Brzozowski Derivatives as Distributive Laws

Dexter Kozen
Cornell University

OWLS
27 May 2020

In Memoriam

Janusz (John) Antoni Brzozowski

10 May 1935 – 24 October 2019

Algebra/Coalgebra Interaction

- Automata/regular expressions [Kleene 56, Silva 10]
- Brzozowski minimization [Brzozowski 64, Adamek et al. 12, Bezhanishvili et al. 12, Bonchi et al. 14]
- Determinization [Bartels 04, Jacobs 06, Silva et al. 10]
- Dynamic logic [Pratt 76]
- Coalgebraic modal logic [Kurz 06, Kupke & Pattinson 11]
- State/predicate transformer duality [Abramsky 91, Bonsangue & Kurz 05]

Q: What is the glue relating the algebraic & coalgebraic structure?

Algebra/Coalgebra Interaction

- Automata/regular expressions [Kleene 56, Silva 10]
- Brzozowski minimization [Brzozowski 64, Adamek et al. 12, Bezhanishvili et al. 12, Bonchi et al. 14]
- Determinization [Bartels 04, Jacobs 06, Silva et al. 10]
- Dynamic logic [Pratt 76]
- Coalgebraic modal logic [Kurz 06, Kupke & Pattinson 11]
- State/predicate transformer duality [Abramsky 91, Bonsangue & Kurz 05]

Q: What is the glue relating the algebraic & coalgebraic structure?

A: A distributive law $\lambda : FG \to GF$

Distributive Laws [Beck 69]

- $F, G : \mathcal{C} \to \mathcal{C}$
- natural transformation $\lambda : FG \to GF$

$$
\begin{array}{ccc}
FGX & \xrightarrow{\lambda_X} & GFX \\
{\scriptstyle FGf}\downarrow & & \downarrow{\scriptstyle GFf} \\
FGY & \xrightarrow[\lambda_Y]{} & GFY
\end{array}
$$

- If F is part of a monad (F, μ, η), also require

$$
\begin{array}{ccccc}
F^2G & \xrightarrow{F\lambda} & FGF & \xrightarrow{\lambda F} & GF^2 \\
{\scriptstyle \mu G}\downarrow & & & & \downarrow{\scriptstyle G\mu} \\
FG & & \xrightarrow[\lambda]{\qquad\qquad} & & GF
\end{array}
\qquad\qquad
\begin{array}{ccc}
& G & \\
{\scriptstyle \eta G}\swarrow & & \searrow{\scriptstyle G\eta} \\
FG & \xrightarrow[\lambda]{} & GF
\end{array}
$$

Distributive Laws [Beck 69]

- originally intended for monad composition
- can lift a G-coalgebra (X, γ) to a G-coalgebra $(FX, \lambda_X \circ F\gamma)$
- can lift an F-algebra (X, α) to an F-algebra $(GX, G\alpha \circ \lambda_X)$
- these are endofunctors

$$\hat{F} : G\text{-Coalg} \to G\text{-Coalg} \qquad \hat{G} : F\text{-Alg} \to F\text{-Alg}$$

$$\begin{array}{ccc} FX & \xrightarrow{\lambda_X \circ F\gamma} & GFX \\ {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle G\alpha} \\ X & \xrightarrow[\gamma]{} & GX \end{array} \qquad \begin{array}{ccc} FGX & \xrightarrow{G\alpha \circ \lambda_X} & GX \\ {\scriptstyle F\gamma}\downarrow & & \downarrow{\scriptstyle \gamma} \\ FX & \xrightarrow[\alpha]{} & X \end{array}$$

F, G-bialgebras [Jacobs 06]

An F, G-bialgebra is a structure (X, α, γ) such that

- (X, α) is an F-algebra
- (X, γ) is a G-coalgebra
- the two structures cohere as expressed by

$$\begin{array}{ccccc} FX & \xrightarrow{\alpha} & X & \xrightarrow{\gamma} & GX \\ {\scriptstyle F\gamma}\downarrow & & & & \uparrow{\scriptstyle G\alpha} \\ FGX & & \xrightarrow[\lambda_X]{} & & GFX \end{array}$$

- α becomes a G-coalgebra morphism $\alpha : \hat{F}(X, \gamma) \to (X, \gamma)$
- γ becomes an F-algebra morphism $\gamma : (X, \alpha) \to \hat{G}(X, \alpha)$

This Talk

- focus on KA-like structures
  - F = variants of regular expressions
  - G = variants of automata
- establish the syntactic Brzozowski derivative as the appropriate distributive law
- a (very) slight generalization of the usual syntactic Brzozowski derivative
- lots of examples!

Lambek's lemma [Lambek 1968]

Lemma

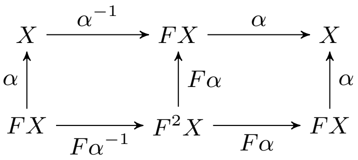
The structure map of an initial F-algebra is invertible. The structure map of a final F-coalgebra is invertible.

Let (X, α) be an initial F-algebra. There is a unique F-algebra morphism $\alpha^{-1}(X, \alpha) \to (FX, F\alpha)$

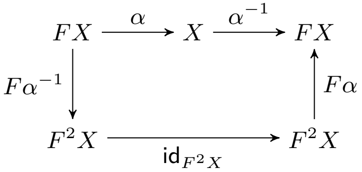
$$\begin{array}{ccccc}
X & \xrightarrow{\alpha^{-1}} & FX & \xrightarrow{\alpha} & X \\
\uparrow{\scriptstyle \alpha} & & \uparrow{\scriptstyle F\alpha} & & \uparrow{\scriptstyle \alpha} \\
FX & \xrightarrow[F\alpha^{-1}]{} & F^2X & \xrightarrow[F\alpha]{} & FX
\end{array}$$

Lambek's lemma [Lambek 1968]

- Key observation: commutativity of the left-hand square



- This is just the bialgebra diagram with $F = G$ and $\lambda_X = \mathrm{id}_{F^2X}$



Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

Ordinary DFA with states X

$$\iota : 1 \to X \qquad \delta_a : X \to X \qquad \varepsilon : X \to 2$$

(X, ε, δ) is a coalgebra for the functor $G = 2 \times (-)^{\Sigma}$

Acceptance

- $\delta : \Sigma \to X \to X$ extends uniquely to a monoid homomorphism $\delta : \Sigma^* \to X \to X$
- for any $w \in \Sigma^*$, $\varepsilon \circ \delta_w \circ \iota : 1 \to 2$
- w is accepted if the value of this function is 1

Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

Nondeterministic automaton: similar, except

$$\iota : 1 \to 2^X \qquad \delta_a : X \to 2^X \qquad \varepsilon : X \to 2$$

(X, ε, δ) is a coalgebra for the functor $GP = 2 \times (P(-))^\Sigma$

Acceptance

- $\delta : \Sigma \to X \to 2^X$ extends uniquely to a monoid homomorphism $\delta : \Sigma^* \to X \to 2^X$ using Kleisli composition $g \bullet f = \mu_X^P \circ Pg \circ f$
- for any $w \in \Sigma^*$, $\varepsilon \bullet \delta_w \bullet \iota : 1 \to 2$
- w is accepted if the value of this function is 1

Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

Classical determinization: subset construction [Rabin & Scott 59]

...which amounts to Kleisli lifting

$$\delta_a : X \to 2^X \quad \Rightarrow \quad \delta_a^\dagger = \mu_X^P \circ P\delta_a : 2^X \to 2^X$$
$$\varepsilon : X \to 2 \quad \Rightarrow \quad \varepsilon^\dagger = \mu_1^P \circ P\varepsilon : 2^X \to 2$$

giving

$$\iota : 1 \to 2^X \qquad \delta_a^\dagger : 2^X \to 2^X \qquad \varepsilon^\dagger : 2^X \to 2$$

$(2^X, \varepsilon^\dagger, \delta^\dagger)$ is a coalgebra for the functor $G = 2 \times (-)^\Sigma$

Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

- The more abstract construction applies to any monad (F, μ, η) on Set
- models an abstract branching structure in the same way the powerset monad models nondeterminism
- many examples in automata theory and coalgebraic modal logic

Let $G = B \times (-)^{\Sigma}$
(B, β) observations, $\beta : FB \to B$
GF-automaton with components

$$\iota : 1 \to FX \qquad \delta_a : X \to FX \qquad \varepsilon : X \to B$$

analog of nondeterministic automata with $F = P$ and $B = 2$

Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

Can determinize by Kleisli lifting to get

$$\iota : 1 \to FX \qquad \delta_a^\dagger : FX \to FX \qquad \varepsilon^\dagger : FX \to B$$

where

$$\delta_a^\dagger = \mu_X \circ F\delta_a \qquad \varepsilon^\dagger : FX \to B$$

$(FX, \delta^\dagger, \varepsilon^\dagger)$ is a G-coalgebra with observations B

Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

What makes this work, and how general is it?

Consider the distributive law $\lambda : FG \to GF$ given by

$$\lambda_Y : F(B \times Y^\Sigma) \xrightarrow{\langle F\pi_1, F\pi_2 \rangle} FB \times F(Y^\Sigma) \xrightarrow{\beta \times \langle F\pi_a \mid a \in \Sigma \rangle} B \times (FY)^\Sigma$$

If Y carries an F-algebra structure $\alpha : FY \to Y$, we have a bialgebra

$$\begin{array}{ccccc}
FY & \xrightarrow{\alpha} & Y & \xrightarrow{\gamma} & B \times Y^\Sigma \\
{\scriptstyle F\gamma}\downarrow & & & & \uparrow{\scriptstyle G\alpha} \\
F(B \times Y^\Sigma) & & \xrightarrow[\lambda_Y]{} & & B \times (FY)^\Sigma
\end{array}$$

Apply with $(Y, \alpha) = (FX, \mu_X)$, $\gamma = (\beta \circ F\varepsilon, \mu_X \circ F\delta_a)$

Determinization [Bartels 04, Jacobs 06, Silva et al. 10]

However, (Y, α) is not limited to free algebras (FX, μ_X); any F-algebra can appear here

Example: alternating automata [Bezhanishvili et al. 20] based on the double contravariant powerset monad

$$\iota : 1 \to 2^{2^X} \qquad \delta_a : X \to 2^{2^X} \qquad \varepsilon : X \to 2$$

determinized to

$$\iota : 1 \to 2^{2^X} \qquad \delta_a^\dagger : 2^{2^X} \to 2^{2^X} \qquad \varepsilon^\dagger : 2^{2^X} \to 2.$$

where

$$\delta_a^\dagger = \mu_X^N \circ N\delta_a \qquad \varepsilon^\dagger = \mu_0^N \circ N\varepsilon$$

Kleene Algebra

Idempotent Semiring Axioms

$$p + (q + r) = (p + q) + r \qquad p(qr) = (pq)r$$
$$p + q = q + p \qquad 1p = p1 = p$$
$$p + 0 = p \qquad p0 = 0p = 0$$
$$p + p = p$$
$$p(q + r) = pq + pr \qquad a \leq b \stackrel{\triangle}{\Longleftrightarrow} a + b = b$$
$$(p + q)r = pr + qr$$

Axioms for *

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

Brzozowski Derivatives [Brzozowski 64, Rutten 99, Silva 10]

A DFA over Σ is a coalgebra for the functor $G = 2 \times (-)^{\Sigma}$

A coalgebra consists of a pair of maps $(\varepsilon, \delta) : X \to GX$

$$\varepsilon : X \to 2 \qquad \delta : X \to X^{\Sigma}$$

observations and actions, respectively

The final coalgebra is the semantic Brzozowski derivative

$$\varepsilon : 2^{\Sigma^*} \to 2 \qquad \delta_a : 2^{\Sigma^*} \to 2^{\Sigma^*}$$

$$\varepsilon(A) = [\varepsilon \in A]^1 \qquad \delta_a(A) = \{x \mid ax \in A\}$$

[1] Iverson bracket: $[\varphi]$ = 1 if φ is true, 0 otherwise

Brzozowski Derivatives [Brzozowski 64, Rutten 99, Silva 10]

$$E : \mathsf{Exp}_\Sigma \to 2 \qquad D_a : \mathsf{Exp}_\Sigma \to \mathsf{Exp}_\Sigma,\ a \in \Sigma$$

$$\begin{aligned} E(e_1 + e_2) &= E(e_1) + E(e_2) & D_a(e_1 + e_2) &= D_a(e_1) + D_a(e_2) \\ E(e_1 e_2) &= E(e_1) \cdot E(e_2) & D_a(e_1 e_2) &= D_a(e_1)e_2 + E(e_1)D_a(e_2) \\ E(e^*) &= 1 & D_a(e^*) &= D_a(e)e^* \\ E(1) &= 1 & D_a(1) &= D_a(0) = 0 \\ E(0) &= E(a) = 0,\ a \in \Sigma & D_a(b) &= [b = a],\ a, b \in \Sigma \end{aligned}$$

- this is a coalgebra $\mathsf{Exp}_\Sigma \to G(\mathsf{Exp}_\Sigma)$
- $L(e) = \{$language represented by $e\}$ is the unique coalgebra morphism $L : \mathsf{Exp}_\Sigma \to 2^{\Sigma^*}$
- used in Brzozowski's proof of Kleene's theorem

KA Bialgebras

To relate KA and finite automata bialgebraically:

- $F = \mathsf{Exp}_\Sigma$, where $\mathsf{Exp}_\Sigma\, X$ is the set of regular expressions over primitive actions X with constant actions Σ
- $G = 2 \times (-)^\Sigma$, the coalgebraic signature of ordinary DFAs

Distributive law:
(a slight generalization of) the syntactic Brzozowski derivative

$$\mathsf{Brz} : \mathsf{Exp}_\Sigma(2 \times (-)^\Sigma) \to 2 \times (\mathsf{Exp}_\Sigma(-))^\Sigma$$

The traditional Brzozowski derivative is $\mathsf{Brz}_\varnothing$

KA Bialgebras

$$\mathsf{Brz} : \mathsf{Exp}_\Sigma(2 \times (-)^\Sigma) \to 2 \times (\mathsf{Exp}_\Sigma(-))^\Sigma$$

usually presented in curried form

$E : \mathsf{Exp}_\Sigma(2 \times (-)^\Sigma) \to 2$ $\qquad D_p : \mathsf{Exp}_\Sigma(2 \times (-)^\Sigma) \to \mathsf{Exp}_\Sigma(-),\ p \in \Sigma$

$E(e_1 + e_2) = E(e_1) + E(e_2)$	$D_p(e_1 + e_2) = D_p(e_1) + D_p(e_2)$
$E(e_1 e_2) = E(e_1)E(e_2)$	$D_p(e_1 e_2) = D_p(e_1)e_2 + E(e_1)D_p(e_2)$
$E(e^*) = 1$	$D_p(e^*) = D_p(e)e^*$
$E(0) = E(p) = 0$	$D_p(0) = D_p(1) = 0$
$E(1) = 1$	$D_p(q) = [p = q]$
$E(i, f) = i$	$D_p(i, f) = f(p)$

where $p, q \in \Sigma$ and $(i, f) \in 2 \times X^\Sigma$

KA Bialgebras

The bialgebra diagram becomes

$$
\begin{array}{ccccc}
\mathsf{Exp}_\Sigma X & \xrightarrow{\ \alpha\ } & X & \xrightarrow{\ (\varepsilon,\delta)\ } & 2 \times X^\Sigma \\
{\scriptstyle (-)[(\varepsilon(x),\delta(x))/x]}\big\downarrow & & & & \big\uparrow{\scriptstyle \mathsf{id}_2 \times (\alpha \circ -)} \\
\mathsf{Exp}_\Sigma(2 \times X^\Sigma) & & \xrightarrow[\ \mathsf{Brz}_X\ ]{} & & 2 \times (\mathsf{Exp}_\Sigma X)^\Sigma
\end{array}
$$

Intuitively,

- if you give me a regular expression $e \in \mathsf{Exp}_\Sigma X$ and tell me how to perform derivatives on elements of X using some $(\varepsilon, \delta) : X \to 2 \times X^\Sigma$, then ...
- I will tell you how to get the derivative of e by substituting $(\varepsilon(x), \delta(x))$ for x in e to get $e' \in \mathsf{Exp}_\Sigma(2 \times X^\Sigma)$, then applying the traditional Brzozowski derivative to e'.

KA Bialgebras

Examples

- Reg_Σ, the family of regular subsets of Σ^*

$$\begin{array}{ccccc}
\mathsf{Exp}_\Sigma\,\mathsf{Reg}_\Sigma & \xrightarrow{\alpha} & \mathsf{Reg}_\Sigma & \xrightarrow{(\varepsilon,\delta)} & 2\times(\mathsf{Reg}_\Sigma)^\Sigma \\
\Big\downarrow{\scriptstyle (-)[(\delta(A),\varepsilon(A))/A]} & & & & \Big\uparrow{\scriptstyle \mathsf{id}_2\times(\alpha\circ -)} \\
\mathsf{Exp}_\Sigma(2\times(\mathsf{Reg}_\Sigma)^\Sigma) & & \xrightarrow[\mathsf{Brz}_{\mathsf{Reg}_\Sigma}]{} & & 2\times(\mathsf{Exp}_\Sigma\,\mathsf{Reg}_\Sigma)^\Sigma
\end{array}$$

- 2^{Σ^*}, the final coalgebra

$$\begin{array}{ccccc}
\mathsf{Exp}_\Sigma\,2^{\Sigma^*} & \xrightarrow{\alpha} & 2^{\Sigma^*} & \xrightarrow{(\varepsilon,\delta)} & 2\times(2^{\Sigma^*})^\Sigma \\
\Big\downarrow{\scriptstyle (-)[(\varepsilon(A),\delta(A))/A]} & & & & \Big\uparrow{\scriptstyle \mathsf{id}_2\times(\alpha\circ -)} \\
\mathsf{Exp}_\Sigma(2\times(2^{\Sigma^*})^\Sigma) & & \xrightarrow[\mathsf{Brz}_{2^{\Sigma^*}}]{} & & 2\times(\mathsf{Exp}_\Sigma\,2^{\Sigma^*})^\Sigma
\end{array}$$

KA Bialgebras

Here $(\varepsilon, \delta) : 2^{\Sigma^*} \to 2 \times (2^{\Sigma^*})^{\Sigma}$ is the semantic Brzozowski derivative

$$\begin{aligned} &\varepsilon : 2^{\Sigma^*} \to 2 && \delta_p : 2^{\Sigma^*} \to 2^{\Sigma^*} \\ &\varepsilon(A) = [\varepsilon \in A] && \delta_p(A) = \{x \in \Sigma^* \mid px \in A\} \end{aligned}$$

and α is the usual evaluation function

$$\begin{aligned} &\alpha(e_1 + e_2) = \alpha(e_1) \cup \alpha(e_2) && \alpha(0) = \varnothing \\ &\alpha(e_1 e_2) = \{xy \mid x \in \alpha(e_1),\ y \in \alpha(e_2)\} && \alpha(1) = \{\varepsilon\} \\ &\alpha(e^*) = \bigcup_n \alpha(e^n) && \alpha(p) = \{p\} \\ & && \alpha(A) = A \end{aligned}$$

To check that Brz is a distributive law …

$$
\begin{array}{ccc}
\mathsf{Exp}_\Sigma(2 \times X^\Sigma) & \xrightarrow{\mathsf{Brz}_X} & 2 \times (\mathsf{Exp}_\Sigma\, X)^\Sigma \\
{\scriptstyle (-)[f(x)/x]}\downarrow & & \downarrow{\scriptstyle (-)[f(x)/x]} \\
\mathsf{Exp}_\Sigma(2 \times Y^\Sigma) & \xrightarrow[\mathsf{Brz}_Y]{} & 2 \times (\mathsf{Exp}_\Sigma\, Y)^\Sigma
\end{array}
$$

$$
\begin{array}{ccccc}
\mathsf{Exp}_\Sigma(\mathsf{Exp}_\Sigma(2 \times X^\Sigma)) & \xrightarrow{F\ \mathsf{Brz}_X} & \mathsf{Exp}_\Sigma(2 \times (\mathsf{Exp}_\Sigma\, X)^\Sigma) & \xrightarrow{\mathsf{Brz}_{FX}} & 2 \times (\mathsf{Exp}_\Sigma(\mathsf{Exp}_\Sigma\, X))^\Sigma \\
{\scriptstyle \mu G}\downarrow & & & & \downarrow{\scriptstyle G\mu} \\
\mathsf{Exp}_\Sigma(2 \times X^\Sigma) & & \xrightarrow[\mathsf{Brz}_X]{} & & 2 \times (\mathsf{Exp}_\Sigma\, X)^\Sigma
\end{array}
$$

$$
\begin{array}{ccc}
 & 2 \times X^\Sigma & \\
{\scriptstyle \eta_{GX}}\swarrow & & \searrow{\scriptstyle G\eta_X} \\
\mathsf{Exp}_\Sigma(2 \times X^\Sigma) & \xrightarrow[\mathsf{Brz}_X]{} & 2 \times (\mathsf{Exp}_\Sigma\, X)^\Sigma
\end{array}
$$

Kleene Algebra with Tests (KAT)

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

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

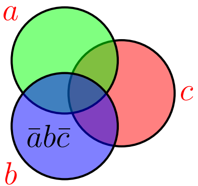
- encodes imperative programming constructs
- subsumes Hoare logic

$p; q$	pq
$\texttt{if}\ b\ \texttt{then}\ p\ \texttt{else}\ q$	$bp + \bar{b}q$
$\texttt{while}\ b\ \texttt{do}\ p$	$(bp)^*\bar{b}$
$\{b\}\ p\ \{c\}$	$bp \leq pc,\ bp = bpc,\ bp\bar{c} = 0$
$\dfrac{\{bc\}\ p\ \{c\}}{\{c\}\ \texttt{while}\ b\ \texttt{do}\ p\ \{\bar{b}c\}}$	$bcp\bar{c} = 0\ \Rightarrow\ (c(bp)^*\bar{b})\,\bar{\bar{b}} = 0$

Guarded Strings [Kaplan 69]

Σ action symbols $\quad T$ test symbols

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



Guarded strings GS $=$ At $\cdot$ $(\Sigma \cdot \text{At})^*$

$$\alpha_0 p_1 \alpha_1 p_2 \alpha_2 \cdots \alpha_{n-1} p_n \alpha_n$$

Standard Language Model for KAT

Regular sets of guarded strings over Σ, T

For $A, B \subseteq$ GS,

$$A + B = A \cup B \qquad AB = \{x\alpha y \mid x\alpha \in A,\ \alpha y \in B\}$$

$$A^* = \bigcup_{n \geq 0} A^n \ = \ A^0 \cup A^1 \cup A^2 \cup \cdots$$

$$1 = \mathsf{At} \qquad 0 = \varnothing$$

- $p \in \Sigma$ interpreted as $\{\alpha p \beta \mid \alpha, \beta \in \mathsf{At}\}$
- $b \in T$ interpreted as $\{\alpha \mid \alpha \leq b\}$

The regular subsets of GS forms the free KAT on generators Σ, T

KAT Coalgebras

KAT automata = automata on guarded strings
coalgebras for the functor $G = 2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}$

$$\varepsilon : X \to 2^{\mathsf{At}} \qquad\qquad \delta : X \to X^{\mathsf{At}\times\Sigma}$$

The final coalgebra is

$$\varepsilon : 2^{\mathsf{GS}} \to 2^{\mathsf{At}} \qquad\qquad \delta : 2^{\mathsf{GS}} \to (2^{\mathsf{GS}})^{\mathsf{At}\times\Sigma}$$

$$\varepsilon_\alpha(A) = [\alpha \in A] \qquad\qquad \delta_{\alpha p}(A) = \{x \mid \alpha p x \in A\}$$

This is the semantic Brzozowski derivative

KAT Bialgebras

Functors:

- $F = \mathsf{Exp}_{\Sigma,B}$, where $\mathsf{Exp}_{\Sigma,B}\, X$ = KAT expressions over indeterminate actions X, constant actions Σ, tests B
- $G = 2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}$, the signature of automata on guarded strings

Distributive law: the syntactic Brzozowski derivative

$$\mathsf{Brz} : \mathsf{Exp}_{\Sigma,B}(2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}) \to 2^{\mathsf{At}} \times (\mathsf{Exp}_{\Sigma,B}(-))^{\mathsf{At}\times\Sigma}$$

usually presented in curried form

$$E_\alpha : \mathsf{Exp}_{\Sigma,B}(2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}) \to 2$$
$$D_{\alpha p} : \mathsf{Exp}_{\Sigma,B}(2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}) \to \mathsf{Exp}_{\Sigma,B}(-)$$

for $\alpha \in \mathsf{At}$ and $p \in \Sigma$

KAT Bialgebras

$$E_\alpha : \mathsf{Exp}_{\Sigma,B}(2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}) \to 2$$
$$D_{\alpha p} : \mathsf{Exp}_{\Sigma,B}(2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}) \to \mathsf{Exp}_{\Sigma,B}(-)$$

$E_\alpha(e_1 + e_2) = E_\alpha(e_1) + E_\alpha(e_2)$	$E_\alpha(0) = E_\alpha(p) = 0$
$E_\alpha(e_1 e_2) = E_\alpha(e_1) E_\alpha(e_2)$	$E_\alpha(1) = 1$
$E_\alpha(e^*) = 1$	$E_\alpha(i, f) = i(\alpha)$
$D_{\alpha p}(e_1 + e_2) = D_{\alpha p}(e_1) + D_{\alpha p}(e_2)$	$D_{\alpha p}(0) = D_{\alpha p}(1) = 0$
$D_{\alpha p}(e_1 e_2) = D_{\alpha p}(e_1) e_2 + E_\alpha(e_1) D_{\alpha p}(e_2)$	$D_{\alpha p}(q) = [p = q]$
$D_{\alpha p}(e^*) = D_{\alpha p}(e) e^*$	$D_{\alpha p}(i, f) = f(\alpha p)$

where $p, q \in \Sigma$ and $(i, f) \in 2^{\mathsf{At}} \times (-)^{\mathsf{At}\times\Sigma}$

Examples

- final coalgebra 2^{GS}

$$
\begin{array}{ccccc}
\mathsf{Exp}_{\Sigma,B}\, 2^{\mathsf{GS}} & \xrightarrow{\sigma} & 2^{\mathsf{GS}} & \xrightarrow{(\varepsilon,\delta)} & 2^{\mathsf{At}} \times (2^{\mathsf{GS}})^{\mathsf{At}\times\Sigma} \\
\Big\downarrow {\scriptstyle (-)[(\varepsilon(A),\delta(A))/A]} & & & & \Big\uparrow {\scriptstyle \mathsf{id}_{2^{\mathsf{At}}} \times (\sigma \circ -)} \\
\mathsf{Exp}_{\Sigma,B}(2^{\mathsf{At}} \times (2^{\mathsf{GS}})^{\mathsf{At}\times\Sigma}) & & \xrightarrow[\mathsf{Brz}_{2^{\mathsf{GS}}}]{} & & 2^{\mathsf{At}} \times (\mathsf{Exp}_{\Sigma,B}\, 2^{\mathsf{GS}})^{\mathsf{At}\times\Sigma}
\end{array}
$$

- Reg_Σ = regular subsets of GS

$$
\begin{array}{ccccc}
\mathsf{Exp}_{\Sigma}\, \mathsf{Reg}_\Sigma & \xrightarrow{\sigma} & \mathsf{Reg}_\Sigma & \xrightarrow{(\varepsilon,\delta)} & 2^{\mathsf{At}} \times (\mathsf{Reg}_\Sigma)^{\Sigma} \\
\Big\downarrow {\scriptstyle (-)[(\delta(A),\varepsilon(A))/A]} & & & & \Big\uparrow {\scriptstyle \mathsf{id}_{2^{\mathsf{At}}} \times (\sigma \circ -)} \\
\mathsf{Exp}_{\Sigma}(2^{\mathsf{At}} \times (\mathsf{Reg}_\Sigma)^{\Sigma} \times 2) & & \xrightarrow[\mathsf{Brz}_{\mathsf{Reg}_\Sigma}]{} & & 2^{\mathsf{At}} \times (\mathsf{Exp}_{\Sigma}\, \mathsf{Reg}_\Sigma)^{\Sigma}
\end{array}
$$

KAT Bialgebras

$(\varepsilon, \delta) : 2^{\mathsf{GS}} \to 2^{\mathsf{At}} \times (2^{\mathsf{GS}})^{\mathsf{At} \times \Sigma}$ is the semantic Brzozowski derivative, where

$$\begin{aligned} &\varepsilon_\alpha : 2^{\mathsf{GS}} \to 2 && \delta_{\alpha p} : 2^{\mathsf{GS}} \to 2^{\mathsf{GS}} \\ &\varepsilon_\alpha(A) = [\alpha \in A] && \delta_{\alpha p}(A) = \{x \in \Sigma^* \mid \alpha p x \in A\} \end{aligned}$$

σ is the usual evaluation function on regular expressions over subsets of GS

$$\begin{aligned} &\sigma(e_1 + e_2) = \sigma(e_1) \cup \sigma(e_2) && \sigma(0) = \varnothing \\ &\sigma(e_1 e_2) = \{x\alpha y \mid x\alpha \in \sigma(e_1),\ \alpha y \in \sigma(e_2)\} && \sigma(1) = \mathsf{At} \\ &\sigma(e^*) = \bigcup_n \sigma(e^n) && \sigma(A) = A \\ &\sigma(p) = \{\alpha p \beta \mid \alpha, \beta \in \mathsf{At}\} \end{aligned}$$

$\sigma(e) = \{$language represented by $e\}$ is the unique coalgebra morphism $e : \mathsf{Exp} \to 2^{\mathsf{GS}}$

NetKAT [Anderson et al. 2013]

A programming language/logic for programmable networks

- primitives for modifying and filtering on packet header values, duplicating and dropping packets
- duplication $(+)$, sequential composition $(\cdot)$, iteration $({}^*)$
- can specify network topology and routing, end-to-end behavior, access control
- integrated as part of the Frenetic suite of network management tools [Foster et al. 10]

NetKAT Axioms

Actions $x := n$, tests $x = n$

- $x := n; y := m \equiv y := m; x := n \;\; (x \neq y)$
- $x := n; y = m \equiv y = m; x := n \;\; (x \neq y)$
- $x = n; \textit{dup} \equiv \textit{dup}; x = n$
- $x := n; x = n \equiv x := n$
- $x = n; x := n \equiv x = n$
- $x := n; x := m \equiv x := m$
- $x = n; x = m \equiv \textit{drop} \;\; (n \neq m)$
- $(\sum_n x = n) \equiv \textit{skip}$

Reduced Axioms

Actions $p \in P$, atoms $\alpha \in \mathsf{At}$

- $p = (x_1 := n_1; \cdots ; x_k := n_k)$
- $\alpha_p = (x_1 = n_1; \cdots ; x_k = n_k)$

- $\alpha\, \textit{dup} \equiv \textit{dup}\, \alpha$
- $p\alpha_p = p$
- $\alpha_p p = \alpha_p$
- $qp = p$

Standard Model

Standard model of NetKAT is a packet-forwarding model

$$[\![e]\!] : H \to 2^H$$

where $H = \{\text{packet traces}\}$

- $+$ is conjunctive
- sequential composition is Kleisli composition

Remarkably, satisfies all the KAT axioms!

Language Model

Regular sets of NetKAT reduced strings

$$\mathsf{NS} = \mathsf{At} \cdot P \cdot (\mathit{dup} \cdot P)^* \qquad \alpha\, p_0\, \mathit{dup}\, p_1\, \mathit{dup} \cdots \mathit{dup}\, p_n$$

For $A, B \subseteq \mathsf{NS}$,

$$A + B = A \cup B \qquad AB = \{\alpha x y q \mid \alpha x p \in A,\ \alpha_p y q \in B\}$$

$$A^* = \bigcup_{n \geq 0} A^n \qquad 1 = \{\alpha_p p \mid p \in P\} \qquad 0 = \varnothing$$

- $p \in P$ interpreted as $\sum_\alpha \alpha p$
- $\alpha \in \mathsf{At}$ interpreted as αp_α
- *dup* interpreted as $\sum_p \alpha_p p\, \mathit{dup}\, \alpha_p$

This is the free NetKAT on its generating set

NetKAT Coalgebra [Foster et al. 14]

NetKAT automata/coalgebras are coalgebras for the functor $G = 2^{\mathsf{At}\times\mathsf{At}} \times (-)^{\mathsf{At}\times\mathsf{At}}$

$$\varepsilon : S \to 2^{\mathsf{At}\times\mathsf{At}} \qquad \delta : S \to S^{\mathsf{At}\times\mathsf{At}}$$

The final coalgebra is

$$\varepsilon : 2^{\mathsf{NS}} \to 2^{\mathsf{At}\times\mathsf{At}} \qquad \delta : 2^{\mathsf{NS}} \to (2^{\mathsf{NS}})^{\mathsf{At}\times\mathsf{At}}$$

$$\varepsilon_{\alpha\beta}(A) = [\alpha p_\beta \in A] \qquad \delta_{\alpha\beta}(A) = \{\beta x \mid \alpha p_\beta\, \textit{\textbf{dup}}\, x \in A\}$$

NetKAT Bialgebras

Functors

- $F = \mathsf{NExp}_{P,B}$ = NetKAT expressions over indeterminate actions X, constant actions P, tests B
- $G = 2^{\mathsf{At}\times\mathsf{At}} \times (-)^{\mathsf{At}\times\mathsf{At}}$, the signature of NetKAT automata

Distributive law: the syntactic Brzozowski derivative

$$\mathsf{Brz} : \mathsf{NExp}_{P,B}(2^{\mathsf{At}\times\mathsf{At}} \times (-)^{\mathsf{At}\times\mathsf{At}}) \to 2^{\mathsf{At}\times\mathsf{At}} \times (\mathsf{NExp}_{P,B}(-))^{\mathsf{At}\times\mathsf{At}}$$

$$E_{\alpha\beta} : \mathsf{NExp}_{P,B}(2^{\mathsf{At}\times\mathsf{At}} \times (-)^{\mathsf{At}\times\mathsf{At}}) \to 2$$
$$D_{\alpha\beta} : \mathsf{NExp}_{P,B}(2^{\mathsf{At}\times\mathsf{At}} \times (-)^{\mathsf{At}\times\mathsf{At}}) \to \mathsf{NExp}_{P,B}(-)$$

for $\alpha, \beta \in \mathsf{At}$

NetKAT Bialgebras

$$E_{\alpha\beta}(p) = [p = p_\beta] \qquad D_{\alpha\beta}(p) = 0$$
$$E_{\alpha\beta}(b) = [\alpha = \beta \leq b] \qquad D_{\alpha\beta}(b) = 0$$
$$E_{\alpha\beta}(\textit{dup}) = 0 \qquad D_{\alpha\beta}(\textit{dup}) = \alpha \cdot [\alpha = \beta]$$
$$E_{\alpha\beta}(g, f) = g(\alpha, \beta) \qquad D_{\alpha\beta}(g, f) = f(\alpha, \beta)$$

where $p \in P$, $b \in B$, and $(g, f) \in 2^{\mathsf{At}\times\mathsf{At}} \times X^{\mathsf{At}\times\mathsf{At}}$

$$E_{\alpha\beta}(e_1 + e_2) = E_{\alpha\beta}(e_1) + E_{\alpha\beta}(e_2)$$
$$E_{\alpha\beta}(e_1 e_2) = \textstyle\sum_\gamma E_{\alpha\gamma}(e_1) \cdot E_{\gamma\beta}(e_2)$$
$$E_{\alpha\beta}(e^*) = [\alpha = \beta] + \textstyle\sum_\gamma E_{\alpha\gamma}(e) \cdot E_{\gamma\beta}(e^*)$$
$$D_{\alpha\beta}(e_1 + e_2) = D_{\alpha\beta}(e_1) + D_{\alpha\beta}(e_2)$$
$$D_{\alpha\beta}(e_1 e_2) = D_{\alpha\beta}(e_1) \cdot e_2 + \textstyle\sum_\gamma E_{\alpha\gamma}(e_1) \cdot D_{\gamma\beta}(e_2)$$
$$D_{\alpha\beta}(e^*) = D_{\alpha\beta}(e) \cdot e^* + \textstyle\sum_\gamma E_{\alpha\gamma}(e) \cdot D_{\gamma\beta}(e^*)$$

NetKAT Bialgebras

$$E_{\alpha\beta}(p) = [p = p_\beta] \qquad D_{\alpha\beta}(p) = 0$$
$$E_{\alpha\beta}(b) = [\alpha = \beta \leq b] \qquad D_{\alpha\beta}(b) = 0$$
$$E_{\alpha\beta}(\textit{dup}) = 0 \qquad D_{\alpha\beta}(\textit{dup}) = \alpha \cdot [\alpha = \beta]$$
$$E_{\alpha\beta}(g, f) = g(\alpha, \beta) \qquad D_{\alpha\beta}(g, f) = f(\alpha, \beta)$$

where $p \in P$, $b \in B$, and $(g, f) \in 2^{\mathsf{At}\times\mathsf{At}} \times X^{\mathsf{At}\times\mathsf{At}}$

$$E_{\alpha\beta}(e_1 + e_2) = E_{\alpha\beta}(e_1) + E_{\alpha\beta}(e_2)$$
$$E_{\alpha\beta}(e_1 e_2) = \sum_\gamma E_{\alpha\gamma}(e_1) \cdot E_{\gamma\beta}(e_2)$$
$$E_{\alpha\beta}(e^*) = [\alpha = \beta] + \sum_\gamma E_{\alpha\gamma}(e) \cdot E_{\gamma\beta}(e^*) \quad \text{circular!}$$
$$D_{\alpha\beta}(e_1 + e_2) = D_{\alpha\beta}(e_1) + D_{\alpha\beta}(e_2)$$
$$D_{\alpha\beta}(e_1 e_2) = D_{\alpha\beta}(e_1) \cdot e_2 + \sum_\gamma E_{\alpha\gamma}(e_1) \cdot D_{\gamma\beta}(e_2)$$
$$D_{\alpha\beta}(e^*) = D_{\alpha\beta}(e) \cdot e^* + \sum_\gamma E_{\alpha\gamma}(e) \cdot D_{\gamma\beta}(e^*) \quad \text{circular!}$$

NetKAT Bialgebras

Use matrix operations on At $\times$ At matrices! [Foster et al. 15]

$$
\begin{aligned}
E(e_1 + e_2) &= E(e_1) + E(e_2) \\
E(e_1 e_2) &= E(e_1) \cdot E(e_2) \\
E(e^*) &= I(1) + E(e) \cdot E(e^*) \\
D(e_1 + e_2) &= D(e_1) + D(e_2) \\
D(e_1 e_2) &= D(e_1) \cdot I(e_2) + E(e_1) \cdot D(e_2) \\
D(e^*) &= D(e) \cdot I(e^*) + E(e) \cdot D(e^*)
\end{aligned}
$$

so for $E(e^*)$ and $D(e^*)$ we can take

$$
E(e^*) = E(e)^* \qquad D(e^*) = E(e)^* \cdot D(e) \cdot I(e^*)
$$

NetKAT Bialgebras

$$
\begin{array}{ccc}
\mathsf{NExp}_{P,B}\, 2^{\mathsf{NS}} \xrightarrow{\ \sigma\ } 2^{\mathsf{NS}} & \xrightarrow{(\varepsilon,\delta)} & 2^{\mathsf{At}\times\mathsf{At}} \times (2^{\mathsf{NS}})^{\mathsf{At}\times\mathsf{At}} \\
{\scriptstyle (-)[(\varepsilon(A),\delta(A))/A]}\Big\downarrow & & \Big\uparrow {\scriptstyle \mathsf{id}_{2^{\mathsf{At}\times\mathsf{At}}} \times (\sigma \circ -)^{\mathsf{At}\times\mathsf{At}}} \\
\mathsf{NExp}_{P,B}(2^{\mathsf{At}\times\mathsf{At}} \times (2^{\mathsf{NS}})^{\mathsf{At}\times\mathsf{At}}) & \xrightarrow[\mathsf{Brz}_{2^{\mathsf{NS}}}]{} & 2^{\mathsf{At}\times\mathsf{At}} \times (\mathsf{NExp}_{P,B}\, 2^{\mathsf{NS}})^{\mathsf{At}\times\mathsf{At}}
\end{array}
$$

$(\varepsilon, \delta) : 2^{\mathsf{NS}} \to 2^{\mathsf{At}\times\mathsf{At}} \times (2^{\mathsf{NS}})^{\mathsf{At}\times\mathsf{At}}$ is the semantic derivative

$$\varepsilon(A)_{\alpha\beta} = [\alpha p_\beta \in A] \qquad \delta(A)_{\alpha\beta} = \{\beta x \mid \alpha p_\beta\ \textit{dup}\ x \in A\}$$

$\sigma : \{\text{NetKAT expressions}\} \to 2^{\mathsf{NS}}$ is the evaluation function

GKAT [Smolka et al. 20]

Guarded KAT (GKAT) restricts KAT to guarded versions of + and *

$$p +_b q \qquad \texttt{if}\ b\ \texttt{then}\ p\ \texttt{else}\ q$$

$$p^{(b)} \qquad \texttt{while}\ b\ \texttt{do}\ p$$

- almost linear time decidability
- Kleene theorem
- completeness over a coequationally-defined language model
- coalgebraic theory

GKAT Automata/Coalgebras

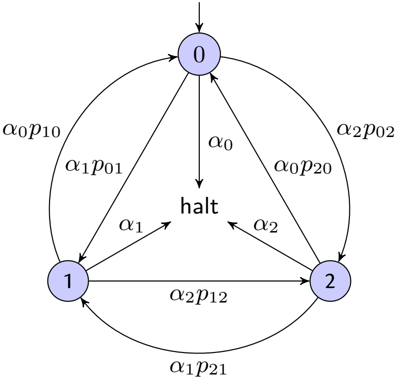
Strictly deterministic automata = coalgebras for the functor $G = (2 + \Sigma \times (-))^{\mathsf{At}}$

Intuitively $\gamma : X \to (2 + \Sigma \times X)^{\mathsf{At}}$ operates as follows:

- atoms $\alpha \in \mathsf{At}$ come in from the environment
- the program responds by either
  - performing an action p and moving to a new state ($\gamma(s)(\alpha) = (p, s)$)
  - halting and accepting ($\gamma(s)(\alpha) = 1$)
  - halting and rejecting ($\gamma(s)(\alpha) = 0$)

A Counterexample [Kozen & Tseng 08]

All GKAT expressions correspond to automata, but not vice versa



GKAT Bialgebras

Functors

- $F = \mathsf{GExp}_{\Sigma,B}$, where $\mathsf{GExp}_{\Sigma,B}\, X$ = GKAT expressions with operators $e_1 \,;\, e_2$, $e_1 +_b e_2$, and $e^{(b)}$ over indeterminate actions X, constant actions Σ, tests B
- $G = (2 + \Sigma \times (-))^{\mathsf{At}}$

Distributive law: syntactic Brzozowski derivative

$$\mathsf{Brz} : \mathsf{GExp}_{\Sigma,B}((2 + \Sigma \times (-))^{\mathsf{At}}) \to (2 + \Sigma \times \mathsf{GExp}_{\Sigma,B}(-))^{\mathsf{At}},$$

$$D_\alpha : \mathsf{GExp}_{\Sigma,B}((2 + \Sigma \times (-))^{\mathsf{At}}) \to 2 + \Sigma \times \mathsf{GExp}_{\Sigma,B}(-)$$

for $\alpha \in \mathsf{At}$

GKAT Bialgebras

$$D_\alpha(e_1 +_b e_2) = \begin{cases} D_\alpha(e_1), & \alpha \leq b \\ D_\alpha(e_2), & \alpha \leq \bar{b} \end{cases}$$

$$D_\alpha(e_1 e_2) = \begin{cases} (p, e_1' e_2), & D_\alpha(e_1) = (p, e_1') \\ D_\alpha(e_2), & D_\alpha(e_1) = 1 \\ 0, & D_\alpha(e_1) = 0 \end{cases}$$

$$D_\alpha(e^{(b)}) = \begin{cases} (p, e' e^{(b)}), & \alpha \leq b \wedge D_\alpha(e) = (p, e') \\ 0, & \alpha \leq b \wedge D_\alpha(e) \in 2 \\ 1, & \alpha \leq \bar{b} \end{cases}$$

$$D_\alpha(0) = 0 \qquad D_\alpha(1) = 1 \qquad D_\alpha(b) = [\alpha \leq b]$$

$$D_\alpha(p) = (p, 1) \qquad D_\alpha(f) = f(\alpha)$$

where $\alpha \in \mathsf{At}$, $b \in B$, $p \in \Sigma$, $f \in (2 + \Sigma \times X)^{\mathsf{At}}$

KAT+B! [Grathwohl et al. 14]

- Add mutable tests $b!$ and $b?$ to KAT whose behavior is specified equationally
- Conservatively extend any KAT with a minimal amount of extra structure sufficient to perform certain program transformations at the propositional level without sacrificing decidability or deductive completeness
- Central result: A representation theorem for the commutative coproduct of an arbitrary KAT K and a finite relation algebra, namely that it is isomorphic to a certain matrix algebra over K

KAT+B! [Grathwohl et al. 14]

- setters $b!, \bar{b}!$ (think: $b :=$ *true*, $b :=$ *false*)
- testers $b?, \bar{b}?$

Axioms

- $b!b? = b!$
- $b?b! = b?$
- $b!\bar{b}! = \bar{b}!$
- $b!c! = c!b! \quad (b \neq \bar{c})$
- $b!c? = c?b! \quad (b \notin \{c, \bar{c}\})$

Consequences

- $b!b! = b!$
- $b!\bar{b}? = 0$

KAT+B! [Grathwohl et al. 14]

- F_n = the free B!-algebra on $b_1, \ldots, b_n$, isomorphic to Mat$(2^n, 2)$ = the full relation algebra on 2^n states
- B! is PSPACE-complete
- can conservatively extend any KAT with mutable tests via a commutative coproduct construction $(K \oplus F_n)/C$
- $(K \oplus F_n)/C \cong$ Mat$(2^n, K)$
- KAT+B! is exponential-space complete

Characterization of F_n

Lemma

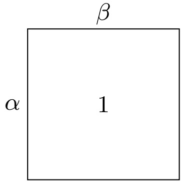
Every element of F_n can be written as a finite sum $\sum_i \alpha_i?\beta_i!$.

$$(\alpha?\beta!)(\gamma?\delta!) = \begin{cases} \alpha?\delta! & \text{if } \beta = \gamma \\ 0 & \text{otherwise} \end{cases}$$

Theorem

$F_n \cong \mathsf{Mat}(2^n, 2)$.

$\alpha?\beta! \quad \mapsto$



Commutative Coproduct

Let $C = \{ab = ba \mid a \in K,\ b \in F\}$.

Lemma

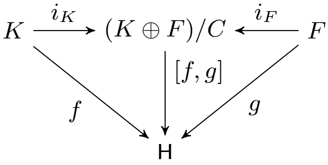
If $f : K \to H$, $g : F \to H$ such that for all $a \in K, b \in F$,

$$f(a)g(b) = g(b)f(a),$$

then there exists a unique universal arrow

$$[f, g] : (K \oplus F)/C \to H$$

commuting with the canonical injections



Commutative Coproduct

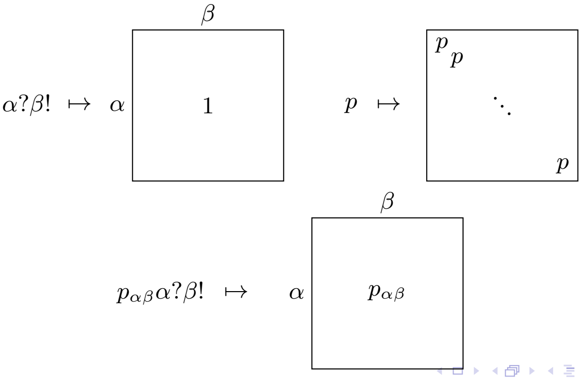
Let K be an arbitrary KAT and let F be a finite KAT.

Lemma

Every element of $(K \oplus F)/C$ *can be written as a finite sum* $\sum_{s \in F} p_s s$.

Theorem

$(K \oplus F_n)/C \cong \mathsf{Mat}(2^n, K)$.



Commutative Coproduct

Corollary

The commutative coproduct $(K \oplus F_n)/C$ *is injective*

(= *the extension of* K *with mutable tests is conservative*)

It is not known whether the coproduct of arbitrary KATs is injective

Complexity

Theorem

KAT + B! *is EXPSPACE-complete.*

A binary counter:

$$
\begin{array}{l}
\overline{b}_0!; \overline{b}_1!; \cdots ; \overline{b}_{n-1}!; \\
\text{while } \overline{b}_0? + \overline{b}_1? + \cdots + \overline{b}_{n-1}? \ \{ \\
\quad \text{if } \overline{b}_0? \text{ then } b_0!; \\
\quad \text{else if } \overline{b}_1? \text{ then } \overline{b}_0!; b_1!; \\
\quad \text{else if } \overline{b}_2? \text{ then } \overline{b}_0!; \overline{b}_1!; b_2!; \\
\quad \text{else } \ldots \\
\quad \text{else if } \overline{b}_{n-1}? \text{ then } \overline{b}_0!; \overline{b}_1!; \cdots ; \overline{b}_{n-2}!; b_{n-1}!; \\
\quad \text{else skip} \\
\}
\end{array}
$$

KAT+B! Bialgebra

Let

- $\mathsf{At}_B = \{\text{atoms of non-mutable tests}\}$
- $\mathsf{At}_T = \{\text{atoms of mutable tests}\}$

Functors:

- $F = \mathsf{Exp}_{\Sigma,B,T}$, where $\mathsf{Exp}_{\Sigma,B,T}\, X$ = KAT expressions over indeterminates X, constant actions Σ, nonmutable tests B, mutable tests T
- $G = (2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T} \times (-)^{\mathsf{At}_B \times \Sigma}$ over $\mathsf{At}_T \times \mathsf{At}_T$ matrices

KAT+B! Bialgebra

Distributive law: syntactic Brzozowski derivative

$$\mathsf{Brz} : \mathsf{Exp}_{\Sigma,B,T}((2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T} \times (-)^{\mathsf{At}_B \times \Sigma}) \\ \to (2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T} \times \mathsf{Exp}_{\Sigma,B,T}(-)^{\mathsf{At}_B \times \Sigma}$$

$$E_{\sigma\tau\alpha} : \mathsf{Exp}_{\Sigma,B,T}((2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T} \times (-)^{\mathsf{At}_B \times \Sigma}) \to 2$$
$$D_{\alpha p} : \mathsf{Exp}_{\Sigma,B,T}((2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T} \times (-)^{\mathsf{At}_B \times \Sigma}) \to \mathsf{Exp}_{\Sigma,B,T}(-)$$

for $\sigma, \tau \in \mathsf{At}_T$, $\alpha \in \mathsf{At}_B$, and $p \in \Sigma$

KAT+B! Bialgebra

$E_{\sigma\tau\alpha}$ and $D_{\alpha p}$ defined exactly like E_α and $D_{\alpha p}$ of KAT, except for the base cases

$$E_{\sigma\tau\alpha}(t!) = [\tau = \sigma[t]] \qquad D_{\alpha p}(t!) = 0^{\mathsf{At}_T \times \mathsf{At}_T}$$
$$E_{\sigma\tau\alpha}(t?) = [\sigma = \tau \leq t] \qquad D_{\alpha p}(t?) = 0^{\mathsf{At}_T \times \mathsf{At}_T}$$
$$E_{\sigma\tau\alpha}(M, f) = M_{\sigma\tau}(\alpha) \qquad D_{\alpha p}(M, f) = f(\alpha p)$$
$$E_{\sigma\tau\alpha}(t!) = [\tau = \sigma[t]] \qquad D_{\alpha p}(t!) = 0^{\mathsf{At}_T \times \mathsf{At}_T}$$
$$E_{\sigma\tau\alpha}(t?) = [\sigma = \tau \leq t] \qquad D_{\alpha p}(t?) = 0^{\mathsf{At}_T \times \mathsf{At}_T}$$
$$E_{\sigma\tau\alpha}(M, f) = M_{\sigma\tau}(\alpha) \qquad D_{\alpha p}(M, f) = f(\alpha p)$$
$$D_{\alpha p}(q) = 0^{\mathsf{At}_T \times \mathsf{At}_T}, \ q \neq p$$
$$D_{\alpha p}(p) = I(1)$$
$$D_{\alpha p}(b) = 0^{\mathsf{At}_T \times \mathsf{At}_T}$$

KAT+B! Bialgebra

Two extremal examples of KAT+B! bialgebras, namely

- $\mathsf{At}_T \times \mathsf{At}_T$ matrices over regular sets of guarded strings
- $\mathsf{At}_T \times \mathsf{At}_T$ matrices over all sets of guarded strings

For the latter with $U = (2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T}$ and $X = (2^{\mathsf{GS}})^{\mathsf{At}_T \times \mathsf{At}_T}$, the bialgebra diagram becomes

$$\begin{array}{ccccc}
\mathsf{Exp}_{\Sigma,B,T}\,X & \xrightarrow{\sigma} & X & \xrightarrow{\zeta} & U \times X^{\mathsf{At}_B \times \Sigma} \\
{\scriptstyle (-)[(\zeta(M))/M]}\downarrow & & & & \uparrow{\scriptstyle \mathsf{id}_U \times (\sigma \circ -)} \\
\mathsf{Exp}_{\Sigma,B,T}(U \times X^{\mathsf{At}_B \times \Sigma}) & & \xrightarrow[\mathsf{Brz}_X]{} & & U \times (\mathsf{Exp}_{\Sigma,B,T}\,X)^{\mathsf{At}_B \times \Sigma}
\end{array}$$

KAT+B! Bialgebra

where

$$\zeta : (2^{\mathsf{GS}})^{\mathsf{At}_T \times \mathsf{At}_T} \to (2^{\mathsf{At}_B})^{\mathsf{At}_T \times \mathsf{At}_T} \times ((2^{\mathsf{GS}})^{\mathsf{At}_T \times \mathsf{At}_T})^{\mathsf{At}_B \times \Sigma}$$

is the componentwise semantic Brzozowski derivative for KAT:

$$\zeta(M) = (\varepsilon_\alpha(M), \delta_{\alpha p}(M))$$

where

$$\varepsilon_\alpha(M)_{\sigma\tau} = [\alpha \in M_{\sigma\tau}] \qquad \delta_{\alpha p}(M) = \{x \mid \alpha p x \in M_{\sigma\tau}\}$$

and σ is the evaluation function on regular expressions over $\mathsf{At}_T \times \mathsf{At}_T$ matrices of subsets of GS

KAT+B! Bialgebra

I think this can be done better!

Break up the KAT+B! derivative into two stages

$$\mathsf{Exp}_{\Sigma,B,T}(\mathsf{Mat}(\mathsf{At}_T, GX)) \to \mathsf{Mat}(\mathsf{At}_T, \mathsf{Exp}_{\Sigma,B}(GX)) \to \mathsf{Mat}(\mathsf{At}_T, G(\mathsf{Exp}_{\Sigma,B}\, X))$$

using the distributive law

$$\mathsf{Exp}(\mathsf{Mat}(S, X)) \to \mathsf{Mat}(S, \mathsf{Exp} X)$$

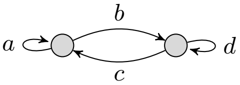
Matrices over a KA(T) form a KA(T)

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

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

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

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



Thanks, and stay safe!