From algebra to logic: There and back again the story of a hierarchy

Pascal Weil (LaBRI, CNRS and Université de Bordeaux)

DLT 2014, Ural Federal University, August 2014

Formal language theory

- Historically: formal language theory (FLT) first dealt with automata — and almost immediately, with logic (Büchi)

Formal language theory

- Historically: formal language theory (FLT) first dealt with automata — and almost immediately, with logic (Büchi)
- Very soon, algebra came into play: syntactic monoid (Elgot, Myhill and later, Schützenberger)

Formal language theory

- Historically: formal language theory (FLT) first dealt with automata — and almost immediately, with logic (Büchi)
- Very soon, algebra came into play: syntactic monoid (Elgot, Myhill and later, Schützenberger)
- Algebra useful for classification (variety theory, Eilenberg *et al.*) and for decision algorithms

Formal language theory

- Historically: formal language theory (FLT) first dealt with automata — and almost immediately, with logic (Büchi)
- Very soon, algebra came into play: syntactic monoid (Elgot, Myhill and later, Schützenberger)
- Algebra useful for classification (variety theory, Eilenberg *et al.*) and for decision algorithms
- FLT later also became useful in algebra and more ingredients came into FLT, especially topology: this will not appear today

Formal language theory

- Historically: formal language theory (FLT) first dealt with automata — and almost immediately, with logic (Büchi)
- Very soon, algebra came into play: syntactic monoid (Elgot, Myhill and later, Schützenberger)
- Algebra useful for classification (variety theory, Eilenberg *et al.*) and for decision algorithms
- FLT later also became useful in algebra and more ingredients came into FLT, especially topology: this will not appear today
- The hierarchy of langages discussed today, is an illustration of this interaction between FLT, algebra and logic

Logic and languages: which logic?

$$\forall x\, \forall y\, ((x < y) \wedge \neg(\exists z\; x < z < y) \Rightarrow (R_a x \Leftrightarrow R_b y))$$
$$\wedge \quad \forall x\, (\neg(\exists y\; y < x) \Rightarrow R_a x$$

specifies the language $(ab)^* \cup (ab)^* b$

- a powerful specification tool

Logic and languages: which logic?

$$\forall x\, \forall y\, ((x < y) \wedge \neg(\exists z\; x < z < y) \Rightarrow (R_a x \Leftrightarrow R_b y))$$
$$\wedge \quad \forall x\, (\neg(\exists y\; y < x) \Rightarrow R_a x$$

specifies the language $(ab)^* \cup (ab)^* b$

- a powerful specification tool
- L is regular if and only if L is MSO-definable (Büchi 1960)

Logic and languages: which logic?

$$\forall x\, \forall y\, ((x < y) \wedge \neg(\exists z\; x < z < y) \Rightarrow (R_a x \Leftrightarrow R_b y))$$
$$\wedge \quad \forall x\, (\neg(\exists y\; y < x) \Rightarrow R_a x$$

specifies the language $(ab)^* \cup (ab)^* b$

- a powerful specification tool
- L is regular if and only if L is MSO-definable (Büchi 1960)
- A natural limitation (also from the algorithmic point of view): restriction to first-order logic

Logic and languages: which logic?

$$\forall x\ \forall y\ ((x < y) \wedge \neg(\exists z\ x < z < y) \Rightarrow (R_a x \Leftrightarrow R_b y))$$
$$\wedge \quad \forall x\ (\neg(\exists y\ y < x) \Rightarrow R_a x$$

specifies the language $(ab)^* \cup (ab)^* b$

- a powerful specification tool
- L is regular if and only if L is MSO-definable (Büchi 1960)
- A natural limitation (also from the algorithmic point of view): restriction to first-order logic
- As it turns out, many important properties of complex systems can be specified in FO

Logic and languages: which logic?

$$\forall x\, \forall y\, ((x < y) \wedge \neg(\exists z\; x < z < y) \Rightarrow (R_a x \Leftrightarrow R_b y))$$
$$\wedge \quad \forall x\, (\neg(\exists y\; y < x) \Rightarrow R_a x$$

specifies the language $(ab)^* \cup (ab)^* b$

- a powerful specification tool
- L is regular if and only if L is MSO-definable (Büchi 1960)
- A natural limitation (also from the algorithmic point of view): restriction to first-order logic
- As it turns out, many important properties of complex systems can be specified in FO
- hence the interest in understanding the expressive power, the algorithmic properties, etc… of FO and some of its fragments

First order logic

- FO is a *variety of languages*: algebraic characterization by means of the syntactic monoid; tool for decidability

First order logic

- FO is a *variety of languages*: algebraic characterization by means of the syntactic monoid; tool for decidability
- L is FO-definable iff L is star-free (McNaughton, Ladner, 1970), iff its syntactic monoid is aperiodic (Schützenberger 1965)

First order logic

- FO is a *variety of languages*: algebraic characterization by means of the syntactic monoid; tool for decidability
- L is FO-definable iff L is star-free (McNaughton, Ladner, 1970), iff its syntactic monoid is aperiodic (Schützenberger 1965)
- A monoid is in **Ap** if the groups in it are all trivial: a decidable property

First order logic

- FO is a *variety of languages*: algebraic characterization by means of the syntactic monoid; tool for decidability
- L is FO-definable iff L is star-free (McNaughton, Ladner, 1970), iff its syntactic monoid is aperiodic (Schützenberger 1965)
- A monoid is in **Ap** if the groups in it are all trivial: a decidable property
- $\mathbf{Ap} = [\![xx^\omega = x^\omega]\!]$

First order logic

- FO is a *variety of languages*: algebraic characterization by means of the syntactic monoid; tool for decidability
- L is FO-definable iff L is star-free (McNaughton, Ladner, 1970), iff its syntactic monoid is aperiodic (Schützenberger 1965)
- A monoid is in **Ap** if the groups in it are all trivial: a decidable property
- $\mathbf{Ap} = [\![xx^\omega = x^\omega]\!]$
- Note: deciding whether an automaton accepts an FO-definable language is PSPACE-complete (Cho, Huynh, 1991)

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^* a_1 A^* \cdots a_k A^*$ (words in which the word $a_1 \cdots a_k$ is a subword)

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^*a_1A^* \cdots a_kA^*$ (words in which the word $a_1 \cdots a_k$ is a subword)
- L is piecewise testable if it is a Boolean combination of elementary piecewise testable languages

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^*a_1A^* \cdots a_kA^*$ (words in which the word $a_1 \cdots a_k$ is a subword)
- L is piecewise testable if it is a Boolean combination of elementary piecewise testable languages
- L is piecewise testable iff it is defined by a Boolean combination of formulas of the form $\exists x_1, \ldots, \exists x_k \; \varphi(x_1, \ldots, x_k)$, where φ is quantifier-free

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^*a_1A^* \cdots a_kA^*$ (words in which the word $a_1 \cdots a_k$ is a subword)
- L is piecewise testable if it is a Boolean combination of elementary piecewise testable languages
- L is piecewise testable iff it is defined by a Boolean combination of formulas of the form $\exists x_1, \ldots, \exists x_k \ \varphi(x_1, \ldots, x_k)$, where φ is quantifier-free
- Piecewise testable languages form a variety. Moreover L is piecewise testable iff its syntactic monoid is in **J**, the class of $\mathcal{J}$-trivial monoids (Simon, 1970)

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^*a_1A^* \cdots a_kA^*$ (words in which the word $a_1 \cdots a_k$ is a subword)
- L is piecewise testable if it is a Boolean combination of elementary piecewise testable languages
- L is piecewise testable iff it is defined by a Boolean combination of formulas of the form $\exists x_1, \ldots, \exists x_k\ \varphi(x_1, \ldots, x_k)$, where φ is quantifier-free
- Piecewise testable languages form a variety. Moreover L is piecewise testable iff its syntactic monoid is in **J**, the class of $\mathcal{J}$-trivial monoids (Simon, 1970)
- A monoid is $\mathcal{J}$-trivial if $x = uyv$ and $y = u'xv'$ implies $x = y$: a decidable property

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^* a_1 A^* \cdots a_k A^*$ (words in which the word $a_1 \cdots a_k$ is a subword)
- L is piecewise testable if it is a Boolean combination of elementary piecewise testable languages
- L is piecewise testable iff it is defined by a Boolean combination of formulas of the form $\exists x_1, \ldots, \exists x_k\ \varphi(x_1, \ldots, x_k)$, where φ is quantifier-free
- Piecewise testable languages form a variety. Moreover L is piecewise testable iff its syntactic monoid is in **J**, the class of $\mathcal{J}$-trivial monoids (Simon, 1970)
- A monoid is $\mathcal{J}$-trivial if $x = uyv$ and $y = u'xv'$ implies $x = y$: a decidable property
- $\mathbf{J} = [\![(xy)^\omega = (yx)^\omega, xx^\omega = x^\omega]\!]$

Piecewise testable languages = only one level of quantifiers

- L is elementary piecewise testable if $L = A^*a_1A^* \cdots a_kA^*$ (words in which the word $a_1 \cdots a_k$ is a subword)
- L is piecewise testable if it is a Boolean combination of elementary piecewise testable languages
- L is piecewise testable iff it is defined by a Boolean combination of formulas of the form $\exists x_1, \ldots, \exists x_k\ \varphi(x_1, \ldots, x_k)$, where φ is quantifier-free
- Piecewise testable languages form a variety. Moreover L is piecewise testable iff its syntactic monoid is in **J**, the class of $\mathcal{J}$-trivial monoids (Simon, 1970)
- A monoid is $\mathcal{J}$-trivial if $x = uyv$ and $y = u'xv'$ implies $x = y$: a decidable property
- $\mathbf{J} = [\![(xy)^\omega = (yx)^\omega, xx^\omega = x^\omega]\!]$
- PTIME decision given a deterministic automaton

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$
- It is *visibly deterministic* if a_i does not occur in L_{i-1}

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$
- It is *visibly deterministic* if a_i does not occur in L_{i-1}
- *co-deterministic* and *visibly co-deterministic*: consider suffixes instead of prefixes

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$
- It is *visibly deterministic* if a_i does not occur in L_{i-1}
- *co-deterministic* and *visibly co-deterministic*: consider suffixes instead of prefixes

- $A^* a A^* = (A \setminus \{a\})^* a A^*$ — the first product is not deterministic, the second is visibly deterministic

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$
- It is *visibly deterministic* if a_i does not occur in L_{i-1}
- *co-deterministic* and *visibly co-deterministic*: consider suffixes instead of prefixes

- $A^* a A^* = (A \setminus \{a\})^* a A^*$ — the first product is not deterministic, the second is visibly deterministic
- The product $(b^* a^+) b A^*$ is deterministic but not visibly deterministic

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$
- It is *visibly deterministic* if a_i does not occur in L_{i-1}
- *co-deterministic* and *visibly co-deterministic*: consider suffixes instead of prefixes
- Let $\mathcal{R}_1 = \mathcal{L}_1 =$ piecewise testable languages, and let $\mathcal{R}_{m+1} = \mathcal{L}_m^{det}$, $\mathcal{L}_{m+1} = \mathcal{R}_m^{co-det}$ — where $\mathcal{V}^{det}$ is the class of all unions and intersections of deterministic products of languages in $\mathcal{V}$

Building a hierarchy from the piecewise testable languages

- A product $L = L_0 a_1 L_1 \cdots a_k L_k$ is *deterministic* if every word $u \in L$ has a unique prefix in $L_0 a_1 L_1 \cdots L_{i-1} a_i$ for $0 \leq i \leq k$
- It is *visibly deterministic* if a_i does not occur in L_{i-1}
- *co-deterministic* and *visibly co-deterministic*: consider suffixes instead of prefixes
- Let $\mathcal{R}_1 = \mathcal{L}_1 =$ piecewise testable languages, and let $\mathcal{R}_{m+1} = \mathcal{L}_m^{det}$, $\mathcal{L}_{m+1} = \mathcal{R}_m^{co-det}$ — where $\mathcal{V}^{det}$ is the class of all unions and intersections of deterministic products of languages in $\mathcal{V}$

- $L = (b,c)^* bc^+ aA^* \in \mathcal{R}_3$ since $L = L_1 aA^*$, with $L_1 = (b,c)^* bc^+$ is a (visibly) deterministic product
- and $L_1 = (b,c)^* bc^+ = (b,c)^* bL_2$, with $L_2 = c^+$ is a (visibly) co-deterministic product and L_2 is piecewise testable

Decidability of membership problems in $\mathcal{R}_m$, $\mathcal{L}_m$

- (Pin 1980 + Krohn, Rhodes, Tilson 1965) If $\mathcal{V}$ is a variety of languages, then $\mathcal{V}^{det}$ and $\mathcal{V}^{co-det}$ are also varieties.

Decidability of membership problems in $\mathcal{R}_m$, $\mathcal{L}_m$

- (Pin 1980 + Krohn, Rhodes, Tilson 1965) If $\mathcal{V}$ is a variety of languages, then $\mathcal{V}^{det}$ and $\mathcal{V}^{co-det}$ are also varieties.
- Moreover, if $\mathbf{V}$ (resp. $\mathbf{V}^{det}$) is the class of finite monoids corresponding to $\mathcal{V}$ (resp. $\mathcal{V}^{det}$), $\mathbf{V}^{det} = \mathbf{K} ⓜ \mathbf{V}$, that is: a monoid M is in $\mathbf{V}^{det}$ if and only if there exists a morphism $\varphi\colon M \to N$ with $N \in \mathbf{V}$ and for every idempotent $e \in N$, $\varphi^{-1}(e) \in \mathbf{K}$ (satisfies $x^\omega y = x^\omega$; is such that if $f = f^2$ and $s \in \varphi^{-1}(e)$, then $fs = f$)

Decidability of membership problems in $\mathcal{R}_m$, $\mathcal{L}_m$

- (Pin 1980 + Krohn, Rhodes, Tilson 1965) If $\mathcal{V}$ is a variety of languages, then $\mathcal{V}^{det}$ and $\mathcal{V}^{co-det}$ are also varieties.
- Moreover, if $\mathbf{V}$ (resp. $\mathbf{V}^{det}$) is the class of finite monoids corresponding to $\mathcal{V}$ (resp. $\mathcal{V}^{det}$), $\mathbf{V}^{det} = \mathbf{K} ⓜ \mathbf{V}$, that is: a monoid M is in $\mathbf{V}^{det}$ if and only if there exists a morphism $\varphi\colon M \to N$ with $N \in \mathbf{V}$ and for every idempotent $e \in N$, $\varphi^{-1}(e) \in \mathbf{K}$ (satisfies $x^\omega y = x^\omega$; is such that if $f = f^2$ and $s \in \varphi^{-1}(e)$, then $fs = f$)
- for co-deterministic product: $\mathbf{D} ⓜ \mathbf{V}$, where $S \in \mathbf{D}$ if for all $f = f^2$ and $s \in S$, then $sf = f$

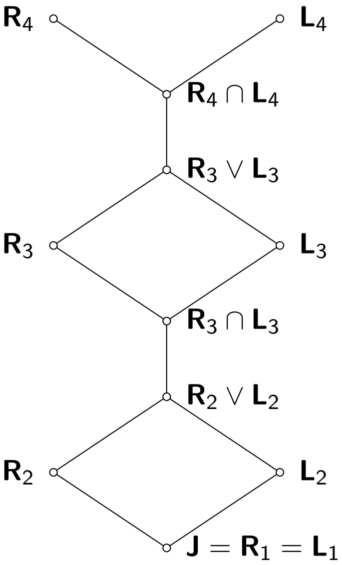
Decidability of membership problems in $\mathcal{R}_m$, $\mathcal{L}_m$

- (Pin 1980 + Krohn, Rhodes, Tilson 1965) If $\mathcal{V}$ is a variety of languages, then $\mathcal{V}^{det}$ and $\mathcal{V}^{co-det}$ are also varieties.
- Moreover, if $\mathbf{V}$ (resp. $\mathbf{V}^{det}$) is the class of finite monoids corresponding to $\mathcal{V}$ (resp. $\mathcal{V}^{det}$), $\mathbf{V}^{det} = \mathbf{K} \textcircled{m} \mathbf{V}$, that is: a monoid M is in $\mathbf{V}^{det}$ if and only if there exists a morphism $\varphi\colon M \to N$ with $N \in \mathbf{V}$ and for every idempotent $e \in N$, $\varphi^{-1}(e) \in \mathbf{K}$ (satisfies $x^\omega y = x^\omega$; is such that if $f = f^2$ and $s \in \varphi^{-1}(e)$, then $fs = f$)
- for co-deterministic product: $\mathbf{D} \textcircled{m} \mathbf{V}$, where $S \in \mathbf{D}$ if for all $f = f^2$ and $s \in S$, then $sf = f$
- If $\mathbf{V}$ is decidable, so are $\mathbf{K} \textcircled{m} \mathbf{V}$ and $\mathbf{D} \textcircled{m} \mathbf{V}$

Decidability of membership problems in $\mathcal{R}_m$, $\mathcal{L}_m$

- (Pin 1980 + Krohn, Rhodes, Tilson 1965) If $\mathcal{V}$ is a variety of languages, then $\mathcal{V}^{det}$ and $\mathcal{V}^{co-det}$ are also varieties.
- Moreover, if $\mathbf{V}$ (resp. $\mathbf{V}^{det}$) is the class of finite monoids corresponding to $\mathcal{V}$ (resp. $\mathcal{V}^{det}$), $\mathbf{V}^{det} = \mathbf{K} \textcircled{m} \mathbf{V}$, that is: a monoid M is in $\mathbf{V}^{det}$ if and only if there exists a morphism $\varphi\colon M \to N$ with $N \in \mathbf{V}$ and for every idempotent $e \in N$, $\varphi^{-1}(e) \in \mathbf{K}$ (satisfies $x^\omega y = x^\omega$; is such that if $f = f^2$ and $s \in \varphi^{-1}(e)$, then $fs = f$)
- for co-deterministic product: $\mathbf{D} \textcircled{m} \mathbf{V}$, where $S \in \mathbf{D}$ if for all $f = f^2$ and $s \in S$, then $sf = f$
- If $\mathbf{V}$ is decidable, so are $\mathbf{K} \textcircled{m} \mathbf{V}$ and $\mathbf{D} \textcircled{m} \mathbf{V}$
- The classes $\mathcal{R}_m$ and $\mathcal{L}_m$ are varieties of languages with decidable membership

The lattice generated by the $\mathcal{R}_m$ and $\mathcal{L}_m$



Second level: $\mathbf{R}_2 = \mathcal{R}$-trivial and $\mathbf{L}_2 = \mathcal{L}$-trivial monoids

joins and meets don't agree

$\mathbf{R}_m$ and $\mathbf{L}_m$ decidable

Infinite hierarchy?

What is its union?

Where **DA** and **B** come into the picture

- Let **DA** be the class of finite monoids in which every regular element is idempotent [x is *regular* if there exists y such that $x = xyx$; x is *idempotent* if $x^2 = x$]

Where **DA** and **B** come into the picture

- Let **DA** be the class of finite monoids in which every regular element is idempotent [x is *regular* if there exists y such that $x = xyx$; x is *idempotent* if $x^2 = x$]
- **DA** is a much studied class in semigroup theory and in FLT

Where **DA** and **B** come into the picture

- Let **DA** be the class of finite monoids in which every regular element is idempotent [x is *regular* if there exists y such that $x = xyx$; x is *idempotent* if $x^2 = x$]
- **DA** is a much studied class in semigroup theory and in FLT
- (Schützenberger 1976) $\mathbf{DA} = L\mathbf{I} ⓜ \mathbf{J} = L\mathbf{I} ⓜ \mathbf{DA}$, where $L\mathbf{I}$ is the class of semigroups S such that, if $e, s, f \in S$, $e = e^2$ and $f = f^2$, then $esf = ef$

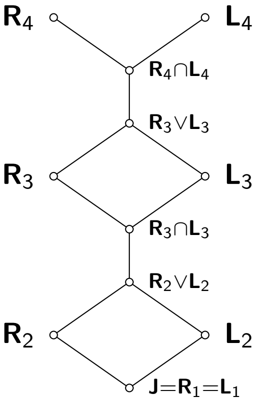
Where **DA** and **B** come into the picture

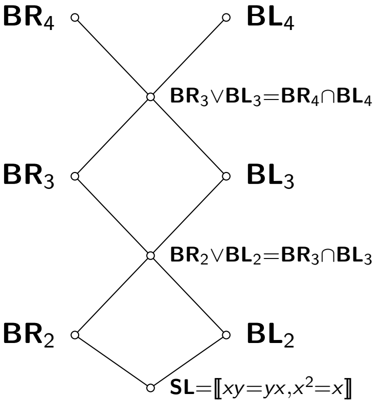
- Let **DA** be the class of finite monoids in which every regular element is idempotent [x is *regular* if there exists y such that $x = xyx$; x is *idempotent* if $x^2 = x$]
- **DA** is a much studied class in semigroup theory and in FLT
- (Schützenberger 1976) $\mathbf{DA} = L\mathbf{I} \textcircled{m} \mathbf{J} = L\mathbf{I} \textcircled{m} \mathbf{DA}$, where $L\mathbf{I}$ is the class of semigroups S such that, if $e, s, f \in S$, $e = e^2$ and $f = f^2$, then $esf = ef$
- So $\mathbf{R}_m, \mathbf{L}_m \subseteq \mathbf{DA}$

Where DA and B come into the picture

- Let **DA** be the class of finite monoids in which every regular element is idempotent [x is *regular* if there exists y such that $x = xyx$; x is *idempotent* if $x^2 = x$]
- **DA** is a much studied class in semigroup theory and in FLT
- (Schützenberger 1976) $\mathbf{DA} = L\mathbf{I} \ⓜ \mathbf{J} = L\mathbf{I} \ⓜ \mathbf{DA}$, where $L\mathbf{I}$ is the class of semigroups S such that, if $e, s, f \in S$, $e = e^2$ and $f = f^2$, then $esf = ef$
- So $\mathbf{R}_m, \mathbf{L}_m \subseteq \mathbf{DA}$
- The lattice of the $\mathbf{R}_m, \mathbf{L}_m$ resembles the lattice of idempotent monoid varieties, i.e. the sub-varieties of $\mathbf{B} = [\![x^2 = x]\!]$ — completely described in the 1970s and 1980s (Birjukov, Fennemore, Gerhard, Petrich)

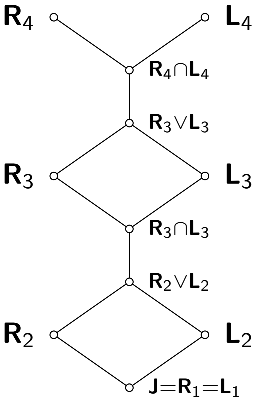
Two similar lattices

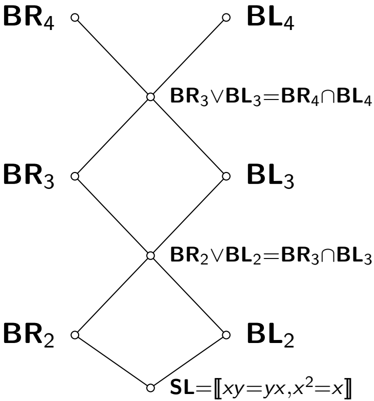




where $\mathbf{BR}_{m+1} = (\mathbf{K} \cap \mathbf{B}) ⓜ \mathbf{BL}_m$ and $\mathbf{BL}_{m+1} = (\mathbf{D} \cap \mathbf{B}) ⓜ \mathbf{BR}_m$

Two similar lattices

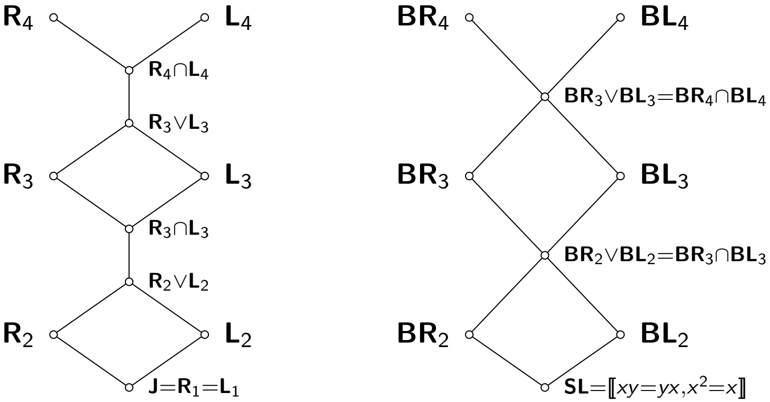




where $\mathbf{BR}_{m+1} = (\mathbf{K} \cap \mathbf{B}) ⓜ \mathbf{BL}_m$ and $\mathbf{BL}_{m+1} = (\mathbf{D} \cap \mathbf{B}) ⓜ \mathbf{BR}_m$

The connection is provided by the map $\mathbf{V} \longmapsto \mathbf{V} \cap \mathbf{B}$, on the sub-varieties of $\mathbf{DA}$

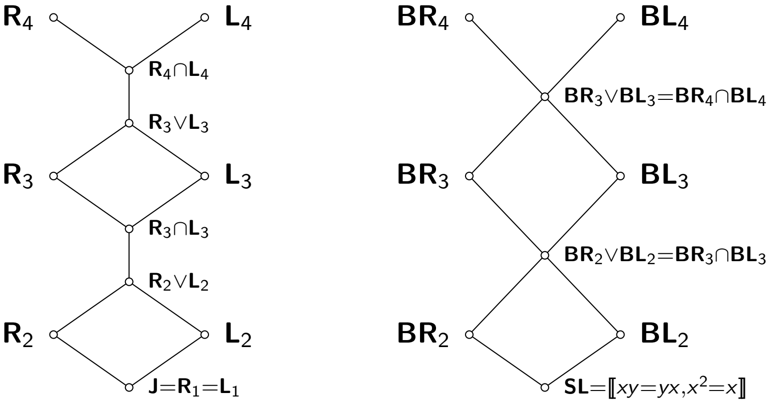
Two similar lattices



where $\mathbf{BR}_{m+1} = (\mathbf{K} \cap \mathbf{B}) \circledm \mathbf{BL}_m$ and $\mathbf{BL}_{m+1} = (\mathbf{D} \cap \mathbf{B}) \circledm \mathbf{BR}_m$

(Trotter, W. 1997 + Kufleitner, W. 2012) $\mathbf{R}_m$ is maximal such that $\mathbf{R}_m \cap \mathbf{B} = \mathbf{BR}_m$ – and dually for $\mathbf{L}_m$

Two similar lattices



where $\mathbf{BR}_{m+1} = (\mathbf{K} \cap \mathbf{B}) \, ⓜ \, \mathbf{BL}_m$ and $\mathbf{BL}_{m+1} = (\mathbf{D} \cap \mathbf{B}) \, ⓜ \, \mathbf{BR}_m$

The union of the $\mathbf{R}_m$, $\mathbf{L}_m$ is **DA** and the hierarchy is infinite – but finite on each finite alphabet

Exploiting our knowledge of idempotent monoids (from the 1970s)

- Every finitely generated idempotent monoid is finite: if A is a finite alphabet, there is a free such monoid over A, $FB(A)$

Exploiting our knowledge of idempotent monoids (from the 1970s)

- Every finitely generated idempotent monoid is finite: if A is a finite alphabet, there is a free such monoid over A, $FB(A)$
- The word problem: given $u, v \in A^*$, are they equal in $FB(A)$ (written $u =_{FB} v$)? Ex: $abacbcacbacabca =_{FB} abacbca$

Exploiting our knowledge of idempotent monoids (from the 1970s)

- Every finitely generated idempotent monoid is finite: if A is a finite alphabet, there is a free such monoid over A, $FB(A)$
- The word problem: given $u, v \in A^*$, are they equal in $FB(A)$ (written $u =_{FB} v$)? Ex: $abacbcacbacabca =_{FB} abacbca$
- A beautiful solution of the word problem considers, for a word w:

Exploiting our knowledge of idempotent monoids (from the 1970s)

- Every finitely generated idempotent monoid is finite: if A is a finite alphabet, there is a free such monoid over A, $FB(A)$
- The word problem: given $u, v \in A^*$, are they equal in $FB(A)$ (written $u =_{FB} v$)? Ex: $abacbcacbacabca =_{FB} abacbca$
- A beautiful solution of the word problem considers, for a word w:
- $w = s(w)\sigma(w)w' = w''\tau(w)t(w)$ where $\sigma(w)$ is the last letter to first appear (from the left), $s(w)$ is the corresponding prefix, and $t(w)$, $\tau(w)$ are the left-right duals.

Exploiting our knowledge of idempotent monoids (from the 1970s)

- Every finitely generated idempotent monoid is finite: if A is a finite alphabet, there is a free such monoid over A, $FB(A)$
- The word problem: given $u, v \in A^*$, are they equal in $FB(A)$ (written $u =_{FB} v$)? Ex: $abacbcacbacabca =_{FB} abacbca$
- A beautiful solution of the word problem considers, for a word w:
- $w = s(w)\sigma(w)w' = w''\tau(w)t(w)$ where $\sigma(w)$ is the last letter to first appear (from the left), $s(w)$ is the corresponding prefix, and $t(w)$, $\tau(w)$ are the left-right duals.
- $u = v$ in $FB(A)$ if and only if $\sigma(u) = \sigma(v)$, $\tau(u) = \tau(v)$ and $s(u) =_{FB} s(v)$, $t(u) =_{FB} t(v)$ (a *recursive* solution)

Exploiting our knowledge of idempotent monoids (from the 1970s)

- Every finitely generated idempotent monoid is finite: if A is a finite alphabet, there is a free such monoid over A, $FB(A)$
- The word problem: given $u, v \in A^*$, are they equal in $FB(A)$ (written $u =_{FB} v$)? Ex: $abacbcacbacabca =_{FB} abacbca$
- A beautiful solution of the word problem considers, for a word w:
- $w = s(w)\sigma(w)w' = w''\tau(w)t(w)$ where $\sigma(w)$ is the last letter to first appear (from the left), $s(w)$ is the corresponding prefix, and $t(w)$, $\tau(w)$ are the left-right duals.
- $u = v$ in $FB(A)$ if and only if $\sigma(u) = \sigma(v)$, $\tau(u) = \tau(v)$ and $s(u) =_{FB} s(v)$, $t(u) =_{FB} t(v)$ (a *recursive* solution)
- That is: find the last letter to have its first occurrence in u from the left (resp. the right), then doing the same in the maximal prefixes and suffixes with alphabetical content one less letter, etc.

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- A *ranker* is a word on the alphabet $\{\mathsf{X}_a, \mathsf{Y}_a \mid a \in A\}$

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- A *ranker* is a word on the alphabet $\{\mathsf{X}_a, \mathsf{Y}_a \mid a \in A\}$
- A ranker r defines (or not!) a position $r(u)$ in a word u of A^*

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- A *ranker* is a word on the alphabet $\{X_a, Y_a \mid a \in A\}$
- A ranker r defines (or not!) a position $r(u)$ in a word u of A^*
- X_a means *go to the next a to the right*
  and Y_a means *go to the next a to the left*

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- A *ranker* is a word on the alphabet $\{\mathsf{X}_a, \mathsf{Y}_a \mid a \in A\}$
- A ranker r defines (or not!) a position $r(u)$ in a word u of A^*
- X_a means *go to the next a to the right* and Y_a means *go to the next a to the left*
- $\mathsf{X}_a\mathsf{Y}_b\mathsf{X}_c$ is defined on *bac* and *bca*, but not on *abc* or *cba*

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- A *ranker* is a word on the alphabet $\{\mathrm{X}_a, \mathrm{Y}_a \mid a \in A\}$
- A ranker r defines (or not!) a position $r(u)$ in a word u of A^*
- X_a means *go to the next a to the right* and Y_a means *go to the next a to the left*
- $\mathrm{X}_a\mathrm{Y}_b\mathrm{X}_c$ is defined on *bac* and *bca*, but not on *abc* or *cba*
- r is well-defined on u if r is defined on u and the trajectory described by u "zooms" in on the position $r(u)$ (= never crosses itself = changes direction as soon as it has passed $r(u)$)

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 1/3

- This in turn reminds us of the rankers, a.k.a. turtle programs (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- A *ranker* is a word on the alphabet $\{\mathrm{X}_a, \mathrm{Y}_a \mid a \in A\}$
- A ranker r defines (or not!) a position $r(u)$ in a word u of A^*
- X_a means *go to the next a to the right* and Y_a means *go to the next a to the left*
- $\mathrm{X}_a\mathrm{Y}_b\mathrm{X}_c$ is defined on *bac* and *bca*, but not on *abc* or *cba*
- r is well-defined on u if r is defined on u and the trajectory described by u "zooms" in on the position $r(u)$ (= never crosses itself = changes direction as soon as it has passed $r(u)$)
- The ranker $\mathrm{X}_a\mathrm{Y}_b\mathrm{X}_c$ is well-defined on *bca*, but not on *bac*

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 2/3

- For a ranker r, notions of *length* and *number of alternating blocks*

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 2/3

- For a ranker r, notions of *length* and *number of alternating blocks*
- Define $\underline{\mathrm{R}}_{m,n}$ as the set of rankers of length at most n, with at most m blocks

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 2/3

- For a ranker r, notions of *length* and *number of alternating blocks*
- Define $\underline{\mathrm{R}}_{m,n}$ as the set of rankers of length at most n, with at most m blocks
- Let $\underline{\mathrm{R}}^{\mathrm{X}}_{m,n}$ be the rankers in $\underline{\mathrm{R}}_{m,n}$ that start with an X-letter, or else that start with a Y-letter but are in $\underline{\mathrm{R}}_{m-1,n-1}$

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 2/3

- For a ranker r, notions of *length* and *number of alternating blocks*
- Define $\underline{R}_{m,n}$ as the set of rankers of length at most n, with at most m blocks
- Let $\underline{R}^{X}_{m,n}$ be the rankers in $\underline{R}_{m,n}$ that start with an X-letter, or else that start with a Y-letter but are in $\underline{R}_{m-1,n-1}$
- Dual definition for $\underline{R}^{Y}_{m,n}$

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 2/3

- For a ranker r, notions of *length* and *number of alternating blocks*
- Define $\underline{R}_{m,n}$ as the set of rankers of length at most n, with at most m blocks
- Let $\underline{R}^{X}_{m,n}$ be the rankers in $\underline{R}_{m,n}$ that start with an X-letter, or else that start with a Y-letter but are in $\underline{R}_{m-1,n-1}$
- Dual definition for $\underline{R}^{Y}_{m,n}$
- The property "the same rankers from $\underline{R}^{X}_{m,n}$ are well-defined on u and v" (written $u \rhd_{m,n} v$) is a finite index congruence. Dually for $\underline{R}^{Y}_{m,n}$, written $u \lhd_{m,n} v$

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band
- $u = abcadbacba$: the value of u in $FB(A)$ is characterized

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band
- $u = abcadbacba$: the value of u in $FB(A)$ is characterized
- by the last letter to occur for the first time from the left and the right:

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band

- $u = abcadbacba$: the value of u in $FB(A)$ is characterized
- by the last letter to occur for the first time from the left and the right:
- X_d is well-defined but $\mathsf{X}_z\mathsf{Y}_d$ $(z = a, b, c)$ and $\mathsf{Y}_z\mathsf{X}_d$ $(z = a, b, c)$ are not

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band

- $u = abcadbacba$: the value of u in $FB(A)$ is characterized
- by the last letter to occur for the first time from the left and the right:
- X_d is well-defined but $\mathsf{X}_z\mathsf{Y}_d$ $(z = a, b, c)$ and $\mathsf{Y}_z\mathsf{X}_d$ $(z = a, b, c)$ are not
- and by the values in $FB(A)$ of $abca$ and $acba$

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band
- and with the notion of visibly deterministic (co-deterministic) product of languages

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band
- and with the notion of visibly deterministic (co-deterministic) product of languages
- This yields a new characterization of the $\mathbf{R}_m$ and $\mathbf{L}_m$ (Kufleitner, W. 2012)

Rankers and the varieties $\mathbf{R}_m$ and $\mathbf{L}_m$ 3/3

- Rankers have a lot to do with the word problem of the free band
- and with the notion of visibly deterministic (co-deterministic) product of languages
- This yields a new characterization of the $\mathbf{R}_m$ and $\mathbf{L}_m$ (Kufleitner, W. 2012)
- **Theorem** $\mathbf{R}_m$ is the variety of monoids generated by the $A^*/\triangleright_{m,n}$ $(n \geq m)$, and $\mathbf{L}_m$ is generated by the $A^*/\triangleleft_{m,n}$ $(n \geq m)$. Moreover, $\mathcal{R}_{m+1} = \mathcal{L}_m^{v-det}$ and $\mathcal{L}_{m+1} = \mathcal{R}_m^{v-det}$

Back to logic: rankers and fragments of first-order logic

- Rankers were introduced in the context of a certain restriction of first-order logic (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)

Back to logic: rankers and fragments of first-order logic

- Rankers were introduced in the context of a certain restriction of first-order logic (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- *Limitation of resources*: a formula is more or less *complex* if it requires a larger or smaller number of nested quantifiers; the quantifier alternation hierarchy

Back to logic: rankers and fragments of first-order logic

- Rankers were introduced in the context of a certain restriction of first-order logic (Schwentick, Thérien, Vollmer 2001; Weis, Immerman 2009)
- *Limitation of resources*: a formula is more or less *complex* if it requires a larger or smaller number of nested quantifiers; the quantifier alternation hierarchy
- *Limitation of resources*: a formula may use more or less variable symbols; at the cost of making the formula longer (Adler & Immerman 2002, Grohe & Schweikardt 2005)

First order logic with two variables

$$A^* a_1 A^* a_2 A^* a_3 A^* a_4 A^*$$

First order logic with two variables

$$A^* a_1 A^* a_2 A^* a_3 A^* a_4 A^*$$

$$\exists x_1\ \exists x_2\ \exists x_3\ \exists x_4\ \ (x_1 < x_2 < x_3 < x_4)$$
$$\wedge R_{a_1} x_1 \wedge R_{a_2} x_2 \wedge R_{a_3} x_3 \wedge R_{a_4} x_4$$

First order logic with two variables

$$A^* a_1 A^* a_2 A^* a_3 A^* a_4 A^*$$

$$\exists x_1\ \exists x_2\ \exists x_3\ \exists x_4\ \ (x_1 < x_2 < x_3 < x_4) \\ \wedge R_{a_1} x_1 \wedge R_{a_2} x_2 \wedge R_{a_3} x_3 \wedge R_{a_4} x_4$$

$$\exists x\ R_{a_1} x \wedge \Big(\exists y\ (x < y) \wedge R_{a_2} y \\ \wedge \Big(\exists x\ (y < x) \wedge R_{a_3} x \\ \wedge \Big(\exists y\ (x < y) \wedge R_{a_4} y \Big) \Big) \Big)$$

First order logic with two variables

- FO^2 is properly contained in $\mathbf{Ap} = \mathrm{FO} = \mathrm{FO}^3$

$$\exists x\ R_{a_1}x \wedge \Big(\exists y\ (x<y) \wedge R_{a_2}y \wedge \Big(\exists x\ (y<x) \wedge R_{a_3}x \wedge \Big(\exists y\ (x<y) \wedge R_{a_4}y\Big)\Big)\Big)$$

First order logic with two variables

- FO^2 is properly contained in $\mathbf{Ap} = FO = FO^3$
- FO^2 is a variety of languages: a language is FO^2-definable if its syntactic monoid is in **DA** (Thérien, Wilke, 1998)

$$\exists x\ R_{a_1}x \wedge \Big(\exists y\ (x < y) \wedge R_{a_2}y \wedge \Big(\exists x\ (y < x) \wedge R_{a_3}x \wedge \Big(\exists y\ (x < y) \wedge R_{a_4}y\Big)\Big)\Big)$$

Quantifier alternation in FO^2

- Every FO^2-formula can be represented by a tree
  whose internal nodes are quantifiers and Boolean connectors
  and whose leaves are atomic formulas (*e.g.* $x < y$, $R_a x$)

Quantifier alternation in FO^2

- Every FO^2-formula can be represented by a tree whose internal nodes are quantifiers and Boolean connectors and whose leaves are atomic formulas (*e.g.* $x < y$, $R_a x$)
- *Quantifier alternation* = maximal number of blocks of quantifiers along a branch. Notation FO^2_m (at most m blocks)

Quantifier alternation in FO^2

- Every FO^2-formula can be represented by a tree whose internal nodes are quantifiers and Boolean connectors and whose leaves are atomic formulas (*e.g.* $x < y$, $R_a x$)
- *Quantifier alternation* = maximal number of blocks of quantifiers along a branch. Notation FO^2_m (at most m blocks)
- **Questions**: Is the quantifier alternation hierarchy infinite? Can we characterize its levels? Can we decide membership in its levels?

Quantifier alternation in FO^2

- Every FO^2-formula can be represented by a tree whose internal nodes are quantifiers and Boolean connectors and whose leaves are atomic formulas (*e.g.* $x < y$, $R_a x$)
- *Quantifier alternation* = maximal number of blocks of quantifiers along a branch. Notation FO^2_m (at most m blocks)
- **Questions**: Is the quantifier alternation hierarchy infinite? Can we characterize its levels? Can we decide membership in its levels?
- A collection of results of Schwentick, Thérien, Vollmer (2001); Weis, Immermann (2007,2009); Kufleitner, Weil (2009, 2012); Straubing and Krebs (2011, 2012)

Rankers and FO^2

- At the cost of long formulas,

Rankers and FO^2

- At the cost of long formulas,
- one can express that a ranker is defined on a word u by an FO^2-formula

Rankers and FO^2

- At the cost of long formulas,
- one can express that a ranker is defined on a word u by an FO^2-formula
- and even that it is well-defined on u

Rankers and FO^2

- At the cost of long formulas,
- one can express that a ranker is defined on a word u by an FO^2-formula
- and even that it is well-defined on u
- (combinatorially highly non trivial)

The final result

Theorem

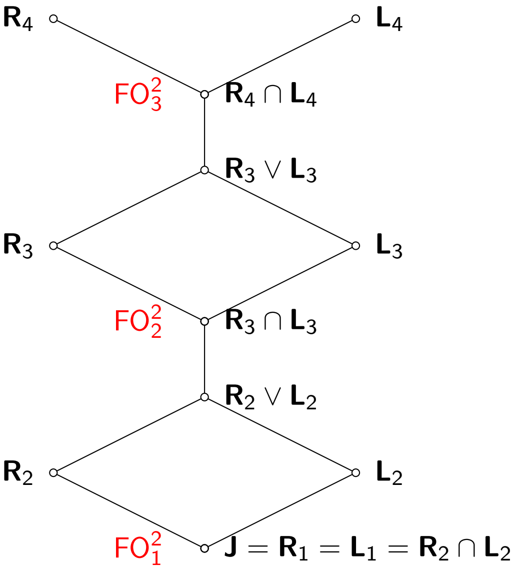
$FO^2_m = \mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$

FO^2_m is decidable

the hierarchy collapses

at level $m + 1$

on an m-letter alphabet



The story is not over!...

- Straubing introduced another algebraic hierarchy which captures exactly FO^2_m

The story is not over!...

- Straubing introduced another algebraic hierarchy which captures exactly FO^2_m
- $\mathbf{V}_1 = \mathbf{J} = \mathsf{FO}^2_1$, $\mathbf{V}_{m+1} = \mathbf{V}_m \,\square\, \mathbf{J}$, where...

The story is not over!...

- Straubing introduced another algebraic hierarchy which captures exactly FO^2_m
- $\mathbf{V}_1 = \mathbf{J} = \mathsf{FO}^2_1$, $\mathbf{V}_{m+1} = \mathbf{V}_m \,\Box\, \mathbf{J}$, where...
- ... $\Box$ is the two-sided wreath product, an operation on varieties of finite monoids which was much studied in the 1980s and 1990s.

The story is not over!...

- Straubing introduced another algebraic hierarchy which captures exactly FO^2_m
- $\mathbf{V}_1 = \mathbf{J} = \mathsf{FO}^2_1$, $\mathbf{V}_{m+1} = \mathbf{V}_m \mathbin{\square} \mathbf{J}$, where...
- ... $\square$ is the two-sided wreath product, an operation on varieties of finite monoids which was much studied in the 1980s and 1990s.
- **Theorem** (Straubing, 2011) For each m, $\mathbf{V}_m = \mathsf{FO}^2_m$

The story is not over!...

- Straubing introduced another algebraic hierarchy which captures exactly FO^2_m
- $\mathbf{V}_1 = \mathbf{J} = \mathrm{FO}^2_1$, $\mathbf{V}_{m+1} = \mathbf{V}_m \,\square\, \mathbf{J}$, where...
- ... $\square$ is the two-sided wreath product, an operation on varieties of finite monoids which was much studied in the 1980s and 1990s.
- **Theorem** (Straubing, 2011) For each m, $\mathbf{V}_m = \mathrm{FO}^2_m$
- **Corollary** (Straubing and Krebs, 2012) Each FO^2_m is decidable

These are (seemingly...) different characterizations

- What is the connection between alternating blocks of quantifiers, and block products with $\mathbf{J}$?

These are (seemingly...) different characterizations

- What is the connection between alternating blocks of quantifiers, and block products with **J**?
- Straubing and Krebs give two identities for FO^2_m
  $xx^\omega = x^\omega$ and $u_n = v_n$

These are (seemingly…) different characterizations

- What is the connection between alternating blocks of quantifiers, and block products with **J**?
- Straubing and Krebs give two identities for FO^2_m
  $xx^\omega = x^\omega$ and $u_n = v_n$
- where $u_1 = (x_1x_2)^\omega, \ \ v_1 = (x_2x_1)^\omega$
  $u_{n+1} = (x_1 \cdots x_{2n}x_{2n+1})^\omega u_n (x_{2n+2}x_1 \cdots x_{2n})^\omega$
  $v_{n+1} = (x_1 \cdots x_{2n}x_{2n+1})^\omega v_n (x_{2n+2}x_1 \cdots x_{2n})^\omega$

These are (seemingly...) different characterizations

- What is the connection between alternating blocks of quantifiers, and block products with **J**?
- Straubing and Krebs give two identities for FO^2_m
  $xx^\omega = x^\omega$ and $u_n = v_n$
- where $u_1 = (x_1x_2)^\omega, \ \ v_1 = (x_2x_1)^\omega$
  $u_{n+1} = (x_1 \cdots x_{2n}x_{2n+1})^\omega u_n(x_{2n+2}x_1 \cdots x_{2n})^\omega$
  $v_{n+1} = (x_1 \cdots x_{2n}x_{2n+1})^\omega v_n(x_{2n+2}x_1 \cdots x_{2n})^\omega$
- We give 3 identities (which are more complicated...):
  $(xy)^\omega x(xy)^\omega = (xy)^\omega$, $\varphi(g_m) = \varphi(h_m)$ and $\varphi(\overline{g_m}) = \varphi(\overline{h_m})$

These are (seemingly. . .) different characterizations

- What is the connection between alternating blocks of quantifiers, and block products with **J**?
- Straubing and Krebs give two identities for FO^2_m
  $xx^\omega = x^\omega$ and $u_n = v_n$
- where $u_1 = (x_1x_2)^\omega, \; v_1 = (x_2x_1)^\omega$
  $u_{n+1} = (x_1 \cdots x_{2n}x_{2n+1})^\omega u_n (x_{2n+2}x_1 \cdots x_{2n})^\omega$
  $v_{n+1} = (x_1 \cdots x_{2n}x_{2n+1})^\omega v_n (x_{2n+2}x_1 \cdots x_{2n})^\omega$
- We give 3 identities (which are more complicated. . .):
  $(xy)^\omega x (xy)^\omega = (xy)^\omega$, $\varphi(g_m) = \varphi(h_m)$ and $\varphi(\overline{g_m}) = \varphi(\overline{h_m})$
- where $g_1 = x_2x_1, \quad h_1 = x_2x_1x_2$
  $g_{m+1} = x_{m+1}\overline{g_m}, \quad h_{m+1} = g_{m+1}x_{m+1}\overline{h_m}$
  $\varphi\colon x_1 \mapsto (x_1^\omega x_2^\omega x_1^\omega)^\omega \;\; x_2 \mapsto x_2^\omega \;\; x_{m+1} \mapsto \left(x_{m+1}^\omega \varphi(g_m\overline{g_m})^\omega x_{m+1}^\omega\right)^\omega$

Thank you for your attention!