Modal Semirings and Kleene Algebras

Georg Struth

University of Sheffield

based on joint work with J Desharnais, P Jipsen, B Möller and others

Motivation

task: to give survey talk on modal semirings and Kleene algebras

disclaimer: present idealised subjective view

- which maths/computing questions motivated us
- which persons/papers influenced us

domain:

- very natural concept
- has been around in many variants in many contexts

Starting Point

DFG project: to develop unified semantics for computing systems

approaches:

- action based: relation algebras, dioids, Kleene algebras,
  quantales, regular algebras, process algebras, refinement calculus, . . .
- proposition based: modal/temporal/dynamic logics/algebras,
  Hoare logic, w(l)p semantics, domain theory (?), . . .

idea: combine two worlds

- focus on Kleene algebras with tests vs dynamic algebras
- use axiomatisation of domain operation as "missing link"

  Kleene algebras $\Rightarrow$ Kleene algebras with domain $\Rightarrow$ modal Kleene algebras

Influences and Aims

influences:

- Kleene algebras: Conway, Kozen, Backhouse
- modal algebras: Pratt, Kozen, Parikh, Németi, Jónsson/Tarski, von Karger
- relational semantics: Berghammer/Zierer, Maddux, Manes, Freyd/Scedrov
- side tracks: Schein, Cockett, Fiore, Hollenberg

aims:

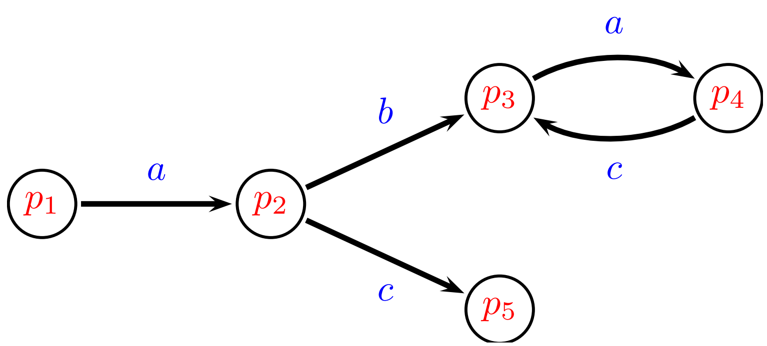
- simple/minimal algebraic structures
- (quasi)equational axioms
- suitable for automated theorem proving

Overview

outline: this survey talk

1. from semirings to modal Kleene algebras
2. connections with logics/semantics of programs
3. program/termination analysis
4. free algebras and representability
5. domain semigroups
6. research questions

Transition System



linear system [Conway, Salomaa] which algebra?

$$
\begin{aligned}
x_1 &= a x_2 \\
x_2 &= b x_3 + c x_5 \\
x_3 &= a x_4 \\
x_4 &= c x_3
\end{aligned}
\qquad
\begin{pmatrix}
0 & a & 0 & 0 & 0 \\
0 & 0 & b & 0 & c \\
0 & 0 & 0 & a & 0 \\
0 & 0 & c & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

solution: regular expression $a(b(ac)^* + c)$ (if p_3 and p_5 final states)

Dioids, Actions and Propositions

semiring: $(S, +, \cdot, 0, 1)$ "ring without minus"

$$x + (y + z) = (x + y) + z \qquad x + y = y + x \qquad x + 0 = x$$

$$x(yz) = (xy)z \qquad x1 = x \qquad 1x = x$$

$$x(y + z) = xy + xz \qquad (x + y)z = xz + yz$$

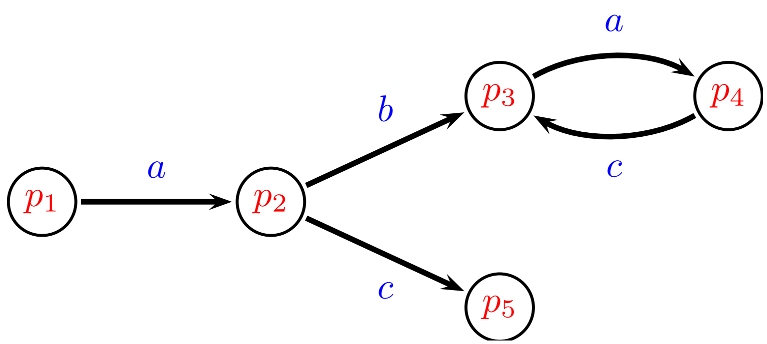
$$x0 = 0 \qquad 0x = 0$$

dioid: (idempotent semiring) $\quad x + x = x$

remarks:

- swapping multiplication yields opposite semiring
- idempotent semirings have natural order $\quad x \leq y \;\Leftrightarrow\; x + y = y$

Dioids, Actions and Propositions



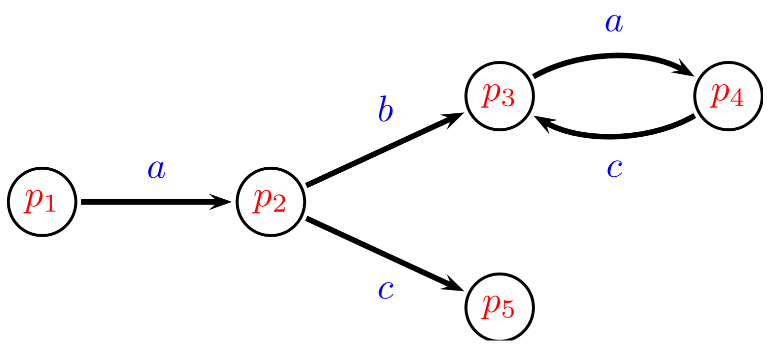
intuition: dioid terms represent action sequences of transition system

$$ab, \quad ac, \quad a(b+c), \quad ab+ac, \quad abac, \quad ab(ac+acac), \quad \dots$$

- $+$ models nondeterministic (angelic) choice
- $\cdot$ models sequential composition
- 0 models abortive action
- 1 models ineffective action

free dioids: isomorphic to sets of words (formal languages)

Dioids, Actions and Propositions



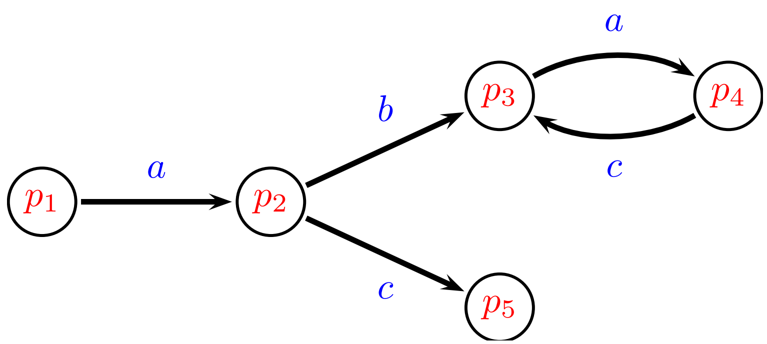
question: what about trace $p_1 a p_2 b p_3 a p_4 c p_3$?

test semiring: [Manes/Arbib] $(S, \text{test}(S), +, \cdot, \neg, 0, 1)$

- Boolean subalgebra $(\text{test}(S), +, \cdot, \neg, 0, 1)$ embedded into $[0, 1]$ of S
- $+/\cdot$ coincide with Boolean join/meet
- $\text{test}(S)$ models state space (sets of states), propositions or tests of program

free test semirings: isomorphic to sets of "guarded strings"

Kleene Algebras



question: what about loop $acacac\ldots$?

Kleene algebra: [Conway, Kozen] dioid with star satisfying

- unfold axiom $1 + xx^* = x^*$
- induction axiom $y + xz = z \Rightarrow x^*y \leq z$
- and their opposites

remark: x^* modelled as least fixpoint

Kleene Algebras

free KAs: isomorphic to regular languages [Salomaa, Conway, Kozen]

- KAs are algebras of "regular events"
- equational theory is decidable by automata! (PSPACE-complete)
- quasiequational theory is undecidable (uniform word problem for semigroups)
- variety not finitely (equationally) axiomatisable [Redko, Salomaa, Conway]

question: axiomatise quasivariety of regular expressions?

1. $x^2 = 1 \Rightarrow x = 1$ holds in regular languages . . .
2. . . . but not for relation $x = \{(0,1),(1,0)\}$
3. relations form KAs (see below)
4. hence KA doesn't work!

Kleene Algebras with Tests

definition: test semiring + star axioms

algebraic semantics of while programs (without assignment):

$\dots$ if p then x else $y = px + \neg py$ while p do $x = (px)^*\neg p$

free KATs: isomorphic to regular languages over guarded strings [Kozen]

- equational theory decidable (PSPACE-complete)
- guarded string models have isomorphic relational models
  1. Cayley map $h : 2^G \to 2^{G\times G}$, $h(L) = \{(a, ab) : a \in G, b \in L\}$ is injective homomorphism
  2. relations form KATs (see below)

Models of Kleene Algebra

trace: alternating sequence $p_0 a_0 p_1 a_1 p_2 \ldots p_{n-2} a_{n-1} p_{n-1}$, $p_i \in P$, $a_i \in A$

trace product: $\sigma.p \cdot p.\sigma' = \sigma.p.\sigma'$ $\sigma.p \cdot q.\sigma'$ undefined

fact: power-set algebra $2^{(P,A)^*}$ forms (full trace) KA

$$T_0 + T_1 = T_0 \cup T_1$$

$$T_0 \cdot T_1 = \{\tau_0 \cdot \tau_1 : \tau_0 \in T_0, \tau_1 \in T_1 \text{ and } \tau_0 \cdot \tau_1 \text{ defined}\}$$

$$T^* = \{\tau_0 \cdot \tau_1 \cdot \cdots \cdot \tau_n : n \geq 0, \tau_i \in T \text{ and prods defined}\}$$

$$0 = \emptyset$$

$$1 = P$$

trace Kleene algebras: subalgebras of full trace KA

Models of Kleene Algebra

special cases: forget structure in traces

- path/language KAs forget actions/propositions
- relation KAs forget sequences between endpoints

property: (equational) properties inherited by (relations), paths, languages

further models: matrices over KAs [Conway, Kozen]

models for KAT: tests are subsets of P/subidentities

Modelling Example: Kleene Algebra and Induction

Church-Rosser theorem: $y^*x^* \leq x^*y^* \Rightarrow (x+y)^* \leq x^*y^*$

proof: induction on number of peaks

$$
\begin{aligned}
(x+y)^* \leq x^*y^* &\Leftrightarrow (y^*x^*)^* \leq x^*y^* && \text{(regular identity)}\\
&\Leftarrow 1 + y^*x^*x^*y^* \leq x^*y^* && \text{(induction)}\\
&\Leftrightarrow 1 \leq x^*y^* \ \wedge \ y^*x^*x^*y^* \leq x^*y^* && \text{(lub)}
\end{aligned}
$$

- base case: $1 \leq x^*y^*$ trivial
- induction step: $y^*x^*x^*y^* = y^*x^*y^* \leq x^*y^*y^* = x^*y^*$

remark: separation theorem for concurrency control

Adding Modalities

motivation:

- many applications require different approach to actions/propositions
- systems dynamics by state transitions; mappings between sets of states
- various logics "use" KAs, but what is precise connection?

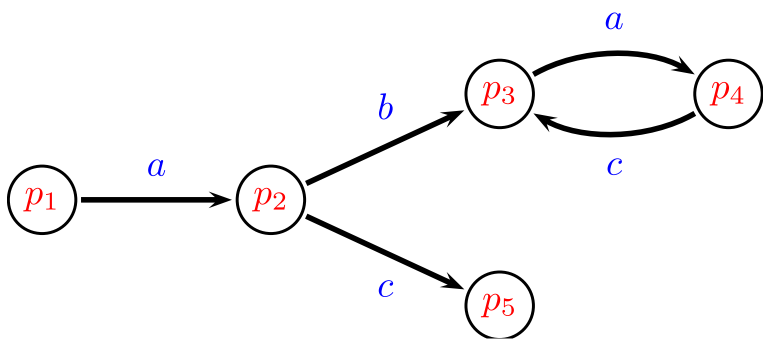
idea: modal approach

- actions/propositions via Kripke frames
- modal operators via preimages/images $|x\rangle p$ / $\langle x|p$
- preimages/images via axioms for domain/codomain

concretely: find equational axioms for domain that

- entail some "natural" properties
- induce "appropriate" state spaces

Properties of Domain



domain concretely: $d(x)$ models states where action x is enabled

- transition systems: $d(a) = \{p : p \xrightarrow{a} q\}$
- relation semirings: $d(R) = \{a : (a, b) \in R\} = R \cdot U \sqcap 1$
- trace semirings: $d(T) = \{p : p = \text{first}(\tau) \text{ and } \tau \in T\}$

domain abstractly: $d(x)$ is least left preserver of x

- so $x = d(x)x$ and even $x \leq px \Leftrightarrow d(x) \leq p$

Domain Semirings

domain semiring: semiring with mapping $d : S \to S$ that satisfies

$$x + d(x)x = d(x)x \qquad d(xy) = d(xd(y)) \qquad d(x+y) = d(x) + d(y)$$

$$d(x) + 1 = 1 \qquad d(0) = 0$$

intuition:

1. domain is left preserver
2. $d(xy)$ is local in y through its domain
3. enabling a choice means enabling one action or the other
4. domain elements are below 1 (see below)
5. abortive action is never enabled

property: d-semirings are automatically idempotent

Domain Semirings

remark: development strongly based on ATP/model search

properties: axioms

- are irredundant (use model generator)
- imply least left preservation (ATP), even $d(x) = \inf(p \in d(S) : x = px)$
- llp $x \leq px \Leftrightarrow d(x) \leq p$ is "almost" Galois connection

domain elements: $d(x) = x$ says "x is domain element"

fixpoint lemma: $x \in f(A) \;\Leftrightarrow\; f(x) = x$ holds for projection $f : A \to A$

Further Natural Properties

fact: let S d-semiring, let $x, y \in S$ and let $p \in d(S)$. then

- $d(x)x = x$ (domain is a left invariant)
- $d(p) = p$ (domain is a projection)
- $d(xy) \leq d(x)$ (domain increases for prefixes)
- $x \leq 1 \Rightarrow x \leq d(x)$ (domain expands subidentities)
- $d(x) = 0 \Leftrightarrow x = 0$ (domain is very strict)
- $d(1) = 1$ (domain is co-strict)
- $x \leq y \Rightarrow d(x) \leq d(y)$ (domain is isotone)
- $d(px) = pd(x)$ (domain elements can be exported)
- $d(x)d(x) = d(x)$ (domain elements are multiplicatively idempotent)
- $d(x)d(y) = d(y)d(x)$ (domain elements commute)
- $xy = 0 \Leftrightarrow xd(y) = 0$ (domain is weakly local)

Domain Algebras

question: how can we relate domain elements with tests/state space?

property: $(d(S), +, \cdot, 0, 1)$ is bounded distributive lattice

1. check closure properties (fixpoint lemma), $d(1) = 1$ and $d(0) = 0$
2. this gives sub-semiring
3. $d(x) \leq 1$ is axiom and $d(x)d(x) = d(x)$
4. semirings satisfying these two properties are DLs [Birkhoff]

notation:

- $(d(S), +, \cdot, 0, 1)$ is called domain algebra of S
- $p, q, r \ldots$ for domain elements

Extension to Domain Semirings

proposition: some semirings cannot be extended to d-semirings

proof: consider $d(2)$ in (idempotent) semiring

$+$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	0

1. $d(2) \neq 0$ since $d(x) = 0 \Leftrightarrow x = 0$
2. $d(2) \neq 1$ since otherwise $1 = d(2 \cdot d(2)) = d(2 \cdot 2) = d(0) = 0$
3. $d(2) \neq 2$ since otherwise $2 = d(2) \cdot 2 = 2 \cdot 2 = 0$

Richer Domain Algebras

remark:

- domain algebras need not be BAs (ex. 3-element S with $d(S)$ a chain)
- but $d(S)$ must contain maximal BA in $[0, 1]$
  ($x, y \in S$ with $x + y = 1$, $xy = 0 = yx$ form BA and $d(x) = x$, $d(y) = y$)

enrichments of domain algebras

1. Heyting algebra: add Galois connection (and closure condition for $\to$)

$$pq \leq r \Leftrightarrow p \leq q \to r$$

2. Boolean algebra: add antidomain operation $a : S \to S$ with axioms

$$d(x) + a(x) = 1 \qquad d(x)a(x) = 0$$

Antidomain Semirings

fact: Boolean case has very compact axiomatisation

antidomain semiring: semiring S with mapping $a : S \to S$ that satisfies

$$a(x)x = 0 \qquad a(xy) \leq a(xa^2(y)) \qquad a^2(x) + a(x) = 1$$

remarks:

- domain definable via $d = a^2$ (Boolean complement)
- $d(S)$ induced is maximal BA in $[0, 1]$
- simple axioms induce rich modal calculus. . .

Modal Semirings

idea: define forward/backward diamonds as preimages/images

$$|x\rangle p = d(xp) \qquad\qquad \langle x|p = d^{\circ}(px)$$

where codomain operation d° satisfies dual domain axioms

consequence: very general way of defining modal logics

- we have $|x\rangle 0 = 0$ and $|x\rangle(p+q) = |x\rangle p + |x\rangle q$
- this yields DLs/HAs/BAs with operators

convention: Kleene algebras with antidomain are called modal Kleene algebras (MKAs)

Modalities, Symmetries, Dualities for Boolean Domain

demodalisation: $|x\rangle p \le q \;\Leftrightarrow\; \neg q x p \le 0 \qquad \langle x| p \le q \;\Leftrightarrow\; p x \neg q \le 0$

dualities:

- de Morgan: $|x]p = \neg|x\rangle\neg p \qquad [x|p = \neg\langle x|\neg p$
- opposition: $\langle x|$, $[x| \Leftrightarrow |x\rangle$, $|x]$

symmetries:

- conjugation: $(|x\rangle p)q = 0 \;\Leftrightarrow\; p(\langle x|q) = 0$
- Galois connection: $|x\rangle p \le q \;\Leftrightarrow\; p \le [x|q$

benefits: rich calculus (automatically verified in Isabelle)

- symmetries as theorem generators
- dualities as theorem transformers

Models

trace: $p_0 a_0 p_1 a_1 p_2 \ldots p_{n-2} a_{n-1} p_{n-1}$, $p_i \in P$, $a_i \in A$

fact: power-set algebra $2^{(P,A)^*}$ forms (full trace) MKA where

$$|T\rangle Q = \{p : p.\sigma.q \in T \text{ and } q \in Q\}$$

trace MKAs: complete subalgebras of full trace MKA

fact: path, language, relation MKAs can again be obtained by forgetting

remark: in relation MKAs, sets are subidentities

Kleene Modules

Kleene module: [Leiß] structure $(K, L, :)$ with

$$(x + y)p = xp + yp \qquad x(p + q) = xp + xq \qquad (xy)p = x(yp)$$
$$1p = p \qquad x0 = 0 \qquad xp + q \leq p \Rightarrow x^*q \leq p$$

remark: scalar product $:$ omitted

fact: MKAs are Kleene modules with $: = \lambda x \lambda p . |x\rangle p$

consequence: close relationship with computational logics

MKA and PDL

fact: MKAs are dynamic/test algebras

proof: (main task) show equivalence of

- module induction law $|x\rangle p + q \leq p \Rightarrow |x^*\rangle q \leq p$
- Segerberg axiom $|x^*\rangle p - p \leq |x^*\rangle(|x\rangle p - p)$

corollary: extensional MKAs are essentially propositional dynamic logics

- extensionality: $(\forall p. |x\rangle p = |y\rangle p) \Rightarrow x = y$

benefits: MKAs offer

- simpler/more modular axioms
- richer model class (beyond Kripke frames)
- more flexible setting, ATP support

MKA and LTL

encoding:

- temporal operators (use one single action x)

$$Xp = |x\rangle p \qquad Fp = |x^*\rangle p \qquad Gp = |x^*]p \qquad pUq = |(px)^*\rangle q$$

- initial state $\mathsf{init}_\mathsf{x} = [x|0$ "there's nothing before the beginning"
- validity of temporal implications $\sigma \models p \to q \Leftrightarrow \mathsf{init}_\mathsf{x} p = q$

MKA and LTL

LTL axioms: von Karger's variant of [Manna/Pnueli]

$$|(px)^*\rangle q = q + p|x\rangle|(px)^*\rangle q \qquad \langle(xp)^*|q = q + p\langle(xp)^*|\langle x|q$$

$$|(px)^*\rangle 0 \leq 0 \qquad \langle x|0 = 1$$

$$|x^*](p \to q) \leq |x^*]p \to |x^*]q \qquad [x^*|(p \to q) \leq [x^*|p \to [x^*|q$$

$$|x^*]p \leq p|x]|x^*]p \qquad |x^*](p \to |x]p) \leq |x^*](p \to |x^*]p)$$

$$p \leq [x||x\rangle p \qquad p \leq |x]\langle x|p$$

$$\mathsf{init}_x \leq |x^*](p \to [x|q) \to |x^*](p \to [x^*|q) \qquad \mathsf{init}_x \leq |x^*]p \to |x^*][x|p$$

$$|x](p \to q) = |x]p \to |x]q \qquad [x|(p \to q) = [x|p \to [x|q$$

$$\langle x|p \leq [x|p \qquad |x\rangle p = |x]p$$

are theorems of MKA or express linearity of time in MKA

MKA and Hoare Logic

fact: MKA subsumes (propositional) Hoare logic

validity of Hoare triple: $\models \{p\}x\{q\} \Leftrightarrow \langle x|p \leq q$

example: validity of while rule $\langle x|pq \leq q \Rightarrow \langle (px)^*\neg p|q \leq \neg pq$

benefits of algebraic approach:

- wlp semantics for free ($\mathsf{wlp}(x,p) = |x]p$)
- soundness and completeness of Hoare logic easy in MKA
- Hoare logic deconstructed to equational modal reasoning

MKA and Hoare Logic

example: validity of while-rule $\langle x|\langle p|q \leq q \Rightarrow \langle (px)^* \neg p|q \leq \langle \neg p|q$

proof: (immediate with ATP)

$$\begin{aligned} \langle x|\langle p|q \leq q &\Leftrightarrow \langle px|q \leq q && (\text{ contravariance }) \\ &\Rightarrow \langle (px)^*|q \leq q && (\text{ induction }) \\ &\Rightarrow \langle \neg p|\langle (px)^*|q \leq \langle \neg p|q && (\text{ isotonicity }) \\ &\Leftrightarrow \langle (px^*) \neg p|q \leq \langle \neg p|q && (\text{ contravariance }) \end{aligned}$$

perspective:

- automated verification in Hoare logic with Isabelle
- numbers or data types require integration of SMT
- approach extends to total/general correctness

Example: Synthesis of Warshall's Algorithm

Hoare logic: (simple while-programs)

1. invariant established by initialisation when precondition is true
2. executions of loop body preserve invariant when test of loop is true
3. invariant establishes postcondition when test of loop is false

synthesis: "a program and its correctness proof should be developed hand-in-hand"

- develop invariant as modification of postcondition
- incrementally establish proof obligations (synthesis of test/assignments)

Initial Specification

spec: given finite binary relation x, find program with relational variable y that stores transitive closure of x after execution

goal: instantiate template

```
... y:=x ...
while ... do
  ... y:=? ... od
```

pre/postcondition: (evident from spec)

```
pre(x) <-> x=x.
post(x,y) <-> y=tc(x).
```

task: use proof obligations to synthesise initialisation, test, body

Invariant, Initialisation and Test

invariant: `inv(x,y,v) <-> (set(v) -> y=rtc(x;v);x).`

initialisation: $v := 0$

test: $v \neq d(x)$

justification: in KA with domain

```
pre(x) -> inv(x,x,0).                    %no time
inv(x,y,v) & v=d(x) -> post(x,y).        %no time
```

Termination and Synthesis of Loop

task: use preservation of invariant to find assignments

result: (development in MKA)

- $v := v + p$ (increment set v by point p)
- $y := y + y; p; y$ (increment y by $y; p; y$ with p)

proof obligation: `wpoint(w) & inv(x,y,v) & y!=d(x) -> inv(x,y+y;(w;y),v+w).`

theorem: Warshall's algorithm is (partially) correct:

```
y,v:=x,0
while v!=d(x) do
  p:=point(v')
  y,v:=y+y;p;y,v+p od
```

Example: Termination Analysis

theorem: [BachmairDershowitz86] *termination of the union of two rewrite systems can be separated into termination of the individual systems if one rewrite system quasicommutes over the other*

remarks: theorem considered difficult

- posed as KA challenge by Ernie Cohen in 2001
- proof by Podelski/Rybalchenko uses infinite version of Ramsey's theorem
- used in MS termination analysis tools

Termination Analysis

formalisation: MKA K with divergence $^\nabla : K \to d(K)$ as greatest fixed point

$$x^\nabla \leq |x\rangle x^\nabla \qquad p \leq |x\rangle p + q \Rightarrow p \leq x^\nabla + |x^*\rangle q$$

encoding:

- quasicommutation $yx \leq x(x+y)^*$
- separation of termination $(x+y)^\nabla = 0 \;\Leftrightarrow\; x^\nabla + y^\nabla = 0$

statement: termination of x and y can be separated if x quasicommutes over y

Termination Analysis

result: extremely short proof reveals new refinement theorem

$$yx \le x(x+y)^* \Rightarrow (x+y)^\nabla = x^\nabla + |x^*\rangle y^\nabla$$

proof: (coinductive)

$$\begin{aligned}
(x+y)^\nabla &= y^\nabla + |y^* x\rangle (x+y)^\nabla \\
&\le y^\nabla + |x(x+y)^*\rangle (x+y)^\nabla \\
&= y^\nabla + |x\rangle (x+y)^\nabla \\
&\le x^\nabla + |x^*\rangle y^\nabla \\
&= 0 + x^* 0 \\
&= 0
\end{aligned}$$

Example: Automating a Modal Correspondence Result

modal logic: Löb's formula $\Box(\Box p \to p) \to \Box p$

translation to MKA: $|x\rangle p \leq |x\rangle(p - |x\rangle p) = |x\rangle \mathsf{max}_x(p)$

intuition: all states with transitions into p are states from which no further transitions are possible

remark: this would correspond to Noethericity if x is transitive ($xx \leq x$)

fact: two more characterisations of termination

- $p \leq |x^*\rangle \mathsf{max}_x(p)$ (x pre-Löbian)
- $\mathsf{max}_x(p) = 0 \Rightarrow p = 0$ (x Noetherian)

Automating a Modal Correspondence Result

property: for every x in some MKA with divergence

(i) x Löbian $\Rightarrow$ x Noetherian
(ii) x Noetherian $\Leftrightarrow$ x pre-Löbian
(iii) x pre-Löbian and $x = xx$ $\Rightarrow$ x Löbian

proofs: by ATP

(i) $\leq 4s$
(ii) $\leq 4s$ and $\leq 20s$ (hypothesis learning)
(iii) $\leq 1s$ (hypothesis learning)

remark: this is a modal correspondence result

- Noethericity corresponds to frame property
- proof is calculational and automated
- model theory is normally used

Free Domain Semirings

polynomials: consider laws

$$x(y+z) = xy + xz, \qquad (x+y)z = xz + yz, \qquad d(x+y) = d(x) + d(y)$$

- every domain semiring term is equivalent to polynomial

$$m_0 + m_1 + \cdots + m_k$$

- every monomial can be written as trace

$$d(s_0)x_0d(s_1)x_1 \ldots d(s_{n-1})x_{n-1}d(s_n)$$

because $d(x)d(y) = d(d(x)y)$ and $d(1) = 1$

One-Generated Case

observation: $d(xt) = d(xd(t))$ and $d(t) \leq 1$ imply $d(1) \geq d(x) \geq d(x^2) \geq \ldots$

consequence: each trace is equivalent to flat trace $d(x^{k_0})xd(x^{k_1})x \ldots xd(x^{k_n})$

- if $s = xt$, then $d(s) = d(xd(t)) = d(xd(x^m)) = d(x^{m+1})$ for some m
- if $s = d(t)u$, then $d(s) = d(d(t)d(u)) = d(t)d(u) = d(x^m)d(x^n) = d(x^{\max(m,n)})$ for some m, n

observation: for each $xd(x^k)$, $d(x^{k+1})$ is least p such that $pxd(x^k) = xd(x^k)$

consequence:

- each flat trace can uniquely be expanded such that $k_i > k_j$ if $i < j$
- trace normal forms isomorphic to strictly decreasing integer sequences

One-Generated Case

fact: sets of interreduced strictly decreasing integer sequences can be made into d-semirings

- multiplication:
  1. merge $(k_1, \ldots, k_m)$ and $(l_1, \ldots, l_n)$ to $(k_1, \ldots, \max(k_m, l_1), \ldots, l_n)$
  2. then expand
- domain: pick first integer from sequence

theorem: d-semiring of sets of inter-reduced decreasing integer sequences is isomorphic to one-generated d-semiring

- if two sets of integer sequences are equal, then the two terms must be eqivalent (by nf construction)
- if two sets of decreasing integer sequences are different, then the two terms are different in some model

n-Generated Case

observation: domain terms in traces are no longer flat

head normal form: domain term $d(xd(s_0)\dots d(s_n))$ and $d(s_i)$ all in hnf

fact: every domain term is equivalent to product of domain terms in hnf

expanded polynomials:

- monomials with hnf domain terms can again be expanded (uniquely)
- use $d(s) = d(s_0)$ if $s = d(s_0)t$ expanded trace

fact: sets of expanded traces form again domain semirings

normal forms: interreduce hnf domain terms recursively via semilattice order

Free Domain Semiring

future work: decidability of equational theory

- a-semirings
- KAs with (anti)domain (interaction of star/domain)

remark: guarded strings arise if domain is not nested

Representability

question: can one extend axiomatisations to characterise relational d-semirings?

fact: [Andréka] for signature $\{+,\cdot\} \subseteq \Sigma \subseteq \{+,\cdot,0,1,{}^*,{}^\circ\}$, the class of representable Σ-algebras is not finitely axiomatisable

consequence: [Hirsch/Mikulás] the class of representable d/a-semirings is not finitely axiomatisable

- appropriately define (antidomain) domain on Σ-algebras above

Domain Semigroups

free domain semirings: interaction of domain and monomials is essential!

domain semigroup: semigroup $(S, \cdot)$ with $d : S \to S$ satisfying

$$d(x)x = x, \quad d(xy) = d(xd(y)), \quad d(d(x)y) = d(x)d(y), \quad d(x)d(y) = d(y)d(x)$$

domain monoid: monoid satisfying same domain axioms

properties:

- axioms hold in relational structures
- $d(S)$ is meet-semilattice
- $x \leq y \Leftrightarrow x = d(x)y$ is fundamental order
- $x = px \Leftrightarrow d(x) \leq p$ (least left preservation)

Representability

fact: representable d-monoids form quasivariety [Schein]

fact: $xy = d(x) \wedge yx = x \wedge d(y) = 1 \Rightarrow x = d(x)$ fails in some d-monoid but holds in relational model

consequence: quasivariety is not a variety

theorem: [Hirsch/Mikulás] class of representable d-semigroups is not finitely axiomatisable

twisted law: [Jackson/Stokes] $xd(y) = d(xy)x$ forces functional models

theorem: [Trokhimenko] twisted d-semigroups/monoids can be emdedded into partial transformation semigroups

Antidomain Monoids

antidomain monoid: $(S, \cdot, 1, ')$ with

$$x'x = 0, \qquad x0 = 0, \qquad x'y' = y'x', \qquad x''x = x,$$
$$x' = (xy)'(xy')', \qquad (xy)'x = (xy)'xy'$$

properties:

- $d(x) = x''$ is domain operation
- $x + y = (x'y')'$ defines join operation
- S' is Boolean algebra
- defining $|x\rangle p = (xp)''$ and $|x]p = (xp')'$, where $p = p''$, yields BAOs
- modal semigroups with conjugation/Galois connections arise in this weak setting
- a-semigroup is twisted iff $|x\rangle p \leq |x]p$ (all x deterministic)

Representability

facts:

- variety of a-monoids is variety of representable a-monoids [Hollenberg]
- quasivariety of representable a-monoids is not a variety [Hollenberg]
- class of representable a-monoids is not finitely axiomatisable [Hirsch/Mikulás]

Variations

domain for pre/near-semirings

- total/general correctness
- refinement calculi
- action systems
- game algebras and multirelations
- process algebras

properies:

- domain axioms essentially as before
- definition of modal operators no longer possible

future work: decidability, free algebras, representability, . . .

Conclusion

(modal) Kleene algebras:

- versatile powerful tools for modelling programs and systems
- easy to combine with ATP systems
- interesting mathematical structures
  (free algebras, decision procedures, representability, axiomatisability)
- some non-representability results a bit disappointing. . .

automated program analysis:

- promising first results
- engineering work to be done
- hypothesis learning/deduction from large dbs seems very interesting

Conclusion

additional material:

- code at `www.dcs.shef.ac.uk/~georg/ka`
  (and in TPTP-library)
- lecture notes at `www.dcs.shef.ac.uk/~georg`

Some Papers

- J Desharnais, G Struth, *Internal Axioms for Domain Semirings*. SCP, 2010.
- J Desharnais, B Möller, G Struth, *Algebraic Notions of Termination*. LMCS, 2010.
- P Höfner, G Struth, *Algebraic Notions of Nontermination*. JLAP, 2010.
- R Berghammer, G Struth, *On Automated Program Construction and Verification*. MPC, 2010.
- J Desharnais, P Jipsen, G Struth, *Domain and Antidomain Semigroups*. RelMiCS/AKA, 2009.
- J Desharnais, G Struth, *Domain Axioms for a Family of Near-Semirings*. AMAST, 2008.
- P Höfner, G Struth, *Automated Reasoning in Kleene Algebra*. CADE, 2007.
- J Desharnais, B Möller, G Struth, *Kleene Algebra with Domain*, ACM TOCL, 2006.
- B Möller, G Struth, *Algebras of Modal Operators and Partial Correctness*. TCS, 2004.