & 0 & \dots & 0 & 0 & x_{0,n} \\ x_{1,0} & x_{1,1} & \dots & x_{1,i-1} & x_{1,i} & x_{1,i+1} & \dots & x_{1,n-2} & x_{1,n-1} & x_{1,n} \end{array}\right)$$

**m = 0:**
Hängt $R_i$ von keinem anderen Register ab, gibt es zwei Fälle für die zugehörige Gleichung:

1. Fall: $x_{1,i} = 0$: $F_i = x_{1,n-1}\frac{zz^{-1}}{z-1} + x_{1,n}z^{-1} + x_{0,n}$
   Jeder dieser drei Terme kann direkt mit den Korrespondenzen (1), (2) und der Verschiebungseigenschaft zurücktransformiert werden.
2. Fall: $x_{1,i} \neq 0$: $F_i = \frac{z}{z-x_{0,i}}\left(x_{1,n-1}\frac{zz^{-1}}{z-1} + x_{1,n}z^{-1} + x_{0,n}\right)$
   Ausmultipliziert: $F_i = x_{1,n-1}\frac{z}{(z-1)(z-x_{0,i})} + x_{1,n}\frac{1}{z-x_{0,i}} + x_{0,n}\frac{z}{z-x_{0,i}}$
   Die hinteren Terme können direkt mit Korrespondenz (4) und der Verschiebungseigenschaft rücktransformiert werden. Für den ersten Term geht dies auch, jedoch muss zuerst eine Partialbruchzerlegung durchgeführt werden. Da alle Polstellen bekannt sind $\Rightarrow$ lösbar.

**m → m+1:**
Sei $R_i$ abhängig von $R_j$ und dies bereits abhängig von bis zu $m$ anderen Registern.

1. Fall: $x_{1,i} = 0$
   Dann hat die zugehörige Gleichung, umgestellt nach $F_i$, die Form:

$$F_i(z) = \sum_{\substack{j=0 \\ j\neq i}}^{n-2}\left(x_{1,j}F_j(z)z^{-1}\right) + \frac{x_{1,n-1}}{z-1} + x_{1,n}z^{-1} + x_{0,n}$$

   Setzt man in diese Gleichung ein $F_j$ ein, so erhöht sich der Grad des Nenners nicht; d.h. es entsteht keine weitere Polstelle, die bei einer Partialbruchzerlegung berücksichtigt werden müsste $\Rightarrow$ Konnte man $F_j$ zerlegen, so kann auch $F_i$ zerlegt werden.

2. Fall: $x_{1,i} \neq 0$
   Die umgestellte Gleichung sieht in diesem Fall wie folgt aus:

$$F_i(z) = \frac{z}{z - x_{1,i}} \left( \sum_{\substack{j=0 \\ j \neq i}}^{n-2} \left( x_{1,j} F_j(z) z^{-1} \right) + \frac{x_{1,n-1}}{z-1} + x_{1,n} z^{-1} + x_{0,n} \right)$$

   Setzt man $F_j$ ein, wird eine Polstelle hinzugefügt. Jedoch ist diese bekannt und kann ohne Probleme bei der Partialbruchzerlegung (durch einen leicht veränderten Ansatz mit einem weiteren Partialbruch) behandelt werden. $F_j$ zerlegbar $\Rightarrow$ $F_i$ zerlegbar.

Somit ist die Behauptung bewiesen.

## 2.5 Fall: Zyklisch ohne Eigenvorkommen

### 2.5.1 Beispiel 2: Fibonacci Zahlen

```
R1 = 0;                          1
R2 = 1;                          2
while (true) {                   3
    R1 = R1 + R2;                4
    R2 = R1 − R2;                5
}                                6
```

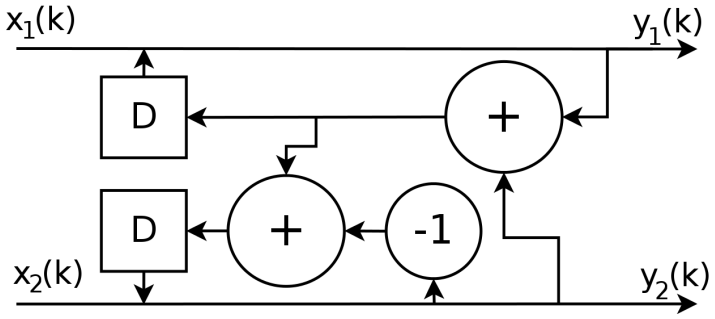



Abbildung 2.13: Ein intuitiv konstruiertes System, das die Zahlenfolgen der Register generiert. $x_1(k) = 0$, $x_2(k) = 1\ \delta(k)$

Das Aufstellen der Registergleichungen ergibt:

$$R1 = \left(\begin{array}{cc|cc} -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{array}\right), R2 = \left(\begin{array}{cc|cc} 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right)$$

Da diese nun beide voneinander abhängen, muss ein anderer Trick angewendet werden. Man multipliziert die Gleichung für $R2$ mit $z^{-1}$, was in der Matrixrepräsentation einer Verschiebung der Zeilen und dem Hinzufügen einer Zeile voller Nullen entspricht, und addiert diese auf R1.

$$R1 + z^{-1}R2 = \left(\begin{array}{cc|cc} -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) + \left(\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right) = \left(\begin{array}{cc|cc} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right)$$

Dies funktioniert jedoch nur, da $R2$ nicht von seinen eigenen Vorgängerwerten abhängt. Somit kann der Eintrag von $F_2(z)$ in der zu $R1$ gehörenden Registermatrix ersetzt werden und man erhält lediglich Terme dazu, die entweder von $F_1(z)$ abhängen oder von anderen festen Werten. Das Prinzip ist für mehrere Register beliebig erweiterbar, jedoch kann sich die Verzögerung und somit der Grad des Nenners der resultierenden Funktion ebenfalls erhöhen. Dies sieht man an diesem Beispiel. Man kann wieder direkt die Gleichung $F_1(z) = \frac{z}{z^2-z-1}$ aufstellen. Um die Gleichung mit den Korrespondenzen transformieren zu können, muss wieder eine Partialbruchzerlegung durchgeführt werden. Die Nullstellen des Nenners können in diesem Beispiel mit Hilfe der *Mitternachtsformel/Lösungsformel für quadratische Gleichungen* gefunden werden. Im Allgemeinen kann das Bestimmen der exakten Stellen jedoch Probleme bereiten und man erhält dieselben Probleme wie in Fall 2.6. Hier erhält man $\frac{1+\sqrt{5}}{2}$ und $\frac{1-\sqrt{5}}{2}$. Nach der Zerlegung erhält man somit:

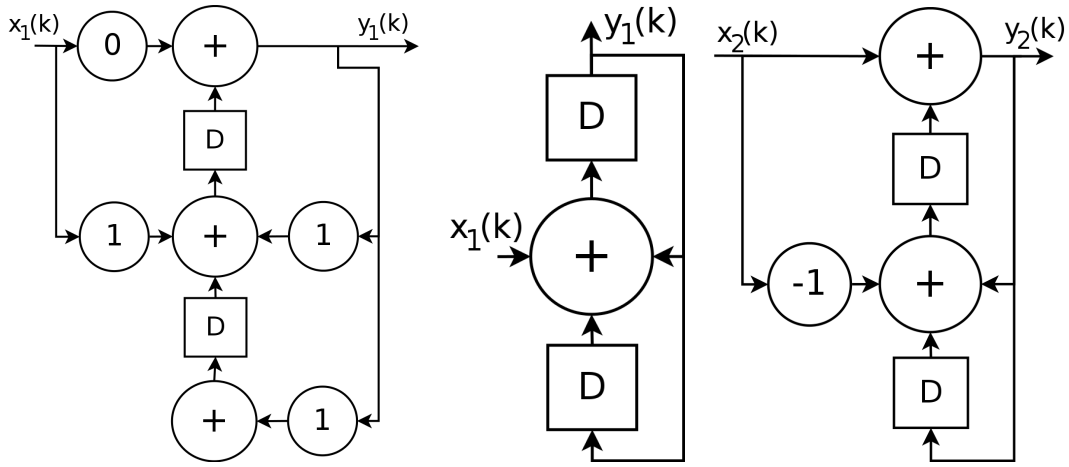
$$F_1(z) = \frac{1}{10}\left(\frac{5+\sqrt{5}}{z - \frac{1+\sqrt{5}}{2}} + \frac{5-\sqrt{5}}{z - \frac{1-\sqrt{5}}{2}}\right)$$

$$f_1(k) = \frac{1}{10}\left((5+\sqrt{5})(\frac{1+\sqrt{5}}{2})^{k-1}\epsilon(k-1) + (5-\sqrt{5})(\frac{1-\sqrt{5}}{2})^{k-1}\epsilon(k-1)\right)$$

Dies lässt sich mit einfachen mathematischen Umformungen und einer Fallunterscheidung für die $\epsilon$-Funktion in die *Moivre-Binet-Formel* umschreiben:

$$f_1(k) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right)$$

Wie man sehr gut an den Abbildungen 2.14(a) und 2.14(b) erkennen kann, hat die Eingabe erst nach einer Verzögerung einen Einfluss auf die Ausgabe. Im Vergleich zum intuitiv konstruierten System (Abb. 2.13) ist hier die Eingabe der Impuls $x_1(k) = \delta(k)$, der nur zum Zeitpunkt $k = 1$ Einfluss auf das System hat. Dies stimmt auch mit dem $z$ im Zähler der Funktion überein. Es wäre also auch denkbar, dass auf der linken Seite des Systems nur ein direkter Eingang gesetzt wird und mit dem angepassten Eingabesignal, hier: $x_1'(k) = \delta(k-1)$, angesteuert wird.




(a) System aus Gleichung für Reg 1 (b) System für Reg 1 (vereinfacht) (c) System für Reg 2

Zum Vergleich ist auf Abbildung 2.14(c) zu erkennen, dass der Eingabeimpuls zu zwei Zeitpunkten Einfluss hat. Interessanterweise liefert dieses System, ab $k = 1$ betrachtet, genau die gleiche Folge (Fibonacci-Zahlen) wie System 2.14(b).

### 2.5.2 Symmetrie der Terme

Aus den Registermatrizen erhalten wir für die Nenner der Funktionen Terme der Form: $N(z) = \sum_{i=0}^{m} a_i z^{m-i}$, mit $a_i \in Z$ und normiert, d.h. $a_0 = 1$. Das Polynom besitzt also nur ganzzahlige Koeffizienten. Von $N(z)$ können nun mittels Polynomdivision oder Hornerschema alle ganzzahligen Nullstellen abdividiert werden. Die resultierenden Polynome nach jeder Division besitzen wiederum nur ganzzahlige Koeffizienten und sind normiert.
Beweis:

- Normierung: Der Koeffizient zur höchsten Potenz bleibt exakt erhalten.
- Ganzzahlige Koeffizienten: Zu den alten Koeffizienten werden lediglich Vielfache der Nullstelle hinzuaddiert. Genauer: Sei $z_0$ ganzzahlige Nullstelle. Dann gilt: $a_{i,neu} = a_{i,alt} + z_0 * a_{i-1,neu}$ für $i > 0$ und $a_{0,neu} = a_{0,alt} = 1$. Da $a_0$ ganzzahlig und das Produkt und die Summe ganzzahliger Werte ebenfalls ganzzahlig ist, folgt, dass $\forall a_{i,neu}$ ganzzahlig.

Weiter kann als Hilfsmittel zum Finden der Nullstellen verwendet werden, dass für Polynome mit ganzzahligen Koeffizienten jede ganzzahlige Nullstelle ein Teiler von $a_m$ sein muss. Mit dem *Lemma von Gauß*, welches besagt, dass für ein normiertes Polynom mit ganzzahligen Koeffizienten jede **rationale** Nullstelle ganzzahlig sein muss, kann geschlossen werden, dass die restlichen Nullstellen entweder irrational oder sogar komplex sein müssen.

Angenommen es bleibt eine Funktion $f(z) = \frac{qz+p}{az^2+bz+c}$, $a$,$b$,$c$,$p$,$q \in Z$ übrig, folgt daraus, dass, falls $b^2 - 4ac > 0$ gilt, die Polstellen von $f(z)$ *symmetrisch* sind und damit die Konstanten nach einer Partialbruchzerlegung für $f(z)$ ebenfalls symmetrisch zueinander sind.
$\alpha$ und $\alpha'$ symmetrisch bedeutet in diesem Fall: $\alpha = r + \sqrt{s} \Rightarrow \alpha' = r - \sqrt{s}$.
Beweis:

- Polstellen symmetrisch: mit Lösungsformel gilt $z_{1/2} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. Ist die Diskriminante nun negativ, so liegt die Lösung in der komplexen Ebene. Für $b^2 - 4ac = 0$ erhält man eine doppelte rationale Polstelle, welche nach dem Lemma von Gauß bereits ganzzahlig hätte sein müssen. Es ist ersichtlich, dass die Symmetrie folgt.

- Konstanten der Partialbruchzerlegung symmetrisch: Seien für den zweiten Teil des Beweises die Polstellen $\alpha = \frac{r+\sqrt{s}}{t}$ und $\alpha' = \frac{r-\sqrt{s}}{t}$. Der Ansatz für die PBZ ist:

$$\frac{A_1}{z-\alpha} + \frac{A_2}{z-\alpha'} = \frac{qz+p}{(z-\alpha)(z-\alpha')}$$

  Dies stellt man um und führt einen Koeffizientenvergleich durch.

$$\begin{aligned} \Leftrightarrow: \quad & A_1(z-\alpha') + A_2(z-\alpha) && = qz+p \\ \Leftrightarrow: \quad & z(A_1+A_2) + (-\alpha' A_1 - \alpha A_2) && = qz+p \end{aligned}$$

$$\begin{aligned} A_1 + A_2 = q \quad & \wedge \quad && -\alpha' A_1 - \alpha A_2 = p \\ A_2 = q - A_1 \quad & \wedge \quad && \frac{-r+\sqrt{s}}{t} A_1 + \frac{-r-\sqrt{s}}{t} A_2 = p \\ & \wedge \quad && (-r+\sqrt{s})A_1 + (-r-\sqrt{s})A_2 = pt \end{aligned}$$

$$\text{mit } A_2 = q - A_1 \Rightarrow \quad (-r+\sqrt{s})A_1 + (-r-\sqrt{s})(q-A_1) = pt$$

$$\begin{aligned} -rA_1 + \sqrt{s}A_1 + -rq - q\sqrt{s} + rA_1 + \sqrt{s}A_1 &= pt \\ 2\sqrt{s}A_1 &= pt + rq + q\sqrt{s} \\ A_1 &= \frac{pt + rq + q\sqrt{s}}{2\sqrt{s}} \\ A_1 &= \frac{qs + (pt+rq)\sqrt{s}}{2s} \end{aligned}$$

$$A_2 = q - A_1 = \frac{2qs}{2s} - \frac{qs + (pt+rq)\sqrt{s}}{2s}$$

$$A_2 = \frac{2qs - qs - (pt+rq)\sqrt{s}}{2s} = \frac{qs - (pt+rq)\sqrt{s}}{2s}$$

Somit ist auch die Symmetrie hierfür bewiesen.

## 2.6 Fall: Zyklisch

### 2.6.1 Beispiel 3

```
R1 = 1;                                  1
R2 = 2;                                  2
R3 = 3;                                  3
while (true) {                           4
    R2 = R2 + 4;                         5
    R1 = R1 + R2;                        6
    R2 = R2 + R3;                        7
    R3 = R3 − R1;                        8
}                                        9
```

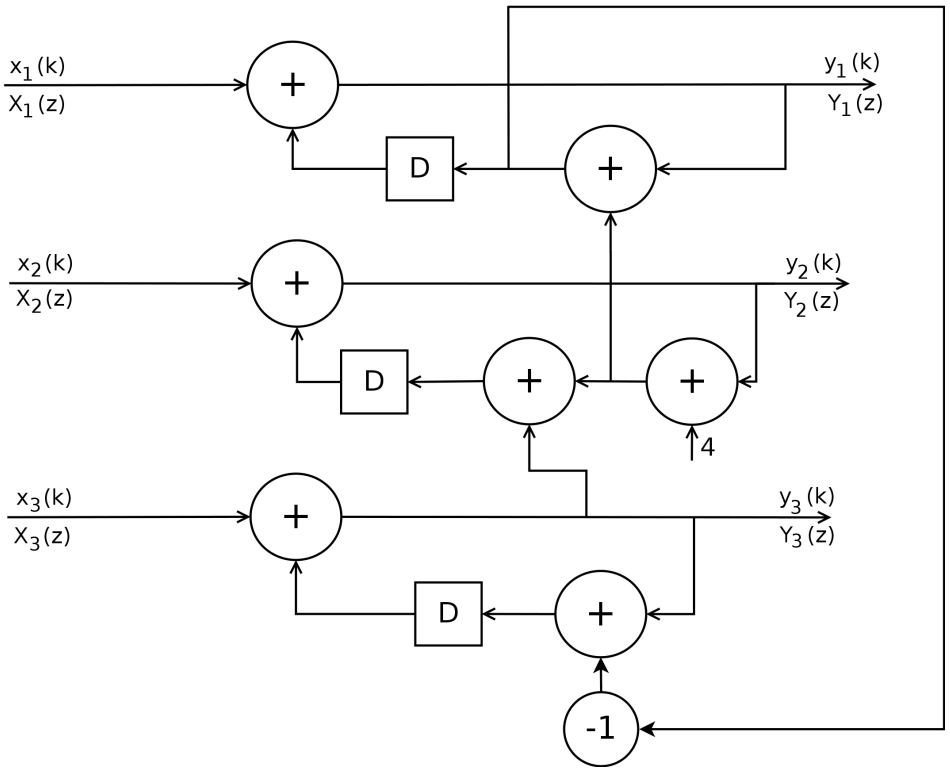



Abbildung 2.14: intuitives Systemblockschaltbild [mit $x_i(k) = i\ \delta(k)$]

Verändert man Beispiel 1 nur minimal, so erhält man deutlich andere Ergebnisse beim Analysieren. Zudem erhöht sich auch der Analyseaufwand und es ist nicht mehr möglich, auf die bisherige Art eine geschlossene Formel zu finden.

Registermatrizen nach dem Aufstellen:

$$R1 = \left(\begin{array}{ccc|cc} -1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 4 & 0 \end{array}\right), R2 = \left(\begin{array}{ccc|cc} 0 & -1 & 0 & 0 & 2 \\ 0 & 1 & 1 & 4 & 0 \end{array}\right), R3 = \left(\begin{array}{ccc|cc} 0 & 0 & -1 & 0 & 3 \\ -1 & -1 & 1 & -4 & 0 \end{array}\right)$$

Da es kein Register gibt, das nicht von einem anderen abhängt, können die bisherigen Methoden nicht angewendet werden. Es müssen die Funktionen aufgestellt, aufgelöst und ineinander eingesetzt werden:

$$\begin{aligned} F_1(z) &= \frac{1}{z-1}(F_2(z) + \frac{4z}{z-1} + z) \\ F_2(z) &= \frac{1}{z-1}(F_3(z) + \frac{4z}{z-1} + 2z) \\ F_3(z) &= \frac{1}{z-1}(-F_1(z) - F_2(z) - \frac{4z}{z-1} + 3z) \end{aligned}$$

$$\begin{aligned} F_1(z) \to F_3(z): \quad F_3(z) &= \frac{-F_2(z)}{(z-1)^2} - \frac{F_2(z)}{z-1} - \frac{4z}{(z-1)^3} - \frac{5z}{(z-1)^2} + \frac{3z}{z-1} \\ F_3(z) \to F_2(z): \quad F_2(z) &= \frac{-F_2(z)}{(z-1)^3} - \frac{F_2(z)}{(z-1)^2} - \frac{4z}{(z-1)^4} - \frac{5z}{(z-1)^3} + \frac{7z}{(z-1)^2} + \frac{2z}{z-1} \\ \Leftrightarrow \quad F_2(z) &= \frac{(z-1)^3}{z^3 - 3z^2 + 4z - 1}\left(-\frac{4z}{(z-1)^4} - \frac{5z}{(z-1)^3} + \frac{7z}{(z-1)^2} + \frac{2z}{z-1}\right) \end{aligned}$$

Man erkennt, dass die Abhängigkeiten aufgelöst werden können, jedoch gibt es trotzdem Probleme. Der erste Term der Gleichung kann nicht mehr einfach mit Hilfe der Korrespondenzen zurücktransformiert werden. (Für Gleichungen bis zu einem Grad von vier existieren noch Lösungsformeln und es wäre somit eine Partialbruchzerlegung denkbar. Da jedoch im Allgemeinen der Grad dieser Funktion beliebig hoch sein kann, werden diese Lösungsformeln hier nicht verwendet.) Die restlichen Terme könnte man in den Zeitbereich transformieren, jedoch würde die Multiplikation dazwischen in eine Faltung übergehen, was wiederum in einer unendlichen Summe resultieren würde.

### 2.6.2 Weitere Rücktransformationsmethoden

#### Polynomdivision

Eine Lösungsmethode ist eine Polynomdivision. Dazu bringen wir $F_2(z)$ in geeignete Form:

$$F_2(z) = \frac{-4z}{z^4 - 4z^3 + 7z^2 - 5z + 1} + \frac{-5z}{z^3 - 3z^2 + 4z - 1} + \frac{7z^2 - 7z}{z^3 - 3z^2 + 4z - 1} + \frac{2z^3 - 4z^2 + 2z}{z^3 - 3z^2 + 4z - 1}$$

Führt man die Division durch, erhält man:

$(-4z) : (z^4 - 4z^3 + 7z^2 - 5z + 1) = -4z^{-3} - 16z^{-4} - 36z^{-5} - 52z^{-6} - 32z^{-7} + 72z^{-8} + \ldots$

Analog durchgeführt für die restlichen Terme resultiert:

1. $0z^0 + 0z^{-1} + 0z^{-2} - 4z^{-3} - 16z^{-4} - 36z^{-5} - 52z^{-6} - 32z^{-7} + 72z^{-8} + \dots$
2. $0z^0 + 0z^{-1} - 5z^{-2} - 15z^{-3} - 25z^{-4} - 20z^{-5} + 25z^{-6} + 130z^{-7} + 270z^{-8} + \dots$
3. $0z^0 + 7z^{-1} + 14z^{-2} + 14z^{-3} - 7z^{-4} - 63z^{-5} - 147z^{-6} - 196z^{-7} - 63z^{-8} + \dots$
4. $2z^0 + 2z^{-1} + 0z^{-2} - 6z^{-3} - 16z^{-4} - 24z^{-5} - 14z^{-6} + 38z^{-7} + 146z^{-8} + \dots$

---

$\sum$: $2z^0 + 9z^{-1} + 9z^{-2} - 11z^{-3} - 64z^{-4} - 143z^{-5} - 188z^{-6} - 60z^{-7} + 425z^{-8} + \dots$

Man hätte die Division für die hinteren Terme auch zusammenfassen können, doch so sieht man direkt die Verzögerung der einzelnen Teile. Die Koeffizienten der Folge entsprechen genau den Werten, die das Register nach dem k-ten Durchlauf der Schleife annimmt.

### Anfangswertsatz

Eine andere Möglichkeit, diese Folge zu berechnen, wird in [Mee] genannt. Dort wird für jeden Zeitschritt der Anfangswertsatz verwendet:

$$\text{Anfangswertsatz:} \qquad x(0) = \lim_{z\to\infty} X(z)$$

Damit gilt

$$X(z) - x(0) = \sum_{k=1}^{\infty} x(k)z^{-k} = z^{-1}\left[x(1) + x(2)z^{-1} + x(3)z^{-2} + \dots\right]$$

Somit entstehen diese Berechnungsvorschriften:

$$\begin{aligned} x(1) &= \lim_{z\to\infty} z\left[X(z) - x(0)\right] \\ x(2) &= \lim_{z\to\infty} z^2\left[X(z) - x(0) - x(1)z^{-1}\right] \\ x(t) &= \lim_{z\to\infty} z^t\left[X(z) - \sum_{l=0}^{t-1} x(l)z^{-l}\right] \end{aligned}$$

Beispiel:

$$\begin{aligned} x(0) &= \lim_{z\to\infty}\left(\frac{-4z}{z^4 - 4z^3 + 7z^2 - 5z + 1} + \frac{2z^3 + 3z^2 - 10z}{z^3 - 3z^2 + 4z - 1}\right) \\ &= \lim_{z\to\infty}\frac{-4z}{z^4 - 4z^3 + 7z^2 - 5z + 1} + \lim_{z\to\infty}\frac{2z^3 + 3z^2 - 10z}{z^3 - 3z^2 + 4z - 1} = 0 + 2 = 2 \end{aligned}$$

$$\begin{aligned} x(1) &= \lim_{z\to\infty} z\left(\frac{-4z}{z^4 - 4z^3 + 7z^2 - 5z + 1} + \frac{2z^3 + 3z^2 - 10z}{z^3 - 3z^2 + 4z - 1} - 2\right) \\ &= \lim_{z\to\infty}\frac{-4z^2}{z^4 - 4z^3 + 7z^2 - 5z + 1} + \lim_{z\to\infty}\frac{2z^4 + 3z^3 - 10z^2 \ - 2z(z^3 - 3z^2 + 4z - 1)}{z^3 - 3z^2 + 4z - 1} \\ &= \lim_{z\to\infty}\frac{-4z^2}{z^4 - 4z^3 + 7z^2 - 5z + 1} + \lim_{z\to\infty}\frac{9z^3 - 18z^2 + 2z}{z^3 - 3z^2 + 4z - 1} = 0 + 9 = 9 \end{aligned}$$

Dies ist beliebig fortsetzbar ...

### Taylorreihenentwicklung

Eine Reihenentwicklung um $a = \infty$ für $n \to \infty$ liefert ebenfalls die Zahlenfolge.

$$\text{Taylorentwicklung:} \qquad f(x) = \sum_{m=0}^{n} \frac{f^{(m)}(a)}{m!}(x-a)^m$$

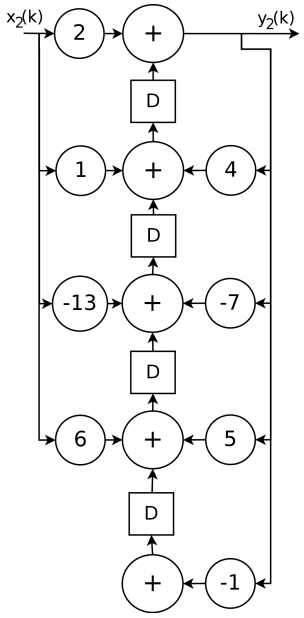
Es gilt somit

$$x(t) = \lim_{a\to\infty} \sum_{m=0}^{t} \frac{X^{(m)}(a)}{m!}(z-a)^m$$

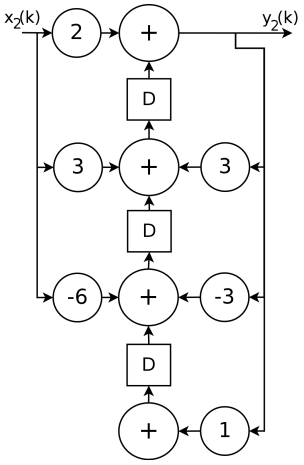
Um den Limes für den Entwicklungspunkt zu umgehen, wird auf [Wol] vorgeschlagen, eine Substitution $z \to \frac{1}{z}$ durchzuführen und dann um den Punkt $a' = 0$ zu entwickeln.

### Bemerkungen

Alle diese Methoden für die Rücktransformation funktionieren, aber keine liefert eine geschlossene Formel. Zusätzlich müssen für jeden Berechnungsschritt die Vorgängerwerte bereits bekannt sein. Es ist also durchaus möglich, auch im zyklischen Fall eine solche Analyse durchzuführen, jedoch nicht im Sinne dieser Arbeit - zumindest ab einem gewissen Grad der Funktionen.




(a) System f. Reg 2 aus Bsp 3




(b) System f. Reg 2 aus Bsp 1, siehe Kap. 2.4

Auch der Vergleich der beiden Systeme liefert leider keine Rückschlüsse über die Ähnlichkeit der Programme. Man sieht unterschiedliche Koeffizienten und zwar ähnliche, aber unterschiedliche Grade.

# Kapitel 3

# Algorithmus

## 3.1 Aufstellen der Gleichungen

---

**Lese** Eingabedatei; 1
**Bestimme** Anzahl benutzter Register ($n$); 2
3
*[ Fuer Initialisierungsblock ]* 4
**Solange** Zuweisung der Form R<i> = <a> gelesen wird: 5
Erstelle leere Registermatrix $M^{(i)}$ der Groesse $2 \times (n+2)$ und 6
initialisiere mit $M^{(i)}_{1,i} = -1,\ M^{(i)}_{1,n+2} = a$; 7
8
*[ Eintritt in Schleife ]* 9
**Fuer** jede Registermatrix $M^{(i)}$: setze $M^{(i)}_{2,i} = 1$; 10
11
*[ Fuer Operationsblock ]* 12
**Solange** Zuweisung $Z_m$ gelesen wird: 13
**Wenn** $Z_m$ die Form: R<i> = R<j> $\pm$ <b> 14
Fuer l=1 bis n+2: $M^{(i)}_{2,l} = M^{(j)}_{2,l}$; 15
$M^{(i)}_{2,n+1} = M^{(i)}_{2,n+1} \pm b$; 16
**Wenn** $Z_m$ die Form: R<i> = R<j> $*$ <b> 17
Fuer l=1 bis n+2: $M^{(i)}_{2,l} = M^{(j)}_{2,l} * b$; 18
**Wenn** $Z_m$ die Form: R<i> = R<j> $\pm$ R<k> 19
Fuer l=1 bis n+2: $M^{(i)}_{2,l} = M^{(j)}_{2,l} \pm M^{(k)}_{2,l}$; 20

---

Dieser einfache Algorithmus erstellt aus der ausgefüllten Eingabemaske die Registermatrizen zu allen verwendeten Registern mit je einer Größe von $n \times (n+2)$.

## 3.2 Auflösen der Gleichungen

**Fuer** jede Registermatrix $M^{(i)}$: stelle zugehoerige Gleichung $G^{(i)}$ auf;

**Fuer** i=1 bis n:
    Loese Gleichung $G^{(i)}$ nach $F_i$ auf und setze Ergebnis in alle anderen Gleichungen $G_{(j)}$, $i \neq j$ ein;
**Fuer** i=n−1 bis 1: Loese Gleichung $G^{(i)}$ erneut nach $F_i$ auf, falls noetig;

**Falls** Gleichung $G^{(i)}$ die Form $F_i = \sum_j \frac{a_j z_j^p}{(z-b)^q}$: $\hat{F}_i = F_i$;
**Sonst**: Fuehre Partialbruchzerlegung (PBZ) durch $\rightarrow \hat{F}_i$

Dieser Algorithmus ist nicht sonderlich trickreich, aber funktioniert in allen Fällen. Trotzdem gibt es Unterschiede: Im azyklischen Fall kann man die Gleichungen immer ohne Probleme zerlegen und rücktransformieren (vgl. 2.4.2). Schwieriger ist dies in den anderen Fällen. Bis zu einem Grad von zwei kann die PBZ mit Hilfe von Lösungsformeln exakt durchgeführt werden (bzw. bis zu einem Grad von 4 mit komplexeren Lösungsformeln). Bei höherem Grad ist dies nicht mehr möglich.

## 3.3 Korrespondenzfindung und Transformation

**Fuer** jede Gleichung $G^{(i)}$:
    **Wenn** $G^{(i)}$ bereits in Partialbrueche zerlegt:
        **Fuer** jeden Partialbruch:
            Suche passende Korrespondenz aus Tabelle und wende diese zusammen mit Verschiebungseigenschaft an;
    **Sonst**:
        Fuehre Reihenentwicklung um $x = \infty$ oder anderes Verfahren fuer Ruecktransformation durch;

Der **Sonst**-Fall funktioniert immer, aber liefert keine geschlossene Form und hat den Nachteil, dass jeder Zwischenwert/schritt berechnet werden muss, was nicht sonderlich effizient und nicht im Sinne dieser Arbeit ist. Vergleiche Kapitel 2.6.2.

# Kapitel 4

# Zusammenfassung

Abschließend kann man sagen, dass es zwar möglich ist, solche Programme, die mit der Eingabemaske erstellt werden können, als Systeme zu betrachten und auch so zu analysieren, um geschlossene Formeln für die einzelnen Register zu erhalten, jedoch stößt man schnell an die Grenzen. Einzig im azyklischen Fall ist es immer möglich, durch diese Analyse und mit Hilfe des in Kapitel 3 beschriebenen Algorithmus eine Lösung zu finden. In allen anderen Fällen erhöht sich der Grad der Funktionen sehr schnell und somit funktioniert der Algorithmus nur auf die Eingaben angewendet, welche einen geringen Grad der Registerfunktionen zur Folge haben.

Ein interessanter Punkt, welcher hier nicht untersucht wurde, wäre die Frage, ob man eine gegebene Zahlenfolge mit diesen Hilfsmitteln auch interpolieren oder approximieren könnte, wie viel Information dazu gespeichert werden müsste und wie gut die Approximation wäre.

Ebenfalls von Interesse wäre eine Weiterführung der Symmetrieeigenschaft, wie in 2.5.2 gezeigt. Oder auch, ob es mit dieser Eigenschaft und der Tatsache, dass alle Koeffizienten ganzzahlig sind, möglich ist, Funktionen höheren Grades zu analysieren. Im beschriebenen Algorithmus könnte die Eigenschaft verwendet werden, dass alle rationalen Nullstellen für ein normiertes Polynom mit ganzzahligen Koeffizienten auch Teiler des Koeffizienten vor dem Faktor $z^0$ sein müssen, um diese herauszuteilen und den Grad des Polynoms zu reduzieren. Weiter können bisher allerdings keine Verbesserungen genutzt werden, trotz dem Wissen, dass alle weiteren Stellen irrational oder komplex sein müssen.

Auch weitere Analysen der Systeme wäre denkbar, um möglicherweise damit noch weitere Informationen zu gewinnen und doch noch eine geschlossene Form erzeugen zu können. Bisher dienen die Systeme in dieser Arbeit nur der Veranschaulichung, da sie sehr eng mit den Z-Transformierten zusammenhängen.

Eine mögliche Anwendung dieser ganzen Arbeit wäre bei der Analyse von einfachen Pseudozufallszahlengeneratoren, wie zum Beispiel den linearen Kongruenzgeneratoren. In [Hal] wird dies von Hand sehr kurz beschrieben und findet im Bereich Reverse Engineering Anwendung. Auch soll diese Arbeit in anderen Gebieten, wie natürlich digitaler Signalverarbeitung oder dem Lösen von Rekursionsgleichungen, einige Hilfsmittel und Tipps bereitstellen.

# Literaturverzeichnis


[Bosa] Martin Bossert. Signale und Systeme - Hilfsblätter.
http://tait.e-technik.uni-ulm.de/~heim/sigsys_WS200607/hall.pdf.

[Bosb] Martin Bossert. Signale und Systeme - Skript.
http://tait.e-technik.uni-ulm.de/~heim/sigsys_WS200607/skript.pdf.

[Bra] Oliver Braeunling. Z-Transformation.
http://www.mathe.braeunling.com. Stand: 25. März 2008 - inzwischen offline.

[Hal] Haldir. Analysis of Algorithms: Recursion Equations.
http://reteam.org/papers/e48.pdf.

[I. 03] I. N. Bronstein, K. A. Semendjajew. Teubner-Taschenbuch der Mathematik, Band 1, 2003. Seite 419 - Regel 5.

[Mee] Klaus Meerkötter. Digitale Signalverarbeitung.
http://nachrichtentheorie.de/skripten/dsv2010/dsv.pdf. Stand: 20. Januar 2012.

[Uwe] Uwe Schoening, Hans A. Kestler. Mathe-Toolbox: Mathematische Notationen, Grundbegriffe und Beweismethoden.

[Wika] Wikipedia. Nullstelle.
http://de.wikipedia.org/wiki/Nullstelle. Stand: 20. Januar 2012.

[Wikb] Wikipedia. Satz ueber rationale Nullstellen.
http://de.wikipedia.org/wiki/Satz_über_rationale_Nullstellen. Stand: 20. Januar 2012.

[Wikc] Wikipedia. Z-Transformation.
http://de.wikipedia.org/wiki/Z-Transformation. Stand: 9. Oktober 2011.

[Wol] Wolfram. Asymptotic Series.
http://mathworld.wolfram.com/AsymptoticSeries.html. Stand: 20. Januar 2012.

# Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Bachelorarbeit selbständig angefertigt, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

.....
Ort, Datum, Unterschrift