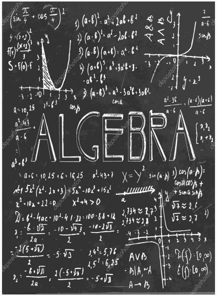
Data Languages: A Categorical Approach

$$(\mathsf{T}, \mu, \eta) +$$ 

Henning Urbat

June 12, 2018

Automata/Languages vs. **Algebraic Structures**



Regular Languages: Four Equivalent Perspectives

Finite Automata

Regular Expressions

$a^* b^*$

Monadic Second-Order Logic

$\neg (\exists x, y : x < y \land b(x) \land a(y))$

Finite Monoids

$e \colon \{a, b\}^* \to M$

Regular Languages: Four Equivalent Perspectives

Finite Automata

Regular Expressions

$a^* b^*$

Monadic Second-Order Logic

$\neg(\exists x, y : x < y \wedge b(x) \wedge a(y))$

Finite Monoids

$e\colon \{a, b\}^* \to M$

Regular Languages: Four Equivalent Perspectives

Finite Automata

Regular Expressions

$a^* b^*$

Monadic Second-Order Logic

$\neg\,(\exists x, y : x < y \land b(x) \land a(y))$

Finite Monoids

$e\colon \{a, b\}^* \to M$

Regular Languages: Four Equivalent Perspectives

Finite Automata

Regular Expressions

$$a^* b^*$$

Monadic Second-Order Logic

$$\neg (\exists x, y : x < y \wedge b(x) \wedge a(y))$$

Finite Monoids

$$e \colon \{a, b\}^* \to M$$

Regular Languages vs. Finite Monoids

Monoid = set M with associative multiplication $\bullet$ und unit 1.

Definition

A monoid morphism $e\colon \Sigma^* \to M$ recognizes the language $L \subseteq \Sigma^*$ if there exists a subset $P \subseteq M$ with $L = e^{-1}[P]$.

Theorem (Kleene, Myhill-Nerode, Schützenberger)

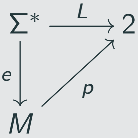
Regular languages = languages recognizable by finite monoids

Regular Languages vs. Finite Monoids

Monoid = set M with associative multiplication $\bullet$ und unit 1.

Definition

A monoid morphism $e \colon \Sigma^* \to M$ recognizes the language $L \subseteq \Sigma^*$ if there exists a subset $P \subseteq M$ with $L = e^{-1}[P]$.



Theorem (Kleene, Myhill-Nerode, Schützenberger)

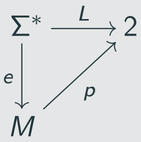
Regular languages = languages recognizable by finite monoids

Regular Languages vs. Finite Monoids

Monoid = set M with associative multiplication $\bullet$ und unit 1.

Definition

A monoid morphism $e\colon \Sigma^* \to M$ recognizes the language $L \subseteq \Sigma^*$ if there exists a subset $P \subseteq M$ with $L = e^{-1}[P]$.



Theorem (Kleene, Myhill-Nerode, Schützenberger)

Regular languages = languages recognizable by finite monoids

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^*$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement} ba \emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^{*}$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement} ba \emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^*$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement} ba \emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^\complement$ but without $(-)^*$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^\complement$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^\complement ba \emptyset^\complement)^\complement$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^*$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement} ba \emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^*$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement} ba \emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^*$.

- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement}ba\emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Application: Star-Free Languages

A language is star-free if it is presentable by a regular expression with $(-)^{\complement}$ but without $(-)^*$.

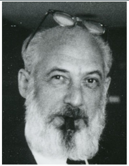
- Is $(a+b)^*$ star-free? Yes: $(a+b)^* = \emptyset^{\complement}$.
- Is a^*b^* star-free? Yes: $a^*b^* = (\emptyset^{\complement} ba \emptyset^{\complement})^{\complement}$.
- Is $(aa)^*$ star-free? No!

Theorem (Schützenberger)

A language is star-free if and only if it is recognized by an aperiodic finite monoid (i.e. satisfying $x^{n+1} = x^n$ for some n).

Corollary: Star-freeness is decidable!

Eilenberg's Variety Theorem (1976)



(Varieties of languages) $\cong$ **(Pseudovarieties of monoids)**

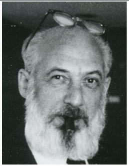
Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

Eilenberg's Variety Theorem (1976)



$$\left(\begin{array}{c} \textbf{Varieties} \\ \textbf{of languages} \end{array} \right) \cong \left(\begin{array}{c} \textbf{Pseudovarieties} \\ \textbf{of monoids} \end{array} \right)$$

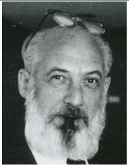
Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

Eilenberg's Variety Theorem (1976)



$$\left(\begin{array}{c}\textbf{Varieties}\\ \textbf{of languages}\end{array}\right) \cong \left(\begin{array}{c}\textbf{Pseudovarieties}\\ \textbf{of monoids}\end{array}\right)$$

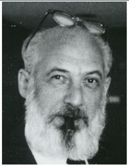
Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

Eilenberg's Variety Theorem (1976)



$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of monoids} \end{pmatrix}$$

Variety of languages

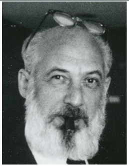
Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

- Boolean operations: $\cup, \cap, (-)^{\complement}$

Eilenberg's Variety Theorem (1976)



$$\begin{pmatrix}\text{Varieties} \\ \text{of languages}\end{pmatrix} \cong \begin{pmatrix}\text{Pseudovarieties} \\ \text{of monoids}\end{pmatrix}$$

Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

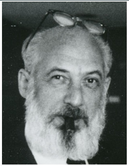
Class of finite monoids closed under quotients, submonoids, and finite products

- Derivatives:

$x^{-1}Ly^{-1} = \{w \in \Sigma^* : xwy \in L\}$

für $L \subseteq \Sigma^*$ und $x, y \in \Sigma^*$

Eilenberg's Variety Theorem (1976)



$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of monoids} \end{pmatrix}$$

Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

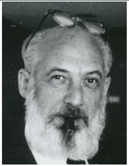
Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

- Homomorphic preimages:

$f^{-1}[L] = \{w \in \Delta^* \ : \ f(w) \in L\}$
for $L \subseteq \Sigma^*$ and $f \colon \Delta^* \to \Sigma^*$
monoid morphism

Eilenberg's Variety Theorem (1976)



$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of monoids} \end{pmatrix}$$

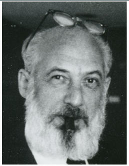
Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

Eilenberg's Variety Theorem (1976)



$$\left(\begin{array}{c} \textbf{Varieties} \\ \textbf{of languages} \end{array} \right) \cong \left(\begin{array}{c} \textbf{Pseudovarieties} \\ \textbf{of monoids} \end{array} \right)$$

Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovarieties of monoids

Class of finite monoids closed under quotients, submonoids, and finite products

$\uparrow$

star-free languages

$\uparrow$

aperiodic monoids

Algebraic Language Theory

Automata theory studies many different types of automata / languages:

- Finite automata / regular languages
- Tree automata / tree languages
- Weighted automata / formal power series
- Büchi automata / ω-regular languages
- ...

Each type of languages with its own algebraic theory, often with closely related ideas, constructions, results (e.g. Eilenberg-type theorems).

$\rightarrow$ Categorical Approach

joint work with J. Adámek, L.-T. Chen, S. Milius, R. Myers

Algebraic Language Theory

Automata theory studies many different types of automata / languages:

- Finite automata / regular languages
- Tree automata / tree languages
- Weighted automata / formal power series
- Büchi automata / ω-regular languages
- . . .

Each type of languages with its own algebraic theory, often with closely related ideas, constructions, results (e.g. Eilenberg-type theorems).

$\to$ Categorical Approach

joint work with J. Adámek, L.-T. Chen, S. Milius, R. Myers

Algebraic Language Theory

Automata theory studies many different types of automata / languages:

- Finite automata / regular languages
- Tree automata / tree languages
- Weighted automata / formal power series
- Büchi automata / ω-regular languages
- ...

Each type of languages with its own algebraic theory, often with closely related ideas, constructions, results (e.g. Eilenberg-type theorems).

$\rightarrow$ **Categorical Approach**

joint work with J. Adámek, L.-T. Chen, S. Milius, R. Myers

Categorical Approach: Idea

Algebraic language theory

$$=$$

(T, μ, η) $\quad + \quad$ 

Algebraic language theory

=

Monads + 

Categorical Approach: Idea

Algebraic language theory

=

Monads + **Duality**

…model languages and recognizing algebraic structures.

Bojańczyk, DLT 2015

…relates languages and recognizing algebras.

Gehrke, Grigorieff, Pin, ICALP 2008

Categorical Approach: Idea

Algebraic language theory

$=$

Monads $+$ **Duality**

. . . model languages and recognizing algebraic structures.

. . . relates languages and recognizing algebras.

Bojańczyk, DLT 2015

Gehrke, Grigorieff, Pin, ICALP 2008

Categorical Approach: Idea

Algebraic language theory

=

Monads + **Duality**

. . . model languages and recognizing algebraic structures.

Bojańczyk, DLT 2015

. . . relates languages and recognizing algebras.

Gehrke, Grigorieff, Pin, ICALP 2008

Categorical Approach: Idea

Algebraic language theory

$=$

Monads $+$ **Duality**

. . . model languages and recognizing algebraic structures.

Bojańczyk, DLT 2015

. . . relates languages and recognizing algebras.

Gehrke, Grigorieff, Pin, ICALP 2008

Categorical Approach: Idea

Algebraic language theory

=

Monads + **Duality**

. . . model languages and recognizing algebraic structures.

. . . relates languages and recognizing algebras.

Languages

Fix a monad $\mathbf{T}$ on a (many-sorted, locally finite) algebraic category $\mathcal{D}$.

Definition

Language = morphism $L\colon T\Sigma \to O$ in $\mathcal{D}$

Σ: free finite object of $\mathcal{D}$ ("alphabet")

O: finite object of $\mathcal{D}$ ("outputs")

- Language of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0, 1\}.$$

- Languages of finite or infinite words: ω-semigroup monad

$$\mathbf{T}(\Sigma, \emptyset) = (\Sigma^+, \Sigma^\omega) \text{ on } \mathbf{Set}^2 \quad \text{and} \quad O = (\{0, 1\}, \{0, 1\}).$$

- Formal power series ($\mathcal{D}$ = vector spaces), tree languages ($\mathcal{D} = \mathbf{Set}^3$), cost functions ($\mathcal{D}$ = posets), ...

Languages

Fix a monad **T** on a (many-sorted, locally finite) algebraic category $\mathcal{D}$.

Definition

Language = morphism $L\colon T\Sigma \to O$ in $\mathcal{D}$

Σ: free finite object of $\mathcal{D}$ ("alphabet")

O: finite object of $\mathcal{D}$ ("outputs")

- Language of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0,1\}.$$

- Languages of finite or infinite words: ω-semigroup monad

$$\mathbf{T}(\Sigma, \emptyset) = (\Sigma^+, \Sigma^\omega) \text{ on } \mathbf{Set}^2 \quad \text{and} \quad O = (\{0,1\},\{0,1\}).$$

- Formal power series ($\mathcal{D}$ = vector spaces), tree languages ($\mathcal{D} = \mathbf{Set}^3$), cost functions ($\mathcal{D}$ = posets), ...

Languages

Fix a monad $\mathbf{T}$ on a (many-sorted, locally finite) algebraic category $\mathcal{D}$.

Definition

Language = morphism $L\colon T\Sigma \to O$ in $\mathcal{D}$

Σ: free finite object of $\mathcal{D}$ ("alphabet")

O: finite object of $\mathcal{D}$ ("outputs")

- Language of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0,1\}.$$

- Languages of finite or infinite words: ω-semigroup monad

$$\mathbf{T}(\Sigma, \emptyset) = (\Sigma^+, \Sigma^\omega) \text{ on } \mathbf{Set}^2 \quad \text{and} \quad O = (\{0,1\}, \{0,1\}).$$

- Formal power series ($\mathcal{D}$ = vector spaces), tree languages ($\mathcal{D} = \mathbf{Set}^3$), cost functions ($\mathcal{D}$ = posets), ...

Languages

Fix a monad $\mathbf{T}$ on a (many-sorted, locally finite) algebraic category $\mathcal{D}$.

Definition

Language = morphism $L\colon T\Sigma \to O$ in $\mathcal{D}$

Σ: free finite object of $\mathcal{D}$ ("alphabet")

O: finite object of $\mathcal{D}$ ("outputs")

- Language of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0,1\}.$$

- Languages of finite or infinite words: ω-semigroup monad

$$\mathbf{T}(\Sigma, \emptyset) = (\Sigma^+, \Sigma^\omega) \text{ on } \mathbf{Set}^2 \quad \text{and} \quad O = (\{0,1\}, \{0,1\}).$$

- Formal power series ($\mathcal{D}$ = vector spaces), tree languages ($\mathcal{D} = \mathbf{Set}^3$), cost functions ($\mathcal{D}$ = posets), ...

Languages

Fix a monad $\mathbf{T}$ on a (many-sorted, locally finite) algebraic category $\mathcal{D}$.

Definition

Language = morphism $L\colon T\Sigma \to O$ in $\mathcal{D}$

Σ: free finite object of $\mathcal{D}$ ("alphabet")

O: finite object of $\mathcal{D}$ ("outputs")

- Language of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0, 1\}.$$

- Languages of finite or infinite words: ω-semigroup monad

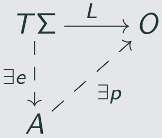
$$\mathbf{T}(\Sigma, \emptyset) = (\Sigma^+, \Sigma^\omega) \text{ on } \mathbf{Set}^2 \quad \text{and} \quad O = (\{0, 1\}, \{0, 1\}).$$

- Formal power series ($\mathcal{D}$ = vector spaces), tree languages ($\mathcal{D} = \mathbf{Set}^3$), cost functions ($\mathcal{D}$ = posets), . . .

Algebraic Recognition

Definition

A language $L\colon T\Sigma \to O$ is **recognizable** if it factorizes through a $\mathbf{T}$-algebra homomorphism $e\colon (\mathbf{T}\Sigma, \mu_\Sigma) \to (A, \alpha)$ with finite codomain.



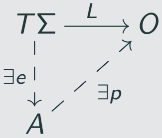
- Languages of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0, 1\}.$$

Recognizable languages = regular languages

Definition

A language $L\colon T\Sigma \to O$ is **recognizable** if it factorizes through a $\mathbf{T}$-algebra homomorphism $e\colon (\mathbf{T}\Sigma, \mu_\Sigma) \to (A, \alpha)$ with finite codomain.



- Languages of finite words: monoid monad

$$\mathbf{T}\Sigma = \Sigma^* \text{ on } \mathbf{Set} \quad \text{and} \quad O = \{0, 1\}.$$

Recognizable languages = regular languages

Definition

A language $L\colon T\Sigma \to O$ is **recognizable** if it factorizes through a $\mathbf{T}$-algebra homomorphism $h\colon (\mathbf{T}\Sigma, \mu_\Sigma) \to (A, \alpha)$ with finite codomain.

$$\begin{array}{ccc} T\Sigma & \xrightarrow{\;L\;} & O \\ {\scriptstyle \exists e}\,\downarrow & \nearrow {\scriptstyle \exists p} & \\ A & & \end{array}$$

- Languages of finite or infinite words: ω-semigroup monad

$$\mathbf{T}(\Sigma, \emptyset) = (\Sigma^+, \Sigma^\omega) \text{ on } \mathbf{Set}^2 \quad \text{and} \quad O = (\{0,1\}, \{0,1\}).$$

Recognizable languages = ∞-regular languages

Categorical Approach

Algebraic language theory

$=$

Monads $+$ **Duality**

. . . model languages and recognizing algebraic structures.

. . . relates languages and recognizing algebras.

Profinite words

- Stone duality between boolean algebras and Stone spaces:

$$\mathbf{BoolAlg}^{op} \xrightarrow{\simeq} \mathbf{Stone} = \text{Pro}{-}\mathbf{Set}_f$$

- Stone space of profinite words:

$$\widehat{\Sigma^*} = \text{limit of all finite quotient monoids } e : \Sigma^* \twoheadrightarrow M.$$

- Dual boolean algebras [Pippenger 1997]:

$$\mathbf{Reg}(\Sigma) = \text{regular languages over } \Sigma.$$

- Key observation: this extends from $\mathbf{T}\Sigma = \Sigma^*$ to arbitrary monads $\mathbf{T}$!

Profinite words

- Stone duality between boolean algebras and Stone spaces:

$$\mathbf{BoolAlg}^{op} \xrightarrow{\simeq} \mathbf{Stone} = \text{Pro}-\mathbf{Set}_f$$

- Stone space of profinite words:

$$\widehat{\Sigma^*} = \text{limit of all finite quotient monoids } e : \Sigma^* \twoheadrightarrow M.$$

- Dual boolean algebras [Pippenger 1997]:

$$\mathbf{Reg}(\Sigma) = \text{regular languages over } \Sigma.$$

- Key observation: this extends from $\mathbf{T}\Sigma = \Sigma^*$ to arbitrary monads $\mathbf{T}$!

Profinite words

- Stone duality between boolean algebras and Stone spaces:

$$\mathbf{BoolAlg}^{op} \xrightarrow{\simeq} \mathbf{Stone} = \text{Pro}-\mathbf{Set}_f$$

- Stone space of profinite words:

$$\widehat{\Sigma^*} = \text{limit of all finite quotient monoids } e : \Sigma^* \twoheadrightarrow M.$$

- Dual boolean algebras [Pippenger 1997]:

$$\mathbf{Reg}(\Sigma) = \text{regular languages over } \Sigma.$$

- Key observation: this extends from $\mathbf{T}\Sigma = \Sigma^*$ to arbitrary monads $\mathbf{T}$!

Profinite words

- Stone duality between boolean algebras and Stone spaces:

$$\mathbf{BoolAlg}^{op} \xrightarrow{\simeq} \mathbf{Stone} = \text{Pro}{-}\mathbf{Set}_f$$

- Stone space of profinite words:

$$\widehat{\Sigma^*} = \text{limit of all finite quotient monoids } e : \Sigma^* \twoheadrightarrow M.$$

- Dual boolean algebras [Pippenger 1997]:

$$\mathbf{Reg}(\Sigma) = \text{regular languages over } \Sigma.$$

- Key observation: this extends from $\mathbf{T}\Sigma = \Sigma^*$ to arbitrary monads $\mathbf{T}$!

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \text{Pro-}\,\mathcal{D}_f$$

$\mathcal{C}$	$\mathcal{D}$	$\widehat{\mathcal{D}}$
boolean algebras	sets	Stone spaces
distributive lattices	posets	Priestley spaces
semilattices	semilattices	Stone semilattices
vector spaces	vector spaces	Stone vector spaces

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} =\!=\!=\!= \mathsf{Pro}\text{–}\,\mathcal{D}_f$$

$\mathcal{C}$	$\mathcal{D}$	$\widehat{\mathcal{D}}$
boolean algebras	sets	Stone spaces
distributive lattices	posets	Priestley spaces
semilattices	semilattices	Stone semilattices
vector spaces	vector spaces	Stone vector spaces

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \text{Pro}-\mathcal{D}_f$$

- $\hat{T}\Sigma \in \widehat{\mathcal{D}}$: limit of all finite quotient algebras $\mathbf{T}\Sigma \twoheadrightarrow A$.
  [This gives a monad $\widehat{\mathbf{T}}$ on $\widehat{\mathcal{D}}$: the profinite monad of $\mathbf{T}$.]
- Choose $O = (\text{dual of } \mathbf{1})$, where $\mathbf{1} \in \mathcal{C}$ is the free object on one generator.
- What is the dual object of $\hat{T}\Sigma$ in $\mathcal{C}$?

Theorem (Generalized Pippenger Theorem)

Let $\mathbf{Rec}(\Sigma)$ *be the set of* $\mathbf{T}$*-recognizable languages. There is a unique* $\mathcal{C}$*-algebra structure on* $\mathbf{Rec}(\Sigma)$ *such that*

$$\mathbf{Rec}(\Sigma) = \textit{dual object of } \hat{T}\Sigma.$$

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} =\!=\!= \text{Pro-}\,\mathcal{D}_f$$

- $\hat{T}\Sigma \in \widehat{\mathcal{D}}$: limit of all finite quotient algebras $\mathbf{T}\Sigma \twoheadrightarrow A$.
  [This gives a monad $\widehat{\mathbf{T}}$ on $\widehat{\mathcal{D}}$: the profinite monad of $\mathbf{T}$.]
- Choose $O =$ (dual of $\mathbf{1}$), where $\mathbf{1} \in \mathcal{C}$ is the free object on one generator.
- What is the dual object of $\hat{T}\Sigma$ in $\mathcal{C}$?

Theorem (Generalized Pippenger Theorem)

Let $\mathbf{Rec}(\Sigma)$ *be the set of* $\mathbf{T}$*-recognizable languages. There is a unique* $\mathcal{C}$*-algebra structure on* $\mathbf{Rec}(\Sigma)$ *such that*

$$\mathbf{Rec}(\Sigma) = \textit{dual object of } \hat{T}\Sigma.$$

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \text{Pro}-\mathcal{D}_f$$

- $\hat{T}\Sigma \in \widehat{\mathcal{D}}$: limit of all finite quotient algebras $\mathbf{T}\Sigma \twoheadrightarrow A$.
  [This gives a monad $\widehat{\mathbf{T}}$ on $\widehat{\mathcal{D}}$: the profinite monad of $\mathbf{T}$.]
- Choose $O =$ (dual of $\mathbf{1}$), where $\mathbf{1} \in \mathcal{C}$ is the free object on one generator.
- What is the dual object of $\hat{T}\Sigma$ in $\mathcal{C}$?

Theorem (Generalized Pippenger Theorem)

Let $\mathbf{Rec}(\Sigma)$ be the set of $\mathbf{T}$-recognizable languages. There is a unique $\mathcal{C}$-algebra structure on $\mathbf{Rec}(\Sigma)$ such that

$$\mathbf{Rec}(\Sigma) = \textit{dual object of } \hat{T}\Sigma.$$

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \text{Pro}-\mathcal{D}_f$$

- $\hat{T}\Sigma \in \widehat{\mathcal{D}}$: limit of all finite quotient algebras $\mathbf{T}\Sigma \twoheadrightarrow A$.
  [This gives a monad $\widehat{\mathbf{T}}$ on $\widehat{\mathcal{D}}$: the profinite monad of $\mathbf{T}$.]
- Choose O = (dual of $\mathbf{1}$), where $\mathbf{1} \in \mathcal{C}$ is the free object on one generator.
- What is the dual object of $\hat{T}\Sigma$ in $\mathcal{C}$?

Theorem (Generalized Pippenger Theorem)

Let $\mathbf{Rec}(\Sigma)$ *be the set of* $\mathbf{T}$*-recognizable languages. There is a unique* $\mathcal{C}$*-algebra structure on* $\mathbf{Rec}(\Sigma)$ *such that*

$$\mathbf{Rec}(\Sigma) = \textit{dual object of } \hat{T}\Sigma.$$

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \text{Pro-}\,\mathcal{D}_f$$

- $\hat{T}\Sigma \in \widehat{\mathcal{D}}$: limit of all finite quotient algebras $\mathbf{T}\Sigma \twoheadrightarrow A$. [This gives a monad $\widehat{\mathbf{T}}$ on $\widehat{\mathcal{D}}$: the profinite monad of $\mathbf{T}$.]
- Choose $O = (\text{dual of } \mathbf{1})$, where $\mathbf{1} \in \mathcal{C}$ is the free object on one generator.
- What is the dual object of $\hat{T}\Sigma$ in $\mathcal{C}$?

Theorem (Generalized Pippenger Theorem)

Let $\mathbf{Rec}(\Sigma)$ be the set of $\mathbf{T}$-recognizable languages. There is a unique $\mathcal{C}$-algebra structure on $\mathbf{Rec}(\Sigma)$ such that

$$\mathbf{Rec}(\Sigma) = \textit{dual object of } \hat{T}\Sigma.$$

Profinite Words: Generalization

Monad $\mathbf{T}$ on $\mathcal{D}$ as before + another locally finite variety $\mathcal{C}$ with $\mathcal{C}_f^{op} \simeq \mathcal{D}_f$.

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \text{Pro}-\mathcal{D}_f$$

- $\hat{T}\Sigma \in \widehat{\mathcal{D}}$: limit of all finite quotient algebras $\mathbf{T}\Sigma \twoheadrightarrow A$.
  [This gives a monad $\widehat{\mathbf{T}}$ on $\widehat{\mathcal{D}}$: the profinite monad of $\mathbf{T}$.]
- Choose $O =$ (dual of $\mathbf{1}$), where $\mathbf{1} \in \mathcal{C}$ is the free object on one generator.
- What is the dual object of $\hat{T}\Sigma$ in $\mathcal{C}$?

Theorem (Generalized Pippenger Theorem)

Let $\mathbf{Rec}(\Sigma)$ *be the set of* $\mathbf{T}$*-recognizable languages. There is a unique* $\mathcal{C}$*-algebra structure on* $\mathbf{Rec}(\Sigma)$ *such that*

$$\mathbf{Rec}(\Sigma) = \textit{dual object of } \hat{T}\Sigma.$$

Generalized Eilenberg Theorem

$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of monoids} \end{pmatrix}$$

Variety of languages

Class of regular languages closed under boolean ops, derivatives and homomorphic preimages.

Pseudovariety of monoids

Class of finite monoids closed under quotients, submonoids and finite products.

Generalized Eilenberg Theorem

$$\left(\begin{array}{c}\textbf{Varieties}\\ \textbf{of languages}\end{array}\right) \quad\cong\quad \left(\begin{array}{c}\textbf{Pseudovarieties}\\ \textbf{of T-algebras}\end{array}\right)$$

Variety of languages

Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

Generalized Eilenberg Theorem

$$\left(\begin{array}{c} \textbf{Varieties} \\ \textbf{of languages} \end{array} \right) \cong \left(\begin{array}{c} \textbf{Pseudovarieties} \\ \textbf{of T-algebras} \end{array} \right)$$

Variety of languages

Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

- $\mathcal{C}$-operations via $\mathbf{Rec}(\Sigma) \in \mathcal{C}$:

$$\mathcal{C}^{op} \xrightarrow{\simeq} \widehat{\mathcal{D}} = \mathrm{Pro}\!-\mathcal{D}_f$$

Generalized Eilenberg Theorem

$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of T-algebras} \end{pmatrix}$$

Variety of languages

Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

- Homomorphic preimages:
$f^{-1}[L] = (T\Delta \xrightarrow{f} T\Sigma \xrightarrow{L} O)$
for $\mathbf{T}$-algebra hom. $f\colon \mathbf{T}\Delta \to \mathbf{T}\Sigma$

Generalized Eilenberg Theorem

$$\left(\begin{array}{c} \textbf{Varieties} \\ \textbf{of languages} \end{array} \right) \cong \left(\begin{array}{c} \textbf{Pseudovarieties} \\ \textbf{of T-algebras} \end{array} \right)$$

Variety of languages

Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

- Derivatives via **unary presentation** of $\mathbf{T}$.

Unary Presentations and Derivatives

Consider the unary operations $\Sigma^* \xrightarrow{x(-)y} \Sigma^*$ $(x, y \in \Sigma^*)$.

- The derivatives of a language $\Sigma^* \xrightarrow{L} \{0,1\}$ are

$$x^{-1}Ly^{-1} = (\ \Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} \{0,1\}\ ).$$

- For any surjective map $e : \Sigma^* \twoheadrightarrow A$:

$$e \text{ carries monoid morphism} \quad \text{iff} \quad \text{each } \Sigma^* \xrightarrow{x(-)y} \Sigma^* \text{ lifts along } e.$$

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{x(-)y} & \Sigma^* \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

Unary Presentations and Derivatives

Consider the unary operations $\Sigma^* \xrightarrow{x(-)y} \Sigma^*$ $(x, y \in \Sigma^*)$.

- The derivatives of a language $\Sigma^* \xrightarrow{L} \{0,1\}$ are

$$x^{-1}Ly^{-1} = (\ \Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} \{0,1\}\ ).$$

- For any surjective map $e : \Sigma^* \twoheadrightarrow A$:

$$e \text{ carries monoid morphism} \quad \text{iff} \quad \text{each } \Sigma^* \xrightarrow{x(-)y} \Sigma^* \text{ lifts along } e.$$

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{x(-)y} & \Sigma^* \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

Unary Presentations and Derivatives

Consider the unary operations $\Sigma^* \xrightarrow{x(-)y} \Sigma^*$ $(x, y \in \Sigma^*)$.

- The derivatives of a language $\Sigma^* \xrightarrow{L} \{0,1\}$ are

$$x^{-1}Ly^{-1} = (\ \Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} \{0,1\}\ ).$$

- For any surjective map $e : \Sigma^* \twoheadrightarrow A$:

$$e \text{ carries monoid morphism} \quad \text{iff} \quad \text{each } \Sigma^* \xrightarrow{x(-)y} \Sigma^* \text{ lifts along } e.$$

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{x(-)y} & \Sigma^* \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

Unary Presentations and Derivatives

Consider the unary operations $\Sigma^* \xrightarrow{x(-)y} \Sigma^*$ $(x, y \in \Sigma^*)$.

- The derivatives of a language $\Sigma^* \xrightarrow{L} \{0,1\}$ are

$$x^{-1}Ly^{-1} = (\ \Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} \{0,1\}\ ).$$

- For any surjective map $e : \Sigma^* \twoheadrightarrow A$:

e carries monoid morphism iff each $\Sigma^* \xrightarrow{x(-)y} \Sigma^*$ lifts along e.

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{x(-)y} & \Sigma^* \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

Unary Presentations and Derivatives

Definition (Unary presentation)

A set $\mathfrak{U} = \{\, T\Sigma \xrightarrow{u} T\Sigma \,\}$ of unary ops such that for all $e : T\Sigma \twoheadrightarrow A$:

e carries **T**-algebra morphism iff each $u \in \mathfrak{U}$ lifts along e.

$$\begin{array}{ccc} T\Sigma & \xrightarrow{\;u\;} & T\Sigma \\ {\scriptstyle e}\big\downarrow & & \big\downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

Definition (Derivative)

For a language $T\Sigma \xrightarrow{L} O$ and a unary operation $T\Sigma \xrightarrow{u} T\Sigma$ in $\mathfrak{U}$, put:

$$u^{-1}L := (\; T\Sigma \xrightarrow{u} T\Sigma \xrightarrow{L} O \;).$$

Unary Presentations and Derivatives

Definition (Unary presentation)

A set $\mathfrak{U} = \{\, T\Sigma \xrightarrow{u} T\Sigma \,\}$ of unary ops such that for all $e : T\Sigma \twoheadrightarrow A$:

$$e \text{ carries } \mathbf{T}\text{-algebra morphism} \quad \text{iff} \quad \text{each } u \in \mathfrak{U} \text{ lifts along } e.$$

$$\begin{array}{ccc} T\Sigma & \xrightarrow{u} & T\Sigma \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

Definition (Derivative)

For a language $T\Sigma \xrightarrow{L} O$ and a unary operation $T\Sigma \xrightarrow{u} T\Sigma$ in $\mathfrak{U}$, put:

$$u^{-1}L := (\; T\Sigma \xrightarrow{u} T\Sigma \xrightarrow{L} O \;).$$

Generalized Eilenberg Theorem

$$\left(\begin{array}{c} \textbf{Varieties} \\ \textbf{of languages} \end{array} \right) \cong \left(\begin{array}{c} \textbf{Pseudovarieties} \\ \textbf{of T-algebras} \end{array} \right)$$

Variety of languages

Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

- Derivatives:

$u^{-1}L := (\ T\Sigma \xrightarrow{u} T\Sigma \xrightarrow{L} O\ )$

for $u \in \mathfrak{U}$

Generalized Eilenberg Theorem

$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of T-algebras} \end{pmatrix}$$

Variety of languages

Class of **T**-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite **T**-algebras closed under quotients, subalgebras and finite products.

Proof of the theorem: Duality!

Generalized Eilenberg Theorem

$$\left(\begin{array}{c}\textbf{Varieties}\\ \textbf{of languages}\end{array}\right) \cong \left(\begin{array}{c}\textbf{Pseudovarieties}\\ \textbf{of T-algebras}\end{array}\right)$$

Variety of languages

Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

Proof of the theorem: Duality!

Generalized Eilenberg Theorem

$$\begin{pmatrix} \textbf{Varieties} \\ \textbf{of languages} \end{pmatrix} \cong \begin{pmatrix} \textbf{Pseudovarieties} \\ \textbf{of T-algebras} \end{pmatrix}$$

Variety of languages

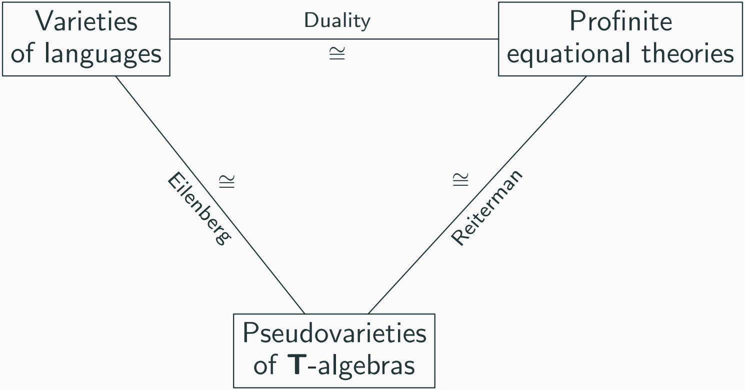
Class of $\mathbf{T}$-recognizable languages closed under $\mathcal{C}$-ops, derivatives and homomorphic preimages.

Pseudovariety of T-algebas

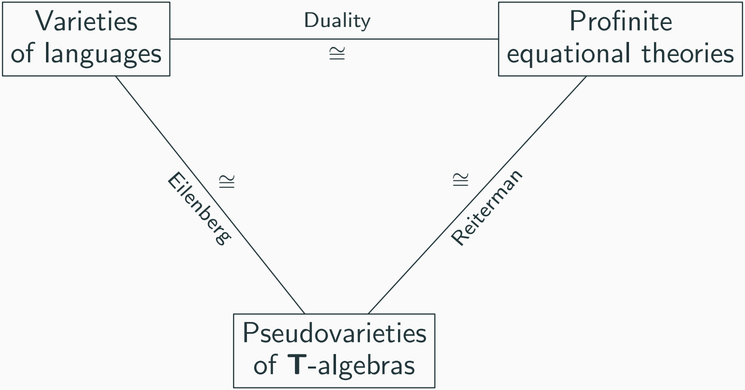
Class of finite $\mathbf{T}$-algebras closed under quotients, subalgebras and finite products.

Proof of the theorem: Duality!

Algebraic Language Theory: Magic Triangle



Thus: **Eilenberg = Reiterman + Duality!**



Thus: **Eilenberg = Reiterman + Duality!**

Towards Data Languages

Goal: Extend the categorical approach to data languages, i.e.

- the category $\mathcal{D} = \mathbf{Nom}$ (nominal sets and finitely supported maps)
- the monad $\mathbf{T}\Sigma = \Sigma^*$ on $\mathbf{Nom}$ (where $\mathbf{Nom}^{\mathbf{T}}$ = nominal monoids)
- languages $L\colon \Sigma^* \to 2$ in $\mathbf{Nom}$ recognized by orbit-finite nominal monoids:

- cf. [Bojańczyk, Klin, Lasota], [Colcombet], ...
- No Eilenberg-Reiterman correspondence known.

Not a direct instance of our categorical framework: $\mathbf{Nom}$ is not algebraic!

Towards Data Languages

Goal: Extend the categorical approach to data languages, i.e.

- the category $\mathcal{D} = \mathbf{Nom}$ (nominal sets and finitely supported maps)
- the monad $\mathbf{T}\Sigma = \Sigma^*$ on $\mathbf{Nom}$ (where $\mathbf{Nom}^{\mathbf{T}}$ = nominal monoids)
- languages $L\colon \Sigma^* \to 2$ in $\mathbf{Nom}$ recognized by orbit-finite nominal monoids:

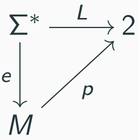
- cf. [Bojańczyk, Klin, Lasota], [Colcombet], ...
- No Eilenberg-Reiterman correspondence known.

Not a direct instance of our categorical framework: **Nom** is not algebraic!

Towards Data Languages

Goal: Extend the categorical approach to data languages, i.e.

- the category $\mathcal{D} = \mathbf{Nom}$ (nominal sets and finitely supported maps)
- the monad $\mathbf{T}\Sigma = \Sigma^*$ on **Nom** (where $\mathbf{Nom}^{\mathbf{T}}$ = nominal monoids)
- languages $L\colon \Sigma^* \to 2$ in **Nom** recognized by orbit-finite nominal monoids:



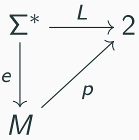
- cf. [Bojańczyk, Klin, Lasota], [Colcombet], ...
- No Eilenberg-Reiterman correspondence known.

Not a direct instance of our categorical framework: **Nom** is not algebraic!

Towards Data Languages

Goal: Extend the categorical approach to data languages, i.e.

- the category $\mathcal{D} = \mathbf{Nom}$ (nominal sets and finitely supported maps)
- the monad $\mathbf{T}\Sigma = \Sigma^*$ on **Nom** (where $\mathbf{Nom}^{\mathbf{T}}$ = nominal monoids)
- languages $L\colon \Sigma^* \to 2$ in **Nom** recognized by orbit-finite nominal monoids:



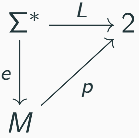
- cf. [Bojańczyk, Klin, Lasota], [Colcombet], ...
- No Eilenberg-Reiterman correspondence known.

Not a direct instance of our categorical framework: **Nom** is not algebraic!

Towards Data Languages

Goal: Extend the categorical approach to data languages, i.e.

- the category $\mathcal{D} = \mathbf{Nom}$ (nominal sets and finitely supported maps)
- the monad $\mathbf{T}\Sigma = \Sigma^*$ on $\mathbf{Nom}$ (where $\mathbf{Nom}^{\mathbf{T}}$ = nominal monoids)
- languages $L\colon \Sigma^* \to 2$ in $\mathbf{Nom}$ recognized by orbit-finite nominal monoids:



- cf. [Bojańczyk, Klin, Lasota], [Colcombet], ...
- No Eilenberg-Reiterman correspondence known.

Not a direct instance of our categorical framework: **Nom** is not algebraic!

Towards Data Languages

Goal: Extend the categorical approach to data languages, i.e.

- the category $\mathcal{D} = \mathbf{Nom}$ (nominal sets and finitely supported maps)
- the monad $\mathbf{T}\Sigma = \Sigma^*$ on $\mathbf{Nom}$ (where $\mathbf{Nom}^{\mathbf{T}}$ = nominal monoids)
- languages $L\colon \Sigma^* \to 2$ in $\mathbf{Nom}$ recognized by orbit-finite nominal monoids:

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{L} & 2 \\ {\scriptstyle e}\big\downarrow & \nearrow{\scriptstyle p} & \\ M & & \end{array}$$

- cf. [Bojańczyk, Klin, Lasota], [Colcombet], ...
- No Eilenberg-Reiterman correspondence known.

Not a direct instance of our categorical framework: **Nom** is not algebraic!

The Category Nom

$\mathcal{D} =$ **Nom** shares many properties with algebraic categories:

- finitely complete and cocomplete
- has generator 1 and cogenerator $2 = \{0, 1\}$
- meaningful notion of "finiteness": orbit-finite sets
- analogue of free algebras: strong nominal sets Σ, serving as alphabets

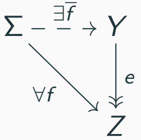
Crucial property: Strong nominal sets Σ are projective w.r.t. support-reflecting quotients $e\colon X \twoheadrightarrow Y$ (i.e. for every $y \in Y$ there exists $x \in X$ with $e(x) = y$ and $\mathrm{supp}(x) = \mathrm{supp}(y) \cup \mathrm{supp}(e)$).

The Category Nom

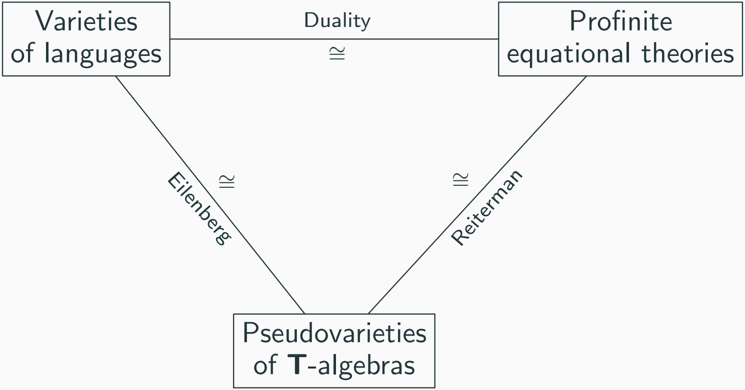
$\mathcal{D} = $ **Nom** shares many properties with algebraic categories:

- finitely complete and cocomplete
- has generator 1 and cogenerator $2 = \{0, 1\}$
- meaningful notion of "finiteness": orbit-finite sets
- analogue of free algebras: strong nominal sets Σ, serving as alphabets

Crucial property: Strong nominal sets Σ are projective w.r.t. support-reflecting quotients $e\colon X \twoheadrightarrow Y$ (i.e. for every $y \in Y$ there exists $x \in X$ with $e(x) = y$ and $\mathrm{supp}(x) = \mathrm{supp}(y) \cup \mathrm{supp}(e)$).



Algebraic Language Theory: Magic Triangle



Thus: **Eilenberg = Reiterman + Duality!**

Duality for Nominal Sets

Let **nCABA** be the category of nominal complete atomic boolean algebras:

- objects are nominal boolean algebras such that (i) every element has an atom below it and (ii) every equivariant subset has a supremum;
- morphisms are finitely supported boolean homomorphisms preserving all suprema of equivariant subsets.

Theorem (Nominal Stone Duality, cf. Petrişan 2011)

$\mathbf{nCABA}^{op} \simeq \mathbf{Nom}$ *given by* $B \mapsto$ *atoms of* B *and* $X \mapsto [X, 2]$.

Duality restricts to one between orbit-finite nominal sets and nCABAs with an orbit-finite set of atoms.

Duality for Nominal Sets

Let **nCABA** be the category of nominal complete atomic boolean algebras:

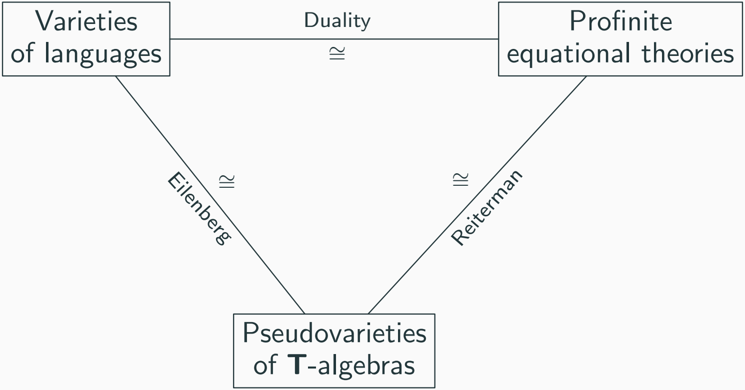
- objects are nominal boolean algebras such that (i) every element has an atom below it and (ii) every equivariant subset has a supremum;
- morphisms are finitely supported boolean homomorphisms preserving all suprema of equivariant subsets.

Theorem (Nominal Stone Duality, cf. Petrișan 2011)

$\mathbf{nCABA}^{op} \simeq \mathbf{Nom}$ *given by* $B \mapsto$ *atoms of* B *and* $X \mapsto [X, 2]$.

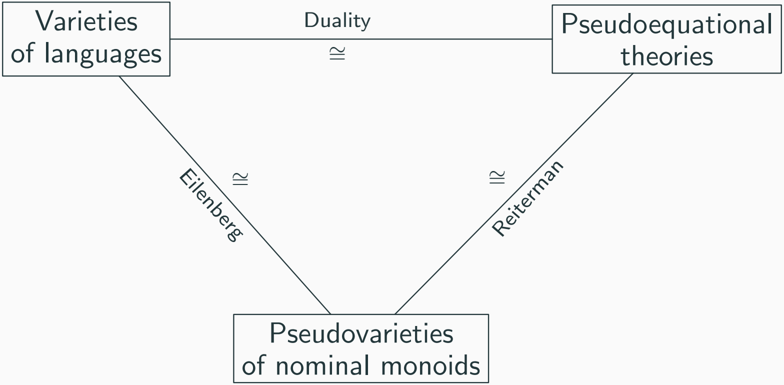
Duality restricts to one between orbit-finite nominal sets and nCABAs with an orbit-finite set of atoms.

Algebraic Language Theory: Magic Triangle



Thus: **Eilenberg = Reiterman + Duality!**

Algebraic Language Theory: Magic Triangle



Thus: **Eilenberg = Reiterman + Duality!**

A Reiterman Correspondence

Theorem

Pseudovarieties of nominal monoids $\cong$ *Pseudoequational theories*

A Reiterman Correspondence

Definition (Pseudovariety)

Pseudovariety of nominal monoids = class of orbit-finite nominal monoids closed under finite products, submonoids, support-reflecting quotients.

A Reiterman Correspondence

A pseudoequational theory associates to every orbit-finite alphabet (= strong nominal set) Σ a filter of orbit-finite quotient monoids

$$(e_i \colon \Sigma^* \twoheadrightarrow M_i)_{i \in I_\Sigma}$$

such that

1. for every $h \colon \Delta^* \to \Sigma^*$ and $i \in I_\Sigma$ there exists $j \in J_\Sigma$ and $\overline{h}$ with

$$\begin{array}{ccc} \Delta^* & \xrightarrow{\forall h} & \Sigma^* \\ {\scriptstyle \exists e_j}\big\downarrow & & \big\downarrow{\scriptstyle \forall e_i} \\ M_j & \xrightarrow[\exists \overline{h}]{} & M_i \end{array}$$

2. for every $e_i \colon \Sigma^* \twoheadrightarrow M_i$ ($i \in I_\Sigma$) and every support-reflecting $e \colon \Delta^* \twoheadrightarrow M_i$, one has $e = e_j$ for some $j \in I_\Delta$.

A Reiterman Correspondence

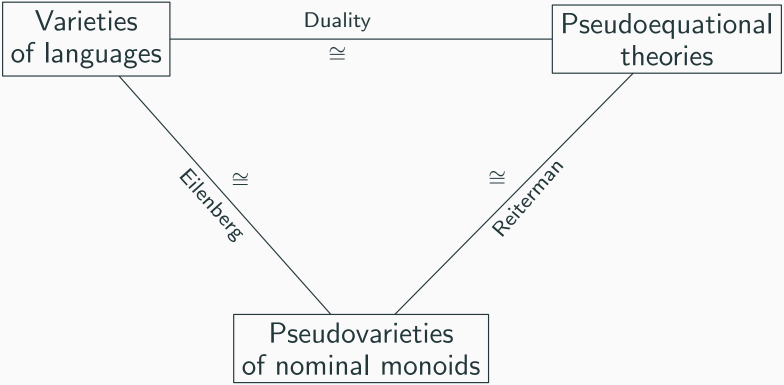
Theorem

Pseudovarieties of nominal monoids $\cong$ *Pseudoequational theories*

Correspondence is not specific to nominal monoids, but can be established in general categories $\mathcal{D}$ with enough limits, a nice factorization system and well-behaved projective objects.

$\rightarrow$ cf. Banaschewski & Herrlich 1976

Algebraic Language Theory: Magic Triangle



Thus: **Eilenberg = Reiterman + Duality!**

Towards an Eilenberg Correspondence

To get the proper notion of variety of languages,

- dualize the concept of a pseudoequational theory from **Nom** to **nCABA**
- interpret this concept in terms of language-theoretic closure properties

Towards an Eilenberg Correspondence

Orbit-finite quotient $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom**

$\cong$ (duality)

Atom-finite nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$

Towards an Eilenberg Correspondence

Orbit-finite quotient monoid $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom**

$\cong$ (unary presentation)

Orbit-finite quotient $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom** such that

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{\forall x(-)y} & \Sigma^* \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

$\cong$ (duality)

Atom-finite nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives $x^{-1}Ly^{-1} = (\Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} 2)$.

Towards an Eilenberg Correspondence

Orbit-finite quotient monoid $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom**

$\cong$ (unary presentation)

Orbit-finite quotient $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom** such that

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{\forall x(-)y} & \Sigma^* \\ {\scriptstyle e}\big\downarrow & & \big\downarrow{\scriptstyle e} \\ A & \underset{\exists}{\dashrightarrow} & A \end{array}$$

$\cong$ (duality)

Atom-finite nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives $x^{-1}Ly^{-1} = (\Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} 2)$.

Towards an Eilenberg Correspondence

Orbit-finite quotient monoid $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom**

$\cong$ (unary presentation)

Orbit-finite quotient $e\colon \Sigma^* \twoheadrightarrow M$ in **Nom** such that

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{\forall x(-)y} & \Sigma^* \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle e} \\ A & \xrightarrow[\exists]{} & A \end{array}$$

$\cong$ (duality)

Atom-finite nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives $x^{-1}Ly^{-1} = (\Sigma^* \xrightarrow{x(-)y} \Sigma^* \xrightarrow{L} 2)$.

Towards an Eilenberg Correspondence

Filter of orbit-finite quotient monoids $e\colon \Sigma^* \twoheadrightarrow M_i$ $(i \in I_\Sigma)$ in **Nom**

$\cong$ (unary presentation)

Filter of atom-finite nominal boolean subalgebras of languages $V_i \rightarrowtail [\Sigma^*, 2]$ $(i \in I_\Sigma)$ closed under derivatives

$\cong$ (identify with union)

Nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives and unions of S-orbits (for every finite set S of names), a.k.a. a local variety of nominal languages over Σ.

This is the **Local Eilenberg Theorem** for nominal languages.

Towards an Eilenberg Correspondence

Filter of orbit-finite quotient monoids $e\colon \Sigma^* \twoheadrightarrow M_i$ ($i \in I_\Sigma$) in **Nom**

$\cong$ (unary presentation)

Filter of atom-finite nominal boolean subalgebras of languages $V_i \rightarrowtail [\Sigma^*, 2]$ ($i \in I_\Sigma$) closed under derivatives

$\cong$ (identify with union)

Nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives and unions of S-orbits (for every finite set S of names), a.k.a. a local variety of nominal languages over Σ.

This is the **Local Eilenberg Theorem** for nominal languages.

Towards an Eilenberg Correspondence

Filter of orbit-finite quotient monoids $e\colon \Sigma^* \twoheadrightarrow M_i$ $(i \in I_\Sigma)$ in **Nom**

$\cong$ (unary presentation)

Filter of atom-finite nominal boolean subalgebras of languages $V_i \rightarrowtail [\Sigma^*, 2]$ $(i \in I_\Sigma)$ closed under derivatives

$\cong$ (identify with union)

Nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives and unions of S-orbits (for every finite set S of names), a.k.a. a local variety of nominal languages over Σ.

This is the **Local Eilenberg Theorem** for nominal languages.

Towards an Eilenberg Correspondence

Filter of orbit-finite quotient monoids $e\colon \Sigma^* \twoheadrightarrow M_i$ $(i \in I_\Sigma)$ in **Nom**

$\cong$ (unary presentation)

Filter of atom-finite nominal boolean subalgebras of languages $V_i \rightarrowtail [\Sigma^*, 2]$ $(i \in I_\Sigma)$ closed under derivatives

$\cong$ (identify with union)

Nominal boolean subalgebra of languages $V \rightarrowtail [\Sigma^*, 2]$ closed under derivatives and unions of S-orbits (for every finite set S of names), a.k.a. a local variety of nominal languages over Σ.

This is the **Local Eilenberg Theorem** for nominal languages.

A Reiterman Correspondence

A pseudoequational theory associates to every orbit-finite alphabet (= strong nominal set) Σ a filter of orbit-finite quotient monoids

$$(e_i \colon \Sigma^* \twoheadrightarrow M_i)_{i \in I_\Sigma}$$

such that

1. for every $h \colon \Delta^* \to \Sigma^*$ and $i \in I_\Sigma$ there exists $j \in J_\Sigma$ and $\overline{h}$ with

$$\begin{array}{ccc} \Delta^* & \xrightarrow{\forall h} & \Sigma^* \\ {\scriptstyle \exists e_j}\big\downarrow & & \big\downarrow{\scriptstyle \forall e_i} \\ M_j & \xrightarrow[\exists \overline{h}]{} & M_i \end{array}$$

2. for every $e_i \colon \Sigma^* \twoheadrightarrow M_i$ ($i \in I_\Sigma$) and every support-reflecting $e \colon \Delta^* \twoheadrightarrow M_i$, one has $e = e_j$ for some $j \in I_\Delta$.

Pseudoequational theory $\Sigma \mapsto (\Sigma^* \twoheadrightarrow M_i)_{i \in I_\Sigma}$

$\cong$ (duality)

Variety of languages: $\Sigma \mapsto$ (local variety $V_\Sigma \rightarrowtail [\Sigma^*, 2]$)

…with closure properties dual to property 1/2 of a pseudoequational theory:

- Property 1 of a theory dualizes to closure under preimages.
- Open problem: State dual of Property 2 in terms of languages.

Towards an Eilenberg Correspondence

Pseudoequational theory $\Sigma \mapsto (\Sigma^* \twoheadrightarrow M_i)_{i \in I_\Sigma}$

$$\cong \text{(duality)}$$

Variety of languages: $\Sigma \mapsto (\text{local variety } V_\Sigma \rightarrowtail [\Sigma^*, 2])$

...with closure properties dual to property 1/2 of a pseudoequational theory:

- Property 1 of a theory dualizes to closure under preimages.
- Open problem: State dual of Property 2 in terms of languages.

Eilenberg Theorem for Data Languages

$$\left(\begin{array}{c} \textbf{Varieties} \\ \textbf{of languages} \end{array} \right) \cong \left(\begin{array}{c} \textbf{Pseudovarieties} \\ \textbf{of nominal monoids} \end{array} \right)$$

Variety of nominal languages

Class of recognizable nominal languages closed under boolean ops, unions of S-orbits, derivatives, homomorphic preimages and ???.

Pseudovar of nominal monoids

Class of orbit-finite nominal monoids closed under finite products, submonoids and support-reflecting quotients.

. . . where ??? is the closure property that dualizes Property 2 of a pseudoequational theory.

Conclusion

- While our previous categorical approach to algebraic language theory does not directly apply to data languages, the idea "monads + duality" still works.
- New Eilenberg-Reiterman correspondence for data languages.
- Nominal Stone duality? (cf. Gabbay, Litak, Petrişan 2011)
- More general framework that includes **Nom** + previous work as a special case?