Stone duality and model theory

Sam v. Gool and Benjamin Steinberg

University of Amsterdam and City College of New York

October 8, 2016
AMS Fall Western Sectional Meeting
Special Session on Algebraic Logic
University of Denver

Introduction

In the theory of regular languages of finite words, profinite monoids are useful canonical objects.

Introduction

In the theory of regular languages of finite words, profinite monoids are useful canonical objects.

Such monoids are Stone dual spaces of certain Boolean algebras.

Introduction

In the theory of regular languages of finite words, profinite monoids are useful canonical objects.

Such monoids are Stone dual spaces of certain Boolean algebras.

These Boolean algebras, in turn, are Lindenbaum algebras of certain logical theories.

Introduction

In the theory of regular languages of finite words, profinite monoids are useful canonical objects.

Such monoids are Stone dual spaces of certain Boolean algebras.

These Boolean algebras, in turn, are Lindenbaum algebras of certain logical theories.

How can we use techniques from logic to study profinite monoids?

Introduction

In the theory of regular languages of finite words, profinite monoids are useful canonical objects.

Such monoids are Stone dual spaces of certain Boolean algebras.

These Boolean algebras, in turn, are Lindenbaum algebras of certain logical theories.

How can we use techniques from logic to study profinite monoids?

Which of these techniques are useful more generally, in other (profinite) contexts?

First-order definable languages

When defining languages with logic, a finite A-word is seen as a structure with a binary relation $<$ and unary letter predicates $(P_a)_{a \in A}$.

First-order definable languages

When defining languages with logic, a finite A-word is seen as a structure with a binary relation $<$ and unary letter predicates $(P_a)_{a \in A}$.

For example, the finite $\{a, b\}$-word

$$abbaa$$

is $(\{1,2,3,4,5\}, <, P_a, P_b)$ with $P_a = \{1,4,5\}$ and $P_b = \{2,3\}$. The first-order sentence

$$\exists x \exists y (x < y \wedge P_a(x) \wedge P_b(x))$$

is true in this word, but false in, e.g., bba.

First-order definable languages

When defining languages with logic, a finite A-word is seen as a structure with a binary relation $<$ and unary letter predicates $(P_a)_{a \in A}$.

For example, the finite $\{a, b\}$-word

$$abbaa$$

is $(\{1, 2, 3, 4, 5\}, <, P_a, P_b)$ with $P_a = \{1, 4, 5\}$ and $P_b = \{2, 3\}$. The first-order sentence

$$\exists x \exists y (x < y \wedge P_a(x) \wedge P_b(x))$$

is true in this word, but false in, e.g., bba.
Any first-order sentence ϕ in the signature $\mathcal{L}^A = \{<\} \cup \{P_a \mid a \in A\}$ defines the language L_ϕ of finite A-words in which ϕ is true, i.e., $L_\phi := \mathrm{Mod}_{fin}(\phi)$.

First-order definability and aperiodicity

Theorem (McNaughton & Papert 1971, Schützenberger 1965)

A language is first-order definable if, and only if, it is recognizable by a finite aperiodic monoid.

First-order definability and aperiodicity

Theorem (McNaughton & Papert 1971, Schützenberger 1965)

A language is first-order definable if, and only if, it is recognizable by a finite aperiodic monoid.

Here,

- a language $L \subseteq A^*$ is recognizable by a finite monoid M if there is a homomorphism $\eta \colon A^* \to M$ with $L = \eta^{-1}\eta(L)$,

First-order definability and aperiodicity

Theorem (McNaughton & Papert 1971, Schützenberger 1965)

A language is first-order definable if, and only if, it is recognizable by a finite aperiodic monoid.

Here,

- a language $L \subseteq A^*$ is recognizable by a finite monoid M if there is a homomorphism $\eta \colon A^* \to M$ with $L = \eta^{-1}\eta(L)$,
- a monoid is aperiodic if it has no non-trivial *subgroups*.

First-order definability and aperiodicity

Theorem (McNaughton & Papert 1971, Schützenberger 1965)

A language is first-order definable if, and only if, it is recognizable by a finite aperiodic monoid.

Here,

- a language $L \subseteq A^*$ is recognizable by a finite monoid M if there is a homomorphism $\eta \colon A^* \to M$ with $L = \eta^{-1}\eta(L)$,
- a monoid is aperiodic if it has no non-trivial *subgroups*.

The class of finite aperiodic monoids is denoted by **A**.

First-order definability and aperiodicity

Theorem (McNaughton & Papert 1971, Schützenberger 1965)

A language is first-order definable if, and only if, it is recognizable by a finite aperiodic monoid.

Here,

- a language $L \subseteq A^*$ is recognizable by a finite monoid M if there is a homomorphism $\eta\colon A^* \to M$ with $L = \eta^{-1}\eta(L)$,
- a monoid is aperiodic if it has no non-trivial *subgroups*.

The class of finite aperiodic monoids is denoted by $\mathbf{A}$.

Observation

The assignment $\phi \mapsto L_\phi$ defines an isomorphism between the Lindenbaum algebra $\mathbb{L}(\mathcal{T}^A_{\mathrm{fin}})$ of the first-order theory of finite A-words and the Boolean algebra $\mathrm{Rec}_{\mathbf{A}}(A)$ of $\mathbf{A}$-recognizable languages.

Profinite monoids

A profinite monoid is a topological monoid whose underlying space is Boolean (compact, T_2, 0-dim).

Profinite monoids

A profinite monoid is a topological monoid whose underlying space is Boolean (compact, T_2, 0-dim).

Alternatively, a profinite monoid is a projective limit of finite discrete monoids in the category of topological monoids.

Profinite monoids

A profinite monoid is a topological monoid whose underlying space is Boolean (compact, T_2, 0-dim).

Alternatively, a profinite monoid is a projective limit of finite discrete monoids in the category of topological monoids.

The free profinite monoid over a finite set A, $\widehat{A^*}$, has as its clopen sets the closures of languages L over A that are recognizable by a finite monoid.

Pro-$\mathbf{V}$ monoids

A (pseudo)variety $\mathbf{V}$ is a collection of finite monoids closed under submonoids, homomorphic images, and finite products.

Pro-**V** monoids

A (pseudo)variety **V** is a collection of finite monoids closed under submonoids, homomorphic images, and finite products.

A pro-**V** monoid is a projective limit of finite monoids in **V**.

Pro-**V** monoids

A (pseudo)variety **V** is a collection of finite monoids closed under submonoids, homomorphic images, and finite products.

A pro-**V** monoid is a projective limit of finite monoids in **V**.

For any finite alphabet A, the free pro-**V** monoid over A, $\widehat{F}_{\mathbf{V}}(A)$, is the pro-**V** monoid characterized by the usual universal property.

Pro-$\mathbf{V}$ monoids

A (pseudo)variety $\mathbf{V}$ is a collection of finite monoids closed under submonoids, homomorphic images, and finite products.

A pro-$\mathbf{V}$ monoid is a projective limit of finite monoids in $\mathbf{V}$.

For any finite alphabet A, the free pro-$\mathbf{V}$ monoid over A, $\widehat{F}_{\mathbf{V}}(A)$, is the pro-$\mathbf{V}$ monoid characterized by the usual universal property.

The clopen sets of $\widehat{F}_{\mathbf{V}}(A)$ are the closures of $\mathbf{V}$-recognizable languages.

Pro-V monoids

A (pseudo)variety **V** is a collection of finite monoids closed under submonoids, homomorphic images, and finite products.

A pro-**V** monoid is a projective limit of finite monoids in **V**.

For any finite alphabet A, the free pro-**V** monoid over A, $\widehat{F}_{\mathbf{V}}(A)$, is the pro-**V** monoid characterized by the usual universal property.

The clopen sets of $\widehat{F}_{\mathbf{V}}(A)$ are the closures of **V**-recognizable languages. Thus, $\widehat{F}_{\mathbf{V}}(A)$ is the Stone dual space of the Boolean algebra $\mathrm{Rec}_{\mathbf{V}}(A)$:

Pro-**V** monoids

A (pseudo)variety **V** is a collection of finite monoids closed under submonoids, homomorphic images, and finite products.

A pro-**V** monoid is a projective limit of finite monoids in **V**.

For any finite alphabet A, the free pro-**V** monoid over A, $\widehat{F}_{\mathbf{V}}(A)$, is the pro-**V** monoid characterized by the usual universal property.

The clopen sets of $\widehat{F}_{\mathbf{V}}(A)$ are the closures of **V**-recognizable languages. Thus, $\widehat{F}_{\mathbf{V}}(A)$ is the Stone dual space of the Boolean algebra $\mathrm{Rec}_{\mathbf{V}}(A)$:

$$\widehat{F}_{\mathbf{V}}(A) = \mathrm{Rec}_{\mathbf{V}}(A)_{\star}$$

In particular, by our earlier observation:

$$\widehat{F}_{\mathbf{A}}(A) = \mathbb{L}(\mathcal{T}_{\mathrm{fin}}^{A})_{\star}$$

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The points of the Stone space $\mathbb{L}(\mathcal{T})_\star$ are the complete theories extending $\mathcal{T}$.

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The points of the Stone space $\mathbb{L}(\mathcal{T})_\star$ are the complete theories extending $\mathcal{T}$.

Alternatively, the points of $\mathbb{L}(\mathcal{T})_\star$ are elementary equivalence classes of $\mathcal{T}$-models.

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The points of the Stone space $\mathbb{L}(\mathcal{T})_\star$ are the complete theories extending $\mathcal{T}$.

Alternatively, the points of $\mathbb{L}(\mathcal{T})_\star$ are elementary equivalence classes of $\mathcal{T}$-models.

The topology is given by the basis of clopen sets $\{\widehat{\phi} \mid \phi \text{ a sentence}\}$, where $\widehat{\phi} = \{[M]_\equiv \mid M \models \phi\}$.

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The points of the Stone space $\mathbb{L}(\mathcal{T})_\star$ are the complete theories extending $\mathcal{T}$.

Alternatively, the points of $\mathbb{L}(\mathcal{T})_\star$ are elementary equivalence classes of $\mathcal{T}$-models.

The topology is given by the basis of clopen sets $\{\widehat{\phi} \mid \phi \text{ a sentence}\}$, where $\widehat{\phi} = \{[M]_\equiv \mid M \models \phi\}$.

In particular, $\widehat{F_{\mathbf{A}}}(A) = \mathbb{L}(\mathcal{T}^A_{\mathrm{fin}})_\star$ is the space of elementary equivalence classes of pseudofinite A-words.

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The points of the Stone space $\mathbb{L}(\mathcal{T})_\star$ are the complete theories extending $\mathcal{T}$.

Alternatively, the points of $\mathbb{L}(\mathcal{T})_\star$ are elementary equivalence classes of $\mathcal{T}$-models.

The topology is given by the basis of clopen sets $\{\widehat{\phi} \mid \phi \text{ a sentence}\}$, where $\widehat{\phi} = \{[M]_\equiv \mid M \models \phi\}$.

In particular, $\widehat{F}_{\mathbf{A}}(A) = \mathbb{L}(\mathcal{T}^A_{\mathrm{fin}})_\star$ is the space of elementary equivalence classes of pseudofinite A-words.

Thus, the above observations give a new perspective on the topological monoid $\widehat{F}_{\mathbf{A}}(A)$.

The Stone space of a first-order theory

Let $\mathcal{T}$ be a first-order theory.

The points of the Stone space $\mathbb{L}(\mathcal{T})_\star$ are the complete theories extending $\mathcal{T}$.

Alternatively, the points of $\mathbb{L}(\mathcal{T})_\star$ are elementary equivalence classes of $\mathcal{T}$-models.

The topology is given by the basis of clopen sets $\{\widehat{\phi} \mid \phi \text{ a sentence}\}$, where $\widehat{\phi} = \{[M]_\equiv \mid M \models \phi\}$.

In particular, $\widehat{F_{\mathbf{A}}}(A) = \mathbb{L}(\mathcal{T}^A_{\text{fin}})_\star$ is the space of elementary equivalence classes of pseudofinite A-words.

Thus, the above observations give a new perspective on the topological monoid $\widehat{F_{\mathbf{A}}}(A)$. How to use it?

Pseudofinite A-words

An *A-word* is a $\mathcal{L}_A$-structure where $<$ is a discrete linear order with endpoints and the predicates P_a form a partition.

Pseudofinite A-words

An A-word is a $\mathcal{L}_A$-structure where $<$ is a discrete linear order with endpoints and the predicates P_a form a partition.

A pseudofinite A-word is, by definition, a model of the theory $\mathcal{T}_{\mathrm{fin}}^A$.

Pseudofinite A-words

An A-word is a $\mathcal{L}_A$-structure where $<$ is a discrete linear order with endpoints and the predicates P_a form a partition.

A pseudofinite A-word is, by definition, a model of the theory $\mathcal{T}^A_{\mathrm{fin}}$.

Theorem

The theory $\mathcal{T}^A_{\mathrm{fin}}$ is not finitely axiomatizable.

Pseudofinite A-words

An A-word is a $\mathcal{L}_A$-structure where $<$ is a discrete linear order with endpoints and the predicates P_a form a partition.

A pseudofinite A-word is, by definition, a model of the theory $\mathcal{T}_{\text{fin}}^A$.

Theorem

The theory $\mathcal{T}_{\text{fin}}^A$ is not finitely axiomatizable.

An A-word is pseudofinite iff it satisfies a certain induction scheme.

Pseudofinite A-words

An A-word is a $\mathcal{L}_A$-structure where $<$ is a discrete linear order with endpoints and the predicates P_a form a partition.

A pseudofinite A-word is, by definition, a model of the theory $\mathcal{T}_{\mathrm{fin}}^{A}$.

Theorem

The theory $\mathcal{T}_{\mathrm{fin}}^{A}$ is not finitely axiomatizable.

An A-word is pseudofinite iff it satisfies a certain induction scheme.

E.g., $aaa\cdots \quad \cdots bbb$ is not pseudofinite.

Concatenation and substitution

Any two A-words can be concatenated in the obvious way. This operation respects elementary equivalence. In this way, $\mathbb{L}(\mathcal{T}^{A}_{\text{fin}})_{\star}$ is a monoid.

Concatenation and substitution

Any two A-words can be concatenated in the obvious way. This operation respects elementary equivalence. In this way, $\mathbb{L}(\mathcal{T}^A_{\mathrm{fin}})_\star$ is a monoid.

Concatenation of pseudofinite A-words corresponds to a residuation operation on the Lindenbaum algebra: if ϕ and ψ are first-order sentences, then ϕ/ψ is a first-order sentence defining the language

$$L_{\phi/\psi} = \{u \mid \text{whenever } v \models \psi, \text{ we have } uv \models \phi\}.$$

Concatenation and substitution

Any two A-words can be concatenated in the obvious way. This operation respects elementary equivalence. In this way, $\mathbb{L}(\mathcal{T}_{\mathrm{fin}}^{A})_{\star}$ is a monoid.

Concatenation of pseudofinite A-words corresponds to a residuation operation on the Lindenbaum algebra: if ϕ and ψ are first-order sentences, then ϕ/ψ is a first-order sentence defining the language

$$L_{\phi/\psi} = \{u \mid \text{whenever } v \models \psi, \text{ we have } uv \models \phi\}.$$

It is not obvious (to me) how to construct this first-order sentence directly from ϕ and ψ.

Concatenation and substitution

Any two A-words can be concatenated in the obvious way. This operation respects elementary equivalence. In this way, $\mathbb{L}(\mathcal{T}^A_{\mathrm{fin}})_\star$ is a monoid.

Concatenation of pseudofinite A-words corresponds to a residuation operation on the Lindenbaum algebra: if ϕ and ψ are first-order sentences, then ϕ/ψ is a first-order sentence defining the language

$$L_{\phi/\psi} = \{u \mid \text{whenever } v \models \psi, \text{ we have } uv \models \phi\}.$$

It is not obvious (to me) how to construct this first-order sentence directly from ϕ and ψ.

More generally, any assignment $B \to \widehat{F}_{\mathbf{A}}(A)$ induces a substitution of pseudofinite A-words into pseudofinite B-words. These substitutions give precisely the continuous homomorphisms $\widehat{F}_{\mathbf{A}}(B) \to \widehat{F}_{\mathbf{A}}(A)$.

Types and weakly saturated A-words

A type is a triple (u, a, v) with $u, v \in \widehat{F}_{\mathbf{A}}(A)$ and $a \in A$. The type space is $\widehat{F}_{\mathbf{A}}(A) \times A \times \widehat{F}_{\mathbf{A}}(A)$.

Types and weakly saturated A-words

A type is a triple (u, a, v) with $u, v \in \widehat{F}_{\mathbf{A}}(A)$ and $a \in A$. The type space is $\widehat{F}_{\mathbf{A}}(A) \times A \times \widehat{F}_{\mathbf{A}}(A)$.

If U is an A-word, and i is a position in U, then

$$\mathrm{tp}^{U}(i) := (U(<i), U(i), U(>i))$$

is the type realized in U by i.

Types and weakly saturated A-words

A type is a triple (u, a, v) with $u, v \in \widehat{F}_{\mathbf{A}}(A)$ and $a \in A$. The type space is $\widehat{F}_{\mathbf{A}}(A) \times A \times \widehat{F}_{\mathbf{A}}(A)$.

If U is an A-word, and i is a position in U, then

$$\mathrm{tp}^{U}(i) := (U(<i), U(i), U(>i))$$

is the type realized in U by i.

A type t is consistent with an A-word U if t is realized in some A-word V that is elementarily equivalent to U.

Types and weakly saturated A-words

A type is a triple (u, a, v) with $u, v \in \widehat{F}_{\mathbf{A}}(A)$ and $a \in A$. The type space is $\widehat{F}_{\mathbf{A}}(A) \times A \times \widehat{F}_{\mathbf{A}}(A)$.

If U is an A-word, and i is a position in U, then

$$\mathrm{tp}^{U}(i) := (U({<}i), U(i), U({>}i))$$

is the type realized in U by i.

A type t is consistent with an A-word U if t is realized in some A-word V that is elementarily equivalent to U. An A-word U is weakly saturated if U realizes all types consistent with it.

Types and weakly saturated A-words

A type is a triple (u, a, v) with $u, v \in \widehat{F}_{\mathbf{A}}(A)$ and $a \in A$. The type space is $\widehat{F}_{\mathbf{A}}(A) \times A \times \widehat{F}_{\mathbf{A}}(A)$.

If U is an A-word, and i is a position in U, then

$$\mathrm{tp}^U(i) := (U(<i), U(i), U(>i))$$

is the type realized in U by i.

A type t is consistent with an A-word U if t is realized in some A-word V that is elementarily equivalent to U. An A-word U is weakly saturated if U realizes all types consistent with it.

Proposition

The set of types consistent with U is the topological closure in the type space of the set of types realized in U. Thus, U is weakly saturated iff the set of realized types is closed.

ω-saturated A-words

An A-word U is ω-saturated if every interval in U is weakly saturated.

ω-saturated A-words

An A-word U is ω-saturated if every interval in U is weakly saturated.

Theorem (Model theory)

Any elementary equivalence class contains an ω-saturated model.

ω-saturated A-words

An A-word U is ω-saturated if every interval in U is weakly saturated.

Theorem (Model theory)

Any elementary equivalence class contains an ω-saturated model.

Theorem (v. Gool & Steinberg)

A substitution of ω-saturated models into ω-saturated models is again ω-saturated. In particular, ω-saturated models are closed under concatenation.

ω-saturated A-words

An A-word U is ω-saturated if every interval in U is weakly saturated.

Theorem (Model theory)

Any elementary equivalence class contains an ω-saturated model.

Theorem (v. Gool & Steinberg)

A substitution of ω-saturated models into ω-saturated models is again ω-saturated. In particular, ω-saturated models are closed under concatenation.

The proof combines the topological characterization of weak saturation and the continuity of substitutions.

First application: equidivisibility

A monoid M is called equidivisible if for any u, v, u', v' in M, $uv = u'v'$ implies that there exists x in M such that $ux = u'$ and $xv' = v$, or $u'x = u$ and $xv = v'$.

First application: equidivisibility

A monoid M is called equidivisible if for any u, v, u', v' in M, $uv = u'v'$ implies that there exists x in M such that $ux = u'$ and $xv' = v$, or $u'x = u$ and $xv = v'$.

Proposition

The monoid $\widehat{F}_{\mathbf{A}}(A)$ is equidivisible.

First application: equidivisibility

A monoid M is called equidivisible if for any u, v, u', v' in M, $uv = u'v'$ implies that there exists x in M such that $ux = u'$ and $xv' = v$, or $u'x = u$ and $xv = v'$.

Proposition

The monoid $\widehat{F_{\mathbf{A}}}(A)$ is equidivisible.

Proof.

Let $w := uv = u'v'$.

First application: equidivisibility

A monoid M is called equidivisible if for any u, v, u', v' in M, $uv = u'v'$ implies that there exists x in M such that $ux = u'$ and $xv' = v$, or $u'x = u$ and $xv = v'$.

Proposition

The monoid $\widehat{F}_{\mathbf{A}}(A)$ is equidivisible.

Proof.

Let $w := uv = u'v'$.

Pick an ω-saturated model W in w.

First application: equidivisibility

A monoid M is called equidivisible if for any u, v, u', v' in M, $uv = u'v'$ implies that there exists x in M such that $ux = u'$ and $xv' = v$, or $u'x = u$ and $xv = v'$.

Proposition

The monoid $\widehat{F_{\mathbf{A}}}(A)$ is equidivisible.

Proof.

Let $w := uv = u'v'$.

Pick an ω-saturated model W in w.

Draw a picture. □

More applications

- A new proof of decidability of the word problem for aperiodic ω-terms.

More applications

- A new proof of decidability of the word problem for aperiodic ω-terms.
  - Key: countably saturated models are unique up to isomorphism in an elementary equivalence class.

More applications

- A new proof of decidability of the word problem for aperiodic ω-terms.
  - Key: countably saturated models are unique up to isomorphism in an elementary equivalence class.
- A detailed analysis of factors of substitutions.

More applications

- A new proof of decidability of the word problem for aperiodic ω-terms.
  - Key: countably saturated models are unique up to isomorphism in an elementary equivalence class.
- A detailed analysis of factors of substitutions.
- Well-quasi-orders of factors, regularity of finite factor languages, are stable under substitutions.

More applications

- A new proof of decidability of the word problem for aperiodic ω-terms.
  - Key: countably saturated models are unique up to isomorphism in an elementary equivalence class.
- A detailed analysis of factors of substitutions.
- Well-quasi-orders of factors, regularity of finite factor languages, are stable under substitutions.
- See our technical report arXiv:1609.07736 (or tomorrow morning. . .).

Where to go from here

- Do these methods extend to Monadic Second Order logic?

Where to go from here

- Do these methods extend to Monadic Second Order logic?
- Are these methods useful in other profinite contexts?

Where to go from here

- Do these methods extend to Monadic Second Order logic?
- Are these methods useful in other profinite contexts?
- Are other logical techniques useful? (Omitting types?)

Where to go from here

- Do these methods extend to Monadic Second Order logic?
- Are these methods useful in other profinite contexts?
- Are other logical techniques useful? (Omitting types?)
- Are there more algorithmic applications in language theory?

Where to go from here

- Do these methods extend to Monadic Second Order logic?
- Are these methods useful in other profinite contexts?
- Are other logical techniques useful? (Omitting types?)
- Are there more algorithmic applications in language theory?
- Your questions?