Algebra of Hybrid Systems

Peter Höfner und Bernhard Möller

Institut für Informatik, Universität Augsburg

1 Überblick

- Motivation
- Trajektorien-basierter Ansatz
- Algebraischer Hintergrund
- Work in Progress
- Ausblick

2 Motivation

Die Dynamik natürlicher und künstlicher Systeme zeichnet sich häufig dadurch aus, dass sie an einigen Stellen *kontinuierlich* und an anderen *diskret* ist. Systeme mit einer solchen Dynamik werden als *hybride Systeme* bezeichnet.

- Zeit-diskret / Zeit-kontinuierlich
- Wert-diskret / Wert-kontinuierlich

Systemarten

Transformationelle Systeme

berechnen eine Funktion

Reaktive Systeme

interagieren mit ihrer Umgebung

Echtzeitsysteme

müssen Ergebnisse in bestimmten Zeitintervallen produzieren

Hybride Systeme

besitzen kontinuierliche und diskrete Komponenten

Quelle: Universität Oldenburg

Anwendungsbereiche

- *Biohybride Systeme:*

  Fraunhofer-Institut für Biomedizinische Technik (IBMT)
- *Verkehrsleitsysteme* (z.B. für Autobahnen)
- *aktive Datenbanken*

  (soweit das Auslösen bestimmter Events betroffen ist)
- *Kontrollsysteme*

  (z.B.: Sensoren in Flugzeugen)
- (Generalized) Railroad Crossing
- Gas Brenner
- ...

Warum algebraische Methoden für hybride Systeme?

Bisherige Ansätze:

- unübersichtlich (z.B.: hybride Automaten)
- beispielorientiert - keine Übertragungsmöglichkeit auf andere Beispiele

Ziel:

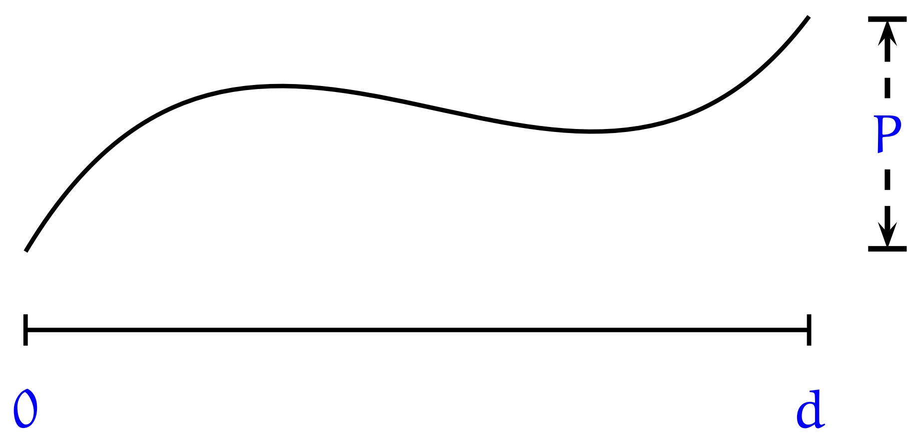
- Strukturen einheitlich und übersichtlich zu beschreiben
- einfache Erweiterung für spezielle Probleme (z.B.: durch zusätzliche Axiome)

3 Trajektorien-basierter Ansatz

- Beschreibung mittels Links-Halbringen
- Trajektorien als Trägermenge
- basierend auf Sintzoff, Henzinger, Davoren, Lynch

Sei V eine Menge von Zuständen (*Wertemenge*) und D eine Menge von Intervalllängen ($\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}, \ldots$).

Definition 3.1 Eine *Trajektorie* t ist ein Paar (d, g), wobei $d \in D$ und $g : [0, d] \to P$, $P \subseteq V$.



- d bezeichnet die *Länge* der Trajektorie, P den *Wertebereich*
- $(D, +, 0)$ kommutatives Monoid
- $+$ kürzbar (cancellative)
- Ordnung auf D: $x \leq y \overset{\text{def}}{\Leftrightarrow} \exists\, z \,.\, x + z = y$
  - 0 kleinstes Element
  - 0 unzerlegbar, d.h. $x + y = 0 \Leftrightarrow x = y = 0$
- $\infty \in D$ möglich
  - größtes Element
  - nicht kürzbar

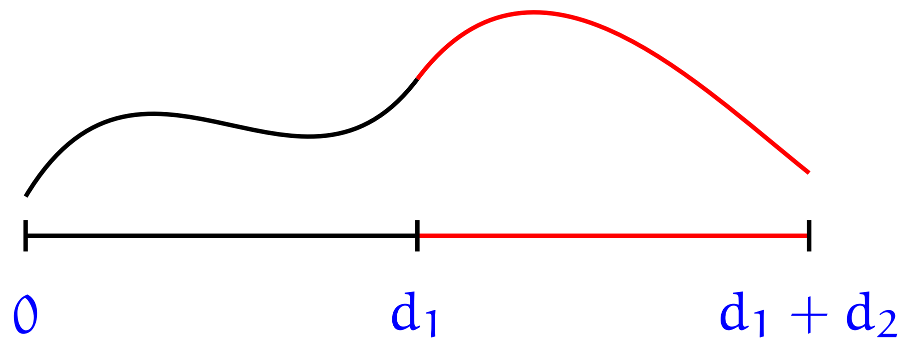
Wir können eine Komposition von Trajektorien definieren.

Seien (d_1, g_1) und (d_2, g_2) Trajektorien über der Wertemenge V

$$(d_1, g_1) \cdot (d_2, g_2) \overset{\text{def}}{=} \begin{cases} (d_1 + d_2, g) & \text{falls } d_1 \neq \infty \wedge g_1(d_1) = g_2(0) \\ (d_1, g_1) & \text{falls } d_1 = \infty \\ \text{undefiniert} & \text{sonst} \end{cases}$$

mit $g(x) = g_1(x) \quad \forall x \in [0, d_1]$

und $g(x + d_1) = g_2(x) \quad \forall x \in [0, d_2]$ bzw. $x \in [0, \infty[$



- $(0, g_1) \cdot (d_2, g_2) = (d_2, g_2)$ oder undefiniert
- TRA: die Menge aller Trajektorien

Definition 3.2 *Prozesse* sind Mengen von Trajektorien, d.h.

$$A \text{ ist Prozess } \overset{\text{def}}{\Leftrightarrow} A \in \mathcal{P}(\mathrm{TRA})$$

$$\mathsf{inf}\, A \overset{\text{def}}{=} \{(d, g) \in A \mid d = \infty\} \qquad \mathsf{fin}\, A \overset{\text{def}}{=} A - \mathsf{inf}\, A$$

Prozesskomposition:

$$A \cdot B \overset{\text{def}}{=} \mathsf{inf}\, A \cup \{a \cdot b \mid a \in \mathsf{fin}\, A, b \in B\}$$

Die Menge I aller Trajektorien der Länge 0 ist neutrales Element.

eingeschränkte Komposition (chop)

$$A \frown B \stackrel{\text{def}}{=} (\mathsf{fin}\, A) \cdot B \ ,$$

- garantiert das Erreichen der zweiten Trajektorie
- $A \cdot B = \mathsf{inf}\, A \cup A \frown B$

Definition 3.3 Mittels Prozessen erhalten wir einen (booleschen) Links-Halbring

$$\mathrm{PRO} \stackrel{\text{def}}{=} (\mathcal{P}(\mathrm{TRA}), \cup, \emptyset, \cdot, \mathrm{I})$$

- durch endliche Iteration ($*$-Operator) zu Links-Kleene-Algebra erweiterbar
- ist $\mathrm{D} = \{0, 1\}$ entspricht PRO den Relationen
- große Ähnlichkeit mit Funktionenräumen der Linearen Algebra
- P diskret, beschreibt diskrete Wert-Verläufe

4 Algebraischer Hintergrund

Definition 4.1 Ein *Links-Halbring* ist ein Quintupel $(A, +, 0, \cdot, 1)$, wobei

- $(A, +, 0)$ ein kommutatives,
- $(A, \cdot, 1)$ ein Monoid ist,
- $(a + b) \cdot c \ = \ a \cdot c + b \cdot c$ und
- $0 \cdot a = 0$ (*Links-Annihilator*) ist.

(Ein Halbring ohne volle Striktheit und Rechtsdistributivität)

Ein Links-Halbring heißt *idempotent*, falls $a + a = a$ gilt und $\cdot$ rechts-isoton ist;
er heißt *boolesch*, falls der zugrunde liegende Verband boolesch ist.

- Links-Distributivität von $\cdot$ impliziert die Links-Isotonie
- natürliche Ordnung: $a \leq b \Leftrightarrow a + b = b$
- 0 kleinstes Element

Definition 4.2 *Links-Test-Halbring* $(S, \mathsf{test}(S))$:

- S ein idempotenter Links-Halbring
- $\mathsf{test}(S) \subseteq [0,1]$ eine Boolesche Subalgebra von $[0,1]$ in S, so dass $0, 1 \in \mathsf{test}(S)$
- in $\mathsf{test}(S) : p + q = p \sqcup q,\ p \cdot q = p \sqcap q$
- $\neg$ beschreibt die Komplementbildung in $\mathsf{test}(S)$

Definition 4.3 *Links-Domain-Halbring* $(S, \ulcorner)$:

- S Links-Test-Halbring
- *domain*-Operation $\ulcorner : S \to \mathsf{test}(S)$

$$a \le \ulcorner a \cdot a\ , \qquad \ulcorner(p \cdot a) \le p\ , \qquad \ulcorner(a \cdot \ulcorner b) \le \ulcorner(a \cdot b)\ .$$

$(a, b \in S, p \in \mathsf{test}(S))$

Es gelten die gleichen Axiome und Konsequenzen wie in Domain-Halbringen [Möller04].

REL, LAN, PRO sind Links-Domain-Halbringe

in PRO:

- $\mathsf{test}(\mathrm{PRO}) = \{(0, g) \mid g(0) = v, v \in V\}$
  (Trajektorien der Länge 0)
- $\ulcorner A = \{(0, g(0)) \mid (d, g) \in A\}$ (Startpunkte)
- $P \in \mathsf{test}(\mathrm{PRO}) \;\Rightarrow\; P \sqcap A \cdot B = (P \sqcap A) \cdot (P \sqcap B)$
- $\cdot$ ist auch recht-distributiv

Definition 4.4 Eine *Links-Kleene-Algebra* ist ein Paar $(A, *)$, wobei

- A ein idempotenter linker Halbring ist und
- folgende Gesetze erfüllt sind:

$$1 + a \cdot a^* \leq a^*, \qquad 1 + a^* \cdot a \leq a^*$$

$$b + a \cdot c \leq c \;\Rightarrow\; a^* \cdot b \leq c, \qquad b + c \cdot a \leq c \;\Rightarrow\; b \cdot a^* b \leq c$$

Beispiele:

- relationale Algebra REL
- formale Sprachen LAN
- Algebra der Prozesse PRO

zusätzliche Operationen (Beispiel)

Wertebereichseinschränkung in PRO

- $\mathsf{P} \in \mathsf{test}(\mathrm{PRO})$ ist isomorph zu $S \subseteq V$
- $\Diamond \mathsf{P} \stackrel{\text{def}}{=} \mathsf{F} \cdot \mathsf{P} \cdot \top$, wobei $\top \stackrel{\text{def}}{=} \mathrm{TRA}$ und $\mathsf{F} \stackrel{\text{def}}{=} \mathsf{fin}\,(\mathrm{TRA})$

  Menge aller Trajektorien, deren Wertebereich P trifft
- $\Box \mathsf{P} \stackrel{\text{def}}{=} \overline{\Diamond \neg \mathsf{P}}$

  Menge aller Trajektorien, deren Wertebereich vollständig in P liegt,

wobei $\top \stackrel{\text{def}}{=} \mathrm{TRA}$ und $\mathsf{F} \stackrel{\text{def}}{=} \mathsf{fin}\,(\mathrm{TRA})$

- geeignet um Sicherheitsanforderungen zu formulieren

Lemma 4.5 Ist S rechts-distributiv und gilt $(\Box p) \cdot (\Box p) \leq \Box p$, dann ist für alle $a, b \in S$ und $p \in \mathsf{test}(S)$,

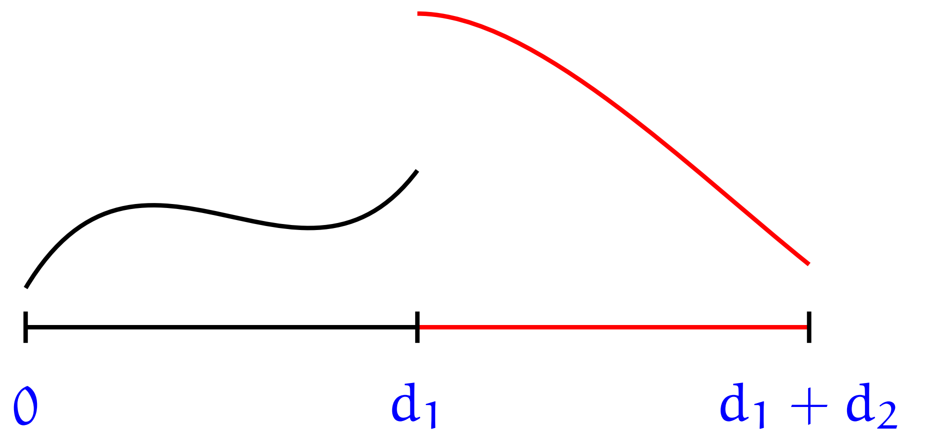
1. $\Box p \sqcap a \cdot b = (\Box p \sqcap a) \cdot (\Box p \sqcap b)$
2. $\Diamond p \sqcap a \cdot b = (\Diamond p \sqcap a) \cdot b + \mathsf{fin}\, a \cdot (\Diamond p \sqcap b)$
3. $(\Box p) \cdot (\Box p) = \Box p$

Weitere mögliche Operatoren

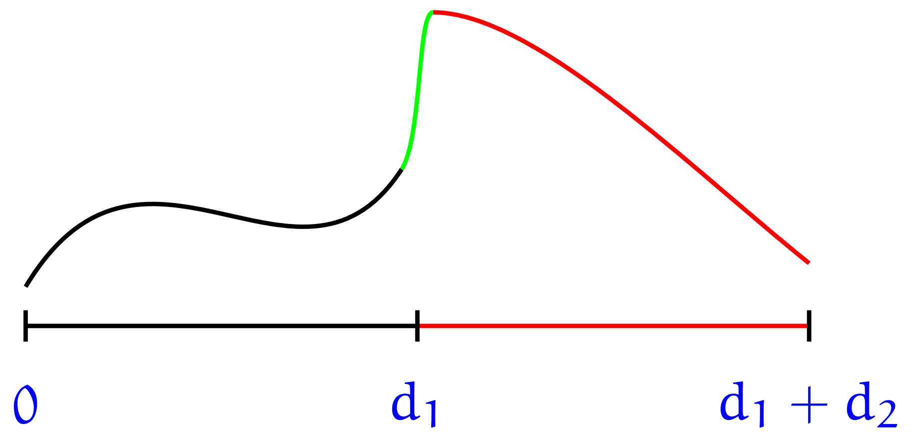
- modale Operationen mittels Quantalen
  (z.B.: „Es existiert eine Trajektorie T in S mit ... “)
- unendliche Iteration
  (Links-ω-Algebra)
- ...

5 Work in Progress

Sprungstellen



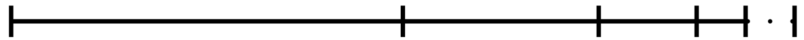
- endliche Sprungstellen ohne Probleme möglich
- wie kann man in PRO Endlichkeit garantieren?
- was passiert bei unendlich vielen Sprungstellen?
- Probleme wie bei Riemann/Lebesque-Integrierbarkeit?



- Näherung durch Trajektorien
- reicht dies für die Praxis aus?
- teilweise realistischer
  (z.B.: Ventile, Messer, ...)

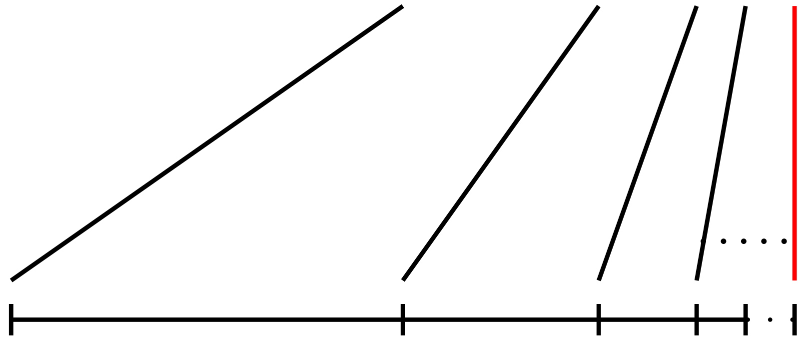
Zeno-Effekte

- beschreiben Situation(en) von unendlicher Iteration ohne die „Unendlichkeit“ zu erreichen



- in Modellen für hybride Systeme meist ausgeschlossen
- in PRO möglich

Sprungstellen und Zeno-Effekte



- beim Zusammenspiel werden Zeno-Effekte auch in PRO problematisch

6 Ausblick

- Einbettung von Neighbourhoodlogic [ZhouHansen1996]
  Möglichkeit der Berücksichtigung der angrenzenden Trajektorien
- Spiel-theoretische Annäherung