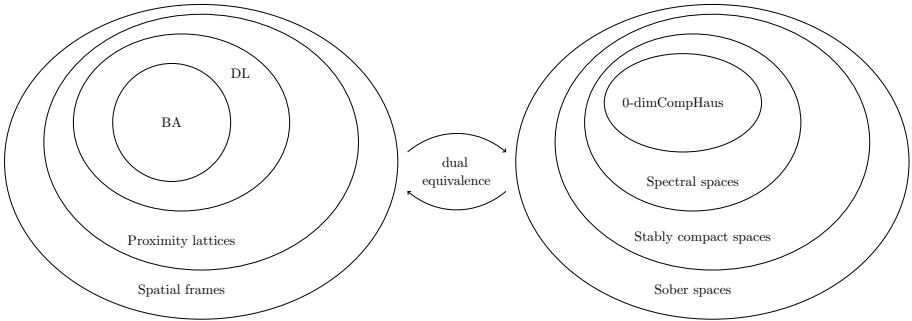
Stone duality in the theory of formal languages

Scandinavian Logic Symposium 2014

Mai Gehrke

CNRS and Paris Diderot

Stone duality

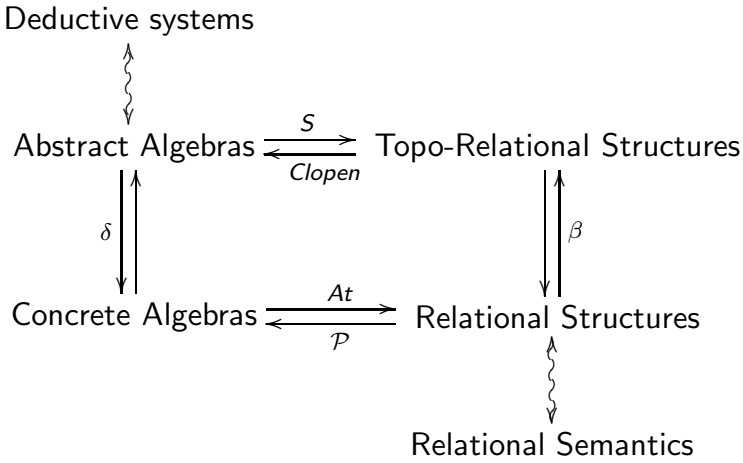


$A \twoheadrightarrow B$ quotient
$A \hookleftarrow B$ subalgebra
$A \oplus B$ coproduct
$A \times B$ product
$f : A^n \to A$ operation

$X \hookleftarrow Y$ subspace
$X \twoheadrightarrow Y$ quotient space
$X \times Y$ product
$X \uplus Y$ sum
$R \subseteq X \times X^n$ dual relation*

* Jonsson-Tarski 1951 for BAOs

Duality theory in semantics I



Duality theory in semantics II

- λ-calculus (a functional calculus allowing self application)

$$\lambda x.xx \xrightarrow{\text{semantics???}} \text{Scott's model}$$

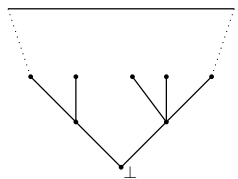
Scott's models are Stone dual spaces

- Domain theory

$$\frac{(M, N) : \sigma \times \tau}{M : \sigma, N : \tau}$$

program logic
for program specification

$\xrightarrow{\text{semantics???}}$



domain

Abramsky: Solutions of domain equations as dual spaces of distributive lattices

Duality theory in logic and computer science

Duality theory has been very successful in semantics. It often plays a role in:

- Completeness: Duality helps in obtaining semantics
- Decidability: Sometimes the dual of a problem is easier to solve.

So far, there have been very few applications of duality theory in complexity theory

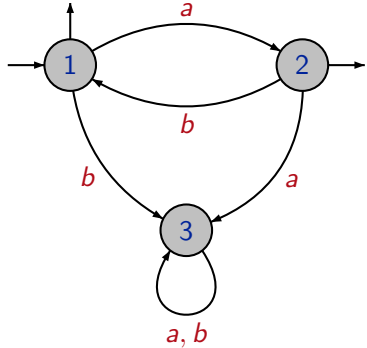
Duality theory in the theory of formal languages

In formal language theory computing machines are studied through corresponding formal languages

Typical problems are decidability, separation, and comparison of complexity classes

In joint work with Serge Grigorieff and Jean-Eric Pin we have shown that duality theory is responsible for the standard tool for proving decidability results in automata theory

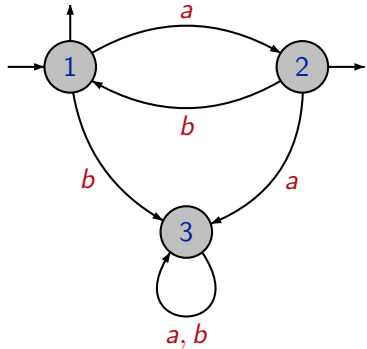
A finite automaton



The states are $\{1, 2, 3\}$.
The initial state is 1, the final states are 1 and 2.
The alphabet is $A = \{a, b\}$ The transitions are

$$1 \cdot a = 2 \qquad 2 \cdot a = 3 \qquad 3 \cdot a = 3$$
$$1 \cdot b = 3 \qquad 2 \cdot b = 1 \qquad 3 \cdot b = 3$$

Recognition by automata



Transitions extend to words: $1 \cdot aba = 2$, $1 \cdot abb = 3$.

The language recognized by the automaton is the set of words u such that $1 \cdot u$ is a final state. Here:

$$L(\mathcal{A}) = (ab)^* \cup (ab)^* a$$

where $*$ means arbitrary iteration of the product.

Rational and recognizable languages

A language is recognizable provided it is recognized by some finite automaton.

A language is rational provided it belongs to the smallest class of languages containing the finite languages which is closed under union, product and star.

Theorem: [Kleene '54] A language is rational iff it is recognizable.

Example: $L(\mathcal{A}) = (ab)^* \cup (ab)^* a$.

Logic on words

To each non-empty word u is associated a structure

$$\mathcal{M}_u = (\{1, 2, \ldots, |u|\}, <, (\mathbf{a})_{a \in A})$$

where $\mathbf{a}$ is interpreted as the set of integers i such that the i-th letter of u is an a, and $<$ as the usual order on integers.

Example:

Let $u = abbaab$ then

$$\mathcal{M}_u = (\{1, 2, 3, 4, 5, 6\}, <, (\mathbf{a}, \mathbf{b}))$$

where $\mathbf{a} = \{1, 4, 5\}$ and $\mathbf{b} = \{2, 3, 6\}$.

Some examples

The formula $\phi = \exists x\ \mathbf{a}x$ interprets as:

> *There exists a position x in u such that the letter in position x is an a.*

This defines the language $L(\phi) = A^*aA^*$.

The formula $\exists x\ \exists y\ (x < y) \wedge \mathbf{a}x \wedge \mathbf{b}y$ defines the language $A^*aA^*bA^*$.

The formula $\exists x\ \forall y\ [(x < y) \vee (x = y)] \wedge \mathbf{a}x$ defines the language aA^*.

Defining the set of words of even length

Macros:

$$(x < y) \vee (x = y) \quad \text{means } x \leqslant y$$
$$\forall y\ x \leqslant y \quad \text{means } x = 1$$
$$\forall y\ y \leqslant x \quad \text{means } x = |u|$$
$$x < y\ \wedge\ \forall z\ (x < z \rightarrow y \leqslant z) \quad \text{means } y = x + 1$$

Let $\phi\ =\ \exists X\ (1 \notin X\ \wedge\ |u| \in X\ \wedge\ \forall x\ (x \in X \leftrightarrow x + 1 \notin X))$

Then $1 \notin X$, $2 \in X$, $3 \notin X$, $4 \in X$, $\ldots$, $|u| \in X$. Thus

$$L(\phi) = \{u \mid |u| \text{ is even}\} = (A^2)^*$$

Monadic second order

Only second order quantifiers over unary predicates are allowed.

Theorem: [Büchi 1960, Elgot 1961]

Monadic second order captures exactly the recognizable languages.

Theorem: [McNaughton-Papert 1971]

First order captures star free languages

(star free = the ones that can be obtained from the alphabet using the Boolean operations on languages and lifted concatenation product only).

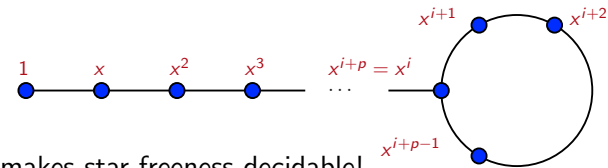
How does one decide whether a given language is star free???

Algebraic theory of automata

Theorem: [Myhill 1953, Rabin-Scott 1959] There is an effective way of associating with each finite automaton, $\mathcal{A}$, a finite monoid, $(M_{\mathcal{A}}, \cdot, 1)$.

Theorem: [Schützenberger 1965] Star free languages correspond to aperiodic monoids, i.e., M such that there exists $n > 0$ with $x^n = x^{n+1}$ for each $x \in M$.

Submonoid generated by x:



This makes star freeness decidable!

An example

$L = (ab)^*$ $\longrightarrow$ $\mathcal{M}(L) =$

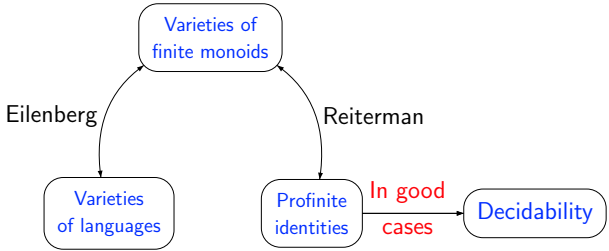
$\cdot$	1	a	ba	b	ab	0
1	1	a	ba	b	ab	0
a	a	0	a	ab	0	0
ba	ba	0	ba	b	0	0
b	b	ba	b	0	b	0
ab	ab	a	0	0	ab	0
0	0	0	0	0	0	0

Syntactic monoid

This monoid is aperiodic since $1 = 1^2$, $a^2 = 0 = a^3$, $ba = ba^2$, $b^2 = 0 = b^3$, $ab = ab^2$, and $0 = 0^2$

Indeed, L is star-free since $L^c = bA^* \cup A^*a \cup A^*aaA^* \cup A^*bbA^*$ and $A^* = \emptyset^c$

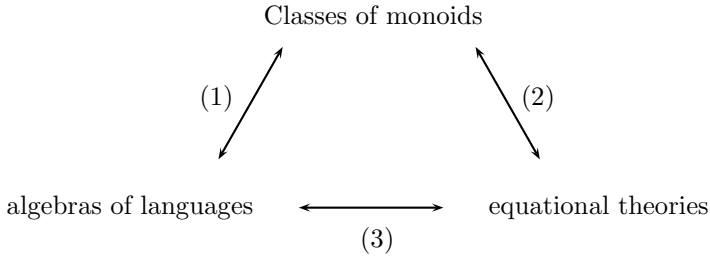
Eilenberg-Reiterman theory



A variety of monoids here means a class of finite monoids closed under homomorphic images, submonoids, and finite products

Various generalisations: [Pin 1995], [Pin-Weil 1996], [Pippenger 1997], [Polák 2001], [Esik 2002], [Straubing 2002], [Kunc 2003]

Eilenberg, Reiterman, and Stone



(1) Eilenberg theorems
(2) Reiterman theorems
(3) extended Stone/Priestley duality

(3) allows generalisation to non-varieties and even to non-regular languages

Connection between duality and Eilenberg-Reiterman I

- The syntactic monoid of a language L is the dual of a certain BAO generated by L in $\mathcal{P}(A^*)$
- The free profinite monoid, $\widehat{A^*}$, is the dual of $\mathrm{Rec}(A^*)$ equipped with certain residuation operations
- Sublattices of $\mathrm{Rec}(A^*)$ correspond via duality to quotients of $\widehat{A^*}$ (and hence equations/pairs in $\widehat{A^*} \times \widehat{A^*}$)

Connection between duality and Eilenberg-Reiterman II

- The dual of a continuous operation

$$\cdot : X \times X \to X$$

should be a coalgebraic structure

$$h : B \to B \oplus B$$

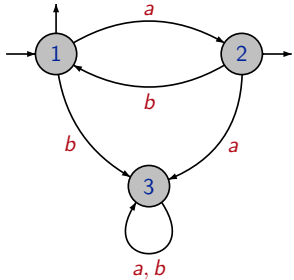
(this is the approach in classical algebra; see also Steinberg and Rhodes)

- It turns out that in an order theoretic setting, the residuals of the product encode this algebra giving and algebraic dual to a topological algebra

- From a lattices and order point of view, residuals are generalised implications, and the pertinent structures are closely related to nuclei.

The residuals of the concatenation product

Consider a finite state automaton



The language recognized by $\mathcal{A}$ is $L(\mathcal{A}) = (ab)^* \cup (ab)^* a$

Quotient operations on languages:

$$a^{-1}L = \{u \in A^* \mid au \in L\} = (ba)^* b \cup (ba)^*$$
$$La^{-1} = \{u \in A^* \mid ua \in L\} = (ab)^*$$
$$b^{-1}L = \{u \in A^* \mid bu \in L\} = \emptyset$$

All recognised by the same underlying machine!

Capturing the underlying machine

Given a recognizable language L the underlying machine is captured by the Boolean algebra $\mathcal{B}(L)$ of languages generated by

$$\left\{ x^{-1}Ly^{-1} \mid x, y \in A^* \right\}$$

NB! This generating set is finite since all the languages are recognized by the same machine with varying sets of initial and final states.

NB! $\mathcal{B}(L)$ is closed under quotients since the quotient operations commute will all the Boolean operations.

The residuation ideal generated by a language

Since $\mathcal{B}(L)$ is finite it is also closed under residuation. That is, for $M \in \mathcal{B}(L)$ and $S \subseteq A^*$

$$S\backslash M = \bigcap_{u \in S} u^{-1}M \in \mathcal{B}(L)$$

$$M/S = \bigcap_{u \in S} Mu^{-1} \in \mathcal{B}(L)$$

These are the upper adjoints in the left and right coordinate of the lifted product on $\mathcal{P}(A^*)$

$$KL \subseteq M \iff L \subseteq K\backslash M \iff K \subseteq M/L$$

$(\mathcal{B}(L), \backslash, /)$ is a Boolean Algebra with additional Operations (BAO)

The syntactic monoid of a recognizable language

[G-Grigorieff-Pin 2008]

The relation dual to $\backslash$ and $/$ on $\mathcal{B}(L)$ is a function

$$f : X \times X \to X$$

Theorem: The dual space of the BAO $(\mathcal{B}(L(\mathcal{A})), \backslash, /)$ is the syntactic monoid of $L(\mathcal{A})$ and the dual of the inclusion $\mathcal{B}(L(\mathcal{A})) \subseteq \mathcal{P}(A^*)$ is a monoid homomorphism $\varphi : A^* \to X$ which satisfies $\varphi^{-1}[\mathcal{P}(X)] = \mathcal{B}(L(\mathcal{A}))$

Boolean topological algebras

We call a topological algebra of some algebraic signature τ Boolean provided the underlying topological space is Boolean (= compact Hausdorff zero-dimensional)

Theorem: Let X be a Boolean space, $f : X^n \to X$ any function, and $R \subseteq X^n \times X$ its graph. The the following are equivalent:

- R is a dual relation with i as the output coordinate for some (and then for all) $1 \leqslant i \leqslant n$
- f is continuous

Corollary: All Boolean topological algebras are dual spaces of certain residuation algebras (as are all Priestley topological algebras)

Duals of topological algebra morphisms...

...are different (and incomparable) to residuation algebra morphisms in general

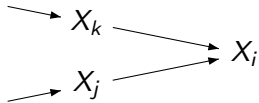
A special well behaved case: the dual of a Boolean topological algebra quotient is a Boolean residuation ideal:

$C \hookrightarrow B$ Boolean residuation subalgebra with $b \backslash c$ and $c / b \in C$ for all $b \in B$ and $c \in C$

Characterization of profinite algebras

The inverse limit system $\mathcal{F}$

$\varprojlim \mathcal{F} = X$



All the X_i's are finite topological algebra quotients, so by duality the dual Boolean residuation algebra is a directed union of finite Boolean residuation ideals

Theorem: A Boolean topological algebra X is profinite iff each finitely generated Boolean residuation ideal of the dual algebra is finite

Profinite completions

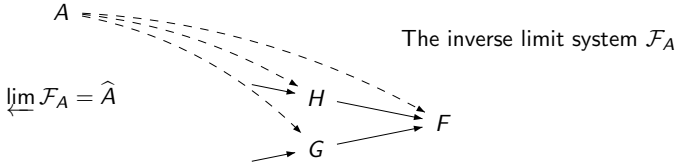
Let A be a (discrete) abstract algebra of any signature (N.B.! A is not an alphabet here!)

We define the recognisable subsets of A to be

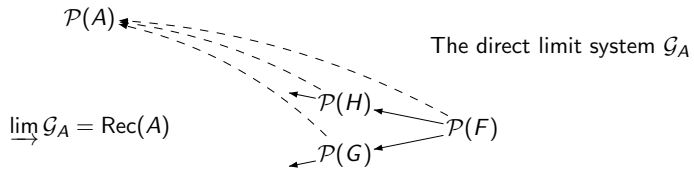
$$\mathrm{Rec}(A) = \{\varphi^{-1}(P) \mid \varphi : A \twoheadrightarrow F \text{ finite quotient and } P \subseteq F\}$$

Theorem: [G-Grigorieff-Pin 2008] The profinite completion of ANY algebra is the dual space of the BAO $\mathrm{Rec}(A)$ with the residuals of the lifted operations

Profinite completions proof sketch

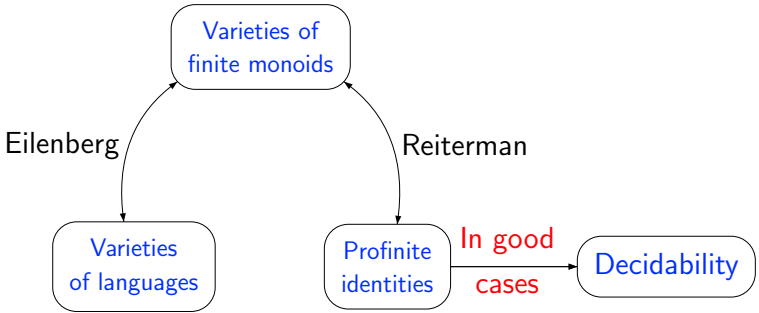


is dual to



$$\varinjlim \mathcal{G}_A = \bigcup \{\varphi^{-1}(\mathcal{P}(F)) \mid \varphi : A \twoheadrightarrow F \text{ finite quotient}\}$$
$$= \{\varphi^{-1}(P) \mid \varphi : A \twoheadrightarrow F \text{ finite quotient and } P \subseteq F\} = \mathrm{Rec}(A)$$

Eilenberg-Reiterman theory



[Eilenberg76] + [Reiterman82]

Characterizing subclasses of languages

[G-Grigorieff-Pin 2008]

subalgebras $\longleftrightarrow$ quotient structures

$\mathcal{C}$ a class of recognizable languages closed under $\cap$ and $\cup$

$$\mathcal{C} \hookrightarrow Rec(A^*)$$

DUALLY

$$X_{\mathcal{C}} \twoheadleftarrow \widehat{A^*}$$

That is, $\mathcal{C}$ is described dually by EQUATING elements of $\widehat{A^*}$.

This is a general form of Eilenberg-Reiterman theorem

A Galois connection for subsets of an algebra

Let B be a Boolean algebra, X the dual space of B.

The maps $\mathcal{P}(B) \leftrightarrows \mathcal{P}(X \times X)$ given by

$$S \mapsto \approx_S = \{(x, y) \in X \mid \forall b \in S \quad (b \in y \iff b \in x)\}$$

and

$$E \mapsto B_E = \{b \in B \mid \forall (x, y) \in E \quad (b \in y \iff b \in x)\}$$

establish a Galois connection whose Galois closed sets are the Boolean equivalence relations and the Boolean subalgebras, respectively.

Example

[Schützenberger 1965]

The equivalence relation on $\widehat{A^*}$ dual to the residuation ideal

Star-free languages $\leqslant$ Rec(A^*)

is generated in the Galois connection of the previous slide by the set

$$\{(ux^{\omega+1}v, ux^{\omega}v) \mid x, u, v \in \widehat{A^*}\}$$

That is, it is given by ONE pair, $(a^{\omega+1}, a^{\omega})$, when closing under:

- substitution
- monoid congruence
- Stone duality subalgebra-quotient adjunction

Beyond regular languages

The goal of circuit complexity theory is to classify problems by the size and/or depth of the Boolean circuits needed to solve them.

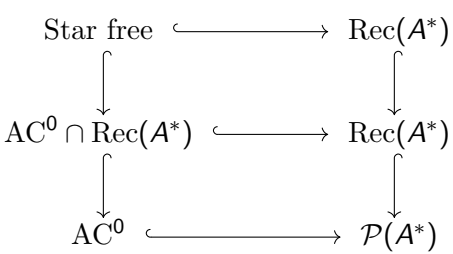
A very low such class is AC^0 which corresponds to constant-depth, unbounded-fanin, polynomial-size circuits with AND, OR, and NOT gates. Let $\mathcal{N}$ denote the set of all numerical predicates

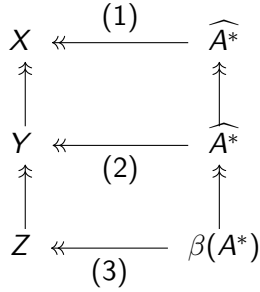
Recall the McNaughton-Papert result: Star free $= FO[<, (\mathbf{a})_{a \in A}]$

[Immerman 1989] and [Stockmeyer and Vishkin 1984]: $\mathrm{AC}^0 = FO[\mathcal{N}, (\mathbf{a})_{a \in A}]$

Research question: Can we develop an equational theory for circuit complexity classes?

Finding an equational basis for AC^0





(1) is given by $x^{\omega+1} = x^{\omega}$

(2) is given by $(x^{\omega-1}y)^{\omega+1} = (x^{\omega-1}y)^{\omega}$ for x and y of the same length
— a very difficult result by [Barrington, Straubing, Thérien 1990]

Can we get equations for (3) and recover (2) from these? How to get β-equations?

A first step

First we consider

$$\mathcal{B} \;=\; FO[\mathcal{N}_0, \mathcal{N}_1, (\mathbf{a})_{a \in A}]$$

That is, arbitrary nullary and unary predicates, no higher arity predicates, not even $=$!

$\mathcal{B}$ is generated as a Boolean algebra by the sets

$$L_P = \{u \in A^* \mid |u| \in P\}$$
$$L_P^a = \{u \in A^* \mid u_i = a \implies i \in P\}$$

for $P \subseteq \mathbb{N}$ and $a \in A$

Equations for $\mathcal{B}$

$A^* \times \mathbb{N}^2$ we think of as 'words with two spots'. Define

$$\begin{aligned} f_{ab} : A^* \times \mathbb{N}^2 &\longrightarrow A^* \\ (u, i, j) &\mapsto u(a@i, b@j) \end{aligned}$$

where the substitutions happen only when $i, j \leqslant |u|$

By duality or Stone-Čech compactification, we obtain

$$\beta f_{ab} : \beta(A^* \times \mathbb{N}^2) \longrightarrow \beta(A^*)$$

$\gamma \in \beta(A^* \times \mathbb{N}^2)$ are generalised 'words with two spots'

N.B.! This is not the same as 'generalised words' with two 'generalised spots'

Equations for $\mathcal{B}$

For $n = 1$ and 2, the maps

$$\beta\pi_n : \beta(A^* \times \mathbb{N}^2) \to \beta(\mathbb{N}), (u, i_1, i_2) \mapsto i_n$$

Give the generalised spots associated with a γ

Theorem: [G-Krebs-Pin 2014] $L \in \mathcal{B}$ if and only if

$$L \vDash \beta f_{ab}(\gamma) = \beta f_{ba}(\gamma)$$

for all $a, b \in A$ and all $\gamma \in \beta(A^* \times \mathbb{N}^2)$ with $\beta\pi_1(\gamma) = \beta\pi_2(\gamma)$ and

$$L \vDash \beta f_{abb}(\gamma) = \beta f_{aab}(\gamma)$$

for all $a, b \in A$ and all $\gamma \in \beta(A^* \times \mathbb{N}^3)$ with $\beta\pi_1(\gamma) = \beta\pi_2(\gamma) = \beta\pi_3(\gamma)$

Equations for $\mathcal{B} \cap \mathrm{Rec}(\mathrm{A}^*)$ by projection

Theorem: [G-Krebs-Pin 2014] $L \in \mathcal{B} \cap \mathrm{Rec}(\mathrm{A}^*)$ if and only if

$$L \models (x^{\omega-1}s)(x^{\omega-1}t) = (x^{\omega-1}t)(x^{\omega-1}s)$$

for all $x, s, t \in \widehat{A^*}$ of the same length and

$$L \models (x^{\omega-1}s)^2 = x^{\omega-1}s$$

for all $x, s \in \widehat{A^*}$ of the same length

References

1. Mai Gehrke, Serge Grigorieff, and Jean-Éric Pin, Duality and Equational Theory of Regular Languages, *LNCS* (ICALP) **5125** (2008), 246–257.

2. Mai Gehrke, Stone duality, topological algebra, and recognition, preprint. See, `http://hal.archives-ouvertes.fr/hal-00859717`

3. Mai Gehrke, Andreas Krebs, and Jean-Éric Pin, From ultrafilters on words to the expressive power of a fragment of logic, to appear in *Proceedings of the 16th International Workshop on Descriptional Complexity of Formal Systems*, 2014.