Determinizing, Forgetting, and Automata in Monoidal Categories

James Worthington

Mathematics Department
Cornell University

ASL North American Annual Meeting
March 17, 2010

Overview

monoidal categories are a natural setting to study automata

- automata based on **actions**, languages are **morphisms**

  connections to bialgebras/hopf algebras

  coassociative operations: “hidden parameters”

  extend work of Grossman & Larson,

  classical algebraic treatment of automata (e.g. Eilenberg)

Overview

monoidal categories are a natural setting to study automata

- automata based on **actions**, languages are **morphisms**
  connections to bialgebras/hopf algebras
  coassociative operations: “hidden parameters”
  extend work of Grossman & Larson,
  classical algebraic treatment of automata (e.g. Eilenberg)

- equivalence proofs based on **morphisms of actions**
  completeness theorem
  determinizing = forgetting
  extend work of Kozen on Kleene algebra,
  Rutten on coalgebraic methods to prove automata equivalent

Monoidal Categories

a monoidal category $\mathcal{A}$ has:

- a bifunctor

$$(- \bullet -) \colon \mathcal{A} \times \mathcal{A} \to \mathcal{A}$$

  which is **naturally associative** (satisfies pentagon condition)
- a **unit object** object $1 \in \mathcal{A}$

Monoidal Categories

a monoidal category $\mathcal{A}$ has:

- a bifunctor

$$(- \bullet -) \colon \mathcal{A} \times \mathcal{A} \to \mathcal{A}$$

  which is **naturally associative** (satisfies pentagon condition)
- a **unit object** object $1 \in \mathcal{A}$

Example

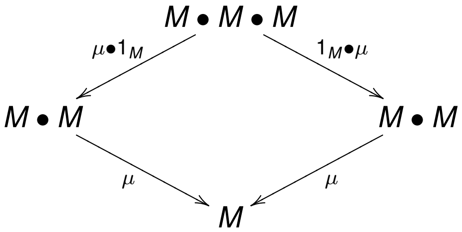
Set, $\times$, $\star$

Example

Mod_K, $\otimes$, K
(K a commutative semiring)

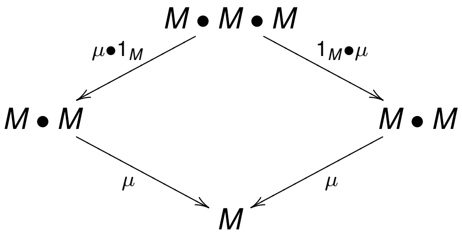
Monoids and Comonoids in Monoidal Categories

a monoid M has an **associative** multiplication μ and a **unit** $\eta\colon 1 \to M$

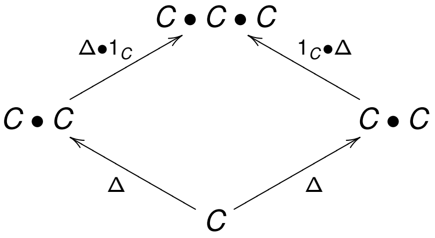


Monoids and Comonoids in Monoidal Categories

a monoid M has an **associative** multiplication μ and a **unit** $\eta\colon 1 \to M$



a comonoid C has a **coassociative** comultiplication Δ and a **counit** $\epsilon\colon C \to 1$



Languages as Morphisms

Fact

Let $\mathcal{A}$ be monoidal category, C a comonoid in $\mathcal{A}$, M a monoid in $\mathcal{A}$. $\mathrm{Hom}(C, M)$ *is a monoid in Set with the* ***convolution product***

$$f * g = \mu_M \circ (f \bullet g) \circ \Delta_C$$

for $f, g \in \mathrm{Hom}(C, M)$

Languages as Morphisms

Fact

Let $\mathcal{A}$ be monoidal category, C a comonoid in $\mathcal{A}$, M a monoid in $\mathcal{A}$. Hom(C, M) *is a monoid in Set with the* ***convolution product***

$$f * g = \mu_M \circ (f \bullet g) \circ \Delta_C$$

for $f, g \in \mathrm{Hom}(C, M)$

generalize languages to **morphisms in monoidal categories**

cf. subsets of $\Sigma^* \leftrightarrow$ formal power series

Example in Set

comonoid

C = words over Σ
$\Delta\colon C \to C \times C$
$\Delta(w) = (w, w)$

monoid

$M = \{0,1\}$
μ = multiplication in two-element Boolean algebra

for $f, g\colon C \to M$,

$$\mu \circ (f \times g) \circ \Delta(w) = f(w)g(w) \quad \text{(intersection)}$$

Example in Set

comonoid

C = words over Σ
$\Delta\colon C \to C \times C$
$\Delta(w) = (w, w)$

monoid

$M = \{0,1\}$
μ = multiplication in two-element Boolean algebra

for $f, g\colon C \to M$,

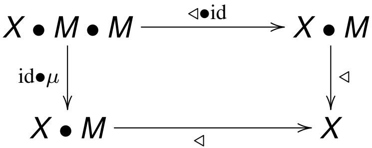
$$\mu \circ (f \times g) \circ \Delta(w) = f(w)g(w) \quad \text{(intersection)}$$

vary category, monoid, comonoid:

- union
- concatenation
- shuffle product

Actions and Automata

monoid M can **act** on object X
(right action)



Actions and Automata

monoid M can **act** on object X
(right action)

$$\begin{array}{ccc} X \bullet M \bullet M & \xrightarrow{\triangleleft \bullet \mathrm{id}} & X \bullet M \\ {\scriptstyle \mathrm{id} \bullet \mu} \downarrow & & \downarrow {\scriptstyle \triangleleft} \\ X \bullet M & \xrightarrow[\triangleleft]{} & X \end{array}$$

to turn (right) action of M on X into automaton, add:

- **start state** $\alpha \colon 1 \to X$
- **observation function** $\Omega \colon X \to O$

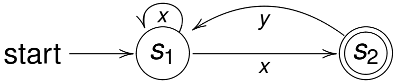
automaton **accepts** morphism

$$\rho \colon M \to O$$

$$M \xrightarrow{\cong} 1 \bullet M \xrightarrow{\alpha \bullet \mathrm{id}} X \bullet M \xrightarrow{\triangleleft} X \xrightarrow{\Omega} O$$

Example in Mod_K

“standard” nfa:



K = two-element idempotent semiring
K**-linear automaton** (automaton in Mod_K)

monoid (input)	Polynomials over $\{x, y\}$ coefficients in K
semimodule (states)	free K-semimodule on $\{s_1, s_2\}$
start	$\alpha(1_K) = 1 s_1$
observation (output in K)	$\Omega(k_1 s_1 + k_2 s_2) = k_2$
action of x	$\begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$

Bimonoids

goal: multiplication of automata consistent with convolution

convolution product requires a **comonoid**

automaton requires **monoid** action

Bimonoids

goal: multiplication of automata consistent with convolution

convolution product requires a **comonoid**

automaton requires **monoid** action

bimonoid: both a monoid and a comonoid, structures compatible

$$\begin{array}{ccccc} B \bullet B & \xrightarrow{\mu} & B & \xrightarrow{\Delta} & B \bullet B \\ \downarrow{\scriptstyle \Delta \bullet \Delta} & & & & \uparrow{\scriptstyle \mu \bullet \mu} \\ B \bullet B \bullet B \bullet B & & \xrightarrow{\mathrm{id} \bullet \sigma \bullet \mathrm{id}} & & B \bullet B \bullet B \bullet B \end{array}$$

braided monoidal category: equipped with a natural isomorphism

$$\sigma : X \bullet Y \cong Y \bullet X$$

(satisfying hexagon conditions)

Multiplying Actions

Fact

The category of right actions of a bimonoid B is a monoidal category:

$$
\begin{array}{c}
X \bullet Y \bullet B \\
\big\downarrow {\scriptstyle id \bullet id \bullet \Delta} \\
X \bullet Y \bullet B \bullet B \\
\big\downarrow {\scriptstyle id \bullet \sigma \bullet id} \\
X \bullet B \bullet Y \bullet B \\
\big\downarrow {\scriptstyle \triangleleft \bullet \triangleleft'} \\
X \bullet Y
\end{array}
$$

Multiplying Automata

multiply automata: “input” a bimonoid B, “output” a monoid M

Automaton C

$\triangleleft \colon X \bullet B \to X$

$\alpha \colon 1 \to X$

$\Omega \colon X \to M$

Automaton D

$\triangleleft' \colon Y \bullet B \to Y$

$\alpha' \colon 1 \to Y$

$\Omega' \colon Y \to M$

Multiplying Automata

multiply automata: "input" a bimonoid B, "output" a monoid M

Automaton C	Automaton D
$\triangleleft\colon X \bullet B \to X$	$\triangleleft'\colon Y \bullet B \to Y$
$\alpha\colon 1 \to X$	$\alpha'\colon 1 \to Y$
$\Omega\colon X \to M$	$\Omega'\colon Y \to M$

product automaton $C \bullet D$

transitions	monoidal category of actions
start state	$\alpha \bullet \alpha'\colon 1 \cong 1 \bullet 1 \to X \bullet Y$
observation	$\mu_M \circ (\Omega \bullet \Omega')\colon X \bullet Y \to M$

Multiplying Automata

multiply automata: "input" a bimonoid B, "output" a monoid M

Automaton C

$\triangleleft\colon X \bullet B \to X$

$\alpha\colon 1 \to X$

$\Omega\colon X \to M$

Automaton D

$\triangleleft'\colon Y \bullet B \to Y$

$\alpha'\colon 1 \to Y$

$\Omega'\colon Y \to M$

product automaton $C \bullet D$

transitions monoidal category of actions

start state $\alpha \bullet \alpha'\colon 1 \cong 1 \bullet 1 \to X \bullet Y$

observation $\mu_M \circ (\Omega \bullet \Omega')\colon X \bullet Y \to M$

Theorem (W.)

$\rho_C * \rho_D = \rho_{C \bullet D}$

(bimonoids for intersection, shuffle)

Morphisms of Automata and Proofs

morphism of actions

$f\colon X \to Y$

preserve start

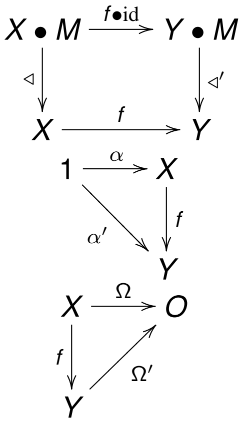
$\alpha\colon 1 \to X$

$\alpha'\colon 1 \to Y$

preserve observation

$\Omega\colon X \to O$

$\Omega'\colon Y \to O$



Fact (soundness)

Morphisms of automata preserve the morphism (language) accepted.

Completeness?

morphisms of automata = **rules of inference**
when is this system **complete**?

Completeness?

morphisms of automata = **rules of inference**
when is this system **complete**?

consider:

	K-linear automata	**det. automata**
category	Mod_K	Set
input monoid	$K\Sigma^*$ (polynomials)	Σ^*
observation	K	$U(K)$

Theorem (W.)

There is an adjunction between the above categories. The functor from this category of K-linear automata to this category of deterministic automata is based on the forgetful functor from Mod_K to Set.

Determinizing = Forgetting

determinize K-linear automaton with state module N:

	K**-linear**	**deterministic**
start: apply adjunction between Mod_K, Set	K-linear map $K \to N$	morphism in Set $\star \to U(K)$
action of x: apply forgetful functor	K-linear map $N \to N$	morphism in Set $U(N) \to U(N)$
observation: apply forgetful functor	K-linear map $N \to K$	morphism in Set $U(N) \to U(K)$

Deterministic Automata

- yields deterministic automaton on $U(N)$ with output set $U(K)$ (n-state nfa: free semimodule on n generators)
- similar determinization for AFA
- free functor in the other direction
- can remove inaccessible states of deterministic automaton
- deterministic automata have unique minimizations

Corollary: Completeness

Let C be a K-linear automaton

$$C \xleftarrow{\epsilon} F(U(C)) \xleftarrow{F(i)} F(U(C)') \xrightarrow{F(m)} F(M(U(C)'))$$

- ϵ: counit of adjunction
- $U(C)'$: accessible part of $U(C)$
- i: inclusion $U(C)' \hookrightarrow U(C)$
- $M(U(C)')$: minimal automaton
- m: unique morphism (finality)

C and D equivalent: same minimization; combine sequences

Corollary: Completeness

Let C be a K-linear automaton

$$C \xleftarrow{\epsilon} F(U(C)) \xleftarrow{F(i)} F(U(C)') \xrightarrow{F(m)} F(M(U(C)'))$$

- ϵ: counit of adjunction
- $U(C)'$: accessible part of $U(C)$
- i: inclusion $U(C)' \hookrightarrow U(C)$
- $M(U(C)')$: minimal automaton
- m: unique morphism (finality)

C and D equivalent: same minimization; combine sequences

Theorem (W.)

For "classical" nfa's, the above proof can be constructed in PSPACE.

References

1. Aguiar, M. and Mahajan, S.
   Monoidal Functors, Species, and Hopf Algebras. 2010.
2. Grossman, R.L. and Larson, R.G.
   Bialgebras and Realizations. 2004.
3. Kozen, D.
   A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events. 1994.
4. Rutten, J.J.M.M.
   Automata and Coinduction (An Exercise in Coalgebra). 1998.
5. Worthington, J.
   A Bialgebraic Approach to Automata and Formal Language Theory. Submitted.