An Algebra of Hybrid Systems

Peter Höfner

University of Augsburg

August 22, 2008

Hybrid Systems

Definition

hybrid systems are heterogeneous systems characterised by the interaction of *discrete* and *continuous* dynamics

Applications

- (air-)traffic controls / traffic management
- chemical and biological processes
- automated manufacturing
- ...

Kinds of Systems

Transformational Systems
determine a function

Reactive Systems
interact with environment

Real–Time Systems
have to produce results within a certain amount of time

Hybrid Systems
discrete and continuous dynamics

source: University of Oldenburg

Kinds of Systems

Transformational Systems
determine a function

Reactive Systems
interact with environment

Real–Time Systems
have to produce results within a certain amount of time

Hybrid Systems
discrete and continuous dynamics

source: University of Oldenburg

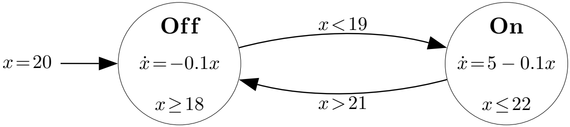
less than 1% of all processors are in PCs;
more than 98% are controllers of hybrid systems

Hybrid Automata

- most common representation type for hybrid systems
- widely popular for designing and modelling
- similar to finite state machines
- states describe continuous dynamics
- edges describe discrete behaviours

Example

Gas Burner:



Hybrid Automata

(Dis-)Advantages

- easy to construct/understand
- growing fast and becoming unreadable
- nearly impossible to check liveness or safety
  (only done partly for a small class of hybrid systems)
- nearly no software-tools available

Hybrid Automata

(Dis-)Advantages

- easy to construct/understand
- growing fast and becoming unreadable
- nearly impossible to check liveness or safety
  (only done partly for a small class of hybrid systems)
- nearly no software-tools available

Question/Idea

is there a relation to an algebra like the relationship between finite statemachines, regular languages and Kleene algebra

Towards an Algebra of Hybrid Systems

Questions

- what are possible elements
- how to describe discrete and continuous behaviour
- how to describe infinity
  (interaction on an on-going, nearly never-ending basis)
- how to compose elements
- how to choose between elements

Towards an Algebra of Hybrid Systems

Questions

- what are possible elements
- how to describe discrete and continuous behaviour
- how to describe infinity
  (interaction on an on-going, nearly never-ending basis)
- how to compose elements
- how to choose between elements

Possible Answers

- elements are trajectories
- continuous behaviour is described by the flow functions
- discrete behaviours are e.g. jumps in the function
- algebra is based on sets of trajectories
- weak Kleene algebra allows modelling infinite elements

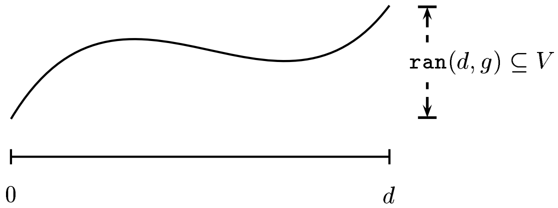
Trajectories

Definition

a trajectory t is a pair (d, g), where $d \in D$ is the duration and

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

the image of $[0, d]$ ($[0, \infty)$) under g is its range $\mathtt{ran}(d, g)$

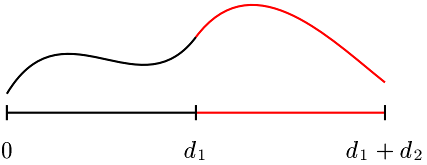


D has to fulfil some properties

Composition of Trajectories

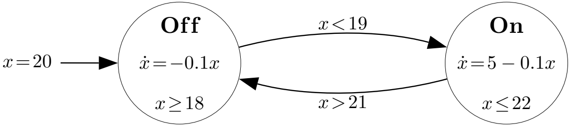
$$(d_1, g_1) \cdot (d_2, g_2) =_{df} \begin{cases} (d_1 + d_2, g) & \text{if } d_1 \neq \infty \wedge g_1(d_1) = g_2(0) \\ (d_1, g_1) & \text{if } d_1 = \infty \\ \text{undefined} & \text{otherwise} \end{cases}$$

with $g(x) = g_1(x)$ for all $x \in [0, d_1]$ and $g(x + d_1) = g_2(x)$ for all $x \in [0, d_2]$ or $x \in [0, \infty)$

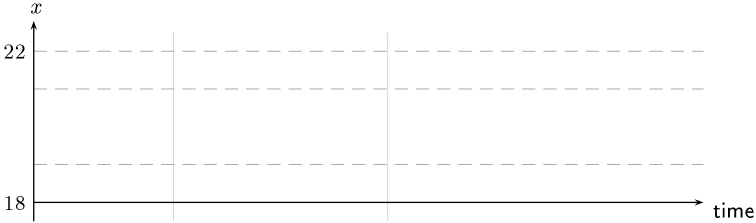


Connection to Hybrid Automata

a trajectory can model a run of a hybrid automaton

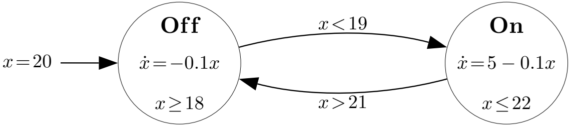


trajectory

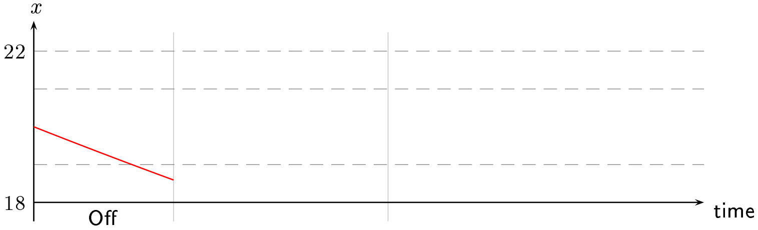


Connection to Hybrid Automata

a trajectory can model a run of a hybrid automaton

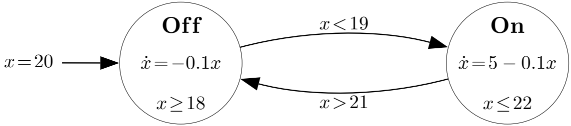


trajectory

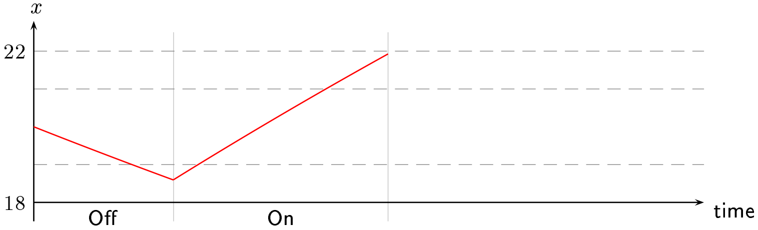


Connection to Hybrid Automata

a trajectory can model a run of a hybrid automaton

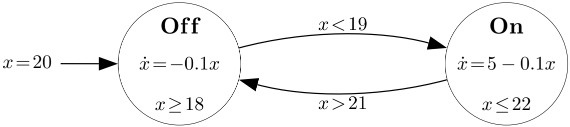


trajectory

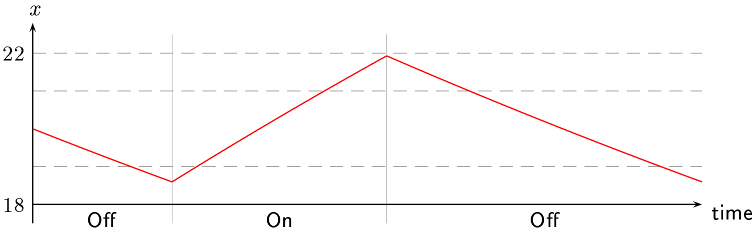


Connection to Hybrid Automata

a trajectory can model a run of a hybrid automaton



trajectory



Getting Algebraic

the algebraic model of regular events is Kleene algebra

Definition
a Kleene algebra is a tuple $(K, +, 0, \cdot, 1, ^*)$ with

- $(K, +, 0)$ idempotent commutative monoid
- $(K, \cdot, 1)$ monoid
- multiplication is distributive
- 0 is an annihilator, $0 \cdot a = 0 = a \cdot 0$
- $*$ satisfies unfold and induction axioms

$$
\begin{array}{ll}
+ & \leftrightarrow \text{ choice} \\
\cdot & \leftrightarrow \text{ sequential composition} \\
* & \leftrightarrow \text{ finite iteration} \\
0 & \leftrightarrow \text{ abort} \\
1 & \leftrightarrow \text{ skip}
\end{array}
$$

Choice, Composition and Neutral Elements

- choice between trajectories is realised by set union over sets of trajectories (also called processes)
- the empty set is neutral element
- composition is lifted pointwise to processes

$$A \cdot B =_{df} \{a \cdot b \,|\, a \in A, b \in B\}$$

- the set of all trajectories with duration 0 (denoted by $\mathbb{1}$) is the neutral element

the algebra of hybrid systems $(\mathcal{P}(\mathrm{TRA}), \cup, \emptyset, \cdot, \mathbb{1}, ^*)$ is nearly a Kleene algebra (TRA is the set of all trajectories)

Choice, Composition and Neutral Elements

- choice between trajectories is realised by set union over sets of trajectories (also called processes)
- the empty set is neutral element
- composition is lifted pointwise to processes

$$A \cdot B =_{df} \{a \cdot b \mid a \in A, b \in B\}$$

- the set of all trajectories with duration 0 (denoted by $\mathbb{1}$) is the neutral element

the algebra of hybrid systems $(\mathcal{P}(\mathrm{TRA}), \cup, \emptyset, \cdot, \mathbb{1}, ^*)$ is nearly a Kleene algebra (TRA is the set of all trajectories)

But

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

Weak Kleene Algebra

Definition

a weak Kleene algebra is a Kleene algebra where 0 is only left annihilator $(0 \cdot a = 0)$

Remark

- relaxation allows to have infinite elements [Möller04]

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

- weak Kleene algebra behaves nearly like Kleene algebra
- adding infinite iteration yields weak omega algebra [Cohen00]
- adding tests to model assertions and guards [Kozen97]
- adding domain/codomain [DesharnaisMöllerStruth03/Möller04]
- in some situations one even needs no right-distributivity law
- weak Kleene algebra generalises predicate transformers [vonWright02,Meinicke08]

Remarks on the Algebra of Hybrid Systems

- similarities to function spaces (linear algebra)
- if $D = \{0, 1\}$ then the algebra of hybrid systems is equivalent to relations
- jumps at composition points possible
- restricted form of composition

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

  the second trajectory is reached
- can be endowed with tests and domain functions.

Safety and Liveness

Safety: “something bad will never happen” [Lamport77]

- conservative in the sense of avoiding bad states
- e.g. do nothing
- something is true forever

Liveness: “something good will eventually happen” [Lamport77]

- progressive in the sense of reaching good states
  or the system will never stop

Algebraic Safety and Liveness

Examples for Range-Restriction Operators

- P will be reached

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

  set of all trajectories, where the range is within P at some point

- P is guaranteed

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

  set of all trajectories, where the range is complete in P
  needs complementation on underlying structure

$\top$ is the set of *all* trajectories;
F is the set of *all finite* trajectories;
P is a set of trajectories *without* duration.

Basic Properties

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

Overview of our work

- **What we have done**
- **What we do**
- **What we will do**

What we have done

- build an algebra of hybrid systems
- show basic properties
- describe and use the Duration Calculus [RavnHoareZhou91] in an algebraic setting
- characterise different useful modal operators in the setting, including the one of vonKarger, Sintzoff, ...
- use of theorem provers (Prover9) [HöfnerStruth07, Höfner08]

(Dis)Advantages of What we have done

- create a “uniform” basis
- algebraic structures like Kleene algebra are well known
- algebra allows easy calculations
- but sometimes domain-knowledge is needed
- not easy to understand (especially for non-computer scientists)
- aid of computer is feasible

What we do

- adapt logics to hybrid systems
  there are algebraic versions of
  - Hoare Logic [Kozen97,MöllerStruth06]
  - LTL [DesharnaisMöllerStruth04]
  - CTL and CTL* [MöllerHöfnerStruth06]
  - Neighbourhood Logic of Zhou and Hansen [Höfner06]

  due to the algebraic versions which also use Kleene algebra, it should be possible
- there are notions of dynamic systems in relation algebra [ScolloFrancoManca06]
  can this be adopted/generalised to our framework?
- handle Zeno Effects

(Dis)Advantages of What we do

- knowlegde transfer
  in CTL, CTL* there are notions of liveness, safety, ...
- use of standard terminology
- support of computers

What we will do

- handle Zeno Effects
- bring game theory into play
  (first steps done in [Sintzoff04])
- add probabilistic [MeinickeHayes08]

Thank you

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