Kleene Algebra with Tests: A Tutorial

Part I

Dexter Kozen
Cornell University

RAMiCS, September 17–18, 2012

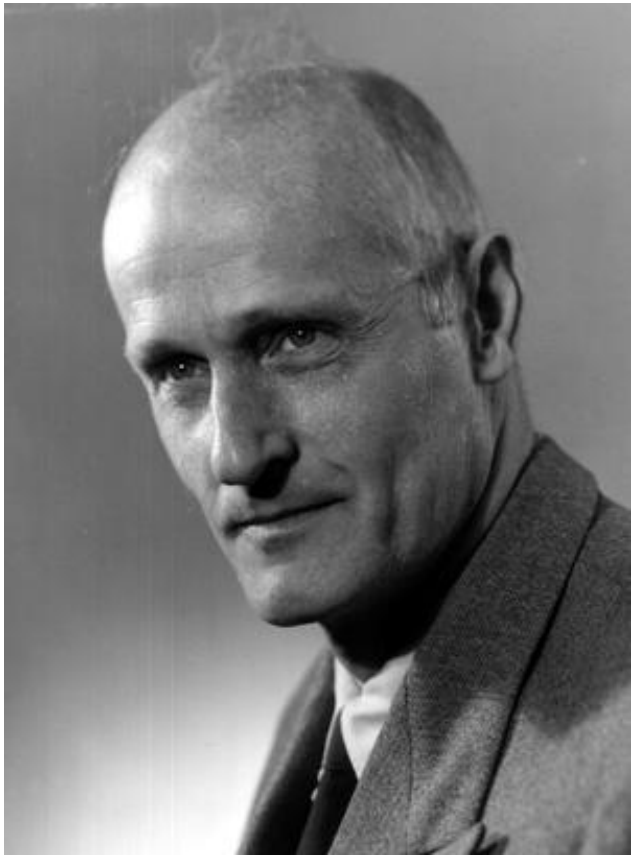
Outline

- Today: Some history. Definitions, models, basic results. Expressiveness, completeness, complexity.
- Tomorrow: The coalgebraic theory. Automata and program schematology. Applications.

Today – Definitions, Models, Basic Results

- Definitions: KA and KAT
- Models: relational models, language models, trace models, matrices over a KAT
- Basic results:
  - KAT and Hoare logic
  - completeness for the equational theory
  - completeness for the Hoare theory (reasoning under assumptions)
  - completeness and incompleteness results for PHL
  - complexity (PSPACE completeness)
  - typed KA and KAT and relation to type theory

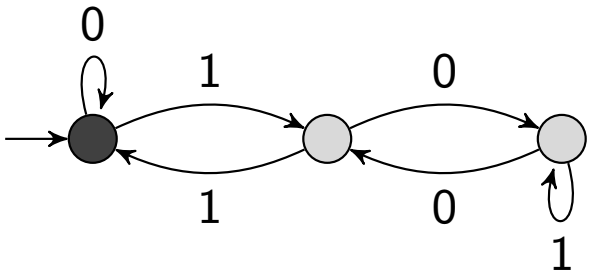
Kleene Algebra (KA)

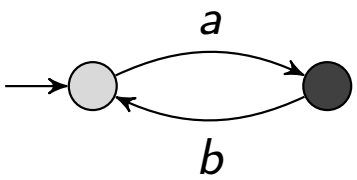


Stephen Cole Kleene
(1909–1994)

- Kleene algebra is an algebraic system that captures axiomatically the properties of a natural class of structures arising in logic and computer science.
- Named for Stephen Cole Kleene, who among his many other achievements, invented finite automata and regular expressions.
- Kleene algebra is the algebraic theory of these objects. It has many natural and useful interpretations.

Kleene's Theorem (1956)





$(0 + 1(01^*0)^*1)^*$
{multiples of 3 in binary}

$(ab)^*a = a(ba)^*$
$\{a, aba, ababa, \ldots\}$

$(a + b)^* = a^*(ba^*)^*$
{all strings over $\{a, b\}$}

Foundations of the Algebraic Theory

John Horton Conway
(1937–)

J. H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, London, 1971 (out of print).

Kleene Algebra

Kleene algebras arise in various guises in many contexts:

- relational algebra,
- semantics and logics of programs,
- program analysis and compiler optimization,
- automata and formal language theory,
- design and analysis of algorithms.

Many authors have contributed to the development of Kleene algebra over the years: Anderaa, Archangelsky, Backhouse, Bloom, Boffa, Braibant, Cohen, Conway, Desharnais, Ésik, Furusawa, Höfner, Hopkins, Jipsen, Kleene, Krob, Kuich, McIver, Meyer, Möller, Morgan, Pous, Pratt, Redko, Sakarovich, Salomaa, Schmidt, Silva, Stockmeyer, Struth, and Tiuryn to name a few.

Kleene Algebra

A Kleene algebra is an algebraic structure

$$(K, +, \cdot, ^*, 0, 1)$$

consisting of a set K with distinguished operations and constants satisfying certain axioms.

operation	*intuition*	*arity*
$+$	addition, choice, join	2
$\cdot$	multiplication, sequential composition, meet	2
*	asterate, iteration	1
0	additive identity, fail, false	0
1	multiplicative identity, skip, true	0

The intuitive meaning of the operations depends on the model.

A regular expression is a term in this language.

Axioms of KA

Idempotent Semiring Axioms

$$p + (q + r) = (p + q) + r \qquad p(qr) = (pq)r$$
$$p + q = q + p \qquad 1p = p1 = p$$
$$p + 0 = p \qquad p0 = 0p = 0$$
$$p + p = p$$
$$p(q + r) = pq + pr \qquad a \leq b \overset{\text{def}}{\iff} a + b = b$$
$$(p + q)r = pr + qr$$

Axioms for *

$$1 + pp^* \leq p^* \qquad q + px \leq x \;\Rightarrow\; p^*q \leq x$$
$$1 + p^*p \leq p^* \qquad q + xp \leq x \;\Rightarrow\; qp^* \leq x$$

Some Basic Consequences

- $p^* = p^* p^*$
- $p^* = p^{**}$
- $p^* = 1 + pp^*$
- $p^* = 1 + p^* p$
- $(p + q)^* = p^*(qp^*)^*$ denesting
- $(pq)^* p = p(qp)^*$ sliding
- for all $n \geq 1$, $p^* = (1 + p)^{n-1}(p^n)^*$
- $pq = qr \Rightarrow p^* q = qr^*$ bisimulation
- $px \leq x \Rightarrow p^* x \leq x$
- $xp \leq x \Rightarrow xp^* \leq x$

Basic Facts about $\leq$

- $\leq$ is a partial order (reflexive, antisymmetric, transitive – depends heavily on idempotence)
- least element 0
- All operations monotone with respect to $\leq$
  that is, if $p \leq q$, then
  - $p + r \leq q + r$
  - $pr \leq qr$
  - $rp \leq rq$
  - $p^* \leq q^*$

Significance of the * Axioms

Axioms for *

$$1 + pp^* \leq p^* \qquad\qquad q + px \leq x \;\Rightarrow\; p^*q \leq x$$

Axioms for *

$$q + pp^*q \leq p^*q \qquad\qquad q + px \leq x \;\Rightarrow\; p^*q \leq x$$

p^*q is the least x such that $q + px \leq x$

Systems of Affine Linear Inequalities

Theorem

Any system of n affine linear inequalities in n unknowns has a unique least solution

$$\begin{aligned} q_1 + p_{11}x_1 + p_{12}x_2 + \cdots p_{1n}x_n &\leq x_1 \\ &\vdots \\ q_n + p_{n1}x_1 + p_{n2}x_2 + \cdots p_{nn}x_n &\leq x_n \end{aligned}$$

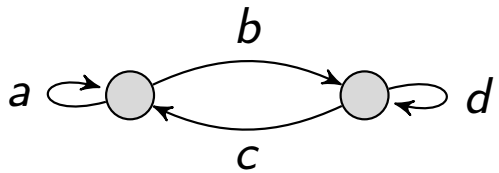
Matrices over a KA form a KA

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

$$0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad 1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^* = \begin{bmatrix} (a+bd^*c)^* & (a+bd^*c)^*bd^* \\ (d+ca^*b)^*ca^* & (d+ca^*b)^* \end{bmatrix}$$



Systems of Affine Linear Inequalities

Theorem

Any system of n affine linear inequalities in n unknowns has a unique least solution

$$\begin{aligned} q_1 + p_{11}x_1 + p_{12}x_2 + \cdots p_{1n}x_n &\leq x_1 \\ &\vdots \\ q_n + p_{n1}x_1 + p_{n2}x_2 + \cdots p_{nn}x_n &\leq x_n \end{aligned}$$

$$\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} + \boxed{P = p_{ij}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Least solution is $P^* q$

Matrices over a KA

- Representation of finite automata
- Construction of regular expressions
- Solution of linear inequalities over a KA
- Connectivity and shortest path algorithms

Language-Theoretic Models

$\Sigma^* = \{\text{finite-length strings over a finite alphabet } \Sigma\}$

For $A, B \subseteq \Sigma^*$:

$$
\begin{aligned}
A + B &\stackrel{\text{def}}{=} A \cup B \\
A \cdot B &\stackrel{\text{def}}{=} \{xy \mid x \in A,\ y \in B\} \\
0 &\stackrel{\text{def}}{=} \varnothing \\
1 &\stackrel{\text{def}}{=} \{\varepsilon\} \qquad \varepsilon = \text{the null string} \\
A^* &\stackrel{\text{def}}{=} \bigcup_{n \geq 0} A^n \;=\; \{x_1 \cdots x_n \mid n \geq 0,\ x_i \in A,\ 1 \leq i \leq n\}
\end{aligned}
$$

where $A^0 \stackrel{\text{def}}{=} \{\varepsilon\}$ and $A^{n+1} \stackrel{\text{def}}{=} A \cdot A^n$.

The operation * on sets of strings is known as (Kleene) asterate.

Language-Theoretic Models

- Any subset of 2^{Σ^*} closed under the operations $\varnothing, \{\varepsilon\}, \cup, \cdot, ^*$
- $\mathrm{Reg}_\Sigma = \{$regular sets over $\Sigma\}$ = smallest subalgebra of 2^{Σ^*} containing $\{a\}$, $a \in \Sigma$.
- Many others!

The standard interpretation is the unique KA homomorphism $R : \mathrm{RExp}_\Sigma \to \mathrm{Reg}_\Sigma$ such that $R(a) = \{a\}$. Examples:

$$\begin{aligned} R(a^*b^*) &= \{a^n b^m \mid n, m \geq 0\} \\ R(a(ba)^*) &= \{a, aba, ababa, abababa, \ldots\} \\ R((a+b)^*) &= \{\text{all strings of } a\text{'s and } b\text{'s}\} \end{aligned}$$

Context-Free Languages

Context-free languages are the algebraic closure of Reg_Σ in 2^{Σ^*}, i.e. solutions of finite systems of algebraic inequalities.

Examples:

- $\{a^n b^n \mid n \geq 0\}$: $1 + axb \leq x$
- Palindromes: $1 + axa + bxb \leq x$
- Balanced parens: $1 + (x) + xx \leq x$

Parikh's theorem = Every CFL is "letter equivalent" to a regular set = Every commutative KA is algebraically closed

Relational Models

Let $R, S \subseteq X \times X$

$$
\begin{aligned}
R + S &\stackrel{\text{def}}{=} R \cup S \\
R \circ S &\stackrel{\text{def}}{=} \{(x,z) \mid \exists y \in X \ (x,y) \in R \wedge (y,z) \in S\} \\
0 &\stackrel{\text{def}}{=} \varnothing && \text{empty relation} \\
1 &\stackrel{\text{def}}{=} \{(x,x) \mid x \in X\} && \text{identity relation} \\
R^* &\stackrel{\text{def}}{=} \bigcup_{n \geq 0} R^n && \text{reflexive-transitive closure}
\end{aligned}
$$

where

$$R^0 \stackrel{\text{def}}{=} 1 \qquad\qquad R^{n+1} \stackrel{\text{def}}{=} R \circ R^n.$$

Relational Models

Relational KA

- Any subset of $2^{X \times X}$ closed under these operations
- Useful in programming language semantics, because they can be used to represent the input/output relations of programs

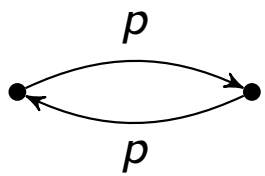
Every language model is isomorphic to a relational model

$$A \;\mapsto\; \{(x, xy) \mid x \in \Sigma^*,\; y \in A\}$$

but not vice versa (language models satisfy

$$p^2 = 1 \;\Rightarrow\; p = 1,$$

relational models not necessarily)



Trace Models

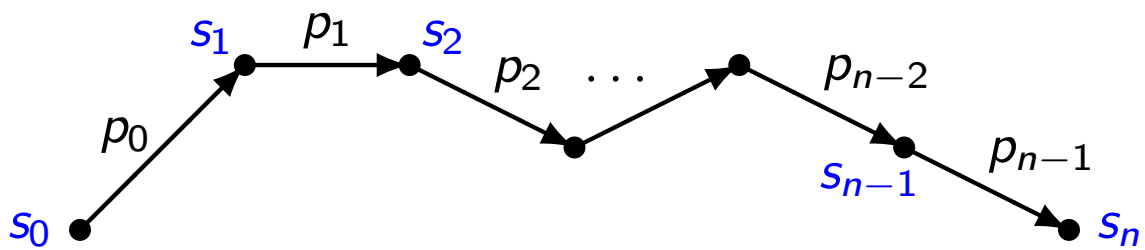
Labeled transition system (LTS)

- a set X of states
- a mapping $\pi : \Sigma \rightarrow 2^{X \times X}$, where Σ is a set of atomic actions

A trace is an alternating sequence of states and atomic actions

$$s_0\ p_0\ s_1\ p_1\ \cdots\ s_{n-1}\ p_{n-1}\ s_n$$

beginning and ending with a state, such that $(s_i, s_{i+1}) \in \pi(p_i)$, $0 \leq i \leq n-1$.



Trace Models

Fusion product

- If $\text{last}\,\sigma = \text{first}\,\tau$, form $\sigma\tau$, suppressing the extra copy of $\text{last}\,\sigma = \text{first}\,\tau$
- If $\text{last}\,\sigma \neq \text{first}\,\tau$, $\sigma\tau$ does not exist

For $A, B \in 2^{\{\text{Traces}\}}$,

$$
\begin{aligned}
A + B &\stackrel{\text{def}}{=} A \cup B \\
A \cdot B &\stackrel{\text{def}}{=} \{\sigma\tau \mid \sigma \in A,\ \tau \in B,\ \sigma\tau \text{ exists}\} \\
0 &\stackrel{\text{def}}{=} \varnothing \\
1 &\stackrel{\text{def}}{=} \{s \mid s \in X\} \ = \ \{\text{traces of length } 0\} \\
A^* &\stackrel{\text{def}}{=} \bigcup_{n \geq 0} A^n.
\end{aligned}
$$

Trace Models

Every language model is isomorphic to a trace model on one state.

Every trace model is isomorphic to a relational model

$$A \;\mapsto\; \{(\sigma, \sigma\tau) \mid \sigma \in \{\text{Traces}\},\ \tau \in A\}$$

but not vice versa (trace models satisfy

$$p^2 = 1 \;\Rightarrow\; p = 1,$$

relational models not necessarily)

The min,+ Algebra (aka Tropical Semiring)

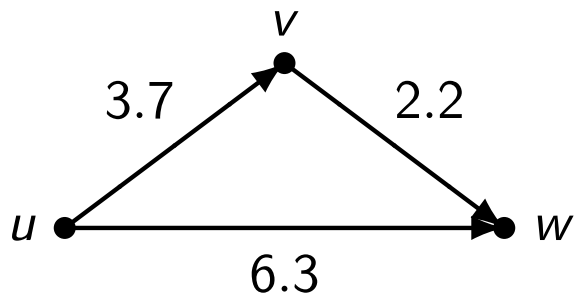
The domain is $\mathbb{R}_+ \cup \{\infty\}$, where

- $\mathbb{R}_+ = \{r \in \mathbb{R} \mid r \geq 0\}$
- $r \leq \infty$ for all $r \in \mathbb{R}$
- $x + \infty = \infty + x = \infty + \infty = \infty$

The Kleene operations are

K	$\mathbb{R}_+ \cup \{\infty\}$
$+$	min
$\cdot$	$+$
0	∞
1	0
$\leq$	$\geq$

and $x^* = 1$ (= the real number 0) for any x.



$$R = \begin{array}{c|ccc} & u & v & w \\ \hline u & 0 & 3.7 & 6.3 \\ v & \infty & 0 & 2.2 \\ w & \infty & \infty & 0 \end{array} \qquad R^* = \begin{array}{c|ccc} & u & v & w \\ \hline u & 0 & 3.7 & 5.9 \\ v & \infty & 0 & 2.2 \\ w & \infty & \infty & 0 \end{array}$$

Deductive Completeness and Complexity

The KA axioms exactly characterize the equational theory of

- the standard interpretation $R : \mathrm{RExp}_\Sigma \to \mathrm{Reg}_\Sigma$
- all language models
- all relational models
- all trace models

That is, $p = q$ holds under all interpretations in that class of models iff $p = q$ is a theorem of KA.

The equational theory is *PSPACE*-complete [(1 + Stock)Meyer 1974]

Salomaa's Axiomatization (1966)

Arto Salomaa
(1934–)

Salomaa (1966) was the first to axiomatize the equational theory of the regular sets and prove completeness. He presented two axiomatizations F_1 and F_2 for the algebra of regular sets and proved their completeness.

Aanderaa (1965) independently presented a system similar to Salomaa's F_1. Backhouse (1975) gave an algebraic version of F_1.

Salomaa's Axiomatization (1966)

Salomaa's system F_1 contains the rule

$$\frac{u + st = t, \quad \varepsilon \notin R(s)}{s^* u = t}$$

This rule is sound under the standard interpretation R, but the premise "$\varepsilon \notin R(s)$" is not preserved under substititution, thus the rule is not valid under nonstandard interpretations.

For example, if s, t, and u are the single letters a, b and c respectively, then the rule holds; but it does not hold after the substitution

$$a \mapsto 1 \qquad b \mapsto 1 \qquad c \mapsto 0.$$

Another way to say this is that the rule must not be interpreted as a universal Horn formula.

Other Axiomatizations (in order of generality)

- Complete semirings (S-algebras, quantales) (Conway 1971)
  - arbitrary suprema & distributivity
- Closed semirings (ω-complete semirings) (Aho, Hopcroft, Ullman 1974)
  - countable suprema & distributivity
- *-continuous KA
  - $pq^*r = \sup_{n \geq 0} pq^n r$

Same equational theory as KA

Motivation for KAT

Propositional Dynamic Logic (PDL) [Fischer & Ladner 1979]

- KA + propositional logic + modalities

$$\varphi \wedge [p^*](\varphi \to [p]\varphi) \;\to\; [p^*]\varphi$$

- subsumes propositional Hoare logic (PHL)

$$\{\varphi\}\, p\, \{\psi\} \;\overset{\text{def}}{\iff}\; \varphi \to [p]\psi$$

- semantically well-grounded and deductively complete, but complex to decide

PDL (Fischer & Ladner 1979)

The test operator ? makes a program out of a test:

$$[\![\varphi?]\!] = \{(s, s) \mid s \vDash \varphi\}$$

Used to model conventional programming constructs:

$$\text{if } \varphi \text{ then } p \text{ else } q \overset{\text{def}}{\Longleftrightarrow} \varphi?; p + \neg\varphi?; q$$

$$\text{while } \varphi \text{ do } p \overset{\text{def}}{\Longleftrightarrow} (\varphi?; p)^*; \neg\varphi?$$

PDL (Fischer & Ladner 1979)

From a practical point of view, many arguments do not require the full power of PDL, but can be carried out in a purely equational subsystem using Kleene algebra

But the Boolean component is essential, as it is needed to model conventional programming constructs

$$\text{if } \varphi \text{ then } p \text{ else } q \iff^{\text{def}} \varphi?; p + \neg\varphi?; q$$

$$\text{while } \varphi \text{ do } p \iff^{\text{def}} (\varphi?; p)^*; \neg\varphi?$$

Kleene Algebra with Tests (KAT)

A Mashup[1] of Kleene and Boolean Algebra

$(K, B, +, \cdot, ^*, ^-, 0, 1), \quad B \subseteq K$

- $(K, +, \cdot, ^*, 0, 1)$ is a Kleene algebra
- $(B, +, \cdot, ^-, 0, 1)$ is a Boolean algebra
- $(B, +, \cdot, 0, 1)$ is a subalgebra of $(K, +, \cdot, 0, 1)$

- $p, q, r, \ldots$ range over K
- $a, b, c, \ldots$ range over B

[1] *Mashup: A web page or web application that uses and combines data, presentation, or functionality from two or more sources to create new services. The term implies easy, fast integration, frequently using data sources to produce enriched results that were not necessarily the original reason for producing the raw source data. –Wikipedia*

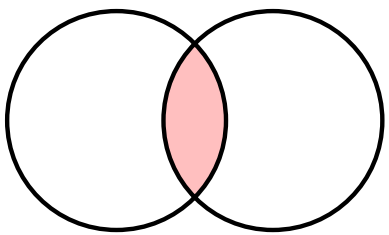
Kleene Algebra with Tests (KAT)

A Mashup of Kleene and Boolean Algebra

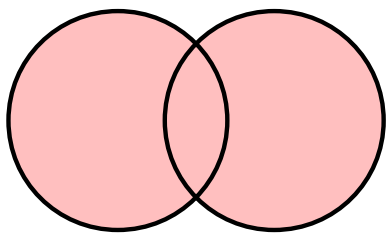
$+, \cdot, 0, 1$ serve double duty

- applied to actions, denote choice, composition, fail, and skip, resp.
- applied to tests, denote disjunction, conjunction, falsity, and truth, resp.
- these usages do not conflict!

$$bc = b \wedge c$$



$$b + c = b \vee c$$



Axioms of Boolean Algebra

$a + (b + c) = (a + b) + c$	$a(bc) = (ab)c$
$a + b = b + a$	$ab = ba$
$a + 0 = a$	$a1 = a$
$a + a = a$	$aa = a$
$a(b + c) = ab + ac$	$(a + b)c = ac + bc$
$a0 = 0$	$a + 1 = 1$
$\overline{a + b} = \overline{a}\,\overline{b}$	$\overline{ab} = \overline{a} + \overline{b}$
$\overline{\overline{a}} = a$	

Models of KAT

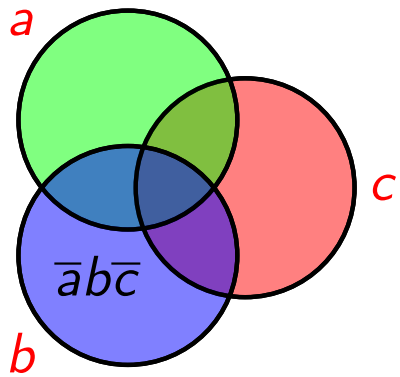
- Language-theoretic models
  - K = sets of guarded strings over Σ, T
  - B = free Boolean algebra generated by T
- Relational models
  - K = binary relations on a set X
  - B = subsets of the identity relation
- Trace models
  - K = sets of traces $s_0 p_0 s_1 p_1 s_2 \cdots s_{n-1} p_{n-1} s_n$
  - B = traces of length 0
- $n \times n$ matrices over a KAT K, B
  - $K' = n \times n$ matrices over K
  - $B' = n \times n$ diagonal matrices over B

Guarded Strings over Σ, T [Kaplan 69]

Σ action symbols T test symbols

B = free Boolean algebra generated by T
At = atoms of $B = \{\alpha, \beta, \ldots\}$

E.g. if $T = \{a, b, c\}$, then $\bar{a}b\bar{c}$ is an atom



Guarded strings $\alpha_0 p_0 \alpha_1 p_1 \alpha_2 \cdots \alpha_{n-1} p_{n-1} \alpha_n \in (\text{At} \cdot \Sigma)^* \cdot \text{At}$

- Guarded strings are the join-irreducible elements of the free KAT on generators Σ, T
- Essentially traces on the state set At

Standard Interpretation of KAT

$\mathsf{GS}_{\Sigma,T} = \{\text{guarded strings over } \Sigma, T\}$

$$
\begin{aligned}
A + B &= A \cup B \\
AB &= \{x\alpha y \mid x\alpha \in A,\ \alpha y \in B\} \\
0 &= \varnothing \\
1 &= \mathsf{At} \\
A^* &= \bigcup_{n \geq 0} A^n \ = \ A^0 \cup A^1 \cup A^2 \cup \cdots \\
\overline{A} &= \mathsf{At} - A, \quad A \subseteq \mathsf{At}
\end{aligned}
$$

$\mathsf{RExp}_{\Sigma,T} = \{\text{KAT terms over } \Sigma, T\}$

Standard interpretation $G : \mathsf{RExp}_{\Sigma,T} \to 2^{\mathsf{GS}_{\Sigma,T}}$:

$$G(p) = \{\alpha p \beta \mid \alpha, \beta \in \mathsf{At}\},\ p \in \Sigma \qquad G(b) = \{\alpha \mid \alpha \leq b\},\ b \in T$$

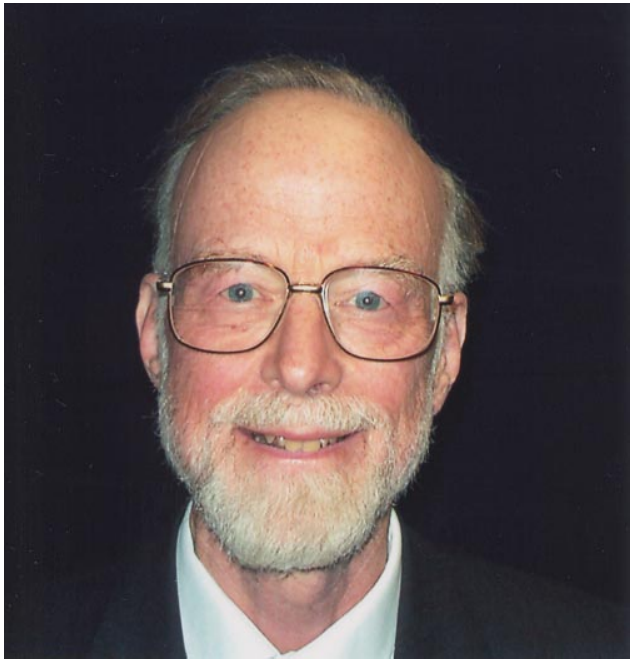
Modeling While Programs

$$p; q \stackrel{\text{def}}{=} pq$$

$$\text{if } b \text{ then } p \text{ else } q \stackrel{\text{def}}{=} bp + \overline{b}q$$

$$\text{while } b \text{ do } p \stackrel{\text{def}}{=} (bp)^* \overline{b}$$

Hoare Logic



C. A. R. "Tony" Hoare

Partial correctness assertion

$$\{\varphi\}\ p\ \{\psi\}$$

"if φ holds of the input state, and if p halts, then ψ must hold of the output state"

KAT Subsumes PHL

$$\{b\}\ p\ \{c\} \overset{\text{def}}{\iff} bp \leq pc$$

$$\iff bp = bpc$$

the test c is always redundant after executing bp

$$\iff bp\overline{c} = 0$$

there is no computation of p with precondition b and postcondition $\overline{c}$

KAT Subsumes PHL

composition rule

$$\frac{\{b\}\ p\ \{c\} \quad \{c\}\ q\ \{d\}}{\{b\}\ p\,;q\ \{d\}} \qquad bp \leq pc \wedge cq \leq qd \ \Rightarrow\ bpq \leq pqd$$

conditional rule

$$\frac{\{bc\}\ p\ \{d\} \quad \{\overline{b}c\}\ q\ \{d\}}{\{c\}\ \text{if } b \text{ then } p \text{ else } q\ \{d\}} \qquad \begin{array}{l} bcp \leq pd \wedge \overline{b}cq \leq qd \\ \quad \Rightarrow\ c(bp + \overline{b}q) \leq (bp + \overline{b}q)d \end{array}$$

while rule

$$\frac{\{bc\}\ p\ \{c\}}{\{c\}\ \text{while } b \text{ do } p\ \{\overline{b}c\}} \qquad bcp \leq pc \ \Rightarrow\ c(bp)^*\overline{b} \leq (bp)^*\overline{b}\,\overline{b}c$$

weakening rule

$$\frac{b' \rightarrow b \quad \{b\}\ p\ \{c\} \quad c \rightarrow c'}{\{b'\}\ p\ \{c'\}} \qquad b' \leq b \wedge bp \leq pc \wedge c \leq c' \ \Rightarrow\ b'p \leq pc'$$

KAT Subsumes PHL

In fact, one can replace the conditional and while rules with

choice rule

$$\frac{\{b\}\ p\ \{c\} \quad \{b\}\ q\ \{c\}}{\{b\}\ p+q\ \{c\}} \qquad bp \leq pc \wedge bq \leq qc \ \Rightarrow\ b(p+q) \leq (p+q)c$$

iteration rule

$$\frac{\{b\}\ p\ \{b\}}{\{b\}\ p^*\ \{b\}} \qquad bp \leq pb \ \Rightarrow\ bp^* \leq p^*b$$

test rule

$$\{b\}\ c\ \{bc\} \qquad bc \leq cbc$$

Proof of the While Rule

The KAT translation of the while rule is

$$bcp \leq pc \;\Rightarrow\; c(bp)^*\overline{b} \leq (bp)^*\overline{b}\,\overline{b}c$$

Assume $bcp \leq pc$. The rhs is equivalent to

$$c(bp)^*\overline{b} \leq (bp)^*c\overline{b}.$$

By monotonicity, it suffices to show

$$c(bp)^* \leq (bp)^*c.$$

By the star rule $x + zy \leq z \Rightarrow xy^* \leq z$, it suffices to show

$$c + (bp)^*cbp \leq (bp)^*c.$$

But

$$\begin{aligned} c + (bp)^*cbp &= c + (bp)^*bbcp \leq c + (bp)^*bpc \\ &= (1 + (bp)^*bp)c \leq (bp)^*c. \end{aligned}$$

Deductive Completeness

The KAT axioms exactly characterize the equational theory of

- the standard interpretation $G : \mathrm{RExp}_{\Sigma,T} \to \mathrm{Reg}_{\Sigma,T}$
- all language models
- all relational models
- all trace models

Deductive Completeness

KAT is deductively complete for all relationally valid Hoare-style rules

$$\frac{\{b_1\}\, p_1\, \{c_1\},\ \ldots,\ \{b_n\}\, p_n\, \{c_n\}}{\{b\}\, p\, \{c\}}$$

That is,

$$b_1 p_1 \overline{c}_1 = 0 \wedge \cdots \wedge b_n p_n \overline{c}_n = 0 \ \Rightarrow\ b p \overline{c} = 0$$

In fact, KAT is deductively complete for all Horn formulas with premises of the form $r = 0$:

$$r_1 = 0 \wedge \cdots \wedge r_n = 0 \ \Rightarrow\ p = q$$

This is called the Hoare theory.

Deductive Completeness

Note that PHL is trivially incomplete; e.g.

$$\frac{\{c\}\ \text{if}\ b\ \text{then}\ p\ \text{else}\ p\ \{d\}}{\{c\}\ p\ \{d\}}$$

is not provable in PHL (but

$$c(bp + \overline{b}p)\overline{d} = 0 \ \Rightarrow\ cp\overline{d} = 0$$

is easily provable in KAT)

Completeness of KA and KAT

To show completeness of KA and KAT, encode classical combinatorial constructions of the theory of finite automata algebraically:

- construction of a transition matrix representing a finite automaton equivalent to a given regular expression (Kleene 1956, Conway 1971)
- elimination of ε-transitions (Kuich and Salomaa 1986, Sakarovitch 1987)

Two other fundamental constructions:

- determinization of an automaton via the subset construction, and
- state minimization via equivalence modulo a Myhill-Nerode equivalence relation

Then use the uniqueness of the minimal deterministic finite automaton to obtain completeness

Finite Automata as Matrices (Conway 1971)

A finite automaton over a KA K is represented by a triple $\mathcal{A} = (u, A, v)$, where $u, v \in \{0,1\}^n$ and A is an $n \times n$ matrix over K for some n.

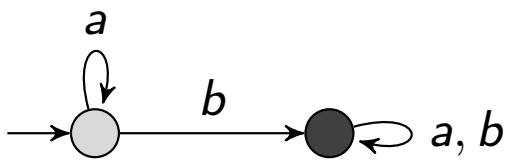
The states are the row and column indices. A start state is an index i for which $u(i) = 1$. A final state is an index i for which $v(i) = 1$. The matrix A is called the transition matrix.

The language accepted by $\mathcal{A}$ is the element $u^T A^* v \in K$.

For automata over the free KA on generators Σ, this is essentially equivalent to the classical combinatorial definition

Example

Consider the two-state automaton



Accepts strings over $\{a, b\}$ containing at least one occurrence of b

$$\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} a & b \\ 0 & a+b \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

Modulo the axioms of KA,

$$\begin{aligned} & \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a & b \\ 0 & a+b \end{bmatrix}^* \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ = \quad & \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a^* & a^* b (a+b)^* \\ 0 & (a+b)^* \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ = \quad & a^* b (a+b)^* \end{aligned}$$

Simple Automata

Definition

Let $\mathcal{A} = (u, A, v)$ be an automaton over $\mathcal{F}_\Sigma$, the free Kleene algebra on free generators Σ. $\mathcal{A}$ is said to be simple if A can be expressed as a sum

$$A \quad = \quad J + \sum_{a \in \Sigma} a \cdot A_a$$

where J and the A_a are 0-1 matrices. In addition, $\mathcal{A}$ is said to be ε-free if J is the zero matrix. Finally, $\mathcal{A}$ is said to be deterministic if it is simple and ε-free, and u and all rows of A_a have exactly one 1.

The automaton of the previous example is simple, ε-free, and deterministic.

Completeness

The first lemma asserts that Kleene's theorem is a theorem of KA.

Lemma

For every regular expression s over Σ (or more accurately, its image in the free KA *under the canonical interpretation), there is a simple automaton (u, A, v) such that*

$$s \quad = \quad u^T A^* v$$

is a theorem of KA*.*

Proof: By induction on the structure of s.

Completeness

For $a \in \Sigma$, the automaton

$$\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

suffices, since

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix}^* \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= a.$$

Completeness

For $s + t$, let $\mathcal{A} = (u, A, v)$ and $\mathcal{B} = (x, B, y)$ be automata such that

$$s = u^T A^* v \qquad t = x^T B^* y.$$

Consider the automaton with transition matrix

$$\left[\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array}\right]$$

and start and final state vectors

$$\left[\begin{array}{c} u \\ \hline x \end{array}\right] \quad \text{and} \quad \left[\begin{array}{c} v \\ \hline y \end{array}\right],$$

respectively. (Corresponds to a disjoint union construction.)

Completeness

Then

$$\left[\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array}\right]^* = \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & B^* \end{array}\right],$$

and

$$\left[\begin{array}{c|c} u^T & x^T \end{array}\right] \cdot \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & B^* \end{array}\right] \cdot \left[\begin{array}{c} v \\ \hline y \end{array}\right]$$

$$= u^T A^* v + x^T B^* y$$

$$= s + t.$$

Completeness

For st, let $\mathcal{A} = (u, A, v)$ and $\mathcal{B} = (x, B, y)$ be automata such that

$$s = u^T A^* v \qquad t = x^T B^* y.$$

Consider the automaton with transition matrix

$$\left[\begin{array}{c|c} A & vx^T \\ \hline 0 & B \end{array}\right]$$

and start and final state vectors

$$\left[\begin{array}{c} u \\ \hline 0 \end{array}\right] \quad \text{and} \quad \left[\begin{array}{c} 0 \\ \hline y \end{array}\right],$$

respectively. (Corresponds to forming the disjoint union and connecting the accept states of $\mathcal{A}$ to the start states of $\mathcal{B}$.)

Completeness

Then

$$\left[\begin{array}{c|c} A & vx^T \\ \hline 0 & B \end{array}\right]^* = \left[\begin{array}{c|c} A^* & A^* vx^T B^* \\ \hline 0 & B^* \end{array}\right],$$

and

$$\begin{aligned} & \left[\begin{array}{c|c} u^T & 0 \end{array}\right] \cdot \left[\begin{array}{c|c} A^* & A^* vx^T B^* \\ \hline 0 & B^* \end{array}\right] \cdot \left[\begin{array}{c} 0 \\ \hline y \end{array}\right] \\ = \; & u^T A^* vx^T B^* y \\ = \; & st. \end{aligned}$$

Completeness

For s^*, let $\mathcal{A} = (u, A, v)$ be an automaton such that $s = u^T A^* v$. First produce an automaton equivalent to the expression ss^*. Consider the automaton

$$(u,\ A + vu^T,\ v).$$

This construction corresponds to the combinatorial construction of adding ε-transitions from the final states of $\mathcal{A}$ back to the start states. Using denesting and sliding,

$$\begin{aligned} u^T(A + vu^T)^* v &= u^T A^* (vu^T A^*)^* v \\ &= u^T A^* v (u^T A^* v)^* \\ &= ss^*. \end{aligned}$$

Once we have an automaton for ss^*, we can get an automaton for $s^* = 1 + ss^*$ by the construction for $+$ given above, using a trivial one-state automaton for 1.

Removing ε-Transitions

This construction models ε-closure.

Lemma

For every simple automaton (u, A, v) over the free KA*, there is a simple ε-free automaton (s, B, t) such that*

$$u^T A^* v \quad = \quad s^T B^* t.$$

Proof.

Write A as a sum $A = J + A'$ where J is 0-1 and A' is ε-free. Then

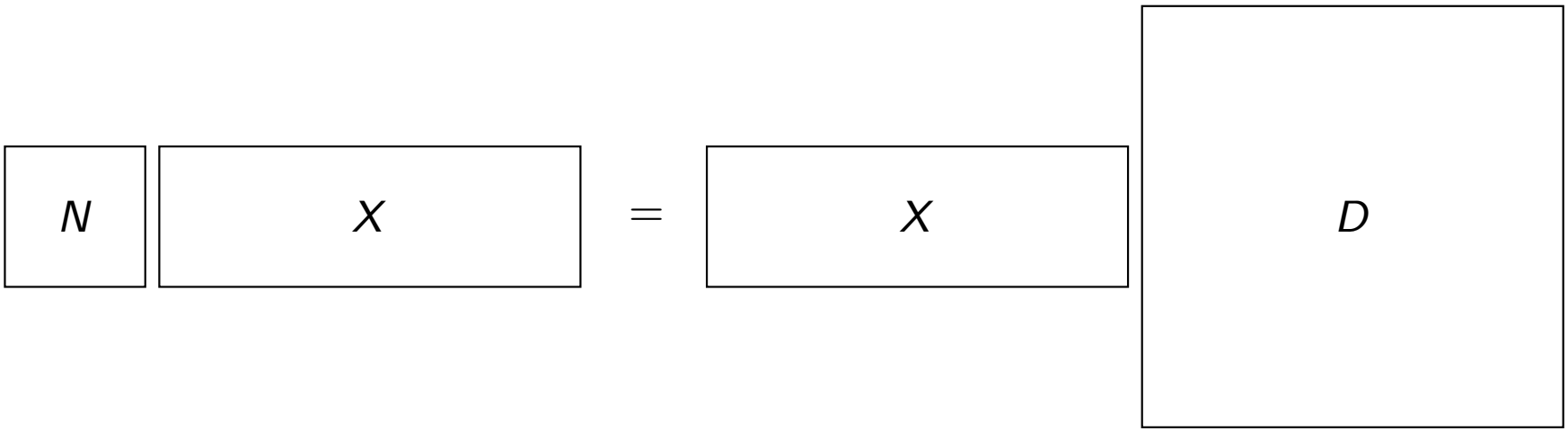
$$u^T A^* v \quad = \quad u^T (A' + J)^* v \quad = \quad u^T J^* (A' J^*)^* v$$

by denesting, so we can take

$$s^T = u^T J^* \qquad\qquad B = A' J^* \qquad\qquad t = v.$$

Then J^* is 0-1 and B is ε-free. □

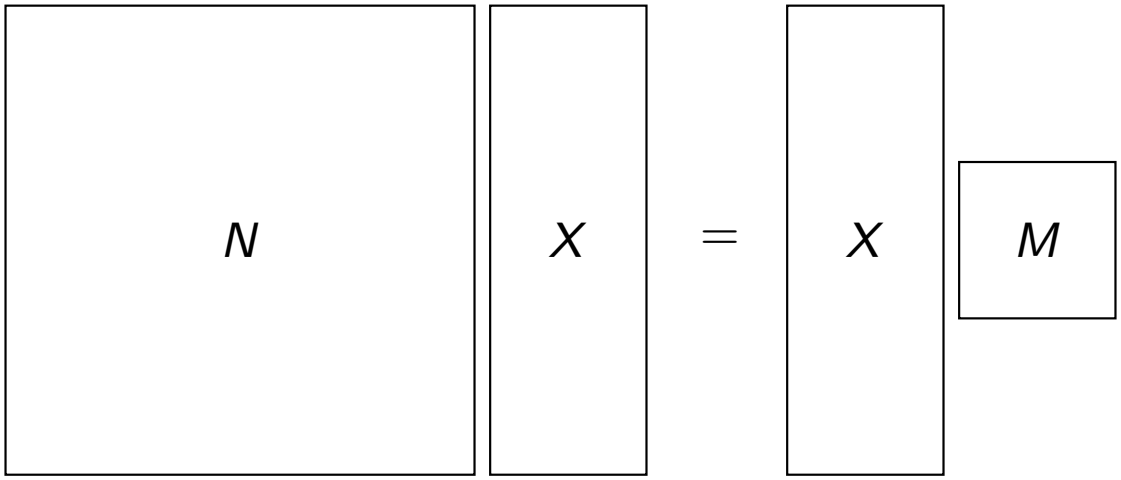
Determinization



$$NX = XD \quad \Rightarrow \quad N^*X = XD^*$$

the bisimulation rule

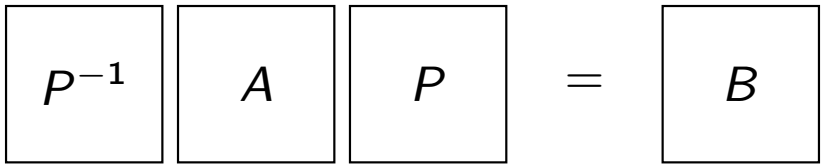
Minimization via a Myhill–Nerode Relation



$$NX = XM \quad \Rightarrow \quad N^*X = XM^*$$

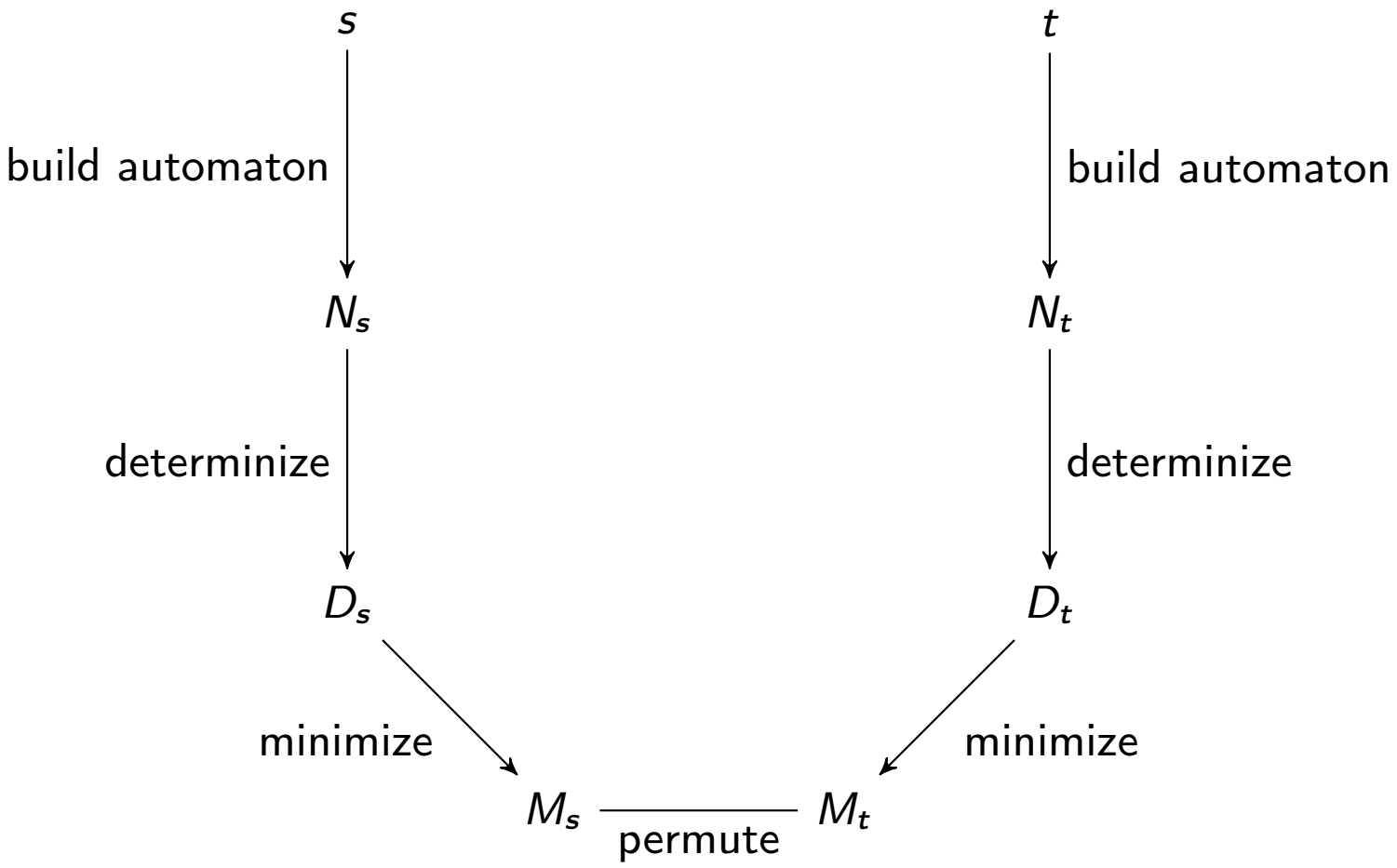
the bisimulation rule again

Isomorphic Automata



$$(P^{-1}AP)^* = P^{-1}A^*P$$

Putting the Steps Together...



Completeness of KAT

Let $\Sigma = \{p_1, \ldots, p_m\}$ and $T = \{b_1, \ldots, b_n\}$. Let $P = p_1 + \cdots + p_m$ and $B = (b_1 + \overline{b}_1) \cdots (b_n + \overline{b}_n)$. A guarded string can be viewed as a string in $(BP)^* B$ over the alphabet $\Sigma \cup T \cup \{\overline{b} \mid b \in T\}$.

Lemma

For every KAT *term* p*, there is a* KAT *term* $\widehat{p}$ *such that*

- $\mathrm{KAT} \vDash p = \widehat{p}$,
- $G(\widehat{p}) = R(\widehat{p})$.

For example,

$$(bpq)^* a \;\mapsto\; ((a + \overline{a})bp(a + b)q)^* a(b + \overline{b})$$

Completeness of KAT

Lemma

For every KAT *term* p*, there is a* KAT *term* $\widehat{p}$ *such that*

- $\mathsf{KAT} \vDash p = \widehat{p}$,
- $G(\widehat{p}) = R(\widehat{p})$.

Theorem

$\mathsf{KAT} \models p = q \quad \Leftrightarrow \quad G(p) = G(q)$.

Proof.

($\Rightarrow$) Immediate, since $\mathsf{GS}_{\Sigma,T}$ is a KAT.
($\Leftarrow$) Suppose $G(p) = G(q)$. Since $\mathsf{KAT} \vDash p = \widehat{p}$ and $\mathsf{GS}_{\Sigma,T}$ is a KAT, $G(\widehat{p}) = G(\widehat{q})$. By the Lemma, $R(\widehat{p}) = R(\widehat{q})$. By the completeness of KA, $\mathsf{KA} \models \widehat{p} = \widehat{q}$. By transitivity, $\mathsf{KAT} \models p = q$. □

Eliminating Assumptions $s = 0$

An ideal of a KA or KAT is a subset $I \subseteq K$ such that

1. $0 \in I$
2. if $x, y \in I$, then $x + y \in I$
3. if $x \in I$ and $r \in K$, then xr and rx are in I
4. if $x \leq y$ and $y \in I$, then $x \in I$.

Given I, set $x \lesssim y$ if $x \leq y + z$ for some $z \in I$, and $x \approx y$ if $x \lesssim y$ and $y \lesssim x$. Equivalently, set $x \approx y$ if $x + z = y + z$ for some $z \in I$, and $x \lesssim y$ if $x + y \approx y$.

$\lesssim$ is a preorder and $\approx$ is an equivalence relation. Let $[x]$ denote the $\approx$-equivalence class of x and let K/I denote the set of all $\approx$-equivalence classes. The relation $\lesssim$ is well-defined on K/I and is a partial order. Note also that $I = [0]$.

Eliminating Assumptions $s = 0$

Theorem

$\approx$ is a KAT *congruence and K/I is a* KAT*. If $A \subseteq K$ and $I = \langle A \rangle$, then K/I is initial among all homomorphic images of K satisfying $x = 0$ for all $x \in A$.*

To show $px \lesssim x \Rightarrow p^*x \lesssim x$:

If $px \lesssim x$, then $px \leq x + z$ for some $z \in I$. Then

$$p(x + p^*z) \;=\; px + pp^*z \;\leq\; x + z + pp^*z \;=\; x + p^*z.$$

Applying the same rule in K, we have $p^*(x + p^*z) \leq x + p^*z$, therefore $p^*x \leq x + p^*z$. Since $p^*z \in I$, $p^*x \lesssim x$.

Eliminating Assumptions $s = 0$

Corollary

Let $\Sigma = \{p_1, \ldots, p_n\}$, $u = (p_1 + \cdots + p_n)^*$. *Then*

$$\mathsf{KAT} \vDash r = 0 \Rightarrow s = t \quad \Leftrightarrow \quad \mathsf{KAT} \vDash s + uru = t + uru.$$

Proof sketch: $\{y \mid y \leq uru\}$ is the ideal generated by r, so $s + uru = t + uru$ iff $s \approx t$ iff $s = t$ in $\mathcal{G}/I$.

Tomorrow

Automata and coalgebras!

Exercises

1. Prove that while b do $(p$; while c do $q) =$
   if b then $(p$; while $b + c$ do if c then q else $p)$ else skip.
2. Prove that the following KAT equations and inequalities are equivalent:
   1. $bp = bpc$
   2. $bp\overline{c} = 0$
   3. $bp \leq pc$
3. Prove that the expression $bp = pc$ is equivalent to the two Hoare partial correctness assertions $\{b\}\, p\, \{c\}$ and $\{\overline{b}\}\, p\, \{\overline{c}\}$.

Exercises

4. Let Σ be a finite alphabet and K a Kleene algebra. A power series in noncommuting variables Σ with coefficients in K is a map $\sigma : \Sigma^* \to K$. The power series σ is often written as a formal sum

$$\sum_{x \in \Sigma^*} \sigma(x) \cdot x.$$

The set of all such power series is denoted $K\langle\langle\Sigma\rangle\rangle$. Addition on $K\langle\langle\Sigma\rangle\rangle$ is defined pointwise, and multiplication is defined as follows:

$$(\sigma \cdot \tau)(x) \stackrel{\text{def}}{=} \sum_{x=yz} \sigma(y) \cdot \tau(z).$$

Define 0 and 1 appropriately and argue that $K\langle\langle\Sigma\rangle\rangle$ forms an idempotent semiring. Then define * as follows:

$$\sigma^*(x) \stackrel{\text{def}}{=} \sum_{x=y_1 \cdots y_n} \sigma(\varepsilon)^* \sigma(y_1) \sigma(\varepsilon)^* \sigma(y_2) \sigma(\varepsilon)^* \cdots \sigma(\varepsilon)^* \sigma(y_n) \sigma(\varepsilon)^*$$

where ε is the null string and the sum is over all ways of expressing x as a product of strings $y_1, \ldots, y_n$. Show that $K\langle\langle\Sigma\rangle\rangle$ forms a KA.

Exercises

5. Strassen's matrix multiplication algorithm can be used to multiply two $n \times n$ matrices over a ring using approximately $n^{\log_2 7} = n^{2.807\ldots}$ multiplications in the underlying ring. The best known result of this form is by Coppersmith and Winograd, who achieve $n^{2.376\ldots}$. Show that over arbitrary semirings, n^3 multiplications are necessary in general. (*Hint*. Interpret over Reg_Σ, where $\Sigma = \{a_{ij}, b_{ij} \mid 1 \leq i, j \leq n\}$. What semiring expressions could possibly be equivalent to $\sum_{j=1}^{n} a_{ij} b_{jk}$?)