TECHNISCHE UNIVERSITÄT
D O R T M U N D
FAKULTÄT FÜR INFORMATIK

Andrej Dudenhefner

Untersuchung und Implementierung des coinduktiven Stromkalküls

November 1, 2011

I N T E R N E B E R I C H T E
I N T E R N A L R E P O R T S

Chair I (functional and rule-based programming)

Supervisors:
Prof. Dr. Peter Padawitz
Dr. Hubert Wagner

GERMANY · D-44221 DORTMUND

Inhaltsverzeichnis

1 Einleitung

1.1 Streams

Unendliche Folgen reeller Zahlen, auch Streams genannt, sind ein grundlegender Baustein der Mathematik. In der Analysis existieren mehrere Möglichkeiten Streams zu beschreiben. Üblich ist die Beschreibung als eine Funktion $f : \mathbb{N} \to \mathbb{R}$ oder eine Rekursionsgleichung (z.B. $a_0 = a_1 = 1, \forall i \in \mathbb{N} : a_{i+2} = a_{i+1} + a_i$).
In der Kategorientheorie ist es möglich, die Menge aller Streams als Träger einer finalen F-Coalgebra mit dem Funktor $F(X) = \mathbb{R} \times X$ zu beschreiben. Des Weiteren lassen sich verschiedene Operatoren auf diesem Objekt coinduktiv definieren und mit deren Hilfe Instanzen von Streams untersuchen. Am bedeutsamstem ist dabei die Stream-Differentiation, die ein Analogon zum Fundamentalsatz der Analysis formulieren lässt. Damit liefert die Kategorientheorie weitere Möglichkeiten Streams zu definieren und zu untersuchen.

2 Kategorientheoretische Strukturen

2.1 Kategorie

Definition 1 (Kategorie)**.** Eine Kategorie $\mathcal{K}$ besteht aus folgenden Komponenten:

- Einer Klasse von Objekten $\mathrm{Ob}(\mathcal{K})$
- Für alle Objekte $A, B \in \mathrm{Ob}(\mathcal{K})$ einer Menge von Morphismen $\mathrm{Mor}_{\mathcal{K}}(A, B)$
- Einer assoziativen Verknüpfung $\circ\colon \mathrm{Mor}_{\mathcal{K}}(A, B) \times \mathrm{Mor}_{\mathcal{K}}(B, C) \to \mathrm{Mor}_{\mathcal{K}}(A, C)$
- Für alle Objekte $A \in \mathrm{Ob}(\mathcal{K})$ einem Identitätsmorphismus $\mathrm{id}_A \in \mathrm{Mor}_{\mathcal{K}}(A, A)$, sodass $\forall B \in \mathrm{Ob}(\mathcal{K}), \forall f \in \mathrm{Mor}_{\mathcal{K}}(A, B) : f = f \circ \mathrm{id}_A = \mathrm{id}_B \circ f$

Im Folgenden wird die Kategorie Set betrachtet, deren Objekte Mengen und deren Morphismen Funktionen sind. Die verlangte assoziative Verknüpfung '$\circ$' entspricht in Set der Verkettung der Funktionen und für jede Menge A entspricht id_A der Identitätsabbildung auf A, die jedes Element von A auf sich selbst abbildet. Offensichtlich erfüllt Set damit alle Forderungen, die an eine Kategorie gestellt sind.

2.2 Funktor

Definition 2 (Funktor)**.** Seien $\mathcal{K}$ und $\mathcal{L}$ zwei Kategorien.
Ein Funktor $F\colon \mathcal{K} \to \mathcal{L}$ definiert

- Eine Abbildung der Objekte $F\colon \mathrm{Ob}(\mathcal{K}) \to \mathrm{Ob}(\mathcal{L})$
- Für alle Objekte $A, B \in \mathrm{Ob}(\mathcal{K})$ eine Abbildung $F\colon \mathrm{Mor}_{\mathcal{K}}(A, B) \to \mathrm{Mor}_{\mathcal{L}}(F(A), F(B))$

Dabei erhält F die Struktur der Kategorie, denn es muss gelten

- $F(\mathrm{id}_A) = \mathrm{id}_{F(A)}$ für alle Objekte $A \in \mathrm{Ob}(\mathcal{K})$
- $F(g \circ f) = F(g) \circ F(f)$ für alle Morphismen $f \in \mathrm{Mor}_{\mathcal{K}}(A, B), g \in \mathrm{Mor}_{\mathcal{K}}(B, C)$

Im Folgenden werden Funktoren $F\colon Set \to Set$ betrachtet, die nur Produkte '$\times$' und Coprodukte '$+$' verwenden. Für Mengen A, B, C, D und Funktionen $f\colon A \to C, g\colon B \to D$ gilt:

$$
\begin{aligned}
A \times B &= \{(a,b) | a \in A, b \in B\} \\
f \times g &\colon A \times B \to C \times D \\
(f \times g)(a,b) &= (f(a), g(b))
\end{aligned}
$$

$$
\begin{aligned}
A + B &= \{(a,0) | a \in A\} \cup \{(b,1) | b \in B\} \\
f + g &\colon A + B \to C + D \\
(f+g)(x) &= \begin{cases} (f(a), 0), & \text{wenn } x = (a,0) \\ (g(b), 1), & \text{wenn } x = (b,1) \end{cases}
\end{aligned}
$$

2.3 F-Coalgebra

Definition 3 (F-Coalgebra)**.** Sei $\mathcal{K}$ eine Kategorie, $A \in \mathrm{Ob}(\mathcal{K})$ ein Objekt und $F\colon \mathcal{K} \to \mathcal{K}$ ein Funktor. Dann heißt ein Morphismus $f \in \mathrm{Mor}_{\mathcal{K}}(A, F(A))$ eine F-Coalgebra.

2.3.1 F-Coalgebra Homomorphismus

Definition 4 (F-Coalgebra Homomorphismus)**.** Seien $\mathcal{K}$ eine Kategorie, $A, B \in \mathrm{Ob}(\mathcal{K})$ zwei Objekte, $F\colon \mathcal{K} \to \mathcal{K}$ ein Funktor und $f \in \mathrm{Mor}_{\mathcal{K}}(A, F(A)), g \in \mathrm{Mor}_{\mathcal{K}}(B, F(B))$ zwei F-Coalgebren.
Ein Morphismus $h \in \mathrm{Mor}_{\mathcal{K}}(A, B)$ heißt F-Coalgebra Homomorphismus, falls gilt $F(h) \circ f = g \circ h$.
Veranschaulicht bedeutet es, dass das folgende Diagramm kommutiert.

$$
\begin{tikzcd}
A \arrow[r, "h"] \arrow[d, "f"] & B \arrow[d, "g"] \\
F(A) \arrow[r, "F(h)"] & F(B)
\end{tikzcd}
$$

Abbildung 1: Kommutatives Diagramm eines F-Coalgebra-Homomorphismus h

2.3.2 Finalität

Definition 5. Sei $\mathcal{K}$ eine Kategorie. Ein Objekt $A \in \mathrm{Ob}(\mathcal{K})$ heißt final genau dann, wenn
$\forall B \in \mathrm{Ob}(\mathcal{K})$ gilt $|\mathrm{Mor}_{\mathcal{K}}(B, A)| = 1$.

3 Stromkalkül

3.1 Streams als Träger einer finalen Coalgebra

Definition 6 (Menge aller Streams)**.** $\mathbb{R}^\omega = \{s \mid s\colon \mathbb{N} \to \mathbb{R}\}$ bezeichnet die Menge aller Streams. Die zu einem Stream $s \in \mathbb{R}^\omega$ äquivalente Folge ist $(s(0), s(1), s(2), \ldots)$.

Satz 1. *Sei $F\colon Set \to Set$ ein Funktor in Set mit $F(X) = \mathbb{R} \times X$. Sei die Menge aller Streams $\mathbb{R}^\omega$ der Träger der F-Coalgebra $\sigma\colon \mathbb{R}^\omega \to \mathbb{R} \times \mathbb{R}^\omega$ mit $\sigma(s) = (s(0), n \mapsto s(n+1))$.*
Dann ist σ eine finale F-Coalgebra.

Bemerkung. Die Coalgebra σ zerlegt einen Stream s in sein Anfangselement und den Rest des Streams. Mit anderen Worden ist $\sigma((s_0, s_1, s_2, \ldots)) = (s_0, (s_1, s_2, s_3, \ldots))$.
Schreibe $\sigma = \langle \sigma_1, \sigma_2 \rangle$ mit $\sigma_1(s) = \pi_1(\sigma(s)) = s(0)$ und $\sigma_2(s) = \pi_2(\sigma(s)) = s'$.
Induktiv lässt sich zeigen: $\forall s \in \mathbb{R}^\omega : s(n) = \sigma_1(\sigma_2^n(s)) = s^{(n)}(0)$.
Bezeichne σ als Stream-Coalgebra.

Um die Finalität von σ zu beweisen wird zunächst der Begriff der Bisimulation auf Streams eingeführt.

Definition 7 (Bisimulation auf Streams)**.** Eine Relation $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ heißt Bisimulation auf Streams, falls $\forall s, t \in \mathbb{R}^\omega : s\, R\, t \Rightarrow \begin{cases} s(0) = t(0) \\ s'\, R\, t' \end{cases}$

Satz 2 (Coinduktion auf Streams)**.** *Für $s, t \in \mathbb{R}^\omega$ gilt:*
$\exists R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ Bisimulation auf Streams mit $s\, R\, t \Rightarrow s = t$

Beweis. Seien $s, t \in \mathbb{R}^\omega$, $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ eine Bisimulation auf Streams und $(s, t) \in R$. Dann gilt induktiv:
(Induktionsanfang) $(\sigma_2^0(s), \sigma_2^0(t)) = (s, t) \in R$,
(Induktionsschritt) $\forall n \in \mathbb{N} : (\sigma_2^n(s), \sigma_2^n(t)) \in R \Rightarrow (\sigma_2^{n+1}(s), \sigma_2^{n+1}(t)) = (\sigma_2^n(s)', \sigma_2^n(t)') \in R$.
Daraus folgt $\forall n \in \mathbb{N} : (\sigma_2^n(s), \sigma_2^n(t)) \in R$.
Da R eine Bisimulation auf Streams ist, gilt schließlich
$\forall n \in \mathbb{N} : \sigma_2^n(s)(0) = \sigma_2^n(t)(0) \Rightarrow s(n) = t(n) \Rightarrow s = t$. □

Bemerkung. Mit Hilfe des obigen Satzes lässt sich die Gleichheit zweier Streams beweisen, indem eine entsprechende Bisimulation auf Streams konstruiert wird.

Korollar 1. *Seien $s, t \in \mathbb{R}^\omega$ mit $s(0) = t(0), s' = t'$. Dann gilt für die Menge*
$R = \{(u, u) \mid u \in \mathbb{R}^\omega\} \cup \{(s, t)\}$:
$\forall (u, v) \in R : u(0) = v(0) \land u' = v'\, R\, v'$.
Damit ist R eine Bisimulation auf Streams und folglich gilt $s = t$.
Mit anderen Worten, falls zwei Streams den gleichen Anfangswert und die gleiche Ableitung besitzen, sind sie gleich.

Beweis von Satz 1. Äquivalente Aussage gezeigt in [1](Theorem 2.2). Sei $\tau\colon A \to F(A)$, $\tau = \langle \tau_1, \tau_2 \rangle$ eine beliebige F-Coalgebra mit dem Träger A. Es gilt zu zeigen, dass genau ein F-Coalgebra Homomorphismus $h\colon A \to \mathbb{R}^\omega$ existiert.
Sei $h\colon A \to \mathbb{R}^\omega$ ein F-Coalgebra Homomorphismus. Dann gilt:

$$\begin{aligned} F(h) \circ \tau &= \sigma \circ h \\ (F(h) \circ \tau)(a) &= (\sigma \circ h)(a) \\ ((id_{\mathbb{R}} \times h) \circ \langle \tau_1, \tau_2 \rangle)(a) &= (\sigma \circ h)(a) \\ (\tau_1(a), h(\tau_2(a))) &= (\sigma_1(h(a)), \sigma_2(h(a))) \end{aligned}$$

Mit anderen Worten gilt (i)

- $\tau_1(a) = \sigma_1(h(a))$
- $h(\tau_2(a)) = \sigma_2(h(a))$

Beweise zunächst die Existenz eines F-Coalgebra Homomorphismus von A nach $\mathbb{R}^\omega$.
Definiere $h\colon A \to \mathbb{R}^\omega$ durch $h(a) = (n \mapsto \tau_1(\tau_2^n(a)))$. Dann gilt:

- $\sigma_1(h(a)) = \sigma_1(n \mapsto \tau_1(\tau_2^n(a))) = \tau_1(\tau_2^0(a)) = \tau_1(a)$.
- $\sigma_2(h(a)) = \sigma_2(n \mapsto \tau_1(\tau_2^n(a))) = n \mapsto \tau_1(\tau_2^{n+1}(a)) = n \mapsto \tau_1(\tau_2^n(\tau_2(a))) = h(\tau_2(a))$

Da (i) erfüllt ist, folgt, dass die definierte Abbildung h ein F-Coalgebra Homomorphismus ist.
Beweise nun die Eindeutigkeit des F-Coalgebra Homomorphismus von A nach $\mathbb{R}^\omega$.
Seien $h_1, h_2\colon A \to \mathbb{R}^\omega$ zwei F-Coalgebra Homomorphismen. Definiere eine Relation $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $R = \{(h_1(a), h_2(a)) \mid a \in A\}$. Aus (i) folgt $\forall a \in A:$
$h_1(a)(0) = \sigma_1(h_1(a)) = \tau_1(a) = \sigma_1(h_2(a)) = h_2(a)(0)$ und

$$\begin{array}{rrr}
h_1(a)' = & \sigma_2(h_1(a)) & \text{Definition von } \sigma \\
= & h_1(\tau_2(a)) & \text{Anwendung von } (i) \\
R & h_2(\tau_2(a)) & \text{Definition von } R \\
= & \sigma_2(h_2(a)) & \text{Anwendung von } (i) \\
= & h_2(a)' & \text{Definition von } \sigma
\end{array}$$

Damit ist R eine Bisimulation auf Streams und $\forall a \in A : h_1(a) = h_2(a)$.
Mit anderen Worten $h_1 = h_2$. □

3.2 Streamoperatoren

In dem beschrieben Kontext der Stream-Coalgebra $\sigma = \langle \sigma_1, \sigma_2 \rangle$ lassen sich Operatoren (siehe [1]) auf $\mathbb{R}^\omega$ definieren. Die ersten beiden Operatoren sind durch die Coalgebra selbst gegeben und wurden bereits als eine vereinfachende Schreibweise verwendet. Andere Operatoren werden durch Gleichheiten, die sie erfüllen müssen, definiert. Seien $r \in \mathbb{R}$ und $s, t \in \mathbb{R}^\omega$, definiere:

- Startwert: $s(0) = \pi_1(\sigma(s)) = \sigma_1(s)$
- Ableitung: $s' = \pi_2(\sigma(s)) = \sigma_2(s)$
- Einbettung: $\underline{\ }\colon \mathbb{R} \to \mathbb{R}^\omega$, mit $\sigma(\underline{r}) = (r, \underline{0})$
- Formale Variable: $X \in \mathbb{R}^\omega$, mit $\sigma(X) = (0, \underline{1})$
- Summe: $+\colon \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(s + t) = (s(0) + t(0), s' + t')$
- Negation: $-\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(-s) = (-s(0), -(s'))$
- Faltungsprodukt: $\times\colon \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(s \times t) = (s(0) * t(0), (s' \times t) + (\underline{s(0)} \times t'))$
- Faltungs-Inverses: $\_^{-1}\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $s(0) \neq 0 \Rightarrow (s \times s^{-1} = \underline{1})$
- Faltungs-Wurzel: $\sqrt[\times]{\_}\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $s(0) > 0 \Rightarrow \sqrt[\times]{s} \times \sqrt[\times]{s} = s$
- Komposition: $\circ\colon \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $t(0) = 0 \Rightarrow \sigma(s \circ t) = (s(0), t \times (s' \circ t))$

Zu einem späteren Zeitpunkt werden weitere Operatoren definiert. Als nächstes soll gezeigt werden, dass die beschriebenen Operatoren wohldefiniert sind.

3.3 Existenz und Eindeutigkeitssatz

Definition 8 (Wohlgeformtes Stream-Differentialgleichungssystem)**.** Sei $\Sigma = \{f, g, \ldots\}$ eine Menge von Funktionssymbolen, und für alle $f \in \Sigma$ sei

- $r_f \in \mathbb{N}$ die Stelligkeit von f.
- t_f ein Term aus Konstanten, Funktionssymbolen aus Σ und den Variablen $(x_1, \ldots, x_{r_f}, x'_1, \ldots, x'_{r_f}, x_1(0), \ldots, x_{r_f}(0))$.
- $h_f \colon \mathbb{R}^{r_f} \to \mathbb{R}$ eine reellwertige Funktion.

Dann heißt das Gleichungssystem $\forall f \in \Sigma$
$f(x_1, \ldots, x_{r_f})(0) = h_f(x_1(0), \ldots, x_{r_f}(0))$,
$f(x_1, \ldots, x_{r_f})' = t_f$
wohl geformt.

Satz 3 (Existenz und Eindeutigkeitssatz)**.** *Sei Σ eine Menge von Funktionssymbolen, dann besitzt jedes wohlgeformte Stream-Differentialgleichungssystem genau eine Lösung.*
Das heißt $\forall f \in \Sigma\ \exists! \hat{f} \colon (\mathbb{R}^\omega)^{r_f} \to \mathbb{R}^\omega$, sodass gilt $\forall s_i \in \mathbb{R}^\omega, 1 \leq i \leq r_f$:
$\hat{f}(s_1, \ldots, s_{r_f})(0) = h_f(s_1(0), \ldots, s_{r_f}(0))$ und $\hat{f}(s_1, \ldots, s_{r_f})' = \hat{t_f}$, wobei $\hat{t_f} \in \mathbb{R}^\omega$ bei der Interpretation der Funktionssymbole $(f, g, \ldots)$ durch die Funktionen $(\hat{f}, \hat{g}, \ldots)$
und Belegung der Variablen $(x_1, \ldots, x_{r_f}, x'_1, \ldots, x'_{r_f}, x_1(0), \ldots, x_{r_f}(0))$ durch
$(s_1, \ldots, s_{r_f}, s'_1, \ldots, s'_{r_f}, s_1(0), \ldots, s_{r_f}(0))$ entsteht.

Beweis nach Skizze in [1](Theorem A.1). Sei Σ eine Menge von Funktionssymbolen, und das Gleichungssystem $\forall f \in \Sigma$
$f(x_1, \ldots, x_{r_f})(0) = h_f(x_1(0), \ldots, x_{r_f}(0))$,
$f(x_1, \ldots, x_{r_f})' = t_f$
wohl geformt.
Definiere induktiv die kleinste Menge U mit Hilfe der Funktionssymbole aus Σ, sodass gilt:

- $\mathbb{R}^\omega \subseteq U$.
- $f \in \Sigma$ mit der Stelligkeit r_f, $u_i \in U, 1 \leq i \leq r_f \Rightarrow f(u_1, \ldots, u_{r_f}) \in U$.

Dabei ist das Element $f(u_1, \ldots, u_{r_f})$ lediglich ein syntaktisches Konstrukt und keine Funktionsanwendung.
Sei $\sigma \colon \mathbb{R}^\omega \to \mathbb{R} \times \mathbb{R}^\omega, \sigma(s) = (s(0), s')$, wie gewohnt, die finale F-Coalgebra (mit dem Funktor $F(X) = \mathbb{R} \times X$). Definiere eine F-Coalgebra $\upsilon \colon U \to \mathbb{R} \times U$, sodass gilt:
$$\upsilon(u) = \begin{cases} \sigma(u), & \text{wenn } u \in \mathbb{R}^\omega \\ (h_f(\pi_1(\upsilon(u_1)), \ldots, \pi_1(\upsilon(u_{r_f}))), \tilde{t_f}), & \text{wenn } u = f(u_1, \ldots, u_{r_f}) \end{cases}$$
Dabei entsteht $\tilde{t_f}$ bei der Belegung der Variablen $(x_1, \ldots, x_{r_f}, x'_1, \ldots, x'_{r_f}, x_1(0), \ldots, x_{r_f}(0))$ in t_f durch $(u_1, \ldots, u_{r_f}, \pi_2(\upsilon(u_1)), \ldots, \pi_2(\upsilon(u_{r_f})), \pi_1(\upsilon(u_1)), \ldots, \pi_1(\upsilon(u_{r_f})))$.
Da σ eine finale F-Coalgebra ist, existiert genau ein F-Coalgebra-Homomorphismus $\phi \colon U \to \mathbb{R}^\omega$.

$$\begin{tikzcd}
U \arrow[r, "\phi"] \arrow[d, "\upsilon"] & \mathbb{R}^\omega \arrow[d, "\sigma"] \\
F(U) = \mathbb{R} \times U \arrow[r, "F(\phi) = id_{\mathbb{R}} \times \phi"] & F(\mathbb{R}^\omega) = \mathbb{R} \times \mathbb{R}^\omega
\end{tikzcd}$$

Abbildung 2: Kommutatives Diagramm zur Veranschaulichung des F-Coalgebra-Homomorphismus ϕ

Nun lässt sich ein Kandidat für die Lösung des geforderten Gleichungssystems konstruieren. Für jedes $f \in \Sigma$ setze $\hat{f}\colon (\mathbb{R}^\omega)^{r_f} \to \mathbb{R}^\omega$, $\hat{f}(s_1, \ldots, s_{r_f}) = \phi(f(s_1, \ldots, s_{r_f}))$.

Zu zeigen bleibt:

- Der Kandidat entspricht einer Lösung des Gleichungssystems.
- Die Lösung ist eindeutig.

Die Eindeutigkeit der Lösung folgt aus der Eindeutigkeit von ϕ und der (hier unbewiesenen) Tatsache, dass aufgrund der Definition von υ jede Lösung des Stream-Differentialgleichungssystems einen F-Coalgebra-Homomorphismus $\psi\colon U \to \mathbb{R}^\omega$ mit
$$\psi(u) = \begin{cases} u & \text{,wenn } u \in \mathbb{R}^\omega \\ \hat{f}(\psi(u_1), \ldots, \psi(u_{r_f})) & \text{,wenn } u = f(u_1, \ldots, u_{r_f}) \end{cases}$$
induzieren würde.

Nun wird gezeigt, dass der Kandidat tatsächlich einer Lösung des Gleichungssystems entspricht. Betrachte zunächst die Relation $R = \{(s, \phi(s)) \mid s \in \mathbb{R}^\omega\}$. Aus der Gleichheit

$$\begin{aligned}
(\phi(s)(0), \phi(s)') &= \sigma(\phi(s)) && \text{Definition von } \sigma \\
&= (\sigma \circ \phi)(s) \\
&= (F(\phi) \circ \upsilon)(s) && \text{Homomorphismuseigenschaft} \\
&= (id_{\mathbb{R}} \times \phi)(\upsilon(s)) && \text{Applikation von } F \\
&= (id_{\mathbb{R}} \times \phi)(\sigma(s)) && \text{Definition von } \upsilon \text{ auf } \mathbb{R}^\omega \\
&= (s(0), \phi(s')) && \text{Definition von } id_{\mathbb{R}} \text{ und } \sigma(s)
\end{aligned}$$

folgt $\forall (s, \phi(s)) \in R : s(0) = \phi(s)(0) \wedge (s', \phi(s)') = (s', \phi(s')) \in R$.
Damit ist die Relation R eine Bisimulation auf Streams und es folgt coinduktiv $\forall s \in \mathbb{R}^\omega : s = \phi(s)$.
Seien nun $s_i \in \mathbb{R}^\omega, 1 \leq i \leq r_f$, dann gilt:

$$\begin{aligned}
\sigma(\hat{f}(s_1, \ldots, s_{r_f})) &= \sigma(\phi(f(s_1, \ldots, s_{r_f}))) && \text{Definition von } \hat{f} \\
&= (\sigma \circ \phi)(f(s_1, \ldots, s_{r_f})) \\
&= (F(\phi) \circ \upsilon)(f(s_1, \ldots, s_{r_f})) && \text{Homomorphismuseigenschaft} \\
&= F(\phi)(\upsilon(f(s_1, \ldots, s_{r_f}))) \\
&= (id_{\mathbb{R}} \times \phi)(h(\pi_1(\upsilon(s_1)), \ldots, \pi_1(\upsilon(s_{r_f})), \tilde{t_f}) && \text{Definitionen von } F \text{ und } \upsilon \\
&= (id_{\mathbb{R}} \times \phi)(h(\pi_1(\sigma(s_1)), \ldots, \pi_1(\sigma(s_{r_f})), \tilde{t_f}) && \upsilon|_{\mathbb{R}^\omega} = \sigma \\
&= (id_{\mathbb{R}} \times \phi)(h(s_1(0), \ldots, s_{r_f}(0)), \tilde{t_f}) && \text{Definition von } \sigma \\
&= (h(s_1(0), \ldots, s_{r_f}(0)), \phi(\tilde{t_f}))
\end{aligned}$$

$= (h(s_1(0), \ldots, s_{r_f}(0)), \hat{t_f})$ Verträglichkeit mit σ, hier unbewiesen, siehe [2](Appendix A, Kompositionalität)

□

Bemerkung. In [4] bezeichnet der Begriff der "admissible equations" eine Verallgemeinerung der wohlgeformten Stream-Differentialgleichungssysteme. Bei diesen werden für die Beschreibung des Startwerts $f(x_1, \ldots, x_{r_f})(0)$ beliebige Terme zugelassen, die bis auf die Variablen $(x_1, \ldots, x_{r_f}, x'_1, \ldots, x'_{r_f}, x_1(0), \ldots, x_{r_f}(0))$, keine Startwerte und Ableitungen enthalten.
Womöglich lässt sich der begriff der "admissible equations" noch weiter verallgemeinern, indem weitere Anwendung des Startwerts und Ableitungen auf die Variablen zugelassen werden z.B. $x'(0), x''(0), \ldots$.

3.3.1 Folgerungen aus dem Existenz und Eindeutigkeitssatz

Mit dem Existenz und Eindeutigkeitssatz lässt sich nun die Wohldefiniertheit der eingeführten Streamoperatoren beweisen. Das Ziel ist es, durch die von den Operatoren geforderten Gleichheiten ein wohlgeformtes Stream-Differentialgleichungssystem aufzustellen. Wähle dazu die Menge der Funktionssymbole $\Sigma = \{\underline{r} \mid r \in \mathbb{R}\} \cup \{X, +, -, \times\}$. Die Stelligkeiten der Funktionssymbole ergeben sich direkt aus den Anforderungen an die entsprechenden Operatoren. Stelle nun ein wohlgeformtes Stream-Differentialgleichungssystem auf:

$$\begin{aligned}
\forall r \in \mathbb{R} : \underline{r}()(0) &= r \\
\forall r \in \mathbb{R} : \underline{r}()' &= \underline{0} \\
X()(0) &= 0 \\
X()' &= \underline{1} \\
+(x_1, x_2)(0) &= x_1(0) + x_2(0) \\
+(x_1, x_2)' &= +(x'_1, x'_2) \\
-(x_1)(0) &= -(x_1(0)) \\
-(x_1)' &= -(x'_1) \\
\times(x_1, x_2)(0) &= x_1(0) * x_2(0) \\
\times(x_1, x_2)' &= +(\times(x'_1, x_2), \times(\underline{x_1(0)}, x'_2)
\end{aligned}$$

Aus dem Existenz und Eindeutigkeitssatz folgt nun unmittelbar, dass die Funktionssymbole aus Σ jeweils genau eine Lösung besitzen, die das Gleichungssystem erfüllt.
Bisher hat sich ein wohlgeformtes Stream-Differentialgleichungssystem direkt aus den Anforderungen an die Operatoren ergeben. Allerdings besitzen geforderte Gleichheiten anderer Operatoren eine andere Form. Um sie zu transformieren, müssen nötige Umformungsregeln der bisherigen Operatoren bewiesen werden.

3.4 Die abelsche Gruppe $(\mathbb{R}^\omega, +)$

Lemma 1. $(\mathbb{R}^\omega, +)$ *ist eine abelsche Gruppe.*

Beweis. Alle Eigenschaften einer abelschen Gruppe lassen sich coinduktiv beweisen.

- Assoziativität: Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{((s+t)+u, s+(t+u)) \mid s,t,u \in \mathbb{R}^\omega\}$.
  Es gilt $\forall s,t,u \in \mathbb{R}^\omega$:

$$\begin{aligned} ((s+t)+u)(0) &= (s(0)+t(0))+u(0) \\ &= s(0)+(t(0)+u(0)) \\ &= (s+(t+u))(0) \end{aligned}$$

$$\begin{aligned} ((s+t)+u)' &= (s'+t')+u' && \text{Definition von } + \\ &R\; s'+(t'+u') && \text{Definition von } R \\ &= (s+(t+u))' && \text{Definition von } + \end{aligned}$$

  Damit ist R eine Bisimulation auf Streams und folglich $\forall s,t,u : (s+t)+u = s+(t+u)$

- Kommutativität: Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s+t, t+s) \mid s,t \in \mathbb{R}^\omega\}$.
  Es gilt $\forall s,t \in \mathbb{R}^\omega$:

$$\begin{aligned} (s+t)(0) &= s(0)+t(0) \\ &= t(0)+s(0) \\ &= (t+s)(0) \end{aligned}$$

$$\begin{aligned} (s+t)' &= s'+t' && \text{Definition von } + \\ &R\; t'+s' && \text{Definition von } R \\ &= (t+s)' && \text{Definition von } + \end{aligned}$$

  Damit ist R eine Bisimulation auf Streams und folglich $\forall s,t : s+t = t+s$

- Neutrales Element: $\underline{0}$. Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s, s+\underline{0}) \mid s \in \mathbb{R}^\omega\}$.
  Es gilt $\forall s \in \mathbb{R}^\omega$:

$$\begin{aligned} s(0) &= s(0)+0 \\ &= s(0)+\underline{0}(0) \\ &= (s+\underline{0})(0) \end{aligned}$$

$$\begin{aligned} s' &R\; s'+\underline{0} && \text{Definition von } R \\ &= s'+\underline{0}' && \text{Definition von } \underline{0}' \\ &= (s+\underline{0})' && \text{Definition von } + \end{aligned}$$

  Damit ist R eine Bisimulation auf Streams und zusammen mit der Kommutativität gilt $\forall s : s = s+\underline{0} = \underline{0}+s$.

- Inverses Element: $-s$. Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s+(-s), \underline{0}) \mid s \in \mathbb{R}^\omega\}$.

Es gilt $\forall s \in \mathbb{R}^\omega$:

$$\begin{aligned}(s + (-s))(0) &= s(0) + (-(s(0))) \\ &= 0 = \underline{0}(0)\end{aligned}$$

$$\begin{aligned}(s + (-s))' &= s' + (-(s')) && \text{Definition von } + \text{ und } - \\ &R\, \underline{0} && \text{Definition von } R \\ &= \underline{0}' && \text{Definition von } \underline{0}'\end{aligned}$$

Damit ist R eine Bisimulation auf Streams und zusammen mit der Kommutativität gilt $\forall s : \underline{0} = s + (-s) = (-s) + s$.

□

Bemerkung. Da $(\mathbb{R}^\omega, +)$ eine abelsche Gruppe ist, ist die Differenz (ebenfalls mit $-$ bezeichnet) implizit definiert durch: $\forall s, t \in \mathbb{R}^\omega : s - t = s + (-t)$.

3.5 Verallgemeinerung der Bisimulation

Bevor weitere Eigenschaften der Operatoren gezeigt werden können, muss der Begriff der Bisimulation (siehe [1]) verallgemeinert werden.

Definition 9 (Σ-Bisimulation auf Streams)**.** Eine Relation $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ heißt Σ-Bisimulation auf Streams, falls $\forall s, t \in \mathbb{R}^\omega : s\, R\, t \Rightarrow s(0) = t(0) \wedge$
$\exists n \in \mathbb{N}, s_0, \ldots, s_n, t_0, \ldots, t_n \in \mathbb{R}^\omega : s' = s_0 + \ldots + s_n, t' = t_0 + \ldots + t_n$ und
$s_i = t_i \vee s_i\, R\, t_i$ für $0 \leq i \leq n$.

Satz 4 (Σ-Coinduktion auf Streams)**.** *Für $s, t \in \mathbb{R}^\omega$ gilt:*
$\exists R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ Σ-Bisimulation auf Streams mit $s\, R\, t \Rightarrow s = t$

Beweis nach Skizze in [1](Theorem 4.2). Sei $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ eine Σ-Bisimulation. Definiere induktiv die kleinste Relation $Q \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$, sodass gilt:

1. $R \subseteq Q$
2. $\{(s, s) \mid s \in \mathbb{R}^\omega\} \subseteq Q$
3. $\forall n \in \mathbb{N}, s_0, \ldots, s_n, t_0, \ldots, t_n \in \mathbb{R}^\omega :$
   $(s_0, t_0) \in Q, \ldots, (s_n, t_n) \in Q \Rightarrow (s_0 + \ldots + s_n, t_0 + \ldots + t_n) \in Q$

Nun wird gezeigt, dass Q eine Bisimulation auf Streams ist. Seien dazu $s, t \in \mathbb{R}^\omega$ mit $(s, t) \in Q$, dann gilt:

Fall 1: $(s, t) \in R \Rightarrow s(0) = t(0) \wedge \exists n \in \mathbb{N}, s_0, \ldots, s_n, t_0, \ldots, t_n \in \mathbb{R}^\omega : s' = s_0 + \ldots + s_n, t' = t_0 + \ldots + t_n$ und
$s_i = t_i \vee s_i\, R\, t_i$ für $0 \leq i \leq n$. Daraus folgt unmittelbar, dass $(s', t') \in Q$.

Fall 2: $(s, t) \in \{(s, s) \mid s \in \mathbb{R}^\omega\} \Rightarrow s(0) = t(0) \wedge (s', t') = (s', s') \in Q$.

Fall 3: $\exists n \in \mathbb{N}, s_0, \ldots, s_n, t_0, \ldots, t_n \in \mathbb{R}^\omega : (s_0, t_0) \in Q, \ldots, (s_n, t_n) \in Q, s = s_0 + \ldots + s_n, t = t_0 + \ldots + t_n$. Induktiv folgt, dass $s(0) = s_0(0) + \ldots + s_n(0) = t_0(0) + \ldots + t_n(0) = t(0)$ und $(s', t') = (s_0' + \ldots + s_n', t_0' + \ldots + t_n') \in Q$.

In allen drei Fällen erfüllt Q die nötigen Forderungen und ist eine Bisimulation auf Streams. Deshalb gilt $\forall s,t \in \mathbb{R}^\omega : (s,t) \in R \Rightarrow (s,t) \in Q \Rightarrow s = t$. $\square$

Im Folgenden wird in Beweisen, die die Σ-Coinduktion verwenden, als Relationssymbol bei n Summanden $\underbrace{R + \ldots + R}_{\text{n Mal}}$ oder ΣR benutzt.

Distributivgesetz

In diesem Abschnitt wird gezeigt, dass das Distributivgesetz zwischen $+$ und $\times$ gilt.
Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s \times (t+u), (s \times t) + (s \times u)) \mid s,t,u \in \mathbb{R}^\omega\}$.
Es gilt $\forall s,t,u \in \mathbb{R}^\omega$:

$$\begin{aligned}
(s \times (t+u))(0) &= s(0) * (t(0) + u(0)) \\
&= s(0) * t(0) + s(0) * u(0) \\
&= ((s \times t) + (s \times u))(0)
\end{aligned}$$

$$\begin{aligned}
(s \times (t+u))' &= (s' \times (t+u)) + (\underline{s(0)} \times (t+u)') && \text{Definition von } \times \\
&\mathrel{R+R} ((s' \times t) + (s' \times u)) + ((\underline{s(0)} \times t') + (\underline{s(0)} \times u')) && \text{Definition von } R+R \\
&= (s' \times t) + (\underline{s(0)} \times t') + (s' \times u) + (\underline{s(0)} \times u') && \text{Assoziativität und Kommutativität von } + \\
&= (s \times t)' + (s \times u)' && \text{Definition von } \times \\
&= ((s \times t) + (s \times u))' && \text{Definition von } +
\end{aligned}$$

Damit ist R eine Σ-Bisimulation auf Streams und $\forall s,t,u \in \mathbb{R}^\omega : s \times (t+u) = (s \times t) + (s \times u)$. Analog lässt sich $\forall s,t,u \in \mathbb{R}^\omega : (t+u) \times s = (t \times s) + (u \times s)$ zeigen.
Zusätzlich gilt für das additive Inverse: $\forall s,t \in \mathbb{R}^\omega : (-s) \times t = -(s \times t) = s \times (-t)$.
Um das zu zeigen, definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(-(s \times t), s \times (-t)) \mid s,t \in \mathbb{R}^\omega\}$.
Es gilt $\forall s,t \in \mathbb{R}^\omega$:

$$\begin{aligned}
(-(s \times t))(0) &= -(s(0) * t(0)) \\
&= s(0) * (-t(0)) \\
&= (s \times (-t))(0)
\end{aligned}$$

$$\begin{aligned}
(-(s \times t))' &= -((s \times t)') && \text{Definition von } - \\
&= -((s' \times t) + (\underline{s(0)} \times t')) && \text{Definition von } \times \\
&= -(s' \times t) + (-(\underline{s(0)} \times t')) && \text{Gruppeneigenschaft von } (\mathbb{R}^\omega, +) \\
&\mathrel{R+R} (s' \times (-t)) + (\underline{s(0)} \times (-(t'))) && \text{Definition von } R \\
&= (s' \times (-t)) + (\underline{s(0)} \times (-(t))') && \text{Definition von } - \\
&= (s \times (-t))' && \text{Definition von } \times
\end{aligned}$$

Damit ist R eine Σ-Bisimulation auf Streams und $\forall s,t \in \mathbb{R}^\omega : -(s \times t) = s \times (-t)$. Analog lässt sich $\forall s,t \in \mathbb{R}^\omega : -(s \times t) = (-s) \times t$ zeigen.
Eine direkte Konsequenz der obigen Aussagen ist
$\forall s \in \mathbb{R}^\omega : s \times \underline{0} = s \times (\underline{1} + (-\underline{1})) = (s \times \underline{1}) + (-(s \times \underline{1})) = \underline{0}$ (Nullelement).

3.6 Der Monoid $(\mathbb{R}^\omega, \times)$

Lemma 2. $(\mathbb{R}^\omega, \times)$ *ist ein Monoid.*

Beweis. Alle Eigenschaften eines Monoids lassen sich coinduktiv beweisen.

- Assoziativität: Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{((s \times t) \times u, s \times (t \times u)) \mid s, t, u \in \mathbb{R}^\omega\}$.
Es gilt $\forall s, t, u \in \mathbb{R}^\omega$:

$$\begin{aligned}
&((s \times t) \times u)(0) \\
&= (s(0) * t(0)) * u(0) \\
&= s(0) * (t(0) * u(0)) \\
&= (s \times (t \times u))(0)
\end{aligned}$$

$$\begin{aligned}
&((s \times t) \times u)' && \\
&= ((s \times t)' \times u) + ((s \times t)(0) \times u') && \text{Definition von } \times \\
&= (((s' \times t) + (\underline{s(0)} \times t')) \times u) + ((s \times t)(0) \times u') && \text{Definition von } \times \\
&= ((s' \times t) \times u) + ((\underline{s(0)} \times t') \times u) + ((\underline{s(0)} \times \underline{t(0)}) \times u') && \text{Distributivgesetz} \\
&R + R + R\ (s' \times (t \times u)) + (\underline{s(0)} \times (t' \times u)) + (\underline{s(0)} \times (\underline{t(0)} \times u')) && \text{Definition von } R \\
&= (s' \times (t \times u)) + (\underline{s(0)} \times ((t' \times u) + (\underline{t(0)} \times u'))) && \text{Distributivgesetz} \\
&= (s' \times (t \times u)) + (\underline{s(0)} \times (t \times u)') && \text{Definition von } \times \\
&= (s \times (t \times u))' && \text{Definition von } \times
\end{aligned}$$

Damit ist R eine Σ-Bisimulation auf Streams und folglich $\forall s, t, u : (s \times t) \times u = s \times (t \times u)$

- Neutrales Element $\underline{1}$: Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s \times \underline{1}, s) \mid s \in \mathbb{R}^\omega\}$.
Es gilt $\forall s \in \mathbb{R}^\omega$:

$$(s \times \underline{1})(0) = s(0)$$

$$\begin{aligned}
(s \times \underline{1})' &= (s' \times \underline{1}) + (\underline{s(0)} \times \underline{1}') && \text{Definition von } \times \\
&= (s' \times \underline{1}) + (\underline{s(0)} \times \underline{0}) && \text{Definition von } \underline{1}' \\
&= (s' \times \underline{1}) && \text{Nullelement } \underline{0} \\
&R\ s' && \text{Definition von } R
\end{aligned}$$

Damit ist R eine Bisimulation auf Streams und $\forall s : s = s \times \underline{1} = s$. Analog lässt sich $\forall s : s = \underline{1} \times s$ beweisen.

□

3.7 Fundamentalsatz

Durch die coalgebraische Struktur des Stromkalküls sind die beiden Operatoren "Startwert" und "Ableitung" unmittelbar gegeben. Die Bezeichnung dieser Operatoren lässt auf die Analogie zur Analysis schließen, die sich mit Differentiation und Integration von Funktionen beschäftigt. Einer der zentralen Sätze der Analysis ist ihr Fundamentalsatz, der den Zusammenhang zwischen Differentiation und Integration beschreibt.

Satz 5 (Fundamentalsatz der Analysis)**.** *Sei* $[a, b] \subset \mathbb{R}, f\colon [a, b] \to \mathbb{R}$ *stetig,* $c \in [a, b]$*, dann ist* $F\colon [a, b] \to \mathbb{R}, F(x) = \int_c^x f(t)dt$ *differenzierbar und es gilt*

1. $F'(x) = f(x)\ \forall x \in [a, b]$

2. $\int_a^b f(t)dt = F(b) - F(a)$

Insbesondere gilt $F(x) = F(a) + \int_a^x F'(t)dt$

Dieser Satz lässt sich (siehe [1](Theorem 4.1)) auf den Stromkalkül übertragen indem die analytische Integration durch das Faltungsprodukt mit der formalen Variable beschrieben wird.

Satz 6 (Fundamentalsatz des Stromkalküls)**.** $\forall s \in \mathbb{R}^\omega : s = \underline{s(0)} + (X \times s')$

Beweis. Sei $s \in \mathbb{R}^\omega$, dann gilt:

- $(\underline{s(0)} + (X \times s'))(0) = s(0) + X(0) * (s')(0) = s(0) + 0 * (s')(0) = s(0)$
- $(\underline{s(0)} + (X \times s'))' = \underline{s(0)}' + (X \times s')' = \underline{0} + (X' \times s') + (\underline{X(0)} \times (s')') = (\underline{1} \times s') + (\underline{0} \times (s')') = (\underline{1} \times s') = s'$

Daraus folgt nach Korollar 1: $s = \underline{s(0)} + (X \times s')$. □

Bemerkung. Der Fundamentalsatz beschreibt den Zusammenhang zwischen einem Stream, seinem Startwert und dem "Integral" seiner Ableitung. Das "Integral" ist dabei das Faltungsprodukt mit der formalen Variable X. Eine ähnliche Aussage lässt sich über die Ableitung des "Integrals" eines Streams formulieren:
$\forall s \in \mathbb{R}^\omega : (X \times s)' = (X' \times s) + (\underline{X(0)} \times s') = (\underline{1} \times s) + (\underline{0} \times s') = s.$
Mit anderen Worten ist die Ableitung des "Integrals" eines Streams der Stream selbst.

3.8 Der Ring $(\mathbb{R}^\omega, +, \times)$

Lemma 3. $(\mathbb{R}^\omega, +, \times)$ *ist ein kommutativer unitärer Ring.*

Beweis. Aus Lemma 1 und Lemma 2 folgt bereits, dass $(\mathbb{R}^\omega, +)$ eine abelsche Gruppe und $(\mathbb{R}^\omega, \times)$ ein Monoid ist. Das Distributivgesetz wurde auch bereits gezeigt. Nun bleibt es, die Kommutativität von $\times$ nachzuweisen. Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s \times t, t \times s) \mid s, t \in \mathbb{R}^\omega\}$.

Es gilt $\forall s, t \in \mathbb{R}^\omega$:

$$
\begin{aligned}
(s \times t)(0) &= s(0) * t(0) \\
&= t(0) * s(0) \\
&= (t \times s)(0)
\end{aligned}
$$

$$
\begin{aligned}
(s \times t)' &= (s \times (\underline{t(0)} + (X \times t')))' && \text{Fundamentalsatz} \\
&= ((s \times \underline{t(0)}) + (s \times (X \times t')))' && \text{Distributivgesetz} \\
&= (s \times \underline{t(0)})' + (s \times (X \times t'))' && \text{Definition von } + \\
&= (s' \times \underline{t(0)}) + (\underline{s(0)} \times \underline{t(0)}') + (s \times (X \times t'))' && \text{Definition von } \times \\
&= (s' \times \underline{t(0)}) + (\underline{s(0)} \times \underline{0}) + (s \times (X \times t'))' && \text{Definition von } \_ \\
&= (s' \times \underline{t(0)}) + (s' \times (X \times t')) + (\underline{s(0)} \times (X \times t')') && \text{Definition von } \times \\
&= (s' \times \underline{t(0)}) + (s' \times (X \times t')) + (\underline{s(0)} \times t') && \text{Fundamentalsatz(Ableitung des "Integrals")} \\
&= (s' \times \underline{t(0)}) + ((s' \times X) \times t') + (\underline{s(0)} \times t') && \text{Assoziativität} \\
&= (s' \times \underline{t(0)}) + ((s' \times X) + \underline{s(0)}) \times t') && \text{Distributivgesetz} \\
&= (s' \times \underline{t(0)}) + (s \times t') && \text{Fundamentalsatz} \\
&R + R\ (\underline{t(0)} \times s') + (t' \times s) && \text{Definition von } R \\
&= (t \times s)' && \text{Definition von } \times
\end{aligned}
$$

Damit ist R eine Σ-Bisimulation auf Streams und folglich $\forall s, t : s \times t = t \times s$ $\square$

3.9 Einbettung als Ringhomomorphismus

Lemma 4. *Die Einbettung* $\_ : \mathbb{R} \to \mathbb{R}^\omega$ *ist ein Ringhomomorphismus zwischen* $(\mathbb{R}, +, *)$ *und* $(\mathbb{R}^\omega, +, \times)$.

Beweis. Seien $s, r \in \mathbb{R}$.

- Verträglichkeit mit Eins: $\underline{1} = \underline{1}$
- Verträglichkeit mit $+$:

$$
\begin{aligned}
(\underline{s+r})(0) = s + r = \underline{s}(0) + \underline{r}(0) = (\underline{s} + \underline{r})(0) \qquad (i) \\
(\underline{s+r})' = \underline{0} = \underline{0} + \underline{0} = \underline{s}' + \underline{r}' = (\underline{s} + \underline{r})' \qquad (ii)
\end{aligned}
$$

  Aus $(i), (ii)$ und Korollar 1 folgt $\underline{s+r} = \underline{s} + \underline{r}$.

- Verträglichkeit mit $\times$:

$$
\begin{aligned}
(\underline{s*r})(0) = s * r = \underline{s}(0) * \underline{r}(0) = (\underline{s} \times \underline{r})(0) \qquad (iii) \\
(\underline{s*r})' = \underline{0} = \underline{0} + \underline{0} = (\underline{0} \times \underline{r}) + (\underline{s} \times \underline{0}) = (\underline{s}' \times \underline{r}) + (\underline{s(0)} \times \underline{r}') = (\underline{s} \times \underline{r})' \qquad (iv)
\end{aligned}
$$

  Aus $(iii), (iv)$ und Korollar 1 folgt: $\underline{s*r} = \underline{s} \times \underline{r}$.

$\square$

3.10 Weitere Eigenschaften von $\times$ in $\mathbb{R}^\omega$

Es wurde bereits gezeigt, dass $(\mathbb{R}^\omega, +, \times)$ ein kommutativer unitärer Ring ist. Leider lässt sich diese Struktur nicht zu einem Körper erweitern. Der Grund dafür ist, dass z.B. die formale Variable X zwar ungleich $\underline{0}$ ist, aber kein Inverses bezüglich $\times$ in $\mathbb{R}^\omega$ besitzt. Die Gleichheit $X^{-1}(0) * X(0) = \underline{1}(0)$ bzw. $X^{-1}(0) * 0 = 1$ kann durch kein $X^{-1}(0) \in \mathbb{R}$ erfüllt werden. Nichtsdestotrotz besitzt $\mathbb{R}^\omega$ eine Einheitenmenge, die im Folgenden identifiziert wird.

Lemma 5. $(\{s \in \mathbb{R}^\omega \mid s(0) \neq 0\}, \times)$ *ist eine abelsche Gruppe.*

Beweis. Bezeichne $(\{s \in \mathbb{R}^\omega \mid s(0) \neq 0\}$ als $\mathbb{R}^\omega_{\neq 0}$.
Zuerst muss gezeigt werden, dass $\mathbb{R}^\omega_{\neq 0}$ unter der Anwendung von $\times$ abgeschlossen ist. Seien dazu $s, t \in \mathbb{R}^\omega_{\neq 0}$, dann gilt $(s \times t)(0) = s(0) * t(0) \neq 0 \Rightarrow (s \times t) \in \mathbb{R}^\omega_{\neq 0}$.
Da bereits gezeigt wurde, dass $(\mathbb{R}^\omega, \times)$ ein Monoid ist, bleibt es nur zu zeigen,
$\forall s \in \mathbb{R}^\omega_{\neq 0}, \exists s^{-1} \in \mathbb{R}^\omega_{\neq 0} : \underline{1} = s \times s^{-1} = s^{-1} \times s$.
Diese Forderung entspricht der Definition des Operators $\_^{-1}$ und wird im Folgenden so umgeformt, dass sie ein wohlgeformtes Stream-Differentialgleichungssystem beschreibt.
Nach Korollar 1 gilt: $\underline{1} = s \times s^{-1} \Leftrightarrow (\underline{1}(0) = (s \times s^{-1})(0)) \wedge (\underline{1}' = (s \times s^{-1})')$

$$
\begin{aligned}
&(s \times s^{-1})(0) &&= \underline{1}(0) \\
\Leftrightarrow\, &s(0) * s^{-1}(0) &&= 1 \\
\Leftrightarrow\, &s^{-1}(0) &&= \frac{1}{s(0)} && (i)
\end{aligned}
$$

$$
\begin{aligned}
&(s \times s^{-1})' &&= \underline{1}' \\
\Leftrightarrow\, &(s' \times s^{-1}) + (\underline{s(0)} \times (s^{-1})') &&= \underline{0} && \text{Definition von } \times \\
\Leftrightarrow\, &\underline{s(0)} \times (s^{-1})' &&= -(s' \times s^{-1}) && \text{Beidseitige Termaddition} \\
\Leftrightarrow\, &\underline{\frac{1}{s(0)}} \times \underline{s(0)} \times (s^{-1})' &&= \underline{\frac{1}{s(0)}} \times (-(s' \times s^{-1})) && \text{Beidseitige Termmultiplikation} \\
\Leftrightarrow\, &\underline{\frac{s(0)}{s(0)}} \times (s^{-1})' &&= \underline{\frac{1}{s(0)}} \times (-(s' \times s^{-1})) && \text{Einbettung als Ringhomomorphie} \\
\Leftrightarrow\, &(s^{-1})' &&= \underline{\frac{1}{s(0)}} \times (-(s' \times s^{-1})) && \rightsquigarrow (ii)
\end{aligned}
$$

Die Gleichheiten $(i), (ii)$ besitzen nun die Form von wohlgeformten Stream-Differentialgleichungen und aus dem Existenz und Eindeutigkeitssatz folgt, dass der auf obige Weise beschriebene Operator existiert und in $\mathbb{R}^\omega_{\neq 0}$ eindeutig ist. $\square$

Auch die Bedingungen an den $\sqrt[\times]{\_}$ Operator lassen sich in wohlgeformte Stream-Differentialgleichungen transformieren. Es soll gelten $\forall s \in \mathbb{R}^\omega, s(0) > 0$:

$$
\begin{aligned}
(\sqrt[\times]{s} \times \sqrt[\times]{s})(0) &= s(0) \\
\sqrt[\times]{s}(0) \times \sqrt[\times]{s}(0) &= s(0) \\
\sqrt[\times]{s}(0) &= \sqrt{s(0)} && (i)
\end{aligned}
$$

$$
(\sqrt[\times]{s} \times \sqrt[\times]{s})' = s'
$$

$$
\begin{array}{rcll}
(\sqrt[\times]{s}' \times \sqrt[\times]{s}) + (\underline{\sqrt[\times]{s(0)}} \times \sqrt[\times]{s}') = & & s' & \text{Definition von } \times \\
\sqrt[\times]{s}' \times (\sqrt[\times]{s} + \underline{\sqrt{s(0)}}) = & & s' & \text{Distributivgesetz} \\
\sqrt[\times]{s}' = & & s' \times (\sqrt[\times]{s} + \underline{\sqrt{s(0)}})^{-1} & \text{Beidseitige Termmultiplikation} \rightsquigarrow (ii)
\end{array}
$$

Die Gleichheiten $(i), (ii)$ besitzen nun die Form von wohlgeformten Stream-Differentialgleichungen.

4 Formale Potenzreihen

Die Menge der formalen Potenzreihen $\mathbb{R}[[t]] = \{\sum_{i=0}^{\infty} a(i) * t^i \mid a \in \mathbb{R}^\omega\}$ verallgemeinert die Menge der Polynome, indem eine nicht endliche Anzahl an Koeffizienten zugelassen wird.
Die Summe $+ \colon \mathbb{R}[[t]] \times \mathbb{R}[[t]] \to \mathbb{R}[[t]]$, mit
$\sum_{i=0}^{\infty} a(i) * t^i + \sum_{i=0}^{\infty} b(i) * t^i = \sum_{i=0}^{\infty} (a(i) + b(i)) * t^i$
und das Faltungsprodukt $\times \colon \mathbb{R}[[t]] \times \mathbb{R}[[t]] \to \mathbb{R}[[t]]$, mit
$\sum_{i=0}^{\infty} a(i) * t^i \times \sum_{i=0}^{\infty} b(i) * t^i = \sum_{i=0}^{\infty} c(i) * t^i$, wobei
$c(n) = \sum_{i=0}^{n} a(i) * b(n-i)$
sind auf formalen Potenzreihen wohldefiniert und bilden mit ihnen einen Ring. Zudem besitzt $\mathbb{R}[[t]]$ eine nichtleere Einheitenmenge bezüglich $\times$.

Satz 7 (Ringisomorphismus)**.** *Die Ringe* $(\mathbb{R}^\omega, +, \times)$ *und* $(\mathbb{R}[[t]], +, \times)$ *sind isomorph.*

Beweis. Definiere $\phi \colon \mathbb{R}[[t]] \to \mathbb{R}^\omega$ durch
$\phi(\sum_{i=0}^{\infty} a(i) * t^i)(0) = a(0)$
$\phi(\sum_{i=0}^{\infty} a(i) * t^i)' = \phi(\sum_{i=0}^{\infty} a(i+1) * t^i)$.
Da die Menge der Funktionssymbole $\{\phi(a) \in \mathbb{R}^\omega \mid a \in \mathbb{R}[[t]]\}$ mit den auf obige Weise definierten Gleichheiten ein wohlgeformtes Stream-Differentialgleichungssystem bildet, folgt aus dem Existenz und Eindeutigkeitssatz die Existenz und Eindeutigkeit von ϕ.
Zeige nun, dass ϕ ein Ring-Homomorphismus ist.

- Verträglichkeit mit der Eins:
  Definiere $R = \{(\phi(\sum_{i=0}^{\infty} a(i) * t^i), \underline{r}) \mid r \in \mathbb{R}, a \in \mathbb{R}^\omega, a(0) = r, \forall i \geq 1 : a(i) = 0\}$. Dann gilt:

$$
\begin{aligned}
\phi(\sum_{i=0}^{\infty} a(i) * t^i)(0) &= r = \underline{r}(0) & \\
\phi(\sum_{i=0}^{\infty} a(i) * t^i)' &= \phi(\sum_{i=0}^{\infty} a(i+1) * t^i) & \text{Definition von } \phi \\
&= \phi(\sum_{i=0}^{\infty} 0 * t^i) & \forall i \geq 1 : a(i) = 0
\end{aligned}
$$

$$
\begin{aligned}
& R\ \underline{0} && \text{Definition von R} \\
& = \underline{r}' && \text{Definition der Einbettung}
\end{aligned}
$$

Damit ist R eine Bisimulation auf Streams und es folgt $\forall r \in \mathbb{R} : \phi(r * t^0) = \underline{r}$. Insbesondere gilt für die Eins im Ring $(\mathbb{R}[[t]], +, \times)$: $\phi(1 * t^0) = \underline{1}$.

- Verträglichkeit mit $+$:
  Definiere $R = \{(\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i + \sum\limits_{i=0}^{\infty} b(i) * t^i), \phi(\sum\limits_{i=0}^{\infty} a(i) * t^i) + \phi(\sum\limits_{i=0}^{\infty} b(i) * t^i)) \mid a, b \in \mathbb{R}^\omega\}$.
  Dann gilt $\forall a, b \in \mathbb{R}^\omega$:

$$
\begin{aligned}
\phi(\sum_{i=0}^{\infty} a(i) * t^i + \sum_{i=0}^{\infty} b(i) * t^i)(0) &= a(0) + b(0) \\
&= (\phi(\sum_{i=0}^{\infty} a(i) * t^i) + \phi(\sum_{i=0}^{\infty} b(i) * t^i))(0)
\end{aligned}
$$

$$
\begin{aligned}
\phi(\sum_{i=0}^{\infty} a(i) * t^i + \sum_{i=0}^{\infty} b(i) * t^i)' &= \phi(\sum_{i=0}^{\infty} (a(i) + b(i)) * t^i)' && \text{Summe in } \mathbb{R}[[t]] \\
&= \phi(\sum_{i=0}^{\infty} (a(i+1) + b(i+1)) * t^i) && \text{Definition von } \phi \\
&= \phi(\sum_{i=0}^{\infty} a(i+1) * t^i + \sum_{i=0}^{\infty} b(i+1) * t^i) && \text{Summe in } \mathbb{R}[[t]] \\
&R\ \phi(\sum_{i=0}^{\infty} a(i+1) * t^i) + \phi(\sum_{i=0}^{\infty} b(i+1) * t^i) && \text{Definition von } R \\
&= \phi(\sum_{i=0}^{\infty} a(i) * t^i)' + \phi(\sum_{i=0}^{\infty} b(i) * t^i)' && \text{Definition von } \phi \\
&= (\phi(\sum_{i=0}^{\infty} a(i) * t^i) + \phi(\sum_{i=0}^{\infty} b(i) * t^i))' && \text{Definition von } +
\end{aligned}
$$

  Damit ist R eine Bisimulation auf Streams und es folgt $\forall a, b \in \mathbb{R}[[t]] : \phi(a + b) = \phi(a) + \phi(b)$.

- Verträglichkeit mit $\times$:
  Definiere $R = \{(\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i \times \sum\limits_{i=0}^{\infty} b(i) * t^i), \phi(\sum\limits_{i=0}^{\infty} a(i) * t^i) \times \phi(\sum\limits_{i=0}^{\infty} b(i) * t^i)) \mid a, b \in \mathbb{R}^\omega\}$.
  Dann gilt $\forall a, b \in \mathbb{R}^\omega$:

$$
\begin{aligned}
&\phi(\sum_{i=0}^{\infty} a(i) * t^i \times \sum_{i=0}^{\infty} b(i) * t^i)(0) = a(0) * b(0) \\
&= (\phi(\sum_{i=0}^{\infty} a(i) * t^i) \times \phi(\sum_{i=0}^{\infty} b(i) * t^i))(0)
\end{aligned}
$$

$$
\phi((\sum_{i=0}^{\infty} a(i) * t^i) \times (\sum_{i=0}^{\infty} b(i) * t^i))'
$$

[Produkt in $\mathbb{R}[[t]]$]

$$= \phi(\sum_{i=0}^{\infty}(\sum_{j=0}^{i} a(j) * b(i-j)) * t^i)'$$

[Definition von ϕ]

$$= \phi(\sum_{i=0}^{\infty}(\sum_{j=0}^{i+1} a(j) * b(i+1-j)) * t^i)$$

[Zerlegung der Summe]

$$= \phi(\sum_{i=0}^{\infty}(a(0) * b(i+1) + \sum_{j=1}^{i+1} a(j) * b(i+1-j)) * t^i)$$

[Indextransformation]

$$= \phi(\sum_{i=0}^{\infty}(a(0) * b(i+1) + \sum_{j=0}^{i} a(j+1) * b(i-j)) * t^i)$$

[Verträglichkeit mit $+$]

$$= \phi(\sum_{i=0}^{\infty}(a(0) * b(i+1)) * t^i) + \phi(\sum_{i=0}^{\infty}(\sum_{j=0}^{i} a(j+1) * b(i-j)) * t^i)$$

[Kommutativität]

$$= \phi(\sum_{i=0}^{\infty}(\sum_{j=0}^{i} a(j+1) * b(i-j)) * t^i) + \phi(\sum_{i=0}^{\infty}(a(0) * b(i+1)) * t^i)$$

[Multiplikation in $\mathbb{R}[[t]]$]

$$= \phi((\sum_{i=0}^{\infty} a(i+1) * t^i) \times (\sum_{i=0}^{\infty} b(i) * t^i)) + \phi((a(0) * t^0) \times (\sum_{i=0}^{\infty} b(i+1) * t^i))$$

[Definition von R]

$$R + R\ (\phi(\sum_{i=0}^{\infty} a(i+1) * t^i) \times \phi(\sum_{i=0}^{\infty} b(i) * t^i)) + (\phi(a(0) * t^0)) \times \phi(\sum_{i=0}^{\infty} b(i+1) * t^i))$$

[Verträglichkeit mit Eins]

$$= (\phi(\sum_{i=0}^{\infty} a(i+1) * t^i) \times \phi(\sum_{i=0}^{\infty} b(i) * t^i)) + (\underline{a(0)} \times \phi(\sum_{i=0}^{\infty} b(i+1) * t^i))$$

[Definition von ϕ]

$$= (\phi(\sum_{i=0}^{\infty} a(i) * t^i)' \times \phi(\sum_{i=0}^{\infty} b(i) * t^i)) + (\underline{\phi(\sum_{i=0}^{\infty} a(i) * t^i)(0)} \times \phi(\sum_{i=0}^{\infty} b(i) * t^i)')$$

[Definition von $\times$]

$$= (\phi(\sum_{i=0}^{\infty} a(i) * t^i) \times \phi(\sum_{i=0}^{\infty} b(i) * t^i))'$$

Damit ist R eine Σ-Bisimulation auf Streams und es folgt $\forall a, b \in \mathbb{R}[[t]] : \phi(a \times b) = \phi(a) \times \phi(b)$.

Zeige nun, dass ϕ bijektiv ist. Definiere $\psi\colon \mathbb{R}^\omega \to \mathbb{R}[[t]]$ mit $\psi(a) = \sum\limits_{i=0}^{\infty} a(i) * t^i$.

Definiere $R = \{(\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i), a) \mid a \in \mathbb{R}^\omega\}$. Dann gilt:

$\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i)(0) = a(0)$ und

$\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i)' = \phi(\sum\limits_{i=0}^{\infty} a(i+1) * t^i) = \phi(\sum\limits_{i=0}^{\infty} (a')(i) * t^i)\ R\ a'$.

Damit ist R eine Bisimulation auf Streams und es folgt

Korollar 2. $\forall a \in \mathbb{R}^\omega : \phi(\sum\limits_{i=0}^{\infty} a(i) * t^i) = a$

Aus dieser Aussage folgt unmittelbar:

$\forall a \in \mathbb{R}^\omega : \phi(\psi(a)) = \phi(\sum\limits_{i=0}^{\infty} a(i) * t^i) = a \wedge \psi(\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i)) = \psi(a) = \sum\limits_{i=0}^{\infty} a(i) * t^i$.

Damit gilt $id_{\mathbb{R}^\omega} = \phi \circ \psi \wedge id_{\mathbb{R}[[t]]} = \psi \circ \phi$. □

Folgerungen aus der Isomorphie

Mit Hilfe der im Beweis verwendeten Aussagen lässt sich auf einfache Weise zeigen, dass für einen Stream $a \in \mathbb{R}^\omega$, mit der Eigenschaft $\exists n \in \mathbb{N} : (i > n \Rightarrow a(i) = 0)$ gilt:

$$
\begin{aligned}
a &= \phi(\sum_{i=0}^{\infty} a(i) * t^i) && \text{,da } \forall s \in \mathbb{R}^\omega : s = \phi(\sum_{i=0}^{\infty} s(i) * t^i) \\
&= \phi(\sum_{i=0}^{n} (a(i) \times t^i)) && \text{,da } i > n \Rightarrow a(i) = 0 \\
&= \sum_{i=0}^{n} (\phi(a(i)) \times \phi(t)^i) && \text{Verträglichkeit mit } + \text{ und } \times \\
&= \sum_{i=0}^{n} (\underline{a(i)} \times X^i) && \text{,da } \phi(t) = X \text{ und Verträglichkeit mit der Eins}
\end{aligned}
$$

Mit anderen Worten ist eine Folge $(a_0, a_1, \ldots, a_n, 0, 0, \ldots)$ gleich dem Polynom $\sum\limits_{i=0}^{n} (\underline{a_i} \times X^i)$.

Die Definitionen von Folgen und Reihen aus der Analysis liefern zusammen mit der obigen Aussage

Korollar 3.

$$
\begin{aligned}
a &= \lim_{n\to\infty} ((a(0), a(1), \ldots, a(n))) && \textit{Analytische Definition von Folgen} \\
&= \lim_{n\to\infty} \sum_{i=0}^{n} (\underline{a(i)} \times X^i) && \textit{Obige Aussage für Polynome} \\
&= \sum_{i=0}^{\infty} (\underline{a(i)} \times X^i) && \textit{Analytische Definition von Reihen}
\end{aligned}
$$

Während in dieser Aussage "$\sum\limits_{i=0}^{\infty}$" der Reihendefinition aus der Analysis entspricht, wird in [1](Theorem 4.3) "$\sum\limits_{i=0}^{\infty}$" als ein separater Streamoperator definiert.

Eine weitere Folgerung ist die explizite Darstellung der Folgenglieder $(a+b)(i)$ und $(a \times b)(i)$ für $a, b \in \mathbb{R}^\omega$. Es gilt $\forall n \in \mathbb{N}$:

$$\begin{aligned}
(a+b)(n) &= (\phi(\sum_{i=0}^{\infty} a(i) * t^i) + \phi(\sum_{i=0}^{\infty} b(i) * t^i))(n) && \text{da } \forall s \in \mathbb{R}^\omega : s = \phi(s) \\
&= (\phi(\sum_{i=0}^{\infty} a(i) * t^i + \sum_{i=0}^{\infty} b(i) * t^i))(n) && \text{Isomorphieeigenschaft} \\
&= \phi(\sum_{i=0}^{\infty} (a(i) + b(i)) * t^i)(n) && \text{Definition von } + \\
&= \phi(\sum_{i=0}^{\infty} c(i) * t^i)(n) && \text{für } c \in \mathbb{R}^\omega : c(i) = a(i) + b(i) \\
&= c(n) = a(n) + b(n)
\end{aligned}$$

$$\begin{aligned}
(a \times b)(n) &= (\phi(\sum_{i=0}^{\infty} a(i) * t^i) \times \phi(\sum_{i=0}^{\infty} b(i) * t^i))(n) && \text{da } \forall s \in \mathbb{R}^\omega : s = \phi(s) \\
&= (\phi(\sum_{i=0}^{\infty} a(i) * t^i \times \sum_{i=0}^{\infty} b(i) * t^i))(n) && \text{Isomorphieeigenschaft} \\
&= \phi(\sum_{i=0}^{\infty} (\sum_{j=0}^{i} a(j) * b(i-j)) * t^i)(n) && \text{Definition von } \times \\
&= \phi(\sum_{i=0}^{\infty} c(i) * t^i)(n) && \text{für } c \in \mathbb{R}^\omega : c(i) = \sum_{j=0}^{i} a(j) * b(i-j) \\
&= c(n) = \sum_{i=0}^{n} a(i) * b(n-i)
\end{aligned}$$

4.1 Geschlossene Form für lineare Rekursionsgleichungen

Formale Potenzreihen werden unter anderem verwendet, um eine geschlossene Form für Folgen zu finden, die durch lineare Rekursionsgleichungen gegeben sind. Eine lineare Rekursionsgleichung k-ter Ordnung mit dem Startvektor $(a_0, \ldots, a_{k-1})$, Koeffizienten $(c_0, \ldots, c_{k-1})$ mit $c_0 \neq 0$ und $b \in \mathbb{R}^\omega$ besitzt die Form
$\forall n < k : a(n) = a_n$
$\forall n \geq 0 : a(n+k) + \sum_{i=0}^{k-1} c_i * a(n+i) = b(n)$
und hat genau eine Lösung $a \in \mathbb{R}^\omega$.
Mit Hilfe des Stromkalküls lässt sich diese Gleichheit auf folgende Weise umformen (Skizze in [1](6. Application: solving difference equations), jedoch wird im Folgenden "$\sum_{i=0}^{\infty}$" im Gegensatz zu [1] als Reihenbegriff verwendet):

$$\forall n \geq 0 : a(n+k) + \sum_{i=0}^{k-1} c_i * a(n+i) = b(n)$$

[Beidseitige Multiplikation mit t^n und anschließende analytische Reihendarstellung]

$$\Rightarrow \sum_{i=0}^{\infty}(a(i+k) + \sum_{j=0}^{k-1} c_j * a(i+j)) * t^i = \sum_{i=0}^{\infty} b(i) * t^i$$

[Distributivgesetz und Summe formaler Potenzreihen]

$$\Rightarrow \sum_{i=0}^{\infty} a(i+k) * t^i + \sum_{j=0}^{k-1} c_j * \sum_{i=0}^{\infty} a(i+j) * t^i = \sum_{i=0}^{\infty} b(i) * t^i$$

[Beidseitige Anwendung des Ringhomomorphismus $\phi\colon \mathbb{R}[[t]] \to \mathbb{R}^{\omega}$]

$$\Rightarrow \phi(\sum_{i=0}^{\infty} a(i+k) * t^i + \sum_{j=0}^{k-1} c_j * \sum_{i=0}^{\infty} a(i+j) * t^i) = \phi(\sum_{i=0}^{\infty} b(i) * t^i)$$

[Homomorphismuseigenschaft von ϕ]

$$\Rightarrow \phi(\sum_{i=0}^{\infty} a(i+k) * t^i) + (\sum_{j=0}^{k-1} \underline{c_j} \times \phi(\sum_{i=0}^{\infty} a(i+j) * t^i)) = \phi(\sum_{i=0}^{\infty} b(i) * t^i)$$

[Eigenschaft der Ableitung]

$$\Rightarrow \phi(\sum_{i=0}^{\infty} a^{(k)}(i) * t^i) + (\sum_{j=0}^{k-1} \underline{c_j} \times \phi(\sum_{i=0}^{\infty} a^{(j)}(i) * t^i)) = \phi(\sum_{i=0}^{\infty} b(i) * t^i)$$

[Korollar 2]

$$\Rightarrow a^{(k)} + (\sum_{j=0}^{k-1} \underline{c_j} \times a^{(j)}) = b$$

[Beidseitige Multiplikation mit X^k und Distributivgesetz]

$$\Rightarrow (a^{(k)} \times X^k) + \sum_{j=0}^{k-1} (\underline{c_j} \times a^{(j)} \times X^k) = b \times X^k$$

[Mehrfache Anwendung des Fundamentalsatzes: $a = \sum_{l=0}^{k-1} (\underline{a(l)} \times X^l) + (a^{(k)} \times X^k)$]

$$\Rightarrow (a - \sum_{l=0}^{k-1} (\underline{a_l} \times X^l)) + \sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j} \times (a - \sum_{l=0}^{j-1} (\underline{a_l} \times X^l))) = b \times X^k$$

[Distributivgesetz]

$$\Rightarrow (a - \sum_{l=0}^{k-1} (\underline{a_l} \times X^l)) + \sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j} \times a) - \sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j} \times (\sum_{l=0}^{j-1} (\underline{a_l} \times X^l))) = b \times X^k$$

[Assoziativität von $+$, Distributivgesetz, beidseitige Termaddition]

$$\Rightarrow a + (a \times \sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j})) = (b \times X^k) + \sum_{l=0}^{k-1} (\underline{a_l} \times X^l) + \sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j} \times (\sum_{l=0}^{j-1} (\underline{a_l} \times X^l)))$$

[Distributivgesetz]

$$\Rightarrow a \times (\underline{1} + (\sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j}))) = (b \times X^k) + \sum_{l=0}^{k-1} (\underline{a_l} \times X^l) + \sum_{j=0}^{k-1} (\underline{c_j} \times X^{k-j} \times (\sum_{l=0}^{j-1} (\underline{a_l} \times X^l)))$$

[Beidseitige Termmultiplikation]

$$\Rightarrow a = (\underline{1} + (\sum_{j=0}^{k-1}(\underline{c_j} \times X^{k-j})))^{-1} \times ((b \times X^k) + \sum_{l=0}^{k-1}(\underline{a_l} \times X^l)) + \sum_{j=0}^{k-1}(\underline{c_j} \times X^{k-j} \times (\sum_{l=0}^{j-1}(\underline{a_l} \times X^l))))$$

So lässt sich a in einen Term des Stromkalküls umformen, der nur den Startvektor $(a_0, \dots, a_{k-1})$, Koeffizienten $(c_0, \dots, c_{k-1})$, b und die Formale Variable X enthält.

Beispiel (Fibonacci-Folge)**.** *Die Fibonacci-Folge fib lässt sich mittels eine linearen Rekursionsgleichung darstellen:*
$fib_0 = 0, fib_1 = 1, fib(n+2) + (-1) * fib(n+1) + (-1) * fib(n) = 0$
Die direkte Anwendung der obigen Aussage liefert $fib^{(2)} + (\underline{-1} \times fib^{(1)}) + (\underline{-1} \times fib^{(0)}) = \underline{0}$*, und:*
$fib = (\underline{1}+(\sum_{j=0}^{k-1}(\underline{-1}\times X^{k-j})))^{-1}\times((\underline{0}\times X^k)+\sum_{l=0}^{k-1}(\underline{fib_l}\times X^l))+\sum_{j=0}^{k-1}(\underline{-1}\times X^{k-j}\times(\sum_{l=0}^{j-1}(\underline{fib_l}\times X^l))))$
$= (\underline{1} - X - X^2)^{-1} \times ((\underline{0} \times X^0) + (\underline{1} \times X^1) + (\underline{-1} \times X^2 \times \underline{0}) + (\underline{-1} \times X^1 \times \underline{0} \times X^0)$
$= (\underline{1} - X - X^2)^{-1} \times X$
Die Auf diese Weise ermittelte geschlossene Form im Stromkalkül liefert durch die Isomorphie zu formalen Potenzreihen eine erzeugende Funktion:
$\sum_{i=0}^{\infty} fib(i) * t^i = \psi(fib) = \psi((\underline{1} - X - X^2)^{-1} \times X) = (1 - t - t^2)^{-1} \times t$

5 Erweiterung des Stromkalküls

Im Vergleich zu anderen bisher eingeführten Operatoren des Stromkalküls besitzt das Faltungsprodukt eine ungewöhnliche Form. In diesem Kapitel wird der Operator Mischprodukt $\otimes$ (siehe [1]) eingeführt, der sich an der analytischen Ableitung des Produktes zweier Funktionen $(f * g)' = (f' * g) + (f * g')$ orientiert.
Sei $\otimes\colon \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ mit, $\forall s, t \in \mathbb{R}^\omega : (s \otimes t)(0) = s(0) * t(0), (s \otimes t)' = (s' \otimes t) + (s \otimes t')$.
Diese Definition besitzt bereits die Form eines wohlgeformten Stream-Differentialgleichungssystems und aus dem Existenz und Eindeutigkeitssatz folgt die Wohldefiniertheit von $\otimes$.

5.1 Der Ring $(\mathbb{R}^\omega, +, \otimes)$

Lemma 6. $(\mathbb{R}^\omega, +, \otimes)$ *ist ein kommutativer unitärer Ring.*

Beweis. Alle nötigen Eigenschaften von $(\mathbb{R}^\omega, +, \otimes)$ lassen sich analog zu den Eigenschaften von $(\mathbb{R}^\omega, +, \times)$ coinduktiv beweisen. Die Eins von $(\mathbb{R}^\omega, +, \otimes)$ ist wieder $\underline{1}$.
Aufgrund der inhärenten Symmetrie der Ableitung des Mischproduktes $\otimes$ unterscheidet sich nur der Beweis seiner Kommutativität. Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s \otimes t, t \otimes s) \mid s, t \in \mathbb{R}^\omega\}$.
Es gilt:
$\forall s, t \in \mathbb{R}^\omega : (s \otimes t)(0) = s(0) * t(0) = t(0) * s(0) = (t \otimes s)(0)$
und

$$\begin{array}{rrr}
(s \otimes t)' = & (s' \otimes t) + (s \otimes t') & \text{Definition von } \otimes \\
R + R & (t \otimes s') + (t' \otimes s) & \text{Definition von } R \\
= & (t' \otimes s) + (t \otimes s') & \text{Kommutativität von von } + \\
= & (t \otimes s)' & \text{Definition von } \otimes
\end{array}$$

Damit ist R eine Σ-Bisimulation auf Streams und folglich $\forall s, t : s \otimes t = t \otimes s$ □

5.2 Analytische Stream-Ableitung

Eine gewöhnliche erzeugende Funktion einer Folge $a \in \mathbb{R}^\omega$ besitzt die Form $\sum\limits_{i=0}^{\infty} a(n) * t^i$. Als Verallgemeinerung von Polynomen lässt sich auch diese Form analytisch ableiten.
Die Ableitung ist definiert durch $(\sum\limits_{i=0}^{\infty} a(i) * t^i)' = \sum\limits_{i=0}^{\infty} a(i+1) * (i+1) * t^i$.
Für diese Operation in $\mathbb{R}[[t]]$ lässt sich mit Hilfe des Mischproduktes eine Operation in $\mathbb{R}^\omega$ beschreiben, die mit dem Ringisomorphismus zwischen $(\mathbb{R}[[t]], +, \times)$ und $(\mathbb{R}^\omega, +, \times)$ verträglich ist.
Sei $\phi\colon \mathbb{R}[[t]] \to \mathbb{R}^\omega$ der Ringisomorphismus aus dem Beweis des Satzes 7 mit
$\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i)(0) = a(0)$
$\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i)' = \phi(\sum\limits_{i=0}^{\infty} a(i+1) * t^i)$.
Definiere die analytische Ableitung (siehe [1]) $\frac{d}{dX} : \mathbb{R}^\omega \to \mathbb{R}^\omega$ mit $\frac{d}{dX}(s) = (X \otimes s')'$.
Es ist zu zeigen, dass $\forall a \in \mathbb{R}^\omega : \phi((\sum\limits_{i=0}^{\infty} a(i) * t^i)') = \frac{d}{dX}(\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i))$.
Definiere $R \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ durch $\{(s, s) \mid s \in \mathbb{R}^\omega\} \cup \{(\phi((\sum\limits_{i=0}^{\infty} a(i) * t^i)'), \frac{d}{dX}(\phi(\sum\limits_{i=0}^{\infty} a(i) * t^i))) \mid a \in \mathbb{R}^\omega\}$.
Dann gilt $\forall a \in \mathbb{R}^\omega$:

$$\frac{d}{dX}(\phi(\sum_{i=0}^{\infty} a(i) * t^i))(0)$$

$= \dfrac{d}{dX}(a)(0)$	Korollar 2
$= ((X \otimes a')')(0)$	Definition von $\dfrac{d}{dX}$
$= ((X' \otimes a') + (X \otimes a''))(0)$	Definition von $\otimes$
$= X'(0) * a'(0) + X(0) * a''(0)$	Definition von $+$ und $\otimes$
$= a(1) = a(1) * 1$	Definition von X
$= \phi(\sum\limits_{i=0}^{\infty} a(i+1) * (i+1) * t^i)(0)$	Definition von ϕ
$= \phi((\sum\limits_{i=0}^{\infty} a(i) * t^i)')(0)$	Definition der analytischen Ableitung

$$\phi((\sum_{i=0}^{\infty} a(i) * t^i)')'$$

$= \phi(\sum\limits_{i=0}^{\infty} a(i+1) * (i+1) * t^i)'$	Definition der analytischen Ableitung
$= \phi(\sum\limits_{i=0}^{\infty} a(i+2) * (i+2) * t^i)$	Definition von ϕ
$= \phi(\sum\limits_{i=0}^{\infty} (a(i+2) + a(i+2) * (i+1)) * t^i)$	Abtrennung des Summanden $a(i+2)$

$$
\begin{aligned}
&= \phi(\sum_{i=0}^{\infty} a(i+2) * t^i + \sum_{i=0}^{\infty} a(i+2) * (i+1) * t^i) && \text{Definition der Summe formaler Potenzreihen}\\
&= \phi(\sum_{i=0}^{\infty} a(i+2) * t^i) + \phi(\sum_{i=0}^{\infty} a(i+2) * (i+1) * t^i) && \text{Homomorphismuseigenschaft von } \phi\\
&= \phi(\sum_{i=0}^{\infty} a''(i) * t^i) + \phi(\sum_{i=0}^{\infty} a'(i+1) * (i+1) * t^i) && \text{Eigenschaft der Ableitung}\\
&= a'' + \phi(\sum_{i=0}^{\infty} a'(i+1) * (i+1) * t^i) && \text{Korollar 2}\\
&= a'' + \phi((\sum_{i=0}^{\infty} a'(i) * t^i)') && \text{Definition analytischer Ableitung}\\
R &+ R\, a'' + \frac{d}{dX}(\phi(\sum_{i=0}^{\infty} a'(i) * t^i)) && \text{Definition von } R\\
&= a'' + \frac{d}{dX}(a') && \text{Korollar 2}\\
&= a'' + (X \otimes a'')' && \text{Definition von } \frac{d}{dX}\\
&= (a' + (X \otimes a''))' && \text{Definition von } +\\
&= ((\underline{1} \otimes a') + (X \otimes a''))' && \text{Multiplikation mit } \underline{1}\\
&= ((X' \otimes a') + (X \otimes a''))' && \text{Definition von } X\\
&= ((X \otimes a')')' && \text{Definition von } \otimes\\
&= \frac{d}{dX}(a)' && \text{Definition von } \frac{d}{dX}\\
&= \frac{d}{dX}(\phi(\sum_{i=0}^{\infty} a(i) * t^i))' && \text{Korollar 2}
\end{aligned}
$$

Damit ist R eine Σ-Bisimulation und es folgt die zu zeigende Aussage.

6 Exponentiell erzeugende Funktionen

Bisher betrachtete gewöhnliche erzeugende Funktionen von Folgen $a \in \mathbb{R}^\omega$ besitzen die Form $\sum_{i=0}^{\infty} a(n) * t^i$. Eine exponentiell erzeugende Funktion einer Folge $a \in \mathbb{R}^\omega$ besitzt hingegen die Form $\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i$. Diese Form erlaubt beispielsweise die Darstellung der exponentiell erzeugenden Funktion der Folge $(1, 1, \ldots)$ als Exponentialfunktion: $\sum_{i=0}^{\infty} \frac{1}{i!} * t^i = exp(t)$.

Offensichtlich gilt $\mathbb{R}[[t]] = \{\sum_{i=0}^{\infty} a(i) * t^i \mid a \in \mathbb{R}^\omega\} = \{\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i \mid a \in \mathbb{R}^\omega\}$.
Die Summe $+\colon \mathbb{R}[[t]] \times \mathbb{R}[[t]] \to \mathbb{R}[[t]]$, mit
$\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i + \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i = \sum_{i=0}^{\infty} \frac{(a(i)+b(i))}{i!} * t^i$
und das Mischprodukt $\otimes\colon \mathbb{R}[[t]] \times \mathbb{R}[[t]] \to \mathbb{R}[[t]]$, mit

$\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i \otimes \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i = \sum_{i=0}^{\infty} \frac{c(i)}{i!} * t^i$, wobei
$c(n) = \sum_{i=0}^{n} \binom{n}{i} * a(i) * b(n-i)$
sind wohldefiniert und bilden mit $\mathbb{R}[[t]]$ einen Ring. Zudem besitzt $\mathbb{R}[[t]]$ eine nichtleere Einheitenmenge bezüglich $\otimes$.

Satz 8 (Ringisomorphismus)**.** *Die Ringe* $(\mathbb{R}^\omega, +, \otimes)$ *und* $(\mathbb{R}[[t]], +, \otimes)$ *sind isomorph.*

Beweis. Definiere $\phi\colon \mathbb{R}[[t]] \to \mathbb{R}^\omega$ durch
$\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)(0) = a(0)$
$\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)' = \phi(\sum_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i)$.
Da die Menge der Funktionssymbole $\{\phi(a) \in \mathbb{R}^\omega \mid a \in \mathbb{R}[[t]]\}$ mit den auf obige Weise definierten Gleichheiten ein wohlgeformtes Stream-Differentialgleichungssystem bildet, folgt aus dem Existenz und Eindeutigkeitssatz die Existenz und Eindeutigkeit von ϕ.
Zeige nun, dass ϕ ein Ring-Homomorphismus ist.

- Verträglichkeit mit der Eins:
  Definiere $R = \{(\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i), \underline{r}) \mid r \in \mathbb{R}, a(0) = r, \forall i \geq 1 : a(i) = 0\}$. Dann gilt:
  $\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)(0) = r = \underline{r}(0)$ und

$$\begin{aligned} \phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)' = &\quad \phi(\sum_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i) && \text{Definition von } \phi \\ = &\quad \phi(\sum_{i=0}^{\infty} \frac{0}{i!} * t^i) && \text{, da } \forall i \geq 1 : a(i) = 0 \\ R &\quad \underline{0} && \text{Definition von } R \\ = &\quad \underline{r}' && \text{Definition der Einbettung} \end{aligned}$$

  Damit ist R eine Bisimulation auf Streams und es folgt $\forall r \in \mathbb{R} : \phi(r * t^0) = \underline{r}$. Insbesondere gilt für die Eins im Ring $(\mathbb{R}[[t]], +, \otimes)$: $\phi(\frac{1}{0!} * t^0) = \underline{1}$.

- Verträglichkeit mit $+$:
  Definiere $R = \{(\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i + \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i), \phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) + \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)) \mid a, b \in \mathbb{R}^\omega\}$.
  Dann gilt $\forall a, b \in \mathbb{R}^\omega$:

$$\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i + \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)(0) = a(0) + b(0) = (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) + \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))(0)$$

$$\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i + \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)'$$

  [Definition der Summe exponentiell erzeugender Funktionen]

$$= \phi(\sum_{i=0}^{\infty} \frac{(a(i) + b(i))}{i!} * t^i)'$$

[Definition von ϕ]

$$= \phi(\sum_{i=0}^{\infty} \frac{(a(i+1) + b(i+1))}{i!} * t^i)$$

[Definition der Summe exponentiell erzeugender Funktionen]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i + \sum_{i=0}^{\infty} \frac{b(i+1)}{i!} * t^i)$$

[Definition von R]

$$R\ \phi(\sum_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i) + \phi(\sum_{i=0}^{\infty} \frac{b(i+1)}{i!} * t^i)$$

[Definition von ϕ]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)' + \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)'$$

[Definition von $+$ in $\mathbb{R}^\omega$]

$$= (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) + \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))'$$

Damit ist R eine Bisimulation auf Streams und es folgt $\forall a, b \in \mathbb{R}[[t]] : \phi(a+b) = \phi(a) + \phi(b)$.

- Verträglichkeit mit $\otimes$:
  Definiere $R = \{(\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i \otimes \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i), \phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)) \mid a, b \in \mathbb{R}^\omega\}$.
  Dann gilt $\forall a, b \in \mathbb{R}^\omega$:

$$\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i \times \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)(0) = a(0) * b(0) = (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \times \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))(0)$$

$$\phi((\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \otimes (\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))'$$

[Definition des Produkts exponentiell erzeugender Funktionen]

$$= \phi(\sum_{i=0}^{\infty} \frac{\sum_{j=0}^{i} \binom{i}{j} * a(j) * b(i-j)}{i!} * t^i)'$$

[Definition von ϕ]

$$= \phi(\sum_{i=0}^{\infty} \frac{\sum_{j=0}^{i+1} \binom{i+1}{j} * a(j) * b(i+1-j)}{i!} * t^i)$$

[Abtrennung des ersten und letzten Summanden der inneren Summe]

$$= \phi(\sum_{i=0}^{\infty} \frac{\binom{i+1}{0} * a(0) * b(i+1) + \binom{i+1}{i+1} * a(i+1) * b(0) + \sum\limits_{j=1}^{i} \binom{i+1}{j} * a(j) * b(i+1-j)}{i!} * t^i)$$

[Vereinfachung der Binomialkoeffizienten und Indextransformation der inneren Summe]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(0) * b(i+1) + a(i+1) * b(0) + \sum\limits_{j=0}^{i-1} \binom{i+1}{j+1} * a(j+1) * b(i-j)}{i!} * t^i)$$

[Zerlegung des Binomialkoeffizienten in Summe zweiter Binomialkoeffizienten]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(0) * b(i+1) + a(i+1) * b(0) + \sum\limits_{j=0}^{i-1} (\binom{i}{j} + \binom{i}{j+1}) * a(j+1) * b(i-j)}{i!} * t^i)$$

[Distributivgesetz]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(0) * b(i+1) + a(i+1) * b(0) + \sum\limits_{j=0}^{i-1} \binom{i}{j} * a(j+1) * b(i-j) + \sum\limits_{j=0}^{i-1} \binom{i}{j+1} * a(j+1) * b(i-j)}{i!} * t^i)$$

[Zusammenfassung der ersten inneren Summe mit dem Summanden $a(i+1) * b(0)$]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(0) * b(i+1) + \sum\limits_{j=0}^{i} \binom{i}{j} * a(j+1) * b(i-j) + \sum\limits_{j=0}^{i-1} \binom{i}{j+1} * a(j+1) * b(i-j)}{i!} * t^i)$$

[Indextransformation der zweiten inneren Summe]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(0) * b(i+1) + \sum\limits_{j=0}^{i} \binom{i}{j} * a(j+1) * b(i-j) + \sum\limits_{j=1}^{i} \binom{i}{j} * a(j) * b(i-(j-1))}{i!} * t^i)$$

[Zusammenfassung der zweiten inneren Summe mit dem Summanden $a(0) * b(i+1)$]

$$= \phi(\sum_{i=0}^{\infty} \frac{\sum\limits_{j=0}^{i} \binom{i}{j} * a(j+1) * b(i-j) + \sum\limits_{j=0}^{i} \binom{i}{j} * a(j) * b(i+1-j)}{i!} * t^i)$$

[Definition der Summe exponentiell erzeugender Funktionen]

$$= \phi(\sum_{i=0}^{\infty} \frac{\sum\limits_{j=0}^{i} \binom{i}{j} * a(j+1) * b(i-j)}{i!} * t^i + \sum_{i=0}^{\infty} \frac{\sum\limits_{j=0}^{i} \binom{i}{j} * a(j) * b(i-j+1)}{i!} * t^i)$$

[Verträglichkeit von ϕ mit $+$]

$$= \phi(\sum_{i=0}^{\infty} \frac{\sum\limits_{j=0}^{i} \binom{i}{j} * a(j+1) * b(i-j)}{i!} * t^i) + \phi(\sum_{i=0}^{\infty} \frac{\sum\limits_{j=0}^{i} \binom{i}{j} * a(j) * b(i-j+1)}{i!} * t^i)$$

[Definition des Produktes exponentiell erzeugender Funktionen]

$$= \phi(\sum_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i \otimes \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i) + \phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i \otimes \sum_{i=0}^{\infty} \frac{b(i+1)}{i!} * t^i)$$

[Definition von R]

$$R + R\ (\phi(\sum_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i) \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)) + (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i+1)}{i!} * t^i))$$

[Definition von ϕ]

$$= (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)' \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)) + (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i)')$$

[Definition von $\otimes$ in $\mathbb{R}^\omega$]

$$= (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))'$$

Damit ist R eine Σ-Bisimulation auf Streams und es folgt $\forall a, b \in \mathbb{R}[[t]] : \phi(a \otimes b) = \phi(a) \otimes \phi(b)$.

Zeige nun, dass ϕ bijektiv ist. Definiere $\psi\colon \mathbb{R}^\omega \to \mathbb{R}[[t]]$ mit $\psi(a) = \sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i$.

Definiere $R = \{(\phi(\sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i), a) \mid a \in \mathbb{R}^\omega\}$. Dann gilt:

$\phi(\sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)(0) = a(0)$ und

$\phi(\sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)' = \phi(\sum\limits_{i=0}^{\infty} \frac{a(i+1)}{i!} * t^i) = \phi(\sum\limits_{i=0}^{\infty} \frac{(a')(i)}{i!} * t^i)\ R\ a'$.

Damit ist R eine Bisimulation auf Streams und es folgt $\forall a \in \mathbb{R}^\omega : \phi(\sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) = a$.

Aus dieser Aussage folgt unmittelbar:

$\forall a \in \mathbb{R}^\omega : \phi(\psi(a)) = \phi(\sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) = a \wedge \psi(\phi(\sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i)) = \psi(a) = \sum\limits_{i=0}^{\infty} \frac{a(i)}{i!} * t^i$.

Damit gilt $id_{\mathbb{R}^\omega} = \phi \circ \psi \wedge id_{\mathbb{R}[[t]]} = \psi \circ \phi$. $\square$

Folgerungen aus der Isomorphie

Mit Hilfe der im Beweis verwendeten Aussagen lässt sich die explizite Darstellung der Folgenglieder $(a \otimes b)(i)$ für $a, b \in \mathbb{R}^\omega$ herleiten. Sei ϕ der im obigen Beweis verwendete Ringisomorphismus, es gilt $\forall n \in \mathbb{N}$:

$$\begin{aligned}
(a \otimes b)(n) &= (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i) \otimes \phi(\sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))(n) && \text{da } \forall s \in \mathbb{R}^\omega : s = \phi(s) \\
&= (\phi(\sum_{i=0}^{\infty} \frac{a(i)}{i!} * t^i \otimes \sum_{i=0}^{\infty} \frac{b(i)}{i!} * t^i))(n) && \text{Isomorphieeigenschaft} \\
&= \phi(\sum_{i=0}^{\infty} \frac{\sum\limits_{j=0}^{i} \binom{i}{j} * a(j) * b(i-j)}{i!} * t^i)(n) && \text{Definition von } \otimes \\
&= \phi(\sum_{i=0}^{\infty} \frac{c(i)}{i!} * t^i)(n) && \text{für } c \in \mathbb{R}^\omega : c(i) = \sum_{j=0}^{i} \binom{i}{j} * a(j) * b(i-j) \\
&= c(n) = \sum_{i=0}^{n} \binom{n}{i} * a(i) * b(n-i) && \text{da } \forall s \in \mathbb{R}^\omega : s = \phi(s)
\end{aligned}$$

6.1 Weitere Streamoperatoren

Analog zum Faltungsprodukt besitzt $\mathbb{R}^\omega$ invertierbare Elemente bezüglich des Mischproduktes. Es lässt sich zeigen, dass genau wie beim Faltungsprodukt $\mathbb{R}^\omega_{\neq 0}$ die Einheitenmenge bezüglich des Mischproduktes ist. Aus diesem Grund ist es sinnvoll, ein Misch-Inverses als Stream-Operator zu definieren.
Des Weiteren lässt sich die Exponentialfunktion aus der Analysis (siehe [1]) strukturell auf den Stromkalkül übertragen. Dieser Idee folgend wird im Folgenden versucht, weitere Funktionen $(\log, \sin, \cos, \sinh, \cosh)$ aus der Analysis auf den Stromkalkül zu übertragen und Zusammenhänge zwischen ihnen aufzuzeigen.
Seien $\exp, \log, \sin, \cos, \sinh, \cosh\colon \mathbb{R} \to \mathbb{R}$ die aus der Analysis bekannten Funktionen und $s, t \in \mathbb{R}^\omega$, definiere:

- Misch-Inverses: $\overline{\_^{-1}}\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $s(0) \neq 0 \Rightarrow s \otimes \overline{s^{-1}} = \underline{1}$
- Misch-Wurzel: $\sqrt[\otimes]{\_}\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $s(0) > 0 \Rightarrow \sqrt[\otimes]{s} \otimes \sqrt[\otimes]{s} = s$
- Exponentialfunktion: $exp\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(exp(s)) = (\exp(s(0)), s' \otimes exp(s))$
- Logarithmus: $log\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $s(0) > 0 \Rightarrow \sigma(log(s)) = (\log(s(0)), s' \otimes \overline{s^{-1}})$
- Sinus: $sin\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(sin(s)) = (\sin(s(0)), s' \otimes cos(s))$
- Cosinus: $cos\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(cos(s)) = (\cos(s(0)), s' \otimes (-sin(s)))$
- Sinus Hyperbolicus: $sinh\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(sinh(s)) = (\sinh(s(0)), s' \otimes cosh(s))$
- Cosinus Hyperbolicus: $cosh\colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, mit $\sigma(cosh(s)) = (\cosh(s(0)), s' \otimes sinh(s))$

6.1.1 Eindeutigkeit als Beweismethode

Eine weitere Möglichkeit induktionslos die Gleichheit zweier Streamoperatoren (oder Streams als nullstellige Streamoperatoren) zu beweisen, ist es, sie durch ein wohlgeformtes Stream-Differentialgleichungssystem zu beschreiben (siehe [4](admissible equations)). Aus dem Existenz und Eindeutigkeitssatz folgt dann unmittelbar ihre Gleichheit.
Zur Demonstration dieser Beweismethode wird eine aus der Analysis bekannte und auf den Stromkalkül übertragene Aussage $\forall a, b \in \mathbb{R}^\omega : exp(a + b) = exp(a) \otimes exp(b)$ gezeigt. Definiere zwei zweistellige Abbildungen $\alpha(a, b) = exp(a + b)$ und $\beta(a, b) = exp(a) \otimes exp(b)$.
Dann gilt $\forall a, b \in \mathbb{R}^\omega$:

$$\alpha(a,b)(0) = exp(a+b)(0) = \exp(a(0)+b(0)) = \exp(a(0)) * \exp(b(0))$$

$$\begin{aligned}
\alpha(a,b)' &= exp(a+b)' && \text{Definition von } \alpha \\
&= (a+b)' \otimes exp(a+b) && \text{Definition von } exp \\
&= (a'+b') \otimes \alpha(a,b) && \text{Definition von } \alpha
\end{aligned}$$

$$\beta(a,b)(0) = (exp(a) \otimes exp(b))(0) = \exp(a(0)) * \exp(b(0))$$

$$\begin{aligned}
\beta(a,b)' &= (exp(a) \otimes exp(b))' && \text{Definition von } \beta \\
&= (exp(a)' \otimes exp(b)) + (exp(a) \otimes exp(b)') && \text{Definition von } \otimes \\
&= (a' \otimes exp(a) \otimes exp(b)) + (exp(a) \otimes b' \otimes exp(b)) && \text{Definition von } exp \\
&= (a'+b') \otimes exp(a) \otimes exp(b) && \text{Distributivgesetz} \\
&= (a'+b') \otimes \beta(a,b) && \text{Definition von } \beta
\end{aligned}$$

Aus den obigen Gleichheiten erkennt man, dass α und β als Lösungen des gleichen wohlgeformten Stream-Differentialgleichungssystems dargestellt werden können. Aus dem Existenz und Eindeutigkeitssatz folgt durch die Eindeutigkeit die Gleichheit der Lösungen α und β.
Im Folgenden wird in Beweisen dieses Typs das Symbol $\rightsquigarrow$ verwendet, um darauf hinzuweisen, dass bei gleichbleibender Termstruktur eine Ersetzung der Funktion stattfindet. Die Termstruktur wird in eckigen Klammern mit Hilfe des Funktionssymbols φ angegeben und ist der essentielle Bestandteil des zugrundeliegenden wohlgeformten Stream-Differentialgleichungssystems. Mit anderen Worten hätte der Beweis der obigen Aussage die Form $\forall a, b \in \mathbb{R}^\omega$:

$$\begin{aligned}
\alpha(a,b)(0) = \ldots = \exp(a(0)) * \exp(b(0)) = \ldots = \beta(a,b)(0) \\
\alpha(a,b)' &= exp(a+b)' \\
&= \ldots = (a' + b') \otimes \alpha(a,b) \\
&[\varphi(x_1,x_2)' = (x_1' + x_2') \otimes \varphi(x_1,x_2)] \\
&\rightsquigarrow (a' + b') \otimes \beta(a,b) \\
&= \ldots = (exp(a) \otimes exp(b))' \\
&= \beta(a,b)'
\end{aligned}$$

Zusammenhang zwischen Operatoren

Eine aus der Analysis bekannte Aussage ist $\forall r \in \mathbb{R} : r > 0 \Rightarrow \sqrt{r} = r^{0.5} = \exp(\log(r) * 0.5)$. Diese Aussage lässt sich ebenfalls auf den Stromkalkül übertragen und so ein Zusammenhang zwischen den Streamoperatoren zeigen.
Zeige nun $\forall s \in \mathbb{R}^\omega : s(0) > 0 \Rightarrow \sqrt[\otimes]{s} = exp(log(s) \otimes \underline{0.5})$.
Seien α, β einstellige Abbildungen mit $s(0) > 0 : \alpha(s) = \sqrt[\otimes]{s}$, $\beta(s) = exp(log(s) \otimes \underline{0.5})$. Es gilt $\forall s \in \mathbb{R}^\omega$ mit $s(0) > 0$:

$$\alpha(s)(0) = \sqrt[\otimes]{s}(0) = \sqrt{s(0)} = exp(log(s(0)) * 0.5) = exp(log(s) \otimes \underline{0.5})(0) = \beta(s)(0)$$

$$\begin{aligned}
&\alpha(s)' \\
&\qquad [\text{Definition von } \alpha] \\
&= \sqrt[\otimes]{s}' \\
&\qquad [\text{Wohlgeformtes Stream-Differentialgleichungssystem für } \sqrt[\otimes]{_} \text{ (siehe [1])}] \\
&= s' \otimes (\sqrt[\otimes]{s} + \sqrt[\otimes]{s})^{\overline{-1}} \\
&\qquad [\text{Erweiterung mit } \sqrt[\otimes]{s} \text{ und Distributivgesetz}] \\
&= s' \otimes \sqrt[\otimes]{s} \otimes (s+s)^{\overline{-1}} \\
&\qquad [\text{Distributivgesetz und Zusammenfassung von } \underline{1} + \underline{1} \text{ zu } \underline{2}] \\
&= s' \otimes \sqrt[\otimes]{s} \otimes (\underline{2} \otimes s)^{\overline{-1}} \\
&\qquad [\text{Eigenschaft des Inversen eines Produktes}] \\
&= s' \otimes \sqrt[\otimes]{s} \otimes \underline{2}^{\overline{-1}} \otimes s^{\overline{-1}} \\
&\qquad [\text{Eigenschaft des Inversen der Einbettung}] \\
&= s' \otimes s^{\overline{-1}} \otimes \underline{0.5} \otimes \sqrt[\otimes]{s} \\
&\qquad [\text{Definition von } \alpha]
\end{aligned}$$

$= s' \otimes \overline{s^{-1}} \otimes \underline{0.5} \otimes \alpha(s)$

$[\varphi(s)' = s' \otimes \overline{s^{-1}} \otimes \underline{0.5} \otimes \varphi(s)]$

$\rightsquigarrow s' \otimes \overline{s^{-1}} \otimes \underline{0.5} \otimes \beta(s)$

[Definition von β]

$= s' \otimes \overline{s^{-1}} \otimes \underline{0.5} \otimes exp(log(s) \otimes \underline{0.5})$

[Addition von $\underline{0}$ und Assoziativität von $\otimes$]

$= (((s' \otimes \overline{s^{-1}}) \otimes \underline{0.5}) + (log(s) \otimes \underline{0})) \otimes exp(log(s) \otimes \underline{0.5})$

[Definition der Einbettung und log]

$= ((log(s)' \otimes \underline{0.5}) + (log(s) \otimes \underline{0.5}')) \otimes exp(log(s) \otimes \underline{0.5})$

[Definition von $\otimes$]

$= (log(s) \otimes \underline{0.5})' \otimes exp(log(s) \otimes \underline{0.5})$

[Definition von exp]

$= exp(log(s) \otimes \underline{0.5})'$

[Definition von β]

$= \beta(s)'$

Aus der Eindeutigkeit folgt $\alpha = \beta$ und damit die obige Aussage.
Eine weitere Aussage der Analysis ist $\forall r \in \mathbb{R} : \sin(r)^2 + \cos(r)^2 = 1$. Auch diese Aussage lässt sich übertragen.
Zeige nun $\forall s \in \mathbb{R}^\omega (sin(s) \otimes sin(s)) + (cos(s) \otimes cos(s)) = \underline{1}$.
Es gilt $\forall s \in \mathbb{R}^\omega$:

$$(sin(s) \otimes sin(s)) + (cos(s) \otimes cos(s))(0) = \sin(s(0))^2 + \cos(s(0))^2 = 1 = \underline{1}(0)$$

$(sin(s) \otimes sin(s)) + (cos(s) \otimes cos(s))'$

[Definition von $+$]

$= (sin(s) \otimes sin(s))' + (cos(s) \otimes cos(s))'$

[Definition von $\otimes$]

$= (sin(s)' \otimes sin(s)) + (sin(s) \otimes sin(s)') + (cos(s)' \otimes cos(s)) + (cos(s) \otimes cos(s)')$

[Definition von sin und cos]

$= (s' \otimes cos(s) \otimes sin(s)) + (sin(s) \otimes s' \otimes cos(s)) + (s' \otimes (-sin(s)) \otimes cos(s)) + (cos(s) \otimes s' \otimes (-sin(s)))$

[Eigenschaft des Produktes mit einer Negation und Kommutativität von $+$ und $\otimes$]

$= (s' \otimes cos(s) \otimes sin(s)) - (s' \otimes cos(s) \otimes sin(s)) + (sin(s) \otimes s' \otimes cos(s)) - (sin(s) \otimes s' \otimes cos(s))$

[Eigenschaft der Subtraktion]

$= \underline{0} + \underline{0} = \underline{0} = \underline{1}'$

Aus der obigen Gleichheit und Korollar 1 folgt $\forall s \in \mathbb{R}^\omega (sin(s) \otimes sin(s)) + (cos(s) \otimes cos(s)) = \underline{1}$.

6.1.2 Eine weitere geschlossene Form der Fibonacci-Folge

Laut [6] besitzt die Fibonacci-Folge die exponentiell erzeugende Funktion $(2/\sqrt{5}) * \exp(x/2) * \sinh(\sqrt{5} * x/2)$. Im Folgenden wird gezeigt, dass sich diese exponentiell erzeugende Funktion in den Stromkalkül übertragen lässt, um eine geschlossene Form der Fibonacci-Folge zu erhalten.

Bisher wurde bereits eine geschlossene Form aus einer linearen Rekursionsgleichung hergeleitet
$fib = (\underline{1} - X - X^2)^{-1} \times X$.
Nun wird gezeigt, dass
$fib_e = \underline{2/\sqrt{5}} \otimes exp(X \otimes \underline{0.5}) \otimes sinh(\underline{\sqrt{5}/2} \otimes X)$
auch der Fibonacci-Folge entspricht ($fib = fib_e$). Zunächst wird die erste Gleichheit näher untersucht.

$$
\begin{aligned}
& fib = (\underline{1} - X - X^2)^{-1} \times X \\
\Rightarrow & fib' = ((\underline{1} - X - X^2)^{-1} \times X)' \\
& \quad \text{[Definition von } \times \text{]} \\
& = (((\underline{1} - X - X^2)^{-1})' \times X) + (\underline{(\underline{1} - X - X^2)^{-1}(0)} \times X') \\
& \quad \text{[Ableitung des Inversen im wohlgeformten Stream-Differentialgleichungssystem]} \\
& = (((\underline{1} - X - X^2)' \times \underline{-(\underline{1} - X - X^2)(0)^{-1}} \times (\underline{1} - X - X^2)^{-1}) \times X) + (\underline{(\underline{1} - X - X^2)^{-1}(0)} \times X') \\
& \quad \text{[Einsetzen von } X(0) = 0 \text{ und } X' = \underline{1}\text{]} \\
& = (((\underline{1} - X - X^2)' \times \underline{-1} \times (\underline{1} - X - X^2)^{-1}) \times X) + (\underline{1} \times \underline{1}) \\
& \quad \text{[Ableitung des ersten Faktors und Vereinfachung]} \\
& = (((\underline{-1} - X) \times \underline{-1} \times (\underline{1} - X - X^2)^{-1}) \times X) + \underline{1} \\
& \quad \text{[Definition von } fib \text{ und Eigenschaft des Produktes mit einer Negation]} \\
& = (((\underline{1} + X) \times fib) + \underline{1}
\end{aligned}
$$

Nun wird fib_e untersucht.

$$
\begin{aligned}
& fib_e = \underline{2/\sqrt{5}} \otimes exp(X \otimes \underline{0.5}) \otimes sinh(\underline{\sqrt{5}/2} \otimes X) \\
\Rightarrow & fib_e' = (\underline{2/\sqrt{5}} \otimes exp(X \otimes \underline{0.5}) \otimes sinh(\underline{\sqrt{5}/2} \otimes X))' \\
& \quad \text{[Definition von } \otimes \text{ und Eigenschaft der Ableitung eines Produktes mit der Einbettung]} \\
& = \underline{2/\sqrt{5}} \otimes ((exp(X \otimes \underline{0.5})' \otimes sinh(\underline{\sqrt{5}/2} \otimes X)) + (exp(X \otimes \underline{0.5}) \otimes sinh(\underline{\sqrt{5}/2} \otimes X)')) \\
& \quad \text{[Definition von } exp \text{ und } sinh\text{]} \\
& = \underline{2/\sqrt{5}} \otimes ((\underline{0.5} \otimes X' \otimes exp(X \otimes \underline{0.5}) \otimes sinh(\underline{\sqrt{5}/2} \otimes X)) + (exp(X \otimes \underline{0.5}) \otimes \underline{\sqrt{5}/2} \otimes X' \otimes cosh(\underline{\sqrt{5}/2} \otimes X))) \\
& \quad \text{[Distributivgesetz und Einsetzung von } X' = \underline{1}\text{]} \\
& = (\underline{2/\sqrt{5}} \otimes \underline{0.5} \otimes exp(X \otimes \underline{0.5}) \otimes sinh(\underline{\sqrt{5}/2} \otimes X)) + (\underline{2/\sqrt{5}} \otimes exp(X \otimes \underline{0.5}) \otimes \underline{\sqrt{5}/2} \otimes cosh(\underline{\sqrt{5}/2} \otimes X))) \\
& \quad \text{[Kommutativität von } \otimes \text{ und Definition von } fib_e\text{]} \\
& = (\underline{0.5} \otimes fib_e) + (\underline{2/\sqrt{5}} \otimes exp(X \otimes \underline{0.5}) \otimes \underline{\sqrt{5}/2} \otimes cosh(\underline{\sqrt{5}/2} \otimes X)) \\
& \quad \text{[Zusammenfassung des Produktes der Einbettungen und Elimination von } \underline{1} \text{ als Faktor]} \\
& = (\underline{0.5} \otimes fib_e) + (exp(X \otimes \underline{0.5}) \otimes cosh(\underline{\sqrt{5}/2} \otimes X)) \\
\Rightarrow & fib_e' - (\underline{0.5} \otimes fib_e) = exp(X \otimes \underline{0.5}) \otimes cosh(\underline{\sqrt{5}/2} \otimes X) \qquad (i)
\end{aligned}
$$

Nun wird fib_e' weiter untersucht.

$$
\begin{aligned}
& fib_e' = (\underline{0.5} \otimes fib_e) + (exp(X \otimes \underline{0.5}) \otimes cosh(\underline{\sqrt{5}/2} \otimes X)) \\
\Rightarrow & fib_e'' = (\underline{0.5} \otimes fib_e)' + (exp(X \otimes \underline{0.5}) \otimes cosh(\underline{\sqrt{5}/2} \otimes X))' \\
& \quad \text{[Definition von } \otimes, exp, cosh\text{]}
\end{aligned}
$$

$= (\underline{0.5} \otimes fib'_e) + (\underline{0.5} \otimes X' \otimes exp(X \otimes \underline{0.5}) \otimes cosh(\underline{\sqrt{5}/2} \otimes X)) + (exp(X \otimes \underline{0.5}) \otimes \underline{\sqrt{5}/2} \otimes X' \otimes sinh(\underline{\sqrt{5}/2} \otimes X))$

[Einsetzung von $X' = \underline{1}$ und fib_e nach Anpassung des Eingebetteten Faktors $5/4 = \dfrac{\sqrt{5}/2}{2/\sqrt{5}}$]

$= (\underline{0.5} \otimes fib'_e) + (\underline{0.5} \otimes exp(X \otimes \underline{0.5}) \otimes cosh(\underline{\sqrt{5}/2} \otimes X)) + (\underline{5/4} \otimes fib_e)$

[Einsetzung von (i)]

$= (\underline{0.5} \otimes fib'_e) + (\underline{0.5} \otimes (fib'_e - (\underline{0.5} \otimes fib_e))) + (\underline{5/4} \otimes fib_e)$

[Distributivgesetz und Zusammenfassung des Produktes von Einbettungen]

$= (\underline{0.5} \otimes fib'_e) + (\underline{0.5} \otimes fib'_e) - (\underline{0.25} \otimes fib_e) + (\underline{5/4} \otimes fib_e)$

[Distributivgesetz und Zusammenfassung der Summen von Einbettungen]

$= (\underline{(0.5 + 0.5)} \otimes fib'_e) + (\underline{(5/4 - 0.25)} \otimes fib_e)$

[Elimination von $\underline{1}$ als neutrales Element in $\otimes$]

$= fib'_e + fib_e$

[Beidseitige Termmultiplikation und Distributivgesetz]

$\Rightarrow fib''_e \times X = (fib'_e \times X) + (fib_e \times X)$

[Fundamentalsatz]

$\Rightarrow fib'_e - \underline{(fib'_e)(0)} = fib_e - \underline{fib_e(0)} + (fib_e \times X)$

[Beidseitige Termaddition und Kommutativität von $+$]

$\Rightarrow fib'_e = fib_e + (fib_e \times X) + \underline{(fib'_e)(0)} - \underline{fib_e(0)}$

[Distributivgesetz]

$\Rightarrow fib'_e = (\underline{1} + X) \times fib_e + \underline{(fib'_e)(0)} - \underline{fib_e(0)}$

[Berechnung von $fib_e(0) = 0$ aus der Definition und $(fib'_e)(0) = 1$ mit Hilfe von (i)]

$\Rightarrow fib'_e = (\underline{1} + X) \times fib_e + \underline{1} - \underline{0}$

$= (\underline{1} + X) \times fib_e + \underline{1}$

Aus den oberen Aussagen folgt:

$$fib(0) = 0 = fib_e(0)$$

$$\begin{aligned} fib' &= (((\underline{1} + X) \times fib) + \underline{1} \\ &\quad [\varphi()' = (((\underline{1} + X) \times \varphi()) + \underline{1}] \\ &\quad \rightsquigarrow (((\underline{1} + X) \times fib_e) + \underline{1} \\ &\quad = fib'_e \end{aligned}$$

Schließlich folgt aus der Eindeutigkeit des durch die oben angegebene Form definierten Streams $fib = fib_e$.

6.1.3 Weitere Streamoperatoren aus der Literatur

Die Streamoperatoren Sinus,Cosinus,Sinus Hyperbolicus und Cosinus Hyperbolicus orientieren sich an den analytischen Vorbildern und wurden in dieser Form bisher nicht in Zusammenhang mit dem Stromkalkül erwähnt. In [5] wurden Konstanten sin und ch beschrieben mit $sin = 0 :: 1 :: 0 :: -1 :: sin, ch = 1 :: 0 :: ch$. Diese Konstanten entsprechen den Streams $sin(X)$ und $cosh(X)$. Exemplarisch wird im Folgenden die zweite Gleichheit bewiesen.

Es gilt:

$$ch(0) = 1 = \cosh(0) = cosh(X)(0)$$

$$\begin{aligned}
ch' &= \underline{(ch')(0)} + (ch'' \times X) && \text{Fundamentalsatz}\\
&= \underline{0} + (ch \times X) && \text{Definition von } ch\\
&[\varphi()' = \varphi() \times X]\\
&\rightsquigarrow cosh(X) \times X\\
&= (\underline{1} \times cosh(X)) \times X\\
&= (X' \times cosh(X)) \times X\\
&= sinh(X)' \times X && \text{Definition von } sinh\\
&= \underline{0} + (sinh(X)' \times X)\\
&= \underline{sinh(X)(0)} + (sinh(X)' \times X) && \text{Definition von } sinh\\
&= sinh(X) && \text{Fundamentalsatz}\\
&= \underline{1} \times sinh(X)\\
&= X' \times sinh(X)\\
&= cosh(X)' && \text{Definition von } cosh
\end{aligned}$$

Aus der Eindeutigkeit des durch die oben angegebene Form definierten Streams folgt $ch = cohs(X)$.
In [1](Theorem 10.1) wird noch ein weiterer Streamoperator $\Lambda_c \colon \mathbb{R}^\omega \to \mathbb{R}^\omega$, die Laplace-Carson-Transformation, definiert. Dieser Streamoperator ist erwähnenswert, da er einem Ringisomorphismus zwischen $(\mathbb{R}^\omega, +, \times)$ und $(\mathbb{R}^\omega, +, \otimes)$ entspricht. Zusätzlich stellt er einen Zusammenhang zwischen der analytischen Ableitung und der coalgebraischen Ableitung eines Streams her.

7 Weitere F-Coalgebren

7.1 Bäume

Ein unendlicher binärer Baum über den reellen Zahlen besteht aus einem Wurzelknoteneintrag und zwei weiteren unendlichen binären Bäumen. Formell gefasst ist die Menge aller unendlichen binären Bäume über den reellen Zahlen $\mathbb{R}^{\mathbb{B}^*} = \{t \colon \mathbb{B}^* \to \mathbb{R}\}$. Für ein $t \in \mathbb{R}^{\mathbb{B}^*}$ und ein Wort $w \in \mathbb{B}^*$ ist $t(w) \in \mathbb{R}$ mit

$$t(w) = \begin{cases} \text{der Wurzelknoteneintrag} & \text{wenn } w = \epsilon \\ u(w') & \text{wenn } w = 0w' \text{ und } u \text{ der linke Unterbaum von } t \text{ ist} \\ v(w') & \text{wenn } w = 1w' \text{ und } v \text{ der rechte Unterbaum von } t \text{ ist} \end{cases}$$

Im Kontext der Stream-Coalgebra mit dem Funktor $X \mapsto \mathbb{R} \times X$, ist es naheliegend, die Menge $\mathbb{R}^{\mathbb{B}^*}$ als Träger einer Coalgebra mit dem Funktor $F(X) = \mathbb{R} \times X \times X$ zu betrachten.
Die F-Coalgebra $\tau \colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R} \times \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*}$ mit
$\tau(t) = \langle \tau_1, \tau_2, \tau_3 \rangle(t) = (t(\epsilon), w \mapsto t(0w), w \mapsto t(1w))$
zerlegt einen unendlichen binären Baum t in seinen Wurzelknoteneintrag, den linken und den rechten Unterbaum. Bezeichne τ als **Tree-Coalgebra**.
Analog zur Stream-Coalgebra, werden die Destruktoren der mit Begriffen aus der Analysis assoziiert. Da ein unendlicher binärer Baum zwei Unterbäume besitzt, ist es sinnvoll, zwei partielle Ableitungen eines Baumes zu definieren. Bezeichne

- $\tau_1(t) = t(\epsilon)$ als Startwert
- $\tau_2(t) = \partial_0 t$ als erste partielle Ableitung
- $\tau_3(t) = \partial_1 t$ als zweite partielle Ableitung
- $\begin{pmatrix} \tau_2(t) \\ \tau_3(t) \end{pmatrix} = \begin{pmatrix} \partial_0 t \\ \partial_1 t \end{pmatrix} = t'$ als Ableitung

7.1.1 Bisimulation auf Trees

Das coalgebraische Prinzip der Coinduktion lässt sich auch für Bäume formulieren.

Definition 10 (Bisimulation auf Trees)**.** Eine Relation $R \subseteq \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*}$ heißt Bisimulation auf Trees, falls
$\forall s, t \in \mathbb{R}^{\mathbb{B}^*} : s\, R\, t \Rightarrow \begin{cases} s(\epsilon) = t(\epsilon) \\ \partial_0 s\, R\, \partial_0 t \\ \partial_1 s\, R\, \partial_1 t \end{cases}$

Satz 9 (Coinduktion auf Trees)**.** *Für $s, t \in \mathbb{R}^{\mathbb{B}^*}$ gilt:*
$\exists R \subseteq \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*}$ *Bisimulation auf Trees mit $s\, R\, t \Rightarrow s = t$*

Beweis. Seien $s, t \in \mathbb{R}^{\mathbb{B}^*}$, $R \subseteq \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*}$ eine Bisimulation auf Trees und $(s, t) \in R$. Dann lässt sich via Induktion nach der Wortlänge zeigen, dass $\forall n \in \mathbb{N}, \forall (w_0, \dots, w_n) \in \mathbb{B}^n : (\partial_{w_0} \dots \partial_{w_n} s, \partial_{w_0} \dots \partial_{w_n} t) \in R$.
Aus der Definition von R folgt dann $\forall w \in \mathbb{B}^* : s(w) = t(w) \Rightarrow s = t$. $\square$

7.1.2 Wohlgeformte Tree-Differentialgleichungssysteme

In [3] wird der Begriff des wohlgeformten Differentialgleichungssystems definiert und anschließend die Existenz und Eindeutigkeit der Lösung eines solchen Systems aus der Finalität der Tree-Coalgebra gefolgert.

Definition 11 (Wohlgeformtes Tree-Differentialgleichungssystem)**.** Sei $\Sigma = \{f, g, \dots\}$ eine Menge von Funktionssymbolen, und für alle $f \in \Sigma$ sei

- $r_f \in \mathbb{N}$ die Stelligkeit von f.
- p_f, q_f Terme aus Konstanten, Funktionssymbolen aus Σ und den Variablen $(x_1, \dots, x_{r_f}, \partial_0 x_1, \dots, \partial_0 x_{r_f}, \partial_1 x_1, \dots, \partial_1 x_{r_f}, x_1(\epsilon), \dots, x_{r_f}(\epsilon))$.
- $h_f \colon \mathbb{R}^{r_f} \to \mathbb{R}$ eine reellwertige Funktion.

Dann heißt das Gleichungssystem $\forall f \in \Sigma$
$f(x_1, \dots, x_{r_f})(\epsilon) = h_f(x_1(\epsilon), \dots, x_{r_f}(\epsilon))$,
$\partial_0 f(x_1, \dots, x_{r_f}) = p_f$
$\partial_1 f(x_1, \dots, x_{r_f}) = q_f$
wohl geformt.

7.1.3 Tree-Operatoren

Unter Verwendung der wohlgeformten Tree-Differentialgleichungssysteme ist es möglich, auf Bäumen zum Stromkalkül analoge Operatoren zu beschreiben.
Beispielsweise besitzt der Operator $+$ im Stromkalkül die Form $\forall s, t \in \mathbb{R}^\omega$:
Startwert: $(s+t)(0) = s(0) + t(0)$, Ableitung: $(s+t)' = s' + t'$.
Eine intuitive syntaktische Übertragung in die Tree-Coalgebra liefert die Form $\forall s, t \in \mathbb{R}^{\mathbb{B}^*}$:
Startwert: $(s+t)(\epsilon) = s(\epsilon) + t(\epsilon)$,
Ableitung: $\begin{pmatrix}\partial_0(s+t)\\ \partial_1(s+t)\end{pmatrix} = \begin{pmatrix}\partial_0 s + \partial_0 t\\ \partial_1 s + \partial_1 t\end{pmatrix}$.
Diese Form entspricht bereits einem wohlgeformten Tree-Differentialgleichungssystem und der auf diese Weise definierte Operator $+\colon \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$ besitzt folglich eine eindeutige Lösung.
In [3] wird mit folgenden Operatoren ein Baumkalkül definiert:

- Einbettung: $\underline{\ }\colon \mathbb{R} \to \mathbb{R}^{\mathbb{B}^*}$, mit $\underline{r}(\epsilon) = r, \underline{r}' = \begin{pmatrix}\underline{0}\\ \underline{0}\end{pmatrix}$
- Erste Formale Variable: $L \in \mathbb{R}^{\mathbb{B}^*}$, mit $L(\epsilon) = 0, L' = \begin{pmatrix}\underline{1}\\ \underline{0}\end{pmatrix}$
- Zweite Formale Variable: $R \in \mathbb{R}^{\mathbb{B}^*}$, mit $R(\epsilon) = 0, R' = \begin{pmatrix}\underline{0}\\ \underline{1}\end{pmatrix}$
- Summe: $+\colon \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $(s+t)(\epsilon) = s(\epsilon) + t(\epsilon), (s+t)' = \begin{pmatrix}\partial_0 s + \partial_0 t\\ \partial_1 s + \partial_1 t\end{pmatrix}$
- Faltungsprodukt: $\times\colon \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $(s \times t)(\epsilon) = s(\epsilon) * t(\epsilon), (s \times t)' = \begin{pmatrix}(\partial_0 s \times t) + (\underline{s(\epsilon)} \times \partial_0 t)\\ (\partial_1 s \times t) + (\underline{s(\epsilon)} \times \partial_1 t)\end{pmatrix}$
- Faltungs-Inverses: $\underline{\ }^{-1}\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $s(\epsilon) \neq 0 \Rightarrow (s \times s^{-1} = \underline{1})$

Ebenfalls in [3] werden die Eigenschaften der Struktur $(\mathbb{R}^{\mathbb{B}^*}, +, \times)$ mit einem Verweis auf Coinduktion aufgelistet, die $(\mathbb{R}^{\mathbb{B}^*}, +, \times)$ zum unitären Ring machen. Der wesentliche Unterschied zum Stromkalkül besteht darin, dass $\times$ auf $\mathbb{R}^{\mathbb{B}^*}$ nicht kommutativ ist.
Nichtsdestotrotz lässt sich laut [3] ein Analogon zum Fundamentalsatz auf Bäumen beschreiben:

Satz 10 (Fundamentalsatz des Baumkalküls). $\forall s \in \mathbb{R}^{\mathbb{B}^*} : s = \underline{s(\epsilon)} + (L \times \partial_0 s) + (R \times \partial_1 s)$

7.1.4 Übertragung weiterer Operatoren vom Strom- auf den Baumkalkül

Die bisher beschriebenen Operatoren des Baumkalküls finden alle eine Entsprechung im Stromkalkül. In diesem Abschnitt wird versucht weitere Operatoren auf den Baumkalkül zu übertragen und einige ihrer Eigenschaften nachzuweisen.

- Negation: $-\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $(-s)(\epsilon) = -(s(\epsilon)), (-s)' = \begin{pmatrix}-(\partial_0 s)\\ -(\partial_1 s)\end{pmatrix}$
- Mischprodukt: $\otimes\colon \mathbb{R}^{\mathbb{B}^*} \times \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $(s \otimes t)(\epsilon) = s(\epsilon) * t(\epsilon)$,
  $(s \otimes t)' = \begin{pmatrix}(\partial_0 s \otimes t) + (s \otimes \partial_0 t)\\ (\partial_1 s \otimes t) + (s \otimes \partial_1 t)\end{pmatrix}$

- Mischinverses: $\overline{\_^{-1}}\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $s(\epsilon) \neq 0 \Rightarrow (\overline{s^{-1}})(\epsilon) = s(\epsilon)^{-1}$,
$(\overline{s^{-1}})' = \begin{pmatrix} -(\partial_0 s) \otimes \overline{s^{-1}} \otimes \overline{s^{-1}} \\ -(\partial_1 s) \otimes \overline{s^{-1}} \otimes \overline{s^{-1}} \end{pmatrix}$
- Exponentialfunktion: $exp\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $exp(s)(\epsilon) = \exp(s(\epsilon)), exp(s)' = \begin{pmatrix} \partial_0 s \otimes exp(s) \\ \partial_1 s \otimes exp(s) \end{pmatrix}$
- Logarithmus: $log\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $s(\epsilon) > 0 \Rightarrow log(s)(\epsilon) = \log(s(\epsilon)), log(s)' = \begin{pmatrix} \partial_0 s \otimes \overline{s^{-1}} \\ \partial_1 s \otimes \overline{s^{-1}} \end{pmatrix}$
- Sinus: $sin\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $sin(s)(\epsilon) = \sin(s(\epsilon)), sin(s)' = \begin{pmatrix} \partial_0 s \otimes cos(s) \\ \partial_1 s \otimes cos(s) \end{pmatrix}$
- Cosinus: $cos\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $cos(s)(\epsilon) = \cos(s(\epsilon)), cos(s)' = \begin{pmatrix} \partial_0 s \otimes -sin(s) \\ \partial_1 s \otimes -sin(s) \end{pmatrix}$
- Sinus Hyperbolicus: $sinh\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $sinh(s)(\epsilon) = \sinh(s(\epsilon)), sinh(s)' = \begin{pmatrix} \partial_0 s \otimes cosh(s) \\ \partial_1 s \otimes cosh(s) \end{pmatrix}$
- Cosinus Hyperbolicus: $cosh\colon \mathbb{R}^{\mathbb{B}^*} \to \mathbb{R}^{\mathbb{B}^*}$, mit $cosh(s)(\epsilon) = \cosh(s(\epsilon)), cosh(s)' = \begin{pmatrix} \partial_0 s \otimes sinh(s) \\ \partial_1 s \otimes sinh(s) \end{pmatrix}$

Diese direkt aus dem Stromkalkül übersetzten Operatoren bilden ein wohlgeformtes Tree-Differentialgleichungssystem und besitzen daher eine eindeutige Lösung in $\mathbb{R}^{\mathbb{B}^*}$.

7.1.5 Untersuchung übertragener Operatoren

Die Existenz und Eindeutigkeit der Lösung eines wohlgeformten Tree-Differentialgleichungssystems liefert dieselbe Beweismethode im Baumkalkül wie im Kontext des Stromkalküls.

Korollar 4 (Eindeutigkeit der Lösung eines wohlgeformten Tree-Differentialgleichungssystems)**.** *Sei $\mathcal{D}$ ein wohlgeformtes Tree-Differentialgleichungssystem für Menge von Funktionssymbolen $\Sigma = \{f, g, \ldots\}$. Seien $\mathcal{A}$ und $\mathcal{B}$ Lösungen von $\mathcal{D}$, dann gilt:*
$\forall f \in \Sigma : (f_{\mathcal{A}}$ *ist Lösung von f in* $\mathcal{A}) \wedge (f_{\mathcal{B}}$ *ist Lösung von f in* $\mathcal{B}) \Rightarrow f_{\mathcal{A}} = f_{\mathcal{B}}$

Im Folgenden wird dieselbe Schreibweise für Beweise solcher Art verwendet, wie für Beweise im Stromkalkül. Das Symbol $\rightsquigarrow$ wird verwendet, um darauf hinzuweisen, dass bei gleichbleibender Termstruktur eine Ersetzung der Funktion stattfindet. Die Termstruktur wird in eckigen Klammern mit Hilfe des Funktionssymbols φ angegeben und ist der essentielle Bestandteil des zugrundeliegenden wohlgeformten Tree-Differentialgleichungssystems.

Der Ring $(\mathbb{R}^{\mathbb{B}^*}, +, \otimes)$

Lemma 7. $(\mathbb{R}^{\mathbb{B}^*}, +, \otimes)$ *ist ein kommutativer unitärer Ring.*

Beweis. In [3] wurde bereits gezeigt, dass $(\mathbb{R}^{\mathbb{B}^*}, +)$ bis auf die Invertierbarkeit alle Eigenschaften einer abelschen Gruppe mit dem neutralen Element $\underline{0}$ besitzt. Für die Invertierbarkeit zeige

$\forall s \in \mathbb{R}^{\mathbb{B}^*} : (-s) + s = \underline{0}$.
Seien dazu α, β einstellige Abbildungen mit $\alpha(s) = (-s) + s$ und $\beta(s) = \underline{0}$. Dann gilt:

$$\alpha(s)(\epsilon) = ((-s) + s)(\epsilon) = -(s(\epsilon)) + s(\epsilon) = 0 = \beta(s)(\epsilon)$$

$$\begin{aligned}
\alpha(s)' = ((-s) + s)' &= \begin{pmatrix} -(\partial_0 s) + \partial_0 s \\ -(\partial_1 s) + \partial_1 s \end{pmatrix} \\
&= \begin{pmatrix} \alpha(\partial_0 s) \\ \alpha(\partial_1 s) \end{pmatrix} \\
[\varphi(x_1)' &= \begin{pmatrix} \varphi(\partial_0 x_1) \\ \varphi(\partial_1 x_1) \end{pmatrix}] \\
&\leadsto \begin{pmatrix} \beta(\partial_0 s) \\ \beta(\partial_1 s) \end{pmatrix} \\
&= \begin{pmatrix} \underline{0} \\ \underline{0} \end{pmatrix} = \underline{0}' \\
&= \beta(s)'
\end{aligned}$$

Da α und β als Lösungen des gleichen Funktionssymbols eines Tree-Differentialgleichungssystems dargestellt werden können, folgt aus Korollar 4
$\forall s \in \mathbb{R}^{\mathbb{B}^*} : (-s) + s = \alpha(s) = \beta(s) = \underline{0}$.
Zeige nun die Kommutativität von $\otimes$. Seien dazu α, β zweistellige Abbildungen mit $\alpha(s,t) = s \otimes t$ und $\beta(s,t) = t \otimes s$. Dann gilt:

$$\alpha(s,t)(\epsilon) = s(\epsilon) * t(\epsilon) = t(\epsilon) * s(\epsilon) = \beta(s,t)(\epsilon)$$

$$\begin{aligned}
\alpha(s,t)' &= (s \otimes t)' \\
&= \begin{pmatrix} (\partial_0 s \otimes t) + (s \otimes \partial_0 t) \\ (\partial_1 s \otimes t) + (s \otimes \partial_1 t) \end{pmatrix} \\
&= \begin{pmatrix} \alpha(\partial_0 s, t) + \alpha(s, \partial_0 t) \\ \alpha(\partial_1 s, t) + \alpha(s, \partial_1 t) \end{pmatrix} \\
[\varphi(x_1, x_2)' &= \begin{pmatrix} \varphi(\partial_0 x_1, x_2) + \varphi(x_1, \partial_0 x_2) \\ \varphi(\partial_1 x_1, x_2) + \varphi(x_1, \partial_1 x_2) \end{pmatrix}] \\
&\leadsto \begin{pmatrix} \beta(\partial_0 s, t) + \beta(s, \partial_0 t) \\ \beta(\partial_1 s, t) + \beta(s, \partial_1 t) \end{pmatrix} \\
&= \begin{pmatrix} (t \otimes \partial_0 s) + (\partial_0 t \otimes s) \\ (t \otimes \partial_1 s) + (\partial_1 t \otimes s) \end{pmatrix} \\
&= \begin{pmatrix} (\partial_0 t \otimes s) + (t \otimes \partial_0 s) \\ (\partial_1 t \otimes s) + (t \otimes \partial_1 s) \end{pmatrix} \\
&= (t \otimes s)' \\
&= \beta(s,t)'
\end{aligned}$$

Da α und β als Lösungen des gleichen Funktionssymbols eines Tree-Differentialgleichungssystems dargestellt werden können, folgt aus Korollar 4
$\forall s, t \in \mathbb{R}^{\mathbb{B}^*} : s \otimes t = \alpha(s,t) = \beta(s,t) = t \otimes s$.

Zeige nun, dass das Distributivgesetz gilt.
Seien dazu α, β dreistellige Abbildungen mit $\alpha(s,t,u) = s \otimes (t+u)$ und $\beta(s,t,u) = (s \otimes t) + (s \otimes u)$. Dann gilt:

$$\alpha(s,t,u)(\epsilon) = s(\epsilon) * (t(\epsilon) + u(\epsilon)) = s(\epsilon) * t(\epsilon) + s(\epsilon) * u(\epsilon) = \beta(s,t,u)(\epsilon)$$

$\alpha(s,t,u)'$

[Definition von α]

$= (s \otimes (t+u))' =$

[Definition von $\otimes$]

$= \begin{pmatrix} (\partial_0 s \otimes (t+u)) + (s \otimes \partial_0(t+u)) \\ (\partial_1 s \otimes (t+u)) + (s \otimes \partial_1(t+u)) \end{pmatrix}$

[Definition von $+$]

$= \begin{pmatrix} (\partial_0 s \otimes (t+u)) + (s \otimes (\partial_0 t + \partial_0 u)) \\ (\partial_1 s \otimes (t+u)) + (s \otimes (\partial_1 t + \partial_1 u)) \end{pmatrix}$

[Definition von α]

$= \begin{pmatrix} \alpha(\partial_0 s, t, u) + \alpha(s, \partial_0 t, \partial_0 u) \\ \alpha(\partial_1 s, t, u) + \alpha(s, \partial_1 t, \partial_1 u) \end{pmatrix}$

$[\varphi(x_1,x_2,x_3)' = \begin{pmatrix} \varphi(\partial_0 x_1, x_2, x_3) + \varphi(x_1, \partial_0 x_2, \partial_0 x_3) \\ \varphi(\partial_1 x_1, x_2, x_3) + \varphi(x_1, \partial_1 x_2, \partial_1 x_3) \end{pmatrix}]$

$\rightsquigarrow \begin{pmatrix} \beta(\partial_0 s, t, u) + \beta(s, \partial_0 t, \partial_0 u) \\ \beta(\partial_1 s, t, u) + \beta(s, \partial_1 t, \partial_1 u) \end{pmatrix}$

[Definition von β]

$= \begin{pmatrix} ((\partial_0 s \otimes t) + (\partial_0 s \otimes u)) + ((s \otimes \partial_0 t) + (s \otimes \partial_0 u)) \\ ((\partial_1 s \otimes t) + (\partial_1 s \otimes u)) + ((s \otimes \partial_1 t) + (s \otimes \partial_1 u)) \end{pmatrix}$

[Assoziativität und Kommutativität von $+$]

$= \begin{pmatrix} (\partial_0 s \otimes t) + (s \otimes \partial_0 t) + (\partial_0 s \otimes u) + (s \otimes \partial_0 u) \\ (\partial_1 s \otimes t) + (s \otimes \partial_1 t) + (\partial_1 s \otimes u) + (s \otimes \partial_1 u) \end{pmatrix}$

[Definition von $\otimes$]

$= \begin{pmatrix} \partial_0(s \otimes t) + \partial_0(s \otimes u) \\ \partial_1(s \otimes t) + \partial_1(s \otimes u) \end{pmatrix}$

[Definition von $+$]

$= ((s \otimes t) + (s \otimes u))'$

[Definition von β]

$= \beta(s,t,u)'$

Da α und β als Lösungen des gleichen Funktionssymbols eines Tree-Differentialgleichungssystems dargestellt werden können, folgt aus Korollar 4
$\forall s,t,u \in \mathbb{R}^{\mathbb{B}^*} : s \otimes (t+u) = \alpha(s,t,u) = \beta(s,t,u) = (s \otimes t) + (s \otimes u)$.
Analog lässt sich zeigen
$\forall s,t,u \in \mathbb{R}^{\mathbb{B}^*} : (t+u) \otimes s = (t \otimes s) + (u \otimes s)$.

Zeige nun die Eigenschaft des Nullelements $\underline{0}$, $\forall s \in \mathbb{R}^{\mathbb{B}^*} : s \otimes \underline{0} = \underline{0}$.

Seien α, β einstellige Abbildungen mit $\alpha(s) = s \otimes \underline{0}$ und $\beta(s) = \underline{0}$. Dann gilt $\forall s \in \mathbb{R}^{\mathbb{B}^*}$:

$$\alpha(s)(\epsilon) = s(\epsilon) * \underline{0}(\epsilon) = 0 = \beta(s)(\epsilon)$$

$$\begin{aligned}
&\alpha(s)' \\
&\quad [\text{Definition von } \alpha] \\
&= (s \otimes \underline{0})' \\
&\quad [\text{Definition von } \otimes] \\
&= \begin{pmatrix} (\partial_0 s \otimes \underline{0}) + (s \otimes \partial_0 \underline{0}) \\ (\partial_1 s \otimes \underline{0}) + (s \otimes \partial_1 \underline{0}) \end{pmatrix} \\
&\quad [\text{Definition der Einbettung}] \\
&= \begin{pmatrix} (\partial_0 s \otimes \underline{0}) + (s \otimes \underline{0}) \\ (\partial_1 s \otimes \underline{0}) + (s \otimes \underline{0}) \end{pmatrix} \\
&\quad [\text{Definition von } \alpha] \\
&= \begin{pmatrix} \alpha(\partial_0 s) + \alpha(s) \\ \alpha(\partial_1 s) + \alpha(s) \end{pmatrix} \\
&[\varphi(x_1)' = \begin{pmatrix} \varphi(\partial_0 x_1) + \varphi(x_1) \\ \varphi(\partial_1 x_1) + \varphi(x_1) \end{pmatrix}] \\
&\rightsquigarrow \begin{pmatrix} \beta(\partial_0 s) + \beta(s) \\ \beta(\partial_1 s) + \beta(s) \end{pmatrix} \\
&\quad [\text{Definition von } \beta] \\
&= \begin{pmatrix} \underline{0} \\ \underline{0} \end{pmatrix} \\
&\quad [\text{Definition der Einbettung}] \\
&= \underline{0}' \\
&\quad [\text{Definition von } \beta] \\
&= \beta(s)'
\end{aligned}$$

Da α und β als Lösungen des gleichen Funktionssymbols eines Tree-Differentialgleichungssystems dargestellt werden können, folgt aus Korollar 4 und der Kommutativität von $\otimes$
$\forall s \in \mathbb{R}^{\mathbb{B}^*} : s \otimes \underline{0} = \underline{0} = \underline{0} \otimes s$

Zeige nun, dass $(\mathbb{R}^{\mathbb{B}^*}, \otimes)$ ein Monoid ist.

- Assoziativität: Seien α, β dreistellige Abbildungen mit $\alpha(s,t,u) = s \otimes (t \otimes u)$ und $\beta(s,t,u) = (s \otimes t) \otimes u$. Dann gilt:

$$\alpha(s,t,u)(\epsilon) = s(\epsilon) * t(\epsilon) * u(\epsilon) = \beta(s,t,u)(\epsilon)$$

$$\begin{aligned}
&\alpha(s,t,u)' \\
&\quad [\text{Definition von } \alpha] \\
&= (s \otimes (t \otimes u))'
\end{aligned}$$

[Definition von $\otimes$]

$= \begin{pmatrix} (\partial_0 s \otimes (t \otimes u)) + (s \otimes \partial_0(t \otimes u)) \\ (\partial_1 s \otimes (t \otimes u)) + (s \otimes \partial_1(t \otimes u)) \end{pmatrix}$

[Definition von $\otimes$]

$= \begin{pmatrix} (\partial_0 s \otimes (t \otimes u)) + (s \otimes ((\partial_0 t \otimes u) + (t \otimes \partial_0 u))) \\ (\partial_1 s \otimes (t \otimes u)) + (s \otimes ((\partial_1 t \otimes u) + (t \otimes \partial_1 u))) \end{pmatrix}$

[Distributivgesetz und Assoziativität von $+$]

$= \begin{pmatrix} (\partial_0 s \otimes (t \otimes u)) + (s \otimes (\partial_0 t \otimes u)) + (s \otimes (t \otimes \partial_0 u))) \\ (\partial_1 s \otimes (t \otimes u)) + (s \otimes (\partial_1 t \otimes u)) + (s \otimes (t \otimes \partial_1 u))) \end{pmatrix}$

[Definition von α]

$= \begin{pmatrix} \alpha(\partial_0 s, t, u) + \alpha(s, \partial_0 t, u) + \alpha(s, t, \partial_0 u) \\ \alpha(\partial_1 s, t, u) + \alpha(s, \partial_1 t, u) + \alpha(s, t, \partial_1 u) \end{pmatrix}$

$[\varphi(x_1, x_2, x_3)' = \begin{pmatrix} \varphi(\partial_0 x_1, x_2, x_3) + \varphi(x_1, \partial_0 x_2, x_3) + \varphi(x_1, x_2, \partial_0 x_3) \\ \varphi(\partial_1 x_1, x_2, x_3) + \varphi(x_1, \partial_1 x_2, x_3) + \varphi(x_1, x_2, \partial_1 x_3) \end{pmatrix}]$

$\rightsquigarrow \begin{pmatrix} \beta(\partial_0 s, t, u) + \beta(s, \partial_0 t, u) + \beta(s, t, \partial_0 u) \\ \beta(\partial_1 s, t, u) + \beta(s, \partial_1 t, u) + \beta(s, t, \partial_1 u) \end{pmatrix}$

[Definition von β]

$= \begin{pmatrix} ((\partial_0 s \otimes t) \otimes u) + ((s \otimes \partial_0 t)) \otimes u) + ((s \otimes t) \otimes \partial_0 u) \\ ((\partial_1 s \otimes t) \otimes u) + ((s \otimes \partial_1 t)) \otimes u) + ((s \otimes t) \otimes \partial_1 u) \end{pmatrix}$

[Distributivgesetz und Assoziativität von $+$]

$= \begin{pmatrix} (((\partial_0 s \otimes t) + (s \otimes \partial_0 t)) \otimes u) + ((s \otimes t) \otimes \partial_0 u) \\ (((\partial_1 s \otimes t) + (s \otimes \partial_1 t)) \otimes u) + ((s \otimes t) \otimes \partial_0 u) \end{pmatrix}$

[Definition von $\otimes$]

$= \begin{pmatrix} (\partial_0 (s \otimes t) \otimes u) + ((s \otimes t) \otimes \partial_0 u) \\ (\partial_1 (s \otimes t) \otimes u) + ((s \otimes t) \otimes \partial_1 u) \end{pmatrix}$

[Definition von $\otimes$]

$= ((s \otimes t) \otimes u)'$

[Definition von β]

$= \beta(s, t, u)'$

Da α und β als Lösungen des gleichen Funktionssymbols eines Tree-Differentialgleichungssystems dargestellt werden können, folgt aus Korollar 4
$\forall s, t, u \in \mathbb{R}^{\mathbb{B}^*} : s \otimes (t \otimes u) = \alpha(s, t, u) = \beta(s, t, u) = (s \otimes t) \otimes u.$

- Neutrales Element $\underline{1}$: Seien α, β einstellige Abbildungen mit $\alpha(s) = s \otimes \underline{1}$ und $\beta(s) = s$. Dann gilt $\forall s \in \mathbb{R}^{\mathbb{B}^*}$:

$$\alpha(s)(\epsilon) = s(\epsilon) * \underline{1}(\epsilon) = s(\epsilon) = \beta(s)(\epsilon)$$

$\alpha(s)'$

[Definition von α]

$= (s \otimes \underline{1})'$

[Definition von $\otimes$]

$$= \begin{pmatrix} (\partial_0 s \otimes \underline{1}) + (s \otimes \partial_0 \underline{1}) \\ (\partial_1 s \otimes \underline{1}) + (s \otimes \partial_1 \underline{1}) \end{pmatrix}$$

[Definition der Einbettung]

$$= \begin{pmatrix} (\partial_0 s \otimes \underline{1}) + (s \otimes \underline{0}) \\ (\partial_1 s \otimes \underline{1}) + (s \otimes \underline{0}) \end{pmatrix}$$

[Eigenschaft des Nullelements]

$$= \begin{pmatrix} \partial_0 s \otimes \underline{1} \\ \partial_1 s \otimes \underline{1} \end{pmatrix}$$

[Definition von α]

$$= \begin{pmatrix} \alpha(\partial_0 s) \\ \alpha(\partial_1 s) \end{pmatrix}$$

$$[\varphi(x_1)' = \begin{pmatrix} \varphi(\partial_0 x_1) \\ \varphi(\partial_1 x_1) \end{pmatrix}]$$

$$\rightsquigarrow \begin{pmatrix} \beta(\partial_0 s) \\ \beta(\partial_1 s) \end{pmatrix}$$

[Definition von β]

$$= \begin{pmatrix} \partial_0 s \\ \partial_1 s \end{pmatrix}$$

[Definition der Ableitung]

$$= s'$$

[Definition von β]

$$= \beta(s)'$$

Da α und β als Lösungen des gleichen Funktionssymbols eines Tree-Differentialgleichungssystems dargestellt werden können, folgt aus Korollar 4
$\forall s \in \mathbb{R}^{\mathbb{B}^*} : s \otimes \underline{1} = \alpha(s) = \beta(s) = s.$
Analog lässt sich zeigen
$\forall s \in \mathbb{R}^{\mathbb{B}^*} : \underline{1} \otimes s = s.$

□

8 Implementierung

Zur Implementierung des Stromkalküls wurde die Programmiersprache Haskell ausgewählt. Haskell bietet zum einen eine einfache Syntax, um coinduktive Datenstrukturen zu beschreiben, und zum anderen die Möglichkeit mit Hilfe von "Lazy Evaluation" auf ihnen zu arbeiten ohne sie ganz auswerten zu müssen.

8.1 Stream-Datenstruktur

Zuerst muss die coinduktive Datenstruktur beschrieben werden. Der Typ der Elemente eines Streams wird variabel gelassen und erst bei weiteren Einschränkungen oder zur Laufzeit spezifiziert.

```
data Stream a = Cons {value :: a, derivative :: Stream a}
```

Die Funktionen `value :: Stream a -> a` und `derivative :: Stream a -> Stream a` sollen den Destruktoren σ_1, σ_2 der Stream-Coalgebra entsprechen. Der Konstruktor `Cons :: a ->`

`Stream a -> Stream a` soll aus einem Startwert und einer Ableitung den entsprechenden Stream erzeugen. Mit anderen Worten soll gelten:
$\forall$`s` $\in$ `Stream a`, `x` $\in$ `a`:
`Cons value(s) derivative(s)` $= s$
`value(Cons x s)` $=$ `x`
`derivative(Cons x s)` $=$ `s`.

8.2 Instanziierung als `Show, Functor, Applicative`

Um eine Ausgabe zu erhalten, wird `Stream a` zur Instanz der Haskell-Klasse `Show` gemacht. Es werden dabei nur die ersten 10 Elemente des Streams ausgegeben.

```
instance (Show a) => Show (Stream a) where
    show = ('(':).(showStream 10) where
        showStream 0 _ = "...)"
        showStream n (Cons s0 s') = (show s0) ++ ", "++ showStream (n-1) s'
```

Haskell erlaubt es, mit Hilfe des Konstruktors `Cons` wohlgeformte Stream-Differentialgleichungssysteme in legale Funktionsdefinitionen zu übersetzen. Dazu müssen die Beschreibungen des Startwerts und Ableitung jedes Operators mit dem Konstruktor `Cons` gekapselt werden. Die Einbettung $\_ : \mathbb{R} \to \mathbb{R}^\omega$ kann verallgemeinert auf folgende Weise implementiert werden.

```
fromNum :: Num a => a -> Stream a
fromNum n = Cons n (fromNum 0)
```

Nach Definition gelten die Gleichheiten, die ein wohlgeformtes Stream-Differentialgleichungssystem auszeichnen:
`value(fromNum n)` $=$ `value(Cons n (fromNum 0))` $=$ `n`
`derivative(fromNum n)` $=$ `derivative(Cons n (fromNum 0))` $=$ `fromNum 0`.

Um die Handhabung zu vereinfachen, wird `Stream` zur Instanz der Haskell-Klassen `Functor` und `Applicative` gemacht. Eine Instanziierung als `Functor` ermöglicht es, Funktionen vom Typ `a->b` in Funktionen vom Typ `Stream a->Stream b` zu transformieren.

```
instance Functor Stream where
    fmap f = r where r (Cons s0 s') = Cons (f s0) (r s')
```

Eine Instanziierung als `Applicative` erlaubt Funktionen zu kapseln und gekapselte Funktionen anzuwenden. `Applicative` in Haskell entspricht der Klasse `Idiom` in [4].

```
instance Applicative Stream where
    pure s0 = r where r = Cons s0 r
    (Cons f0 f') <*> (Cons s0 s') = Cons (f0 s0) (f' <*> s')
```

Es ist durch Haskell gefordert (siehe [7]), dass folgende Gleichheiten erfüllt werden:
`fmap f x` $=$ `pure f <*> x`
`u <*> pure y = pure ($ y) <*> u`
`pure id <*> v = v`
`pure (.) <*> u <*> v <*> w = u <*> (v <*> w)`
`pure f <*> pure x = pure (f x)`
Die erste Gleichheit folgt coinduktiv mit der Bisimulation $\{($`fmap f x`, `pure f <*> x`$) \mid$ `x` $\in$ `Stream a`, `f :: a -> a`$\}$. Für die restlichen Gleichheiten siehe die Idiom-Axiome in [4].

8.3 Instanziierung als `Num`

Haskell besitzt einige Klassen zur Beschreibung von Strukturen mit Ring-Eigenschaften.
Die Klasse `Num` erlaubt in Haskell die Beschreibung von Strukturen mit Ring-Eigenschaften und fordert die Definition folgender Funktionen:

- Addition `(+) :: Num a => a -> a -> a`
- Negation `negate :: Num a => a -> a`
- Subtraktion `(-) :: Num a => a -> a -> a`
- Multiplikation `(*) :: Num a => a -> a -> a`
- Betrag `abs :: Num a => a -> a`
- Vorzeichen `signum :: Num a => a -> a`
  geknüpft an die Bedingung `abs x * signum x == x`
- Integer-Einbettung `fromInteger :: Num a => Integer -> a`

Sowohl der Ring $(\mathbb{R}^\omega, +, \times)$ als auch der Ring $(\mathbb{R}^\omega, +, \otimes)$ lässt sich als `Num` instantiieren. In dieser Arbeit wird der zweit-genannte Ring aus mehreren Gründen vorgezogen. Zum einen steht das Mischprodukt im direkten Zusammenhang mit weiteren Stream Operatoren (z.B. exp), zum anderen verliert es, im Gegensatz zum Faltungsprodukt, bei der Übertragung auf den Baumkalkül nicht die Kommutativität. Das Vorzeichen und der Betrag eines Streams haben bei der bisherigen Betrachtung keine Rolle gespielt und werden deshalb nur definiert, um die o.g. Forderung zu erfüllen. Alle anderen Definitionen ergeben sich direkt aus ihrer Beschreibung.

```haskell
instance (Num a) => Num (Stream a) where
   (Cons s0 s') + (Cons t0 t') = Cons (s0 + t0) (s' + t')
   negate (Cons s0 s') = Cons (negate s0) (negate s')
   (Cons s0 s') - (Cons t0 t') = Cons (s0 - t0) (s' - t')
   s@(Cons s0 s') * t@(Cons t0 t') = Cons (s0 * t0) ((s' * t)+(s * t'))
   abs s = s
   signum s = 1
   fromInteger n = Cons (fromInteger n) 0
```

8.4 Instanziierung als `Fractional`

Die Klasse `Fractional` ist eine Unterklasse von `Num` und fordert Invertierbarkeit der Elemente
`recip :: Fractional a => a -> a`.
Die Einheitenmenge von $\mathbb{R}^\omega$ bezüglich $\otimes$ ist $\mathbb{R}^\omega_{\neq 0}$ und ist in Haskell einfach identifizierbar. Das Inverse ist durch ein wohlgeformtes Stream-Differentialgleichungssystem beschreibbar (siehe [1]). Folglich ergibt sich die Instanziierung direkt aus der Theorie.

```haskell
instance (Fractional a) => Fractional (Stream a) where
   recip (Cons 0 _) = error "Non-invertible Stream"
   recip (Cons s0 s') = r where r = Cons (recip s0) (negate (s'*(r*r)))
   fromRational q = Cons (fromRational q) 0
```

Mit Hilfe von `recip` lässt sich Die Division `(/) :: Fractional a => a -> a -> a` als Multiplikation mit dem Inversen definieren.

8.5 Instanziierung als `Floating`

Die Klasse `Floating` ist eine Unterklasse von `Fractional` und fordert die Definition analytischer Konstanten und Funktionen `pi, exp, log, sin, cos, sinh, cosh, asin, acos, atan, asinh, acosh, atanh`. Die ersten sieben Funktionen wurden im Kontext des Stromkalküls behandelt und besitzen eine Definition in Form eines wohldefinierten Stream-Differentialgleichungssystems. Die letztgenannten 6 Funktionen werden als `undefined` definiert.

```
instance (Ord a, Floating a) => Floating (Stream a) where
    pi = Cons pi 0
    exp (Cons s0 s') = r where r = Cons (exp s0) (s' * r)
    log s@(Cons s0 s') | s0 > 0 = Cons (log s0) (s' / s)
                       | otherwise = error "Logarithm not defined for this Stream"
    sin s@(Cons s0 s') = Cons (sin s0) (s' * (cos s))
    cos s@(Cons s0 s') = Cons (cos s0) (negate (s' * (sin s)))
    asin = undefined
    atan = undefined
    acos = undefined
    sinh s@(Cons s0 s') = Cons (sinh s0) (s' * (cosh s))
    cosh s@(Cons s0 s') = Cons (cosh s0) (s' * (sinh s))
    asinh = undefined
    atanh = undefined
    acosh = undefined
```

8.6 Weitere Streamoperatoren

In dem folgenden Abschnitt werden die restlichen theoretisch behandelten Operatoren definiert.
Formale Variable `var`

```
var :: Num a => Stream a
var = Cons 0 (Cons 1 0)
```

Faltungsprodukt `(#)`:

```
(#) :: Num a => Stream a -> Stream a -> Stream a
s@(Cons s0 s') # t@(Cons t0 t') = Cons (s0 * t0) ((s' # t)+(fmap (*s0) t'))
```

Faltungs-Inverses `crecip`:

```
crecip :: Fractional a => Stream a -> Stream a
crecip (Cons 0 _) = error "Non-invertible Stream"
crecip (Cons s0 s') = r where r = Cons (recip s0) (fmap (*(recip (-s0))) (s' # r))
```

Faltungs-Division `(/#)`:

```
(/#) :: Fractional a => Stream a -> Stream a -> Stream a
s /# t = s # (crecip t)
```

Faltungs-Wurzel `csqrt`:

```
csqrt :: (Ord a, Floating a) => Stream a -> Stream a
csqrt (Cons s0 s') | s0 <= 0 = error "Cannot compute Stream squareroot"
                   | otherwise = r where r = Cons (sqrt s0) (s' /# (((fromNum.sqrt) s0)+r))
```

Komposition `(<.>)`:

```
(<.>) :: Num a => Stream a -> Stream a -> Stream a
(Cons s0 s') <.> t@(Cons 0 t') = Cons s0 (t' # (s' <.> t))
_ <.> _ = error "Non composable second argument"
```

Analytische Ableitung `anaDerivative`:

```
anaDerivative :: Num a ⇒ Stream a → Stream a
anaDerivative (Cons _ s') = derivative ( var ∗ s' )
```

8.7 Implementierungen des Fibonacci-Streams

Der Fibonacci-Stream `fib` kann auf verschiedene Arten implementiert werden.
Eine Möglichkeit ist es, die Gleichheit `fib'' = fib' + fib` zu nutzen, um ein wohlgeformtes Stream-Differentialgleichungssystem zu formen.

```
fib1 :: Num a ⇒ Stream a
fib1 = Cons 0 fib1'
fib1' :: Num a ⇒ Stream a
fib1' = Cons 1 (fib1 + fib1')
```

Dieses Gleichungssystem lässt sich in Haskell kompakt formulieren.

```
fib2 :: Num a ⇒ Stream a
fib2 = Cons 0 r where r = (Cons 1 (fib2 + r))
```

Eine andere Möglichkeit ist es, mit einer weiteren Methode eine Liste mit dem Inhalt des Fibonacci-Streams zu erstellen (siehe [9]) und diese dann in einen Stream zu konvertieren.

```
fibs :: [Integer]
fibs = map fst ( iterate (λ(a,b) → (b,a+b)) (0,1) )
fib3 = foldr Cons undefined fibs
```

Des Weiteren lässt sich der Fibonacci-Stream direkt durch seine geschlossene Form darstellen.

```
fib4 :: Fractional a ⇒ Stream a
fib4 = var /# (1-var-(var # var))
```

Die exponentiell erzeugende Funktion liefert eine weitere geschlossene Darstellung.

```
fib5 :: (Ord a, Floating a) ⇒ Stream a
fib5 = fmap (∗(2/sqrt(5))) (exp(fmap (∗0.5) var)∗sinh(fmap (∗(0.5∗sqrt(5))) var))
```

Auswertungsgeschwindigkeit

Um die Auswertungsgeschwindigkeiten der unterschiedlichen Implementierungen zu betrachten, wird eine Funktion definiert, die einen Stream in eine einstellige Funktion transformiert.

```
(<!) :: Num a ⇒ Stream t → (a → t)
(Cons s0 _) <! 0 = s0
(Cons _ s') <! n = s' <! (n-1)
```

In der folgenden Tabelle sind die Ergebnisse der Auswertung an den Positionen 10,30 und 1000 an einem Referenzcomputer. In machen Fällen konnte die Auswertung nicht ausgeführt werden, da der Systemspeicher nicht ausreichte.

Implementierung	**Laufzeit (10,30,1000)**			**Speicherverbrauch (10,30,1000)**		
fib1	0.03 secs	12.81 secs	Fehler	4316064 bytes	1678695848 bytes	Fehler
fib2	0.02 secs	0.02 secs	0.51 secs	2114172 bytes	2115484 bytes	114089508 bytes
fib3	0.02 secs	0.02 secs	0.03 secs	2112444 bytes	3794852 bytes	3848964 bytes
fib4	0.02 secs	0.03 secs	6.36 secs	2112752 bytes	4839308 bytes	739503228 bytes
fib5	0.05 secs	Fehler	Fehler	3686468 bytes	Fehler	Fehler

Es ist zu erkennen, dass das direkte Differentialgleichungssystem (fib1) und die Darstellungen als geschlossene Form (fib4, fib5) nicht für die Praxis gedacht sind. Zudem können fib4, fib5 nicht auf ganzen Zahlen operieren, da ihre Definition Funktionen beinhaltet, die auf ganzen Zahlen nicht geschlossen sind.

8.8 BinTree-Datenstruktur

Ein binärer Baum besitzt neben einem Wurzelknoteneintrag (Startwert) zwei Kinder (Ableitungen), daher muss die BinTree-Datenstruktur genau diese Attribute besitzen.

```
data BinTree a = Node {value :: a, derivative1 :: BinTree a, derivative2 :: BinTree a}
```

Analog zur Stream-Datenstruktur lassen sich wohlgeformte Differentialgleichungen direkt in Haskell-Definitionen umformen. Dazu müssen der Startwert und die beiden Ableitungen an den Konstruktor Node übergeben werden. Die Einbettung $\_ : \mathbb{R} \to \mathbb{R}^{\mathbb{B}^*}$ kann in Haskell durch die folgende Funktion beschrieben werden:

```
fromNum :: Num a => a -> BinTree a
fromNum n = Node n (fromNum 0) (fromNum 0)
```

Dank dieser Umformungsmöglichkeit lässt sich BinTree a analog zu Stream a als Num, Fractional, Floating instantiieren. Auch weitere Operatoren auf BinTree a (z.B. Faltungsprodukt (#)) lassen sich damit in Haskell definieren. Zusätzlich lässt sich eine Funktion definieren, die einen Baum in die Abbildung transformiert, die er darstellt.

```
(<!) :: BinTree a -> ([Char] -> a)
Node s0 _ _ <! "" = s0
(Node _ d1s _) <! ('0':path) = d1s <! (path)
(Node _ _ d2s) <! ('1':path) = d2s <! (path)
_ <! _ = error "Invalid path"
```

Die Implementierung des Baumkalküls befindet sich im Anhang.

8.8.1 BinTree-Beispiele

In diesem Abschnitt werden einige Bäume definiert und gezeichnet. Für das Zeichnen der Bäume wurde **Graphviz** in der Funktion showBinTree verwendet. var1 und var2 entsprechen den formalen Variablen.
Das erste Beispiel ist ein Baum, dessen Einträge der Anzahl der Nullen im Pfad zum jeweiligen Eintrag entsprechen. Mit anderen Worten wächst der Eintrag mit der ersten partiellen Ableitung und mit der zweiter partiellen Ableitung bleibt er konstant.

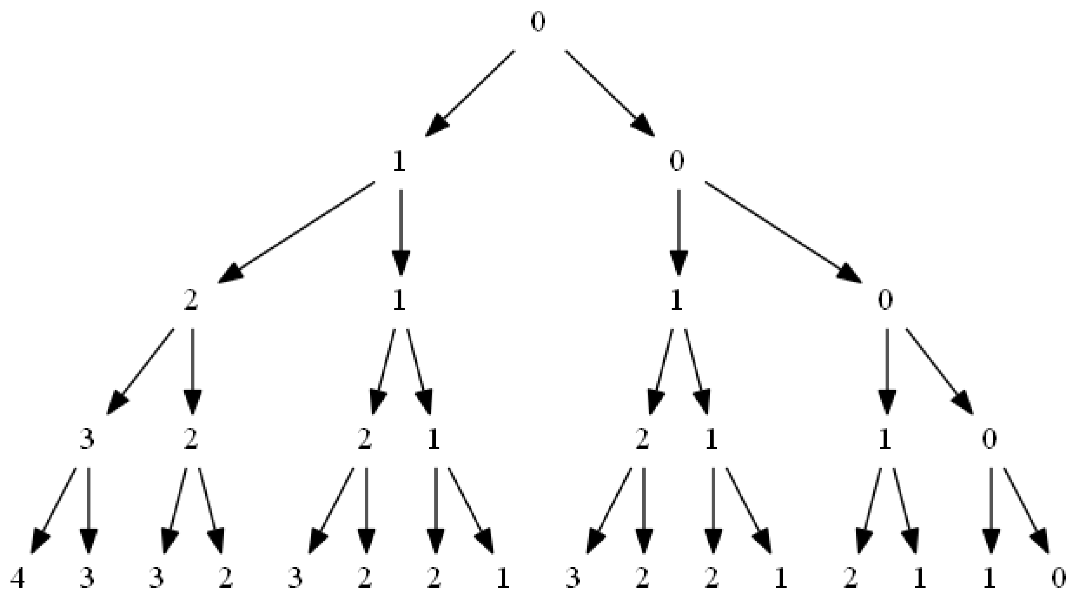

Abbildung 3: `tree1 = exp(var1+var2)*var1`

Das zweite Beispiel ist ein Baum, dessen Einträge der Zweierpotenz der Länge des Pfades zum jeweiligen Eintrag entsprechen. Mit anderen Worten sind die Einträge auf der Baumebene n alle 2^n.

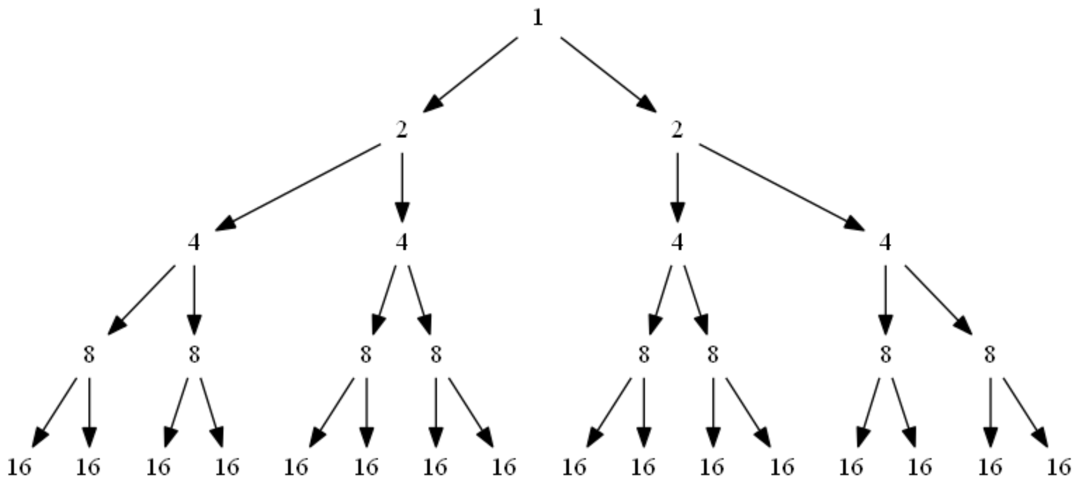

Abbildung 4: `tree2 = exp(var1+var1+var2+var2)`

Das dritte Beispiel ist ein Baum, dessen Einträge auf jeder Ebene den ersten natürlichen Zahlen entsprechen.

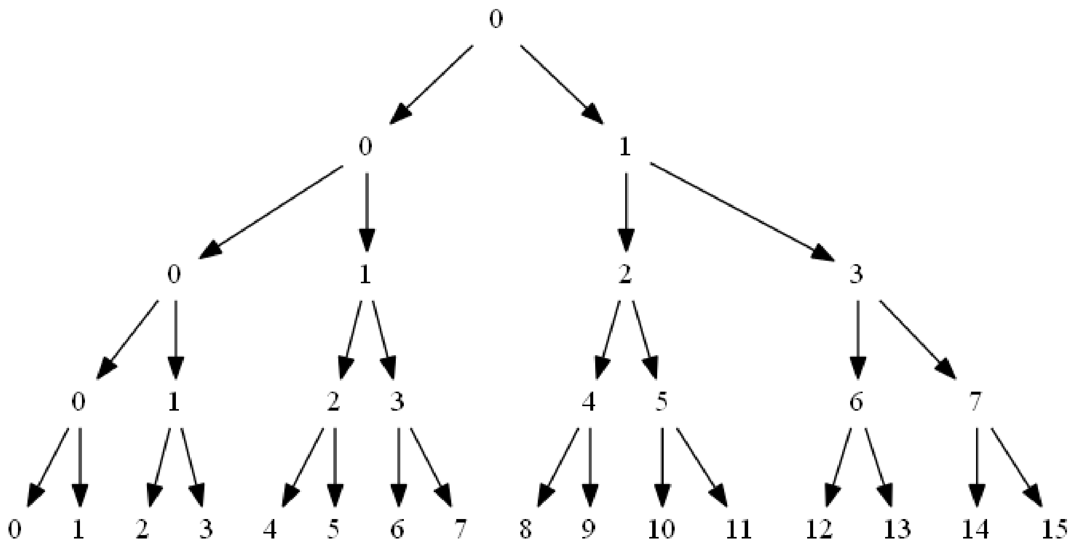

Abbildung 5: `tree3 = Node 0 tree3 tree3 + ((var2) # exp(var1+var1+var2+var2))`

Das vierte Beispiel ist ein Baum, dessen Eintrage Ebene für Ebene fortlaufend nummeriert sind.

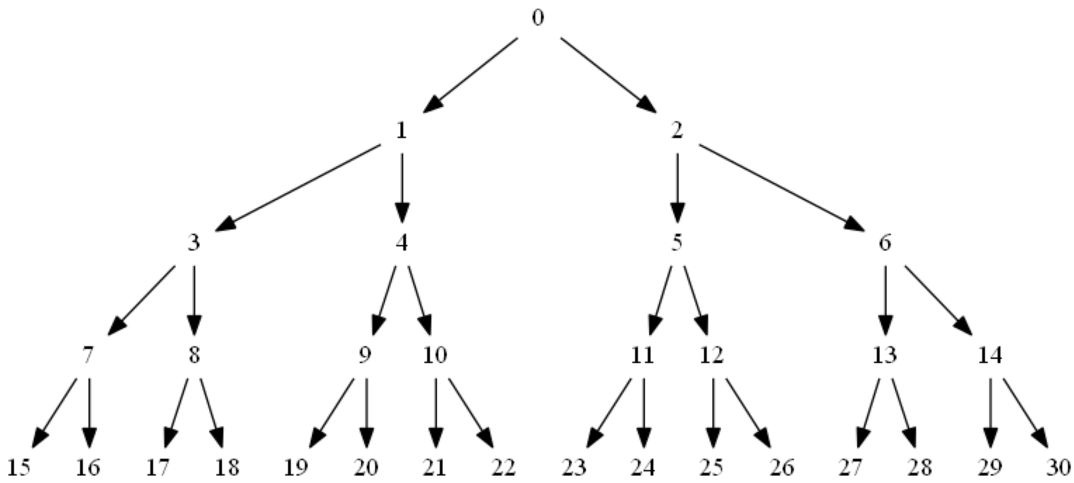

Abbildung 6: `tree4 = Node 0 tree4 tree4 + ((var1+var2+var2) # exp(var1+var1+var2+var2))`

Das fünfte Beispiel ist ein Baum, dessen Einträge auf der Ebene n der n-ten Fibonacci-Zahl entsprechen.

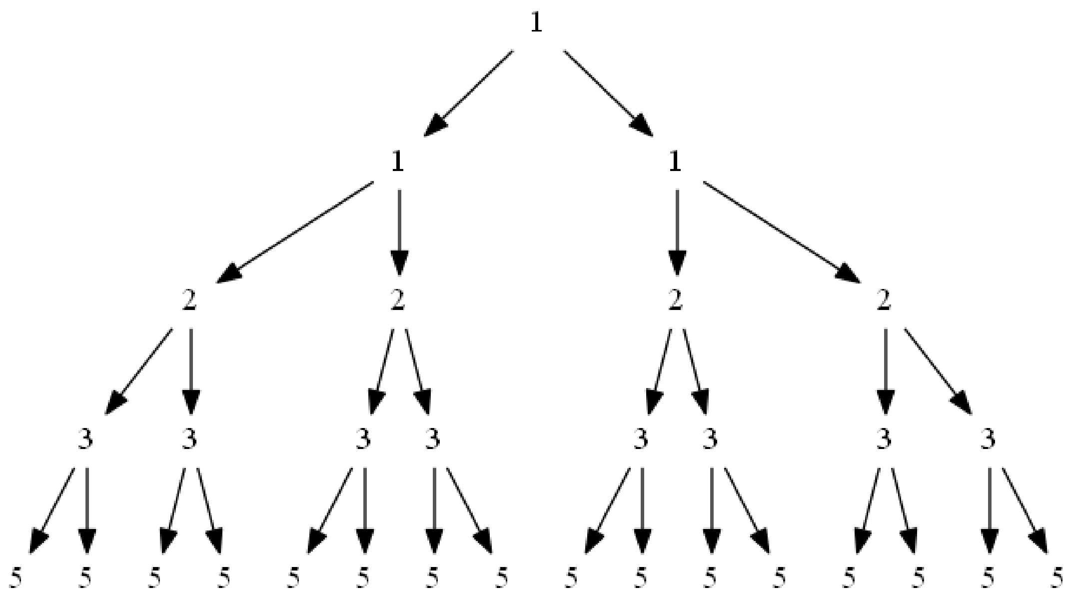

Abbildung 7: `tree5 = (1/#(1-var1-var2-(var1 # var1)-(var2 # var2)-(var1 * var2)))`

Das sechste Beispiel ist ein Baum, bei welchem der Wurzeleintrag im linken Teilbaum der Summe der Einträge der beiden Vorgängerknoten entsprechen und der Wurzeleintrag im rechten Teilbaum der Differenz.

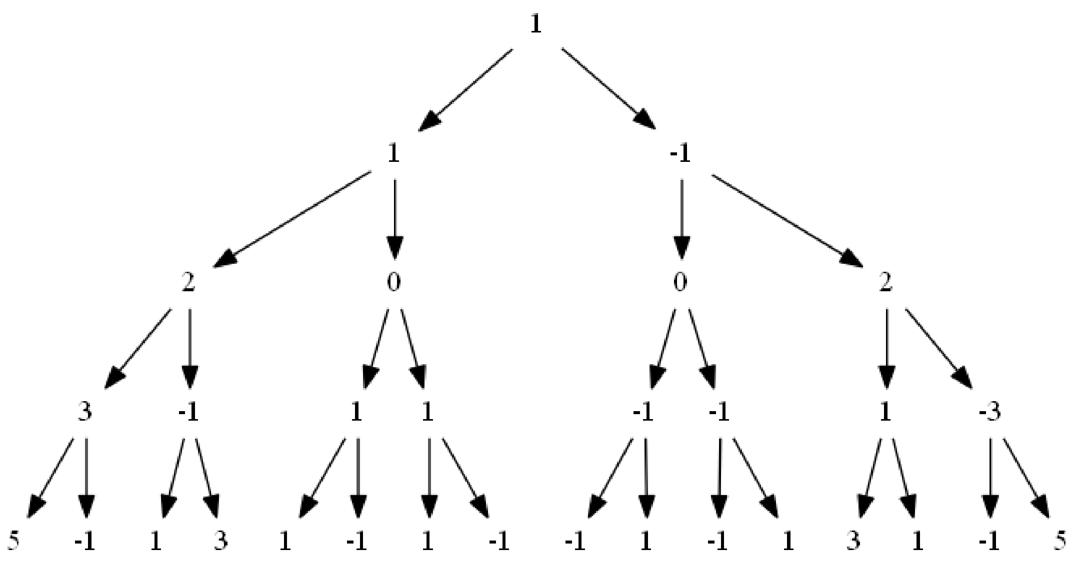

Abbildung 8: `tree6 = (1/#(1-var1+var2-(var1 # var1)-(var2 # var2)-(var1 * var2)))`

9 Zusammenfassung

In dieser Arbeit wird der Stromkalkül über die finale Stream-Coalgebra σ eingeführt. Der Existenz- und Eindeutigkeitssatz, der auf der coalgebraischen Struktur des Stromkalküls beruht, liefert die Grundlage zur Definition einer Vielzahl von Operatoren. Die Menge der Streams kann mit einigen der Operatoren algebraisch klassifiziert werden (Ringe $(\mathbb{R}^\omega, +, \times)$ und $(\mathbb{R}^\omega, +, \otimes)$). Des Weiteren können Isomorphismen zwischen formalen Potenzreihen, exponentiell erzeugenden Funktionen aus der Analysis und Teilen des Stromkalküls gezeigt werden. Zusätzlich wurden in dieser Arbeit weitere Streamoperatoren definiert und ihr Zusammenhang gezeigt. Am Ende wurde die Struktur der binären Bäume betrachtet und einige Aussagen und Operatoren des Stromkalküls auf diese Struktur übertragen. Zum Schluss wurden die Operatoren des Strom- und des Baumkalküls in der Programmiersprache Haskell implementiert und anhand einiger Beispiele verwendet.

10 Anhang

Stromkalkülimplementierung in Haskell

```haskell
import Control.Applicative

data Stream a = Cons {value :: a, derivative :: Stream a}

instance (Show a) => Show (Stream a) where
    show = ('(':).(showStream 10) where
        showStream 0 _ = "...)"
        showStream n (Cons s0 s') = (show s0) ++ ", "++ showStream (n-1) s'

--Stream embed num
fromNum :: Num a => a -> Stream a
fromNum n = Cons n (fromNum 0)

instance Functor Stream where
    fmap f = r where r (Cons s0 s') = Cons (f s0) (r s')

instance Applicative Stream where
    pure s0 = r where r = Cons s0 r
    (Cons f0 f') <*> (Cons s0 s') = Cons (f0 s0) (f' <*> s')

instance (Eq a) => Eq (Stream a) where
    (==) = compareStream 50 where
        compareStream 0 _ _ = True
        compareStream n (Cons s0 s') (Cons t0 t') =
         s0==t0 && (compareStream (n-1) s' t')

instance (Num a) => Num (Stream a) where
    --Stream addition
    (Cons s0 s') + (Cons t0 t') = Cons (s0 + t0) (s' + t')
    --Stream negation
    negate (Cons s0 s') = Cons (negate s0) (negate s')
    --Stream difference
    (Cons s0 s') - (Cons t0 t') = Cons (s0 - t0) (s' - t')
    --Stream shuffle product
    s@(Cons s0 s') * t@(Cons t0 t') = Cons (s0 * t0) ((s' * t)+(s * t'))
    abs s = s
    signum s = 1
    fromInteger n = Cons (fromInteger n) 0

--Stream to function transformation
(<!) :: Num a => Stream t -> (a -> t)
(Cons s0 _) <! 0 = s0
(Cons _ s') <! n = s' <! (n-1)
```

```haskell
--Formal variable
var :: Num a => Stream a
var = Cons 0 (Cons 1 0)

instance (Fractional a) => Fractional (Stream a) where
    --Stream shuffle inverse
    recip (Cons 0 _) = error "Non-invertible Stream"
    recip (Cons s0 s') = r where r = Cons (recip s0) (negate (s'*(r*r)))
    fromRational q = Cons (fromRational q) 0

instance (Ord a, Floating a) => Floating (Stream a) where
    pi = Cons pi 0
    --Stream exp
    exp (Cons s0 s') = r where r = Cons (exp s0) (s' * r)
    --Stream log
    log s@(Cons s0 s') | s0 > 0 = Cons (log s0) (s' / s)
                       | otherwise = error "Logarithm not defined for this Stream"
    --Stream sin
    sin s@(Cons s0 s') = Cons (sin s0) (s' * (cos s))
    --Stream cos
    cos s@(Cons s0 s') = Cons (cos s0) (negate (s' * (sin s)))
    asin = undefined
    atan = undefined
    acos = undefined
    --Stream sinh
    sinh s@(Cons s0 s') = Cons (sinh s0) (s' * (cosh s))
    --Stream cosh
    cosh s@(Cons s0 s') = Cons (cosh s0) (s' * (sinh s))
    asinh = undefined
    atanh = undefined
    acosh = undefined


--Stream convolution product
(#) :: Num a => Stream a -> Stream a -> Stream a
s@(Cons s0 s') # t@(Cons t0 t') = Cons (s0 * t0) ((s' # t)+(fmap (*s0) t'))

--Stream convolution reciprocal
crecip :: Fractional a => Stream a -> Stream a
crecip (Cons 0 _) = error "Non-invertible Stream"
crecip (Cons s0 s') = r where r = Cons (recip s0) (fmap (*(recip (-s0))) (s' # r))

--Stream convolution division
(/#) :: Fractional a => Stream a -> Stream a -> Stream a
s /# t = s # (crecip t)

--Stream convolution squareroot
csqrt :: (Ord a, Floating a) => Stream a -> Stream a
csqrt (Cons s0 s') | s0 <= 0 = error "Cannot compute Stream squareroot"
```

```haskell
                     | otherwise = r where
                        r = Cons (sqrt s0) (s' /# (((fromNum.sqrt) s0)+r))

--Stream composition
(<.>) :: Num a => Stream a -> Stream a -> Stream a
(Cons s0 s') <.> t@(Cons 0 t') = Cons s0 (t' # (s' <.> t))
_ <.> _ = error "Non composable second argument"

--Stream analytical derivative
anaDerivative :: Num a => Stream a -> Stream a
anaDerivative (Cons _ s') = derivative $ var * s'

--Stream Laplace-Carson transform
lcTransform :: Num a => Stream a -> Stream a
lcTransform s = Cons (value s) (lcTransform(anaDerivative(s)))

--Stream difference
difference :: Num a => Stream a -> Stream a
difference s = (derivative s) - s


--Fibonacci Streams

--Fibonacci-Stream from a differential equation system
fib1 :: Num a => Stream a
fib1 = Cons 0 fib1'
fib1' :: Num a => Stream a
fib1' = Cons 1 (fib1 + fib1')

--Fibonacci-Stream from combined differential equations
fib2 :: Num a => Stream a
fib2 = Cons 0 r where r = (Cons 1 (fib2 + r))

--http://www.haskell.org/haskellwiki/The_Fibonacci_sequence
fibs :: [Integer]
fibs = map fst $ iterate (\(a,b) -> (b,a+b)) (0,1)
--Fibonacci-Stream from list
fib3 = foldr Cons undefined fibs

--Fibonacci-Stream from a closed form
fib4 :: Fractional a => Stream a
fib4 = var /# (1-var-(var # var))
fib41 = Cons 0 (crecip(1-var-(var # var)))

--Fibonacci-Stream from an exponential closed form
fib5 :: (Ord a, Floating a) => Stream a
fib5 = fmap (*(2/sqrt(5))) (exp(fmap (*0.5) var)*sinh(fmap (*(0.5*sqrt(5))) var))
```

Baumkalkülimplementierung in Haskell

```haskell
import System

import Control.Applicative

data BinTree a =
 Node {value :: a, derivative1 :: BinTree a, derivative2 :: BinTree a}

instance (Show a) => Show (BinTree a) where
    show = ('(':).(showBinTree 4) where
        showBinTree 0 _ = ""
        showBinTree 1 (Node s0 _ _) = (show s0)
        showBinTree n (Node s0 d1s d2s) =
         "["++ showBinTree (n-1) d1s ++ "] " ++ (show s0) ++ " ["++
         showBinTree (n-1) d2s ++ "]"

--Tree embed num
fromNum :: Num a => a -> BinTree a
fromNum n = Node n (fromNum 0) (fromNum 0)

instance Functor BinTree where
    fmap f (Node s0 d1s d2s) = Node (f s0) (fmap f d1s) (fmap f d2s)

instance Applicative BinTree where
    pure s0 = r where r = Node s0 r r
    (Node f0 d1f d2f) <*> (Node s0 d1s d2s) =
     Node (f0 s0) (d1f <*> d1s) (d2f <*> d2s)

instance (Eq a) => Eq (BinTree a) where
    (==) = compareBinTree 50 where
        compareBinTree 0 _ _ = True
        compareBinTree n (Node s0 d1s d2s) (Node t0 d1t d2t) =
         s0==t0 && (compareBinTree (n-1) d1s d1t) && (compareBinTree (n-1) d2s d2t)

instance (Num a) => Num (BinTree a) where
    --BinTree addition
    (Node s0 d1s d2s) + (Node t0 d1t d2t) = Node (s0+t0) (d1s+d1t) (d2s+d2t)
    --BinTree negation
    negate (Node s0 d1s d2s) = Node (negate s0) (negate d1s) (negate d2s)
    --BinTree difference
    (Node s0 d1s d2s) - (Node t0 d1t d2t) = Node (s0-t0) (d1s-d1t) (d2s-d2t)
    --BinTree shuffle product
    s@(Node s0 d1s d2s) * t@(Node t0 d1t d2t) =
     Node (s0*t0) (d1s*t + s*d1t) (d2s*t + s*d2t)
    abs s = s
    signum s = fromInteger 1
    fromInteger n = Node (fromInteger n) 0 0
```

```haskell
--Formal variables
var1 :: Num a => BinTree a
var1 = Node 0 1 0

var2 :: Num a => BinTree a
var2 = Node 0 0 1

instance (Fractional a) => Fractional (BinTree a) where
    --BinTree shuffle inverse
    recip (Node 0 _ _) = error "Non-invertible BinTree"
    recip (Node s0 d1s d2s) = r where r =
     Node (recip s0) (negate (d1s*(r*r))) (negate (d2s*(r*r)))
    fromRational q = Node (fromRational q) 0 0

instance (Ord a, Floating a) => Floating (BinTree a) where
    pi = Node pi 0 0
    --BinTree exp
    exp (Node s0 d1s d2s) = r where r = Node (exp s0) (d1s * r) (d2s * r)
    --BinTree log
    log s@(Node s0 d1s d2s) | s0 > 0 = Node (log s0) (d1s / s) (d2s / s)
                            | otherwise =
                              error "Logarithm not defined for this BinTree"
    --BinTree sin
    sin s@(Node s0 d1s d2s) = Node (sin s0) (d1s * (cos s)) (d2s * (cos s))
    --BinTree cos
    cos s@(Node s0 d1s d2s) =
     Node (cos s0) (negate (d1s * (sin s))) (negate (d2s * (sin s)))
    asin = undefined
    atan = undefined
    acos = undefined
    --BinTree sinh
    sinh s@(Node s0 d1s d2s) = Node (sinh s0) (d1s * (cosh s)) (d2s * (cosh s))
    --BinTree cosh
    cosh s@(Node s0 d1s d2s) = Node (cosh s0) (d1s * (sinh s)) (d2s * (sinh s))
    asinh = undefined
    atanh = undefined
    acosh = undefined

--BinTree convolution product
(#) :: Num a => BinTree a -> BinTree a -> BinTree a
s@(Node s0 d1s d2s) # t@(Node t0 d1t d2t) =
 Node (s0*t0) (d1s # t + (fmap (*s0) d1t)) (d2s # t + (fmap (*s0) d2t))

--BinTree convolution reciprocal
crecip :: Fractional a => BinTree a -> BinTree a
crecip (Node 0 _ _) = error "Non-invertible BinTree"
crecip (Node s0 d1s d2s) = r where r =
 Node (recip s0) (fmap (*(recip (-s0))) (d1s # r)) (fmap (*(recip (-s0))) (d2s # r))
```

```haskell
--BinTree convolution division
(/#) :: Fractional a => BinTree a -> BinTree a -> BinTree a
s /# t = s # (crecip t)

--BinTree convolution squareroot
csqrt :: (Ord a, Floating a) => BinTree a -> BinTree a
csqrt (Node s0 d1s d2s) | s0 <= 0 = error "Cannot compute BinTree squareroot"
                        | otherwise = r where r =
                          Node (sqrt s0) (d1s /# (((fromNum.sqrt) s0)+r))
                                         (d2s /# (((fromNum.sqrt) s0)+r))

(<!) :: BinTree a -> ([Char] -> a)
Node s0 _ _ <! "" = s0
(Node _ d1s _) <! ('0':path) = d1s <! (path)
(Node _ _ d2s) <! ('1':path) = d2s <! (path)
_ <! _ = error "Invalid path"

--BinTree printing functions require graphviz/dot (http://www.graphviz.org/)
--Write BinTree s to "tree.gv", compile "tree.gv" to "tree.png" and show "tree.png"
showBinTree s = do writeBinTree "tree.gv" s
                   system "dot -Tpng -o tree.png tree.gv"
                   system "tree.png"

--Write BinTree s to file including level 5
writeBinTree file s = writeFile file
 ("digraph tree {node [shape=none,width=0,height=0];nodesep=0.1;" ++
  binTreeToDot 5 "s" s ++ "}")

--Convert BinTree to dot string
binTreeToDot 0 _ _ = ""
binTreeToDot 1 path (Node s0 _ _) = path ++ "[label=\"" ++ show s0 ++ "\"];"
binTreeToDot n path (Node s0 d1s d2s) =
 path ++ "[label=\"" ++ show s0 ++ "\"];" ++ path ++ "->" ++
 path ++ "0;" ++ path ++ "->" ++ path ++ "1;" ++
 (binTreeToDot (n-1) (path++"0") d1s) ++ (binTreeToDot (n-1) (path++"1") d2s)

--left: increase, right:constant
tree1 = exp(var1+var2)*var1
--each level: 2^level
tree2 = exp(var1+var1+var2+var2)
--each level: [0,1..]
tree3 = Node 0 tree3 tree3 + ((var2) # exp(var1+var1+var2+var2))
--consecutive numbered
tree4 = Node 0 tree4 tree4 + ((var1+var2+var2) # exp(var1+var1+var2+var2))
--each level: fibonacci number
tree5 = (1/#(1-var1-var2-(var1 # var1)-(var2 # var2)-(var1 * var2)))
--left: parent(parent(x))+parent(x)=x, right: parent(parent(x))-parent(x)=x
tree6 = (1/#(1-var1+var2-(var1 # var1)-(var2 # var2)-(var1 * var2)))
```

Literatur

[1] J.J.M.M. Rutten: *A coinductive calculus of streams.*

[2] J.J.M.M. Rutten: *Behavioural differential equations: a coinductive calculus of streams, automata, and power series* .

[3] Alexandra Silva, Jan Rutten: *A coinductive calculus of binary trees.*

[4] Ralf Hinze: *Reasoning about Codata.*

[5] D. Pavlovic, M.H. Escardo: *Calculus in coinductive form.*

[6] *http://http://oeis.org*

[7] *http://en.wikibooks.org/wiki/Haskell/Applicative_Functors*

[8] *http://www.haskell.org/ghc/docs/7.0.4/html/libraries/base-4.3.1.0/Prelude.html*

[9] *http://www.haskell.org/haskellwiki/The_Fibonacci_sequence*

[10] *http://tunes.org/wiki/initiality_20and_20finality.html*

[11] *http://www.haskell.org/ghc/docs/6.12.1/html/libraries/base/src/GHC-List.html*