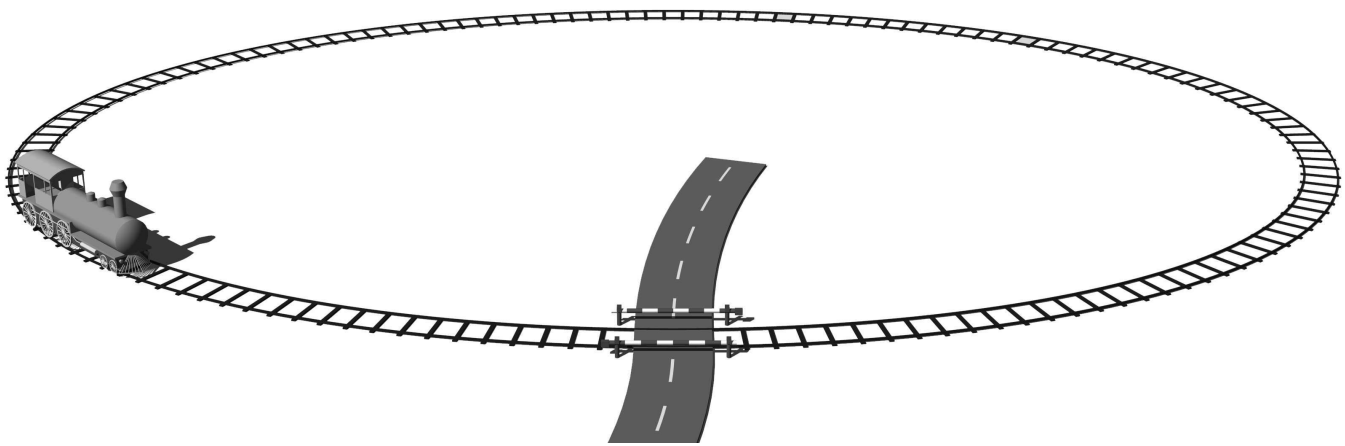
Algebraic Calculi for Hybrid Systems
— Algebraische Kalküle für Hybride Systeme —

Peter Höfner

Universität Augsburg



Motivation

Fehlerhafte Software

- fordert Menschenleben
- ruft Umweltschäden hervor
- führt zu wirtschaftlichen Verlusten

Flughafen Heathrow (Terminal 5, April 2008)

- über 500 gestrichene Flüge,
- über 28.000 Koffer falsch sortiert,
- über $£16$ Mio. Schaden

(trotz mehr als 400.000 Stunden Entwicklungszeit der Software)

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

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

weniger als 1% aller Prozessoren sind in PCs;
mehr als 98% sind Kontroller in hybriden Systemen

Hybride Automaten

"[Bezogen auf hybride Systeme] scheint die Automatentheorie das richtige Werkzeug zu sein. [...] jedoch ist die Behandlung von Automaten mit nicht diskreten Zuständen recht schwierig."

K. Zuse, 1967

Hybride Automaten

"[Bezogen auf hybride Systeme] scheint die Automatentheorie das richtige Werkzeug zu sein. [...] jedoch ist die Behandlung von Automaten mit nicht diskreten Zuständen recht schwierig."

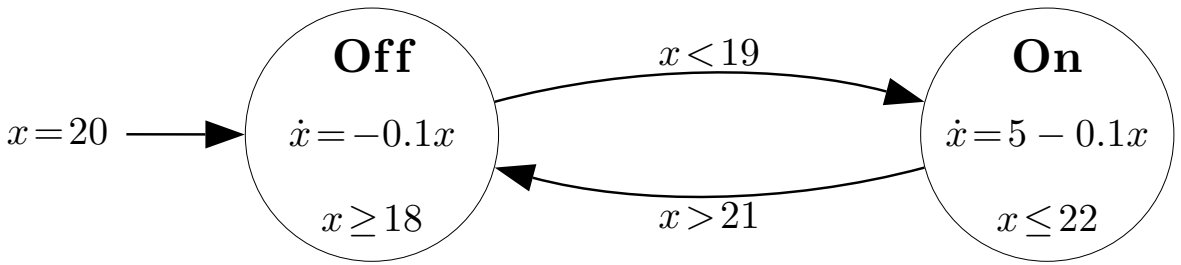
K. Zuse, 1967

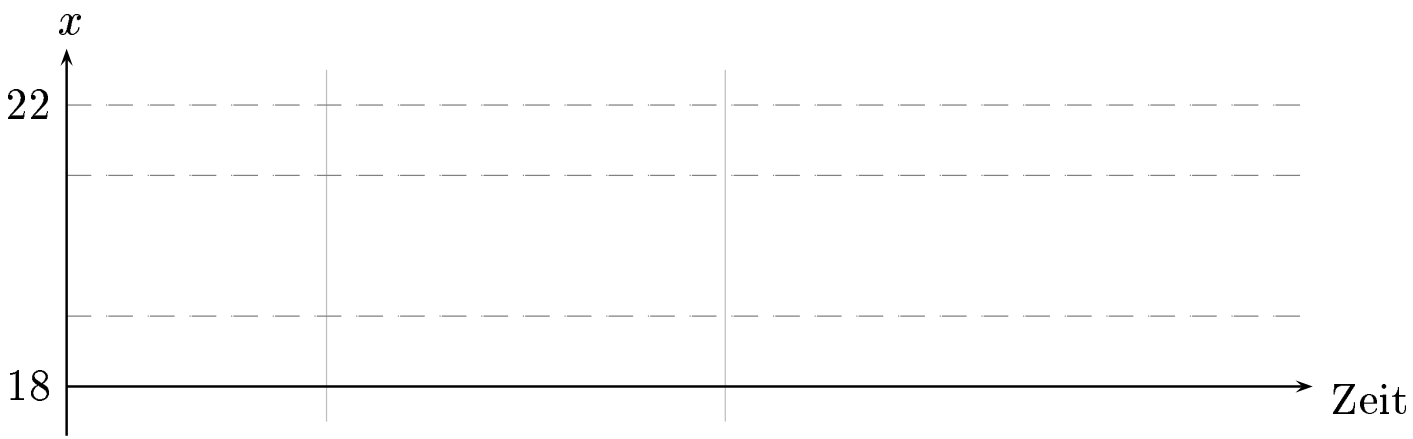
- bekannteste Darstellung von hybriden Systemen
- häufig verwendet für Design und Modellierung
- ähnlich zu endlichen Automaten
- Zustände beschreiben kontinuierliches Systemverhalten
- Kanten beschreiben diskretes Verhalten
- beschreiben systematisch alle Abläufe

Hybride Automaten

Beispiel

Heizung:

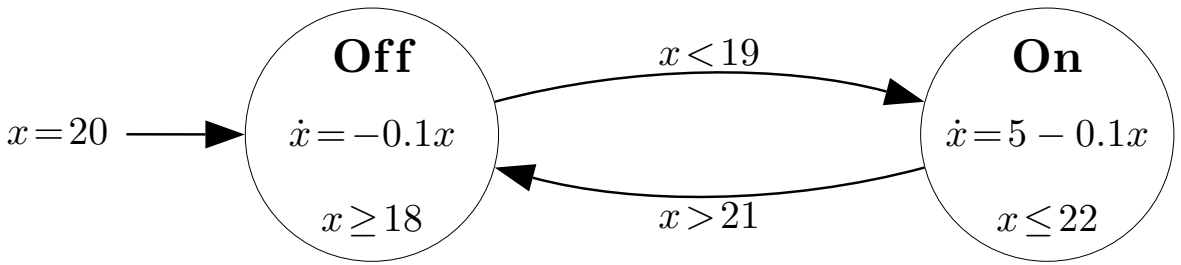


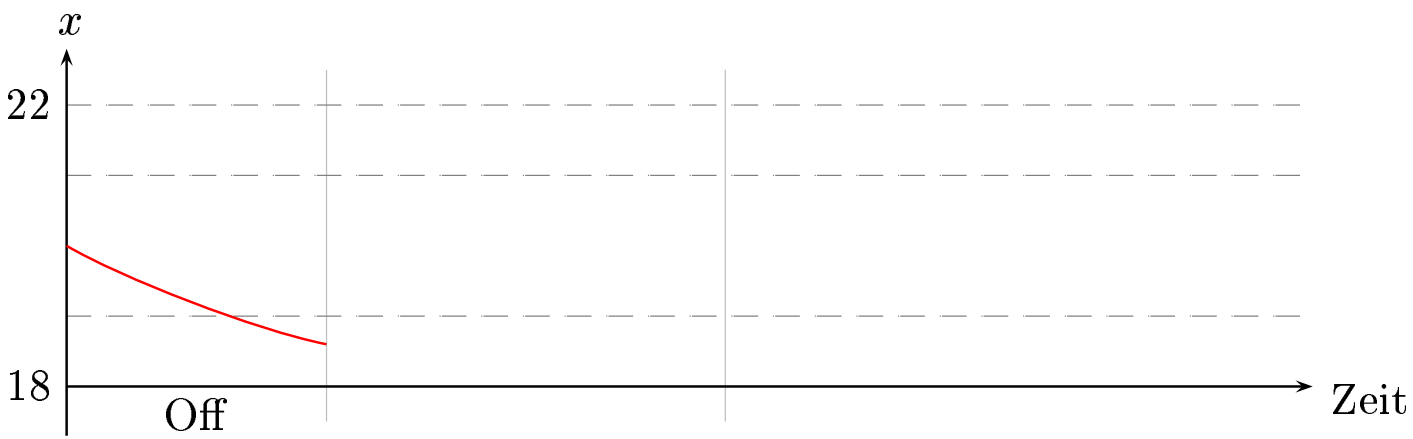


Hybride Automaten

Beispiel

Heizung:

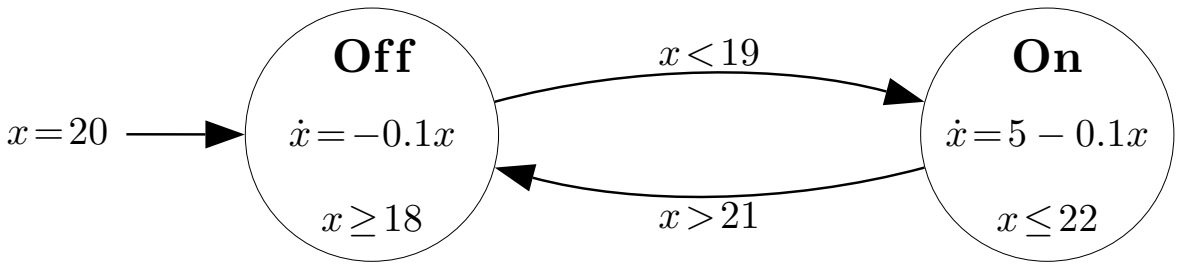


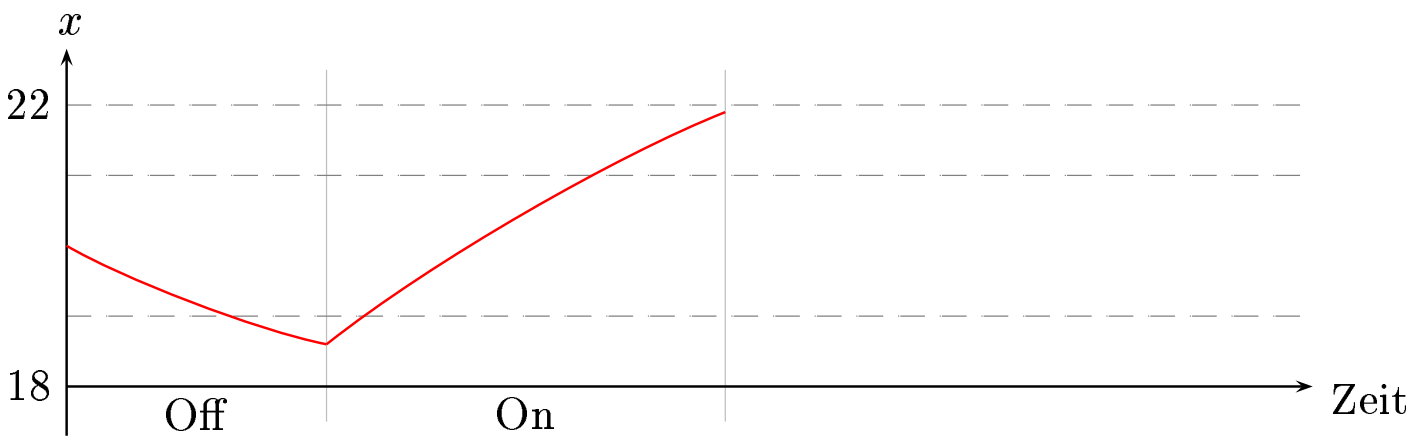


Hybride Automaten

Beispiel

Heizung:

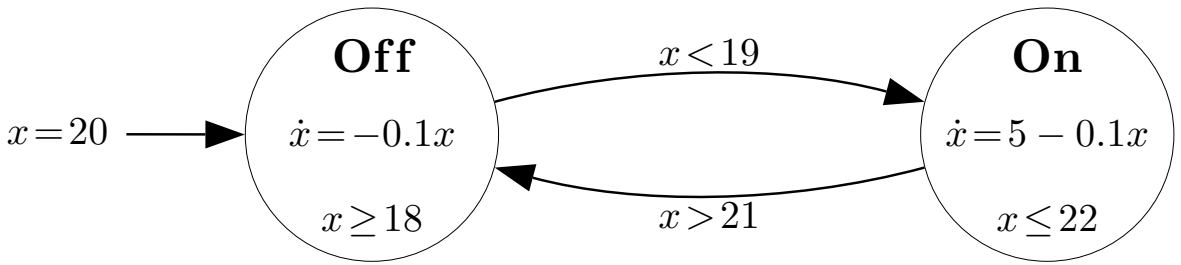


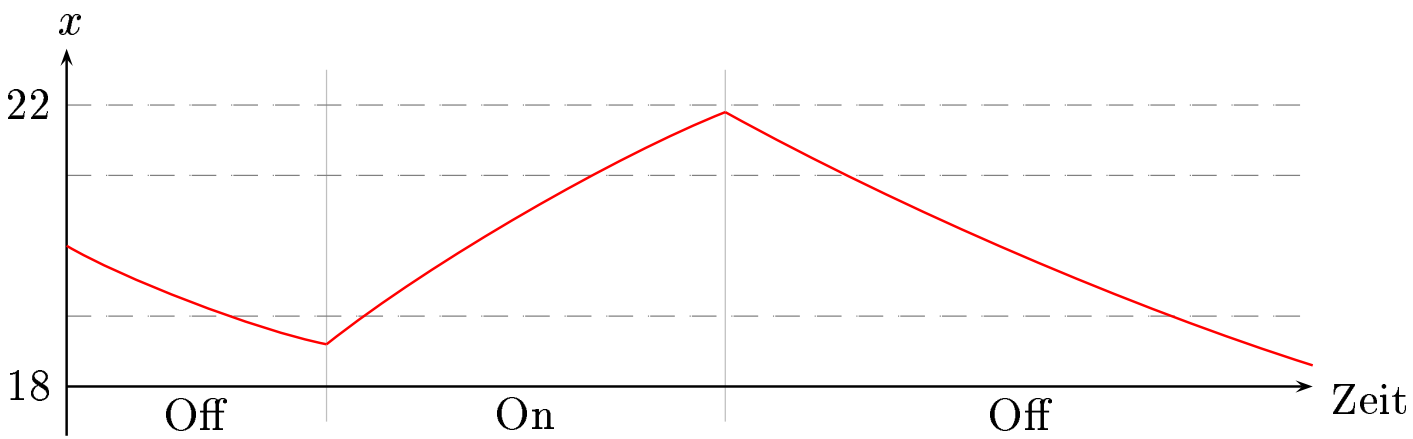


Hybride Automaten

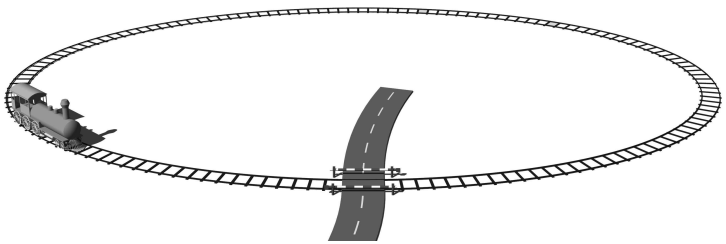
Beispiel

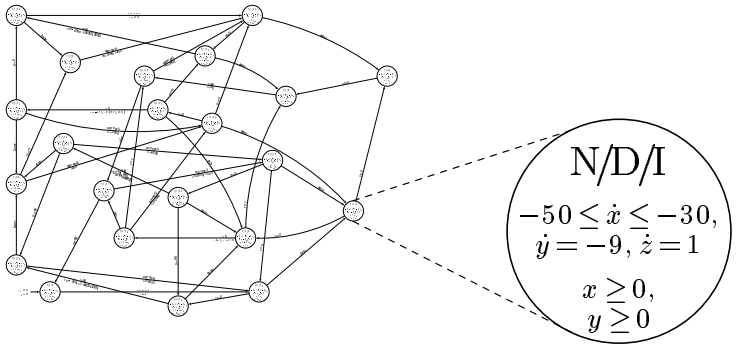
Heizung:





Beispiel





Kontrollsystem für einen beschrankten Bahnübergang

- Bahnschranken müssen geschlossen sein, sobald der Zug den Bahnübergang passiert
- Steuerung des Schrankensystems durch einen Computer
- Gleichungen beschreiben Schranke und Zug

Hybride Automaten

Vor-/Nachteile

- leicht zu konstruieren und zu verstehen
- wachsen schnell und werden unlesbar
  (Zustandsexplosion)
- Verifikation von Eigenschaften oft kompliziert
  (z.B. Sicherheit und Lebendigkeit nur für eine Unterklasse)
- wenig Softwareunterstützung
  spezielle Entwicklung der Software
- enorme Rechenleistung wird benötigt

Auf dem Weg zur Algebra

enge Verbindung zwischen endlichen Automaten und regulären Ausdrücken

Gibt es einen algebraischen Ansatz für hybride Systeme?

Auf dem Weg zur Algebra

enge Verbindung zwischen endlichen Automaten und regulären Ausdrücken

Gibt es einen algebraischen Ansatz für hybride Systeme?

Fragen

- was sind die Elemente
- wie beschreibt man kontinuierliches und diskretes Verhalten
- wie beschreibt man unendliche Abläufe
- wie verknüpft man Elemente

Auf dem Weg zur Algebra

Antworten

- Grundobjekte: Trajektorien
- Kontinuität wird durch Fluß-Gleichungen beschrieben
- Diskrete Zustandswechsel können Sprungstellen in der Funktion sein
- Algebra basiert auf Mengen von Trajektorien

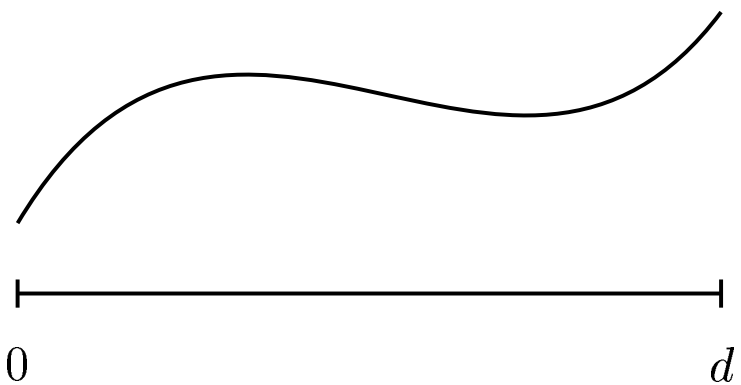
Trajektorien

Definition

Eine Trajektorie t ist ein Paar (d, g), wobei $d \in D$ die Zeitdauer und

$$g : [0, d] \to V \text{ oder } g : [0, \infty) \to V$$

das Bild von $[0, d]$ ($[0, \infty)$) unter g ist

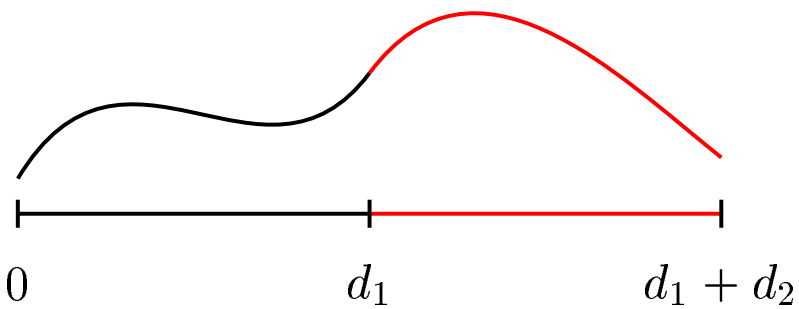


D muss wenige Bedingungen erfüllen

Komposition von Trajektorien

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

wobei $g(x) = g_1(x)$ für alle $x \in [0, d_1]$ und $g(x + d_1) = g_2(x)$ für $x \in [0, d_2]$ oder $x \in [0, \infty)$



Algebra für hybride Systeme

das algebraische Modell für reguläre Ausdrücke ist Kleene-Algebra

Definition

eine Kleene-Algebra ist ein Sextupel $(K, +, 0, \cdot, 1, ^*)$ mit

- $(K, +, 0)$ ist idempotentes kommutatives Monoid
- $(K, \cdot, 1)$ ist Monoid
- Multiplikation ist distributiv
- 0 ist ein Annihilator, $0 \cdot a = 0 = a \cdot 0$
- $*$ erfüllt Entfaltungs- und Induktionsgesetze

$+$	$\longleftrightarrow$	Auswahl
$\cdot$	$\longleftrightarrow$	sequentielle Komposition
*	$\longleftrightarrow$	endliche Iteration
0	$\longleftrightarrow$	Abbruch
1	$\longleftrightarrow$	skip

Algebra für hybride Systeme

- Auswahl zwischen Trajektorien ist modelliert durch Vereinigung von Mengen von Trajektorien
- die leere Menge ist neutral bzgl. Vereinigung
- Komposition von Trajektorien wird auf Mengen fortgesetzt

$$A \cdot B \;=_{df}\; \inf A \;\cup\; \{a \cdot b \,|\, a \in \text{fin}\, A, b \in B\}$$

wobei $\text{fin}\, A = \{(d, g) \in A \,|\, d < \infty\}$ und $\inf A = \{(d, g) \in A \,|\, d = \infty\}$

- die Menge $\mathbb{1}$ aller Trajektorien mit Zeitdauer 0 ist neutrales Element bzgl. Komposition

Struktur der Algebra $(\mathcal{P}(\mathrm{TRA}), \cup, \emptyset, \cdot, \mathbb{1}, ^*)$ ist beinahe eine Kleene-Algebra (TRA ist die Menge aller Trajektorien)

Algebra für hybride Systeme

- Auswahl zwischen Trajektorien ist modelliert durch Vereinigung von Mengen von Trajektorien
- die leere Menge ist neutral bzgl. Vereinigung
- Komposition von Trajektorien wird auf Mengen fortgesetzt

$$A \cdot B \;=_{df}\; \inf A \;\cup\; \{a \cdot b \,|\, a \in \text{fin}\, A, b \in B\}$$

wobei $\text{fin}\, A = \{(d, g) \in A \,|\, d < \infty\}$ und $\inf A = \{(d, g) \in A \,|\, d = \infty\}$

- die Menge $\mathbb{1}$ aller Trajektorien mit Zeitdauer 0 ist neutrales Element bzgl. Komposition

Struktur der Algebra $(\mathcal{P}(\mathrm{TRA}), \cup, \emptyset, \cdot, \mathbb{1}, ^*)$ ist beinahe eine Kleene-Algebra (TRA ist die Menge aller Trajektorien)

ABER

$$A \cdot \emptyset \neq \emptyset$$

Schwache Kleene-Algebra

Definition

eine schwache Kleene-Algebra ist eine Kleene-Algebra bei der 0 nur ein Links-Annihilator ist $(0 \cdot a = 0)$

Bemerkung

- Relaxation erlaubt unendliche Elemente

$$\inf a = a \cdot 0 \qquad \text{fin}\, a = a - \inf a$$

- schwache Kleene-Algebra verhält sich ähnlich zur Kleene-Algebra
- unendliche Iteration ist möglich
- oft kann auch die Rechts-Distributivität weggelassen werden
- Testelemente erlauben die Charakterisierung von Zusicherungen

Bemerkungen

- ähnlich zu Funktionenräumen der Linearen Algebra
- im Fall $D = \{0, 1\}$ ist die Algebra äquivalent zu Relationen
- Sprünge an Kompositionspunkten ebenfalls möglich
- Einschränkung der Komposition

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

  die zweite Trajektorie wird garantiert erreicht
- Erweiterung um Tests und Vorbereichsfunktion möglich

Sicherheit und Lebendigkeit

Sicherheit: "something bad will never happen" [Lamport77]

- konservativ (Vermeidung von "schlechten" Zuständen)
- z.B. "mache gar nichts"
- irgendetwas ist für immer wahr

Lebendigkeit: "something good will eventually happen" [Lamport77]

- progressiv ("gute" Zustände werden erreicht)
- z.B. das System wird nicht abbrechen

Algebraische Sicherheit und Lebendigkeit

Operatoren die den Wertebereich einschränken

- P wird erreicht

$$\Diamond P =_{df} \mathsf{F} \cdot P \cdot \top$$

  Menge aller Trajektorien, die Werte aus P an einem Punkt erfüllen

- P wird garantiert

$$\Box P =_{df} \overline{\Diamond \neg P}$$

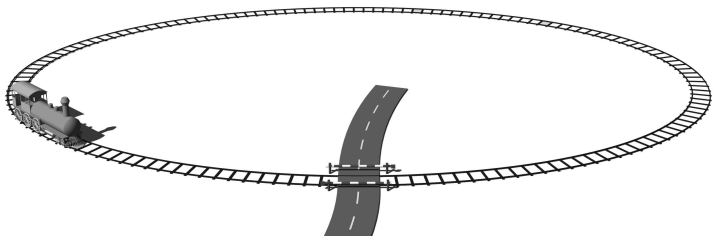
  Menge aller Trajektorien, die komplett in Werten aus P liegen

$\top$: Menge *aller* Trajektorien;
F: Menge *aller endlichen* Trajektorien;
P: eine Menge mit Trajektorien der Länge 0 (ein Test)

Grundlegende Eigenschaften

- $\Box P \sqcap A \cdot B = (\Box P \sqcap A) \cdot (\Box P \sqcap B)$
- $\Diamond P \sqcap A \cdot B = (\Diamond P \sqcap A) \cdot B + \mathsf{fin}\, A \cdot (\Diamond P \sqcap B)$
- $(\Box P) \cdot (\Box P) = \Box P$

Beispiel

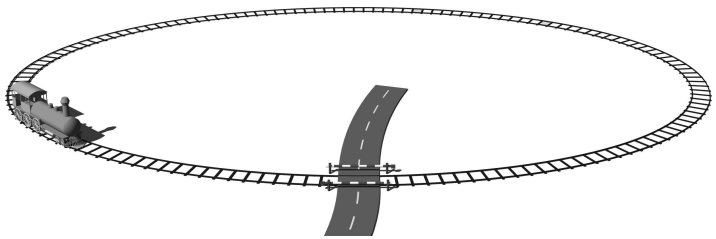


$$\big(O||(P_1 \cdot T_1 \cdot P_2)\big) \cdot \Big(\big((M_1 \cdot C)||(T_2 \cdot P_3)\big) \cdot \big(C||(T_2 \cdot P_4)_{\asymp}\big) \cdot \big((M_2 \cdot O)||(T_1 \cdot P_2)\big)\Big)^{\omega}$$

Vorteile

- kompakte Beschreibung
- Verwendung von Standard-Computer-Algebrasystemen
- Gleichungsregeln zur Überprüfung von Sicherheitsaspekten
- Modellierung von Lebendigkeit und Invarianten

Beispiel



$$\big(O||(P_1 \cdot T_1 \cdot P_2)\big) \cdot \Big(\big((M_1 \cdot C)||(T_2 \cdot P_3)\big) \cdot \big(C||(T_2 \cdot P_4)_{\asymp}\big) \cdot \big((M_2 \cdot O)||(T_1 \cdot P_2)\big)\Big)^{\omega}$$

Vorteile

- kompakte Beschreibung
- Verwendung von Standard-Computer-Algebrasystemen
- Gleichungsregeln zur Überprüfung von Sicherheitsaspekten
- Modellierung von Lebendigkeit und Invarianten

Wichtiges algebraisches Konstrukt

$$P_1 \cdot T \cdot P_2$$

Vor- und Nachbedingungen

- erfüllt das hybride System die Vorbedingungen von P_1 vor der Ausführung von T, dann gilt danach P_2
- eng verwandt mit "Wenn-dann-Konstrukten" (Logik)

Algebraische Logik

Logiken können auch in den algebraischen Rahmen eingebettet werden

- Hoare-Logik
- Lineare Temporallogik LTL
- Baumlogiken CTL and CTL*
- Nachbarschaftslogik von Zhou und Hansen

Vorteile

- einheitliche mathematische Basis
- Wissenstransfer
- Standardterminologie
- Computerunterstützung

Zusammenfassung

- Entwurf einer Algebra für hybride Systeme
- grundlegende Eigenschaften bewiesen
- Einbettung/Definition von modalen Operatoren in die Algebra, einschließlich derer von Sintzoff und von Karger
- kompakte und einheitliche Behandlung von diversen Logiken
- erste Fallstudien
- Computerunterstützung zur Verifikation

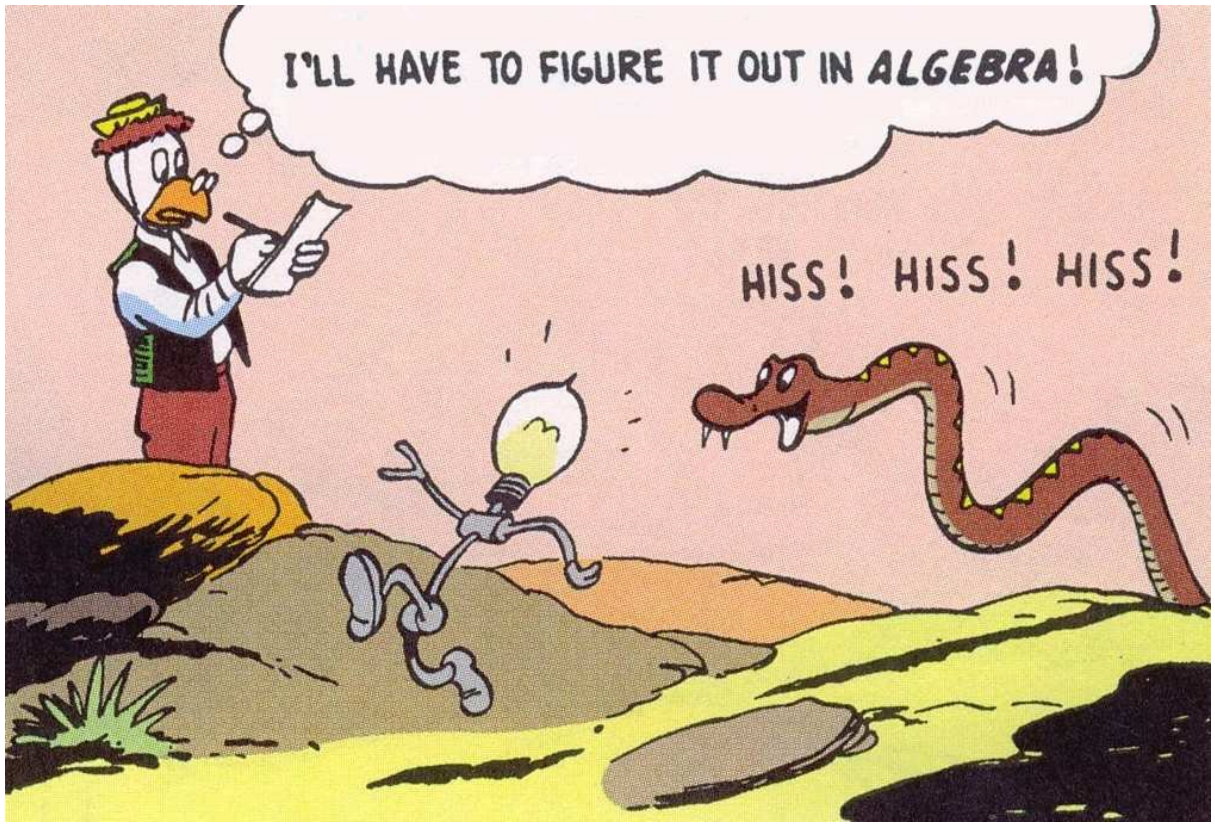
kohärente Familie algebraischer Kalküle für hybride Systeme

Vor-/Nachteile

- einheitliche Grundlage
- algebraische Strukturen sind wohl verstanden
- Algebra ermöglicht "einfache" Rechnungen/Beweise
- oft Domänen-spezifisches Wissen nötig
- oft Formeln "unverständlich"
  (insbesondere für Nicht-Mathematiker/Nicht-Informatiker)
- Softwareunterstützung mittels Standard-Software

Ausblick

- Kombination von Theorembeweisern mit Algebra-Systemen (Waldmeister ist bereits in Mathematica integriert)
- reale Fallstudien
- Integration von Spieltheorie



If you are faced by a difficulty or a controversy in science,
an ounce of algebra is worth a ton of verbal argument.

J.B.S. Haldane