Finite algebras and regular tree languages

Matt Valeriote

McMaster University

15 January 2020

joint work with Mikołaj Bojańczyk

Regular tree languages

Definition

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.
- For Σ a ranked alphabet, the set of Σ-trees, denoted treesΣ, is the set of variable-free Σ terms.

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.
- For Σ a ranked alphabet, the set of Σ-trees, denoted treesΣ, is the set of variable-free Σ terms.
- A Σ-language is any subset of treesΣ.

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.
- For Σ a ranked alphabet, the set of Σ-trees, denoted treesΣ, is the set of variable-free Σ terms.
- A Σ-language is any subset of treesΣ.

Definition

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.
- For Σ a ranked alphabet, the set of Σ-trees, denoted treesΣ, is the set of variable-free Σ terms.
- A Σ-language is any subset of treesΣ.

Definition

- A Σ-reduct of an algebra $\mathbb{A}$ is any algebra of type Σ that has universe A and whose basic operations are all polynomial operations of $\mathbb{A}$.

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.
- For Σ a ranked alphabet, the set of Σ-trees, denoted $\mathrm{trees}\Sigma$, is the set of variable-free Σ terms.
- A Σ-language is any subset of $\mathrm{trees}\Sigma$.

Definition

- A Σ-reduct of an algebra $\mathbb{A}$ is any algebra of type Σ that has universe A and whose basic operations are all polynomial operations of $\mathbb{A}$.
- A finite algebra $\mathbb{A}$ recognizes the Σ-language L if there is some Σ-reduct $\mathbb{A}'$ of $\mathbb{A}$ and some subset F of A such that

$$L = \{t \in \mathrm{trees}\Sigma \mid t^{\mathbb{A}'} \in F\}.$$

Regular tree languages

Definition

- A finite set of function symbols that includes at least one constant symbol will be referred to as a ranked alphabet.
- For Σ a ranked alphabet, the set of Σ-trees, denoted $\text{trees}\Sigma$, is the set of variable-free Σ terms.
- A Σ-language is any subset of $\text{trees}\Sigma$.

Definition

- A Σ-reduct of an algebra $\mathbb{A}$ is any algebra of type Σ that has universe A and whose basic operations are all polynomial operations of $\mathbb{A}$.
- A finite algebra $\mathbb{A}$ recognizes the Σ-language L if there is some Σ-reduct $\mathbb{A}'$ of $\mathbb{A}$ and some subset F of A such that

$$L = \{t \in \text{trees}\Sigma \mid t^{\mathbb{A}'} \in F\}.$$

- A Σ-language L is regular if it is recognized by some finite algebra.

Regular tree language example

Example

Regular tree language example

Example

1. Let Σ be the ranked alphabet that has one rank 2 symbol b, one rank 1 symbol a, and one rank 0 symbol c.

Regular tree language example

Example

1. Let Σ be the ranked alphabet that has one rank 2 symbol b, one rank 1 symbol a, and one rank 0 symbol c.
2. Let $\mathcal{E}$ be the tree language consisting of all Σ-trees whose branches all have even length.

Regular tree language example

Example

1. Let Σ be the ranked alphabet that has one rank 2 symbol b, one rank 1 symbol a, and one rank 0 symbol c.
2. Let $\mathcal{E}$ be the tree language consisting of all Σ-trees whose branches all have even length.
3. Then the following algebra $\mathbb{E}$ recognizes $\mathcal{E}$:

Regular tree language example

Example

1. Let Σ be the ranked alphabet that has one rank 2 symbol b, one rank 1 symbol a, and one rank 0 symbol c.
2. Let $\mathcal{E}$ be the tree language consisting of all Σ-trees whose branches all have even length.
3. Then the following algebra $\mathbb{E}$ recognizes $\mathcal{E}$:

Regular tree language example

Example

1. Let Σ be the ranked alphabet that has one rank 2 symbol b, one rank 1 symbol a, and one rank 0 symbol c.
2. Let $\mathcal{E}$ be the tree language consisting of all Σ-trees whose branches all have even length.
3. Then the following algebra $\mathbb{E}$ recognizes $\mathcal{E}$:

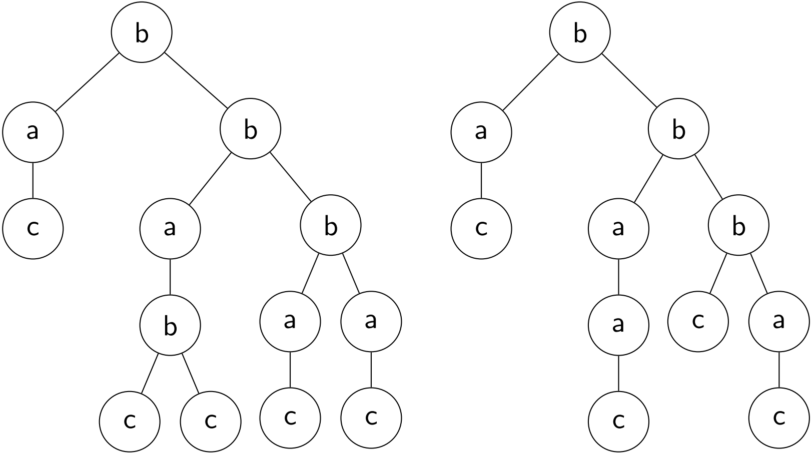
$\mathbb{E} = \langle \{0, 1, 2\}, a^{\mathbb{E}}, b^{\mathbb{E}}, c^{\mathbb{E}} \rangle$,

$b^{\mathbb{E}}$	0	1	2
0	1	2	2
1	2	0	2
2	2	2	2

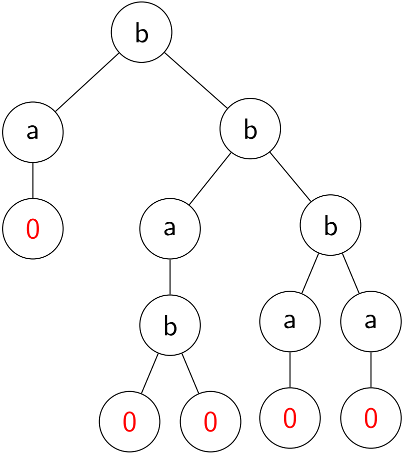
x	$a^{\mathbb{E}}(x)$
0	1
1	0
2	2

$c^{\mathbb{E}} = 0$

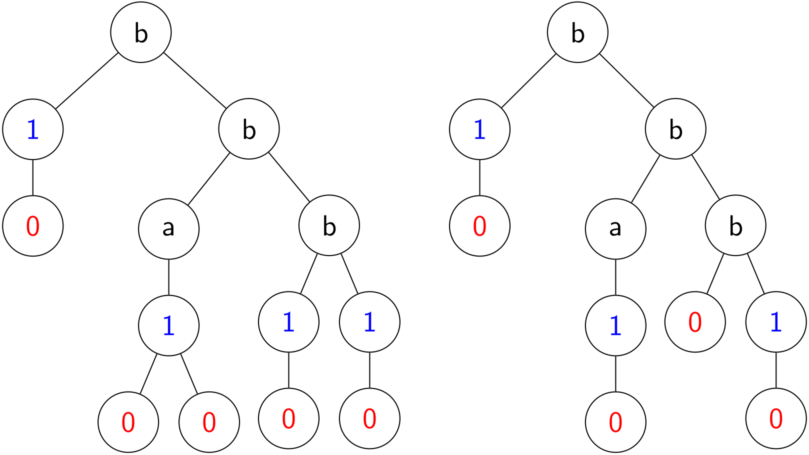
Regular tree language example



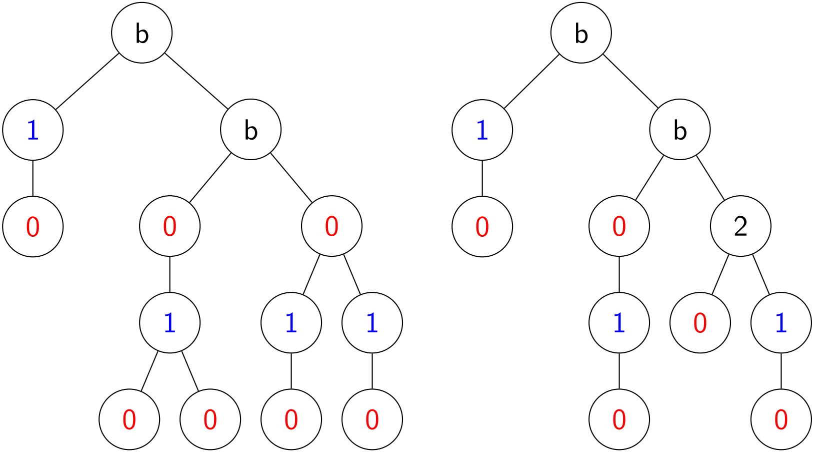
Regular tree language example



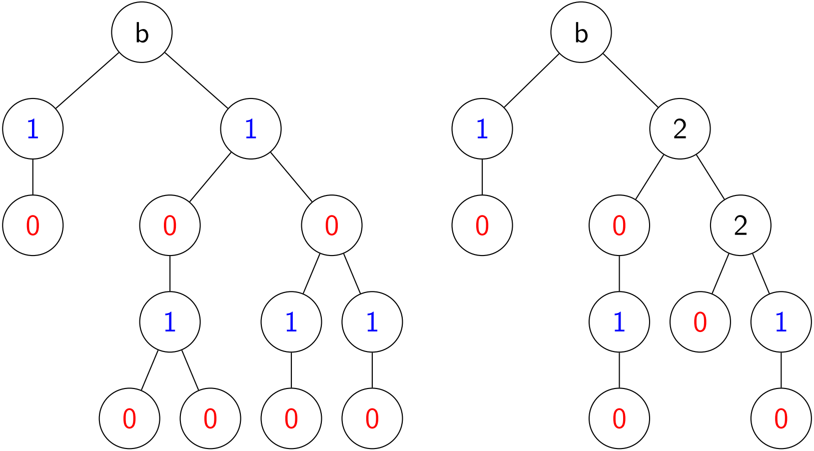
Regular tree language example



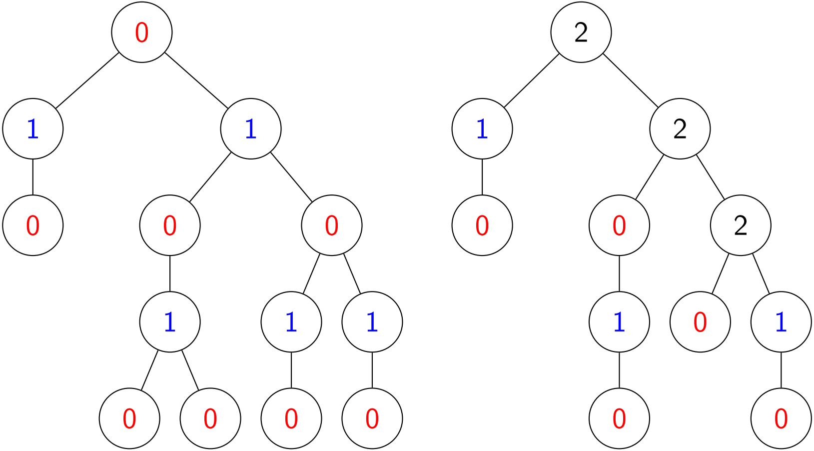
Regular tree language example



Regular tree language example



Regular tree language example



Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with

Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with
  - universe the set of nodes of t,

Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with
  - universe the set of nodes of t,
  - the binary descendant relation $\leq$ of t,

Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with
  - universe the set of nodes of t,
  - the binary descendant relation $\leq$ of t,
  - unary predicates for each symbol from Σ, and

Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with
  - universe the set of nodes of t,
  - the binary descendant relation $\leq$ of t,
  - unary predicates for each symbol from Σ, and
  - an ith child relation for each suitable i.

Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with
  - universe the set of nodes of t,
  - the binary descendant relation $\leq$ of t,
  - unary predicates for each symbol from Σ, and
  - an ith child relation for each suitable i.
- Using various logics, we can express certain tree properties and define some tree languages using these structures.

Trees as models

- Any tree $t \in \text{trees}\Sigma$ for some ranked alphabet Σ can be viewed as a finite relational structure $\mathbb{M}_t$ with
  - universe the set of nodes of t,
  - the binary descendant relation $\leq$ of t,
  - unary predicates for each symbol from Σ, and
  - an ith child relation for each suitable i.
- Using various logics, we can express certain tree properties and define some tree languages using these structures.

Example

The first order sentence:

$$\exists x\,((\forall y \quad y \leq x) \wedge a(x)) \wedge \exists z(b(z))$$

defines the set of Σ-trees that have the symbol a as their root and in which b occurs at least once.

Definable tree languages

Definition

Definable tree languages

Definition

- Monadic second order logic (MSO) is the extension of first order logic that allows for quantification over arbitrary subsets of the universe.

Definable tree languages

Definition

- Monadic second order logic (MSO) is the extension of first order logic that allows for quantification over arbitrary subsets of the universe.
- For ordered structures, chain logic is a version of MSO where quantification only over linearly ordered subsets is allowed.

Definable tree languages

Definition

- Monadic second order logic (MSO) is the extension of first order logic that allows for quantification over arbitrary subsets of the universe.
- For ordered structures, chain logic is a version of MSO where quantification only over linearly ordered subsets is allowed.

Theorem (Thatcher and Wright)

A tree language L is regular if and only if it is definable by an MSO sentence.

Definable tree languages

Definition

- Monadic second order logic (MSO) is the extension of first order logic that allows for quantification over arbitrary subsets of the universe.
- For ordered structures, chain logic is a version of MSO where quantification only over linearly ordered subsets is allowed.

Theorem (Thatcher and Wright)

A tree language L is regular if and only if it is definable by an MSO sentence.

Definability problem

The definability problem for a given fragment of MSO, such as first-order or chain logic, is: given a regular tree language, decide if it is definable by some formula of the logic.

Chain logic definability

- The Σ-tree language $\mathcal{E}$, whose members all have branches of even length, can be defined in Chain logic.

Chain logic definability

- The Σ-tree language $\mathcal{E}$, whose members all have branches of even length, can be defined in Chain logic.
- A Σ-tree t will be in $\mathcal{E}$ if and only if

Chain logic definability

- The Σ-tree language $\mathcal{E}$, whose members all have branches of even length, can be defined in Chain logic.
- A Σ-tree t will be in $\mathcal{E}$ if and only if

  Every maximal chain of t can be partitioned into two subchains X and Y that interleave such that

  - X contains the leaf of the branch and

  - Y contains the root.

Chain logic definability

- The Σ-tree language $\mathcal{E}$, whose members all have branches of even length, can be defined in Chain logic.
- A Σ-tree t will be in $\mathcal{E}$ if and only if

  Every maximal chain of t can be partitioned into two subchains X and Y that interleave such that

  - X contains the leaf of the branch and

  - Y contains the root.
- This tree language is not first order definable, since the property of being of even length cannot be defined in the first order language of Σ-trees.

Definability and the syntactic algebra

Syntactic Algebra

If L is a regular Σ-language, then there is a smallest finite algebra $\mathbb{A}_L$ of type Σ that recognizes L, called the syntactic algebra of L.

Definability and the syntactic algebra

Syntactic Algebra

If L is a regular Σ-language, then there is a smallest finite algebra $\mathbb{A}_L$ of type Σ that recognizes L, called the syntactic algebra of L.

Theorem

Let L be a regular tree language.

Definability and the syntactic algebra

Syntactic Algebra

If L is a regular Σ-language, then there is a smallest finite algebra $\mathbb{A}_L$ of type Σ that recognizes L, called the syntactic algebra of L.

Theorem

Let L be a regular tree language.

- *Being first-order definable is a property of $\mathbb{A}_L$ (but not of $\mathrm{Clo}(\mathbb{A}_L)$).*

Definability and the syntactic algebra

Syntactic Algebra

If L is a regular Σ-language, then there is a smallest finite algebra $\mathbb{A}_L$ of type Σ that recognizes L, called the syntactic algebra of L.

Theorem

Let L be a regular tree language.

- *Being first-order definable is a property of* $\mathbb{A}_L$ *(but not of* $\mathrm{Clo}(\mathbb{A}_L)$*).*
- *Being chain logic definable is a property of* $\mathrm{Pol}(\mathbb{A}_L)$*, the set of polynomials of* $\mathbb{A}_L$*.*

Definability and the syntactic algebra

Syntactic Algebra

If L is a regular Σ-language, then there is a smallest finite algebra $\mathbb{A}_L$ of type Σ that recognizes L, called the syntactic algebra of L.

Theorem

Let L be a regular tree language.

- *Being first-order definable is a property of* $\mathbb{A}_L$ *(but not of* $\mathrm{Clo}(\mathbb{A}_L)$*).*
- *Being chain logic definable is a property of* $\mathrm{Pol}(\mathbb{A}_L)$*, the set of polynomials of* $\mathbb{A}_L$*.*

Theorem (Schützenberger)

Let L be a regular language over an alphabet of rank 1. Then L is first-order definable if and only if there is some $n > 0$ *such that* $\mathbb{A}_L \models t^n(x) \approx t^{n+1}(x)$ *for all unary terms* t*.*

Two problems

First Order definability

Is there an algorithm to decide if a given regular tree language is first order definable?

Two problems

First Order definability

Is there an algorithm to decide if a given regular tree language is first order definable?

Chain Logic definability

Is there an algorithm to decide if a given regular tree language is chain logic definable?

Two problems

First Order definability

Is there an algorithm to decide if a given regular tree language is first order definable?

Chain Logic definability

Is there an algorithm to decide if a given regular tree language is chain logic definable?

Remark

In light of the previous slide, both problems can be re-phrased in terms of properties of syntactic algebras of regular languages.

Two problems

First Order definability

Is there an algorithm to decide if a given regular tree language is first order definable?

Chain Logic definability

Is there an algorithm to decide if a given regular tree language is chain logic definable?

Remark

In light of the previous slide, both problems can be re-phrased in terms of properties of syntactic algebras of regular languages.

Two problems

First Order definability

Is there an algorithm to decide if a given regular tree language is first order definable?

Chain Logic definability

Is there an algorithm to decide if a given regular tree language is chain logic definable?

Remark

In light of the previous slide, both problems can be re-phrased in terms of properties of syntactic algebras of regular languages.
Both problems are still open.

Simulation classes

Remarks

Simulation classes

Remarks

- *Bojańczyk has shown that a tree language is chain logic definable if and only if its syntactic algebra belongs a particular class of finite algebras that is closed under the operations of division, matrix powers, and wreath products.*

Simulation classes

Remarks

- *Bojańczyk has shown that a tree language is chain logic definable if and only if its syntactic algebra belongs a particular class of finite algebras that is closed under the operations of division, matrix powers, and wreath products.*
- *We call such a class of algebras a simulation class and denote the smallest simulation class that contains a class $\mathcal{K}$ of finite algebras by* $\mathbf{Sim}(\mathcal{K})$.

Simulation classes

Remarks

- *Bojańczyk has shown that a tree language is chain logic definable if and only if its syntactic algebra belongs a particular class of finite algebras that is closed under the operations of division, matrix powers, and wreath products.*
- *We call such a class of algebras a simulation class and denote the smallest simulation class that contains a class $\mathcal{K}$ of finite algebras by* $\mathbf{Sim}(\mathcal{K})$.
- *Simulation classes were first studied by Joel VanderWerf in his 1994 PhD thesis.*

Simulation classes

Remarks

- *Bojańczyk has shown that a tree language is chain logic definable if and only if its syntactic algebra belongs a particular class of finite algebras that is closed under the operations of division, matrix powers, and wreath products.*
- *We call such a class of algebras a simulation class and denote the smallest simulation class that contains a class $\mathcal{K}$ of finite algebras by* $\mathbf{Sim}(\mathcal{K})$.
- *Simulation classes were first studied by Joel VanderWerf in his 1994 PhD thesis.*

Definition

An algebra $\mathbb{B}$ is a divisor of an algebra $\mathbb{A}$ if it is a homomorphic image of a subalgebra of a reduct of $\mathbb{A}$.

Simulation classes

Remarks

- *Bojańczyk has shown that a tree language is chain logic definable if and only if its syntactic algebra belongs a particular class of finite algebras that is closed under the operations of division, matrix powers, and wreath products.*
- *We call such a class of algebras a simulation class and denote the smallest simulation class that contains a class $\mathcal{K}$ of finite algebras by* $\mathbf{Sim}(\mathcal{K})$*.*
- *Simulation classes were first studied by Joel VanderWerf in his 1994 PhD thesis.*

Definition

An algebra $\mathbb{B}$ is a divisor of an algebra $\mathbb{A}$ if it is a homomorphic image of a subalgebra of a reduct of $\mathbb{A}$.

Simulation classes

Remarks

- *Bojańczyk has shown that a tree language is chain logic definable if and only if its syntactic algebra belongs a particular class of finite algebras that is closed under the operations of division, matrix powers, and wreath products.*
- *We call such a class of algebras a simulation class and denote the smallest simulation class that contains a class $\mathcal{K}$ of finite algebras by* **Sim**$(\mathcal{K})$.
- *Simulation classes were first studied by Joel VanderWerf in his 1994 PhD thesis.*

Definition

An algebra $\mathbb{B}$ is a divisor of an algebra $\mathbb{A}$ if it is a homomorphic image of a subalgebra of a reduct of $\mathbb{A}$. $\mathbf{D}(\mathcal{K})$ denotes the class of divisors of the algebras in $\mathcal{K}$.

Matrix powers

Definition

Let $\mathbb{A}$ be an algebra with clone $\mathcal{C} = \mathrm{Clo}(\mathbb{A})$ and let $n > 0$. The *n-th matrix power of $\mathbb{A}$* is the algebra with universe A^n and whose basic operations are all functions $f(\bar{x}_1, \ldots, \bar{x}_k)$ of the form: for some choice of nk-ary functions $f_1, f_2, \ldots, f_n \in \mathcal{C}$,

$$(\bar{a}_1, \ldots, \bar{a}_k) \mapsto (f_1(\bar{a}_1, \ldots, \bar{a}_k), \ldots, f_n(\bar{a}_1, \ldots, \bar{a}_k))$$

We denote this algebra by $\mathbb{A}^{[n]}$. For $\mathcal{K}$ a class of algebras, let $\mathbf{M}(\mathcal{K})$ denote the set of matrix powers of algebras in $\mathcal{K}$.

Matrix powers

Definition

Let $\mathbb{A}$ be an algebra with clone $\mathcal{C} = \mathrm{Clo}(\mathbb{A})$ and let $n > 0$. The *n-th matrix power of $\mathbb{A}$* is the algebra with universe A^n and whose basic operations are all functions $f(\bar{x}_1, \dots, \bar{x}_k)$ of the form: for some choice of nk-ary functions $f_1, f_2, \dots, f_n \in \mathcal{C}$,

$$(\bar{a}_1, \dots, \bar{a}_k) \mapsto (f_1(\bar{a}_1, \dots, \bar{a}_k), \dots, f_n(\bar{a}_1, \dots, \bar{a}_k))$$

We denote this algebra by $\mathbb{A}^{[n]}$. For $\mathcal{K}$ a class of algebras, let $\mathbf{M}(\mathcal{K})$ denote the set of matrix powers of algebras in $\mathcal{K}$.

Matrix powers

Definition

Let $\mathbb{A}$ be an algebra with clone $\mathcal{C} = \mathrm{Clo}(\mathbb{A})$ and let $n > 0$. The n-th matrix power of $\mathbb{A}$ is the algebra with universe A^n and whose basic operations are all functions $f(\bar{x}_1, \ldots, \bar{x}_k)$ of the form: for some choice of nk-ary functions f_1, f_2, $\ldots$, $f_n \in \mathcal{C}$,

$$(\bar{a}_1, \ldots, \bar{a}_k) \mapsto (f_1(\bar{a}_1, \ldots, \bar{a}_k), \ldots, f_n(\bar{a}_1, \ldots, \bar{a}_k))$$

We denote this algebra by $\mathbb{A}^{[n]}$. For $\mathcal{K}$ a class of algebras, let $\mathbf{M}(\mathcal{K})$ denote the set of matrix powers of algebras in $\mathcal{K}$.

Remarks

Matrix powers

Definition

Let $\mathbb{A}$ be an algebra with clone $\mathcal{C} = \mathrm{Clo}(\mathbb{A})$ and let $n > 0$. The n-th matrix power of $\mathbb{A}$ is the algebra with universe A^n and whose basic operations are all functions $f(\bar{x}_1, \ldots, \bar{x}_k)$ of the form: for some choice of nk-ary functions $f_1, f_2, \ldots, f_n \in \mathcal{C}$,

$$(\bar{a}_1, \ldots, \bar{a}_k) \mapsto (f_1(\bar{a}_1, \ldots, \bar{a}_k), \ldots, f_n(\bar{a}_1, \ldots, \bar{a}_k))$$

We denote this algebra by $\mathbb{A}^{[n]}$. For $\mathcal{K}$ a class of algebras, let $\mathbf{M}(\mathcal{K})$ denote the set of matrix powers of algebras in $\mathcal{K}$.

Remarks

- *The collection of operations of $\mathbb{A}^{[n]}$ is a clone.*

Matrix powers

Definition

Let $\mathbb{A}$ be an algebra with clone $\mathcal{C} = \mathrm{Clo}(\mathbb{A})$ and let $n > 0$. The n-th matrix power of $\mathbb{A}$ is the algebra with universe A^n and whose basic operations are all functions $f(\bar{x}_1, \ldots, \bar{x}_k)$ of the form: for some choice of nk-ary functions $f_1, f_2, \ldots, f_n \in \mathcal{C}$,

$$(\bar{a}_1, \ldots, \bar{a}_k) \mapsto (f_1(\bar{a}_1, \ldots, \bar{a}_k), \ldots, f_n(\bar{a}_1, \ldots, \bar{a}_k))$$

We denote this algebra by $\mathbb{A}^{[n]}$. For $\mathcal{K}$ a class of algebras, let $\mathbf{M}(\mathcal{K})$ denote the set of matrix powers of algebras in $\mathcal{K}$.

Remarks

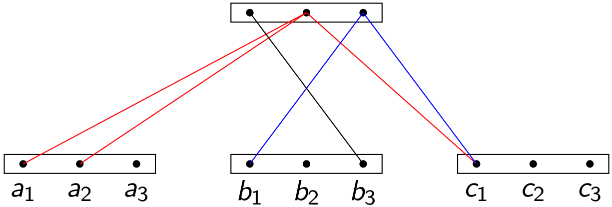
- *The collection of operations of* $\mathbb{A}^{[n]}$ *is a clone.*
- *The matrix power construction is a generalization of the cartesian power of an algebra in the sense that* $\mathbb{A}^n$ *is a reduct of* $\mathbb{A}^{[n]}$.

Matrix Power Example

The function

$$((a_1, a_2, a_3), (b_1, b_2, b_3), (c_1, c_2, c_3)) \to (b_3, a_1 \vee a_2 \vee c_1, b_1 \vee c_1)$$

is in the clone of $\langle \{0, 1\}, \vee \rangle^{[3]}$.



Wreath products

Definition

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F} \rangle$ and $\mathbb{B} = \langle B, \mathcal{G} \rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F} \rangle$ and $\mathbb{B} = \langle B, \mathcal{G} \rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.
- The wreath product of $\mathbb{A}$ and $\mathbb{B}$ is the algebra with universe $A \times B$ that has the following set of basic operations:

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F}\rangle$ and $\mathbb{B} = \langle B, \mathcal{G}\rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.
- The wreath product of $\mathbb{A}$ and $\mathbb{B}$ is the algebra with universe $A \times B$ that has the following set of basic operations:
    - For $k \geq 1$, let $g(x_1, \ldots, x_k) \in \mathcal{G}$, and

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F} \rangle$ and $\mathbb{B} = \langle B, \mathcal{G} \rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.
- The wreath product of $\mathbb{A}$ and $\mathbb{B}$ is the algebra with universe $A \times B$ that has the following set of basic operations:
  - For $k \geq 1$, let $g(x_1, \ldots, x_k) \in \mathcal{G}$, and
  - $\bar{h}$ be a tuple of k-ary functions in $\mathcal{F}$ indexed by B^k (so $\bar{h} \in \mathcal{F}_k^{B^k}$).

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F} \rangle$ and $\mathbb{B} = \langle B, \mathcal{G} \rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.
- The wreath product of $\mathbb{A}$ and $\mathbb{B}$ is the algebra with universe $A \times B$ that has the following set of basic operations:
  - For $k \geq 1$, let $g(x_1, \ldots, x_k) \in \mathcal{G}$, and
  - $\bar{h}$ be a tuple of k-ary functions in $\mathcal{F}$ indexed by B^k (so $\bar{h} \in \mathcal{F}_k^{B^k}$).
- Define $(\bar{h}, g)$ to be the k-ary operation on $A \times B$ that maps $((a_1, b_1), \ldots, (a_k, b_k))$ to

$$(\bar{h}(b_1, \ldots, b_k)(a_1, \ldots, a_k), g(b_1, \ldots, b_k)).$$

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F} \rangle$ and $\mathbb{B} = \langle B, \mathcal{G} \rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.
- The wreath product of $\mathbb{A}$ and $\mathbb{B}$ is the algebra with universe $A \times B$ that has the following set of basic operations:
  - For $k \geq 1$, let $g(x_1, \dots, x_k) \in \mathcal{G}$, and
  - $\bar{h}$ be a tuple of k-ary functions in $\mathcal{F}$ indexed by B^k (so $\bar{h} \in \mathcal{F}_k^{B^k}$).
- Define $(\bar{h}, g)$ to be the k-ary operation on $A \times B$ that maps $((a_1, b_1), \dots, (a_k, b_k))$ to

$$(\bar{h}(b_1, \dots, b_k)(a_1, \dots, a_k), g(b_1, \dots, b_k)).$$

- We denote this algebra by $\mathbb{A} \circ \mathbb{B}$

Wreath products

Definition

- Let $\mathbb{A} = \langle A, \mathcal{F} \rangle$ and $\mathbb{B} = \langle B, \mathcal{G} \rangle$ be algebras with $\mathcal{F}$ and $\mathcal{G}$ clones.
- The wreath product of $\mathbb{A}$ and $\mathbb{B}$ is the algebra with universe $A \times B$ that has the following set of basic operations:
  - For $k \geq 1$, let $g(x_1, \ldots, x_k) \in \mathcal{G}$, and
  - $\bar{h}$ be a tuple of k-ary functions in $\mathcal{F}$ indexed by B^k (so $\bar{h} \in \mathcal{F}_k^{B^k}$).
- Define $(\bar{h}, g)$ to be the k-ary operation on $A \times B$ that maps $((a_1, b_1), \ldots, (a_k, b_k))$ to

$$(\bar{h}(b_1, \ldots, b_k)(a_1, \ldots, a_k), g(b_1, \ldots, b_k)).$$

- We denote this algebra by $\mathbb{A} \circ \mathbb{B}$
- For $\mathcal{K}$ a class of algebras, let $\mathbf{Wr}(\mathcal{K})$ denote the class of all wreath products of members from $\mathcal{K}$.

Wreath products

Remarks

Wreath products

Remarks

- *In general the wreath product is not commutative.*

Wreath products

Remarks

- *In general the wreath product is not commutative.*
- $\mathbb{B}$ *is a homomorphic image of* $\mathbb{A} \circ \mathbb{B}$ *via the projection map onto the second coordinate.*

Wreath products

Remarks

- *In general the wreath product is not commutative.*
- $\mathbb{B}$ *is a homomorphic image of* $\mathbb{A} \circ \mathbb{B}$ *via the projection map onto the second coordinate.*
- $\mathrm{Con}\,(\mathbb{A} \circ \mathbb{B})$ *is isomorphic to a copy of* $\mathrm{Con}\,(\mathbb{B})$ *glued on top of a copy of* $\mathrm{Con}\,(\mathbb{A})$.

Wreath products

Remarks

- *In general the wreath product is not commutative.*
- $\mathbb{B}$ *is a homomorphic image of* $\mathbb{A} \circ \mathbb{B}$ *via the projection map onto the second coordinate.*
- $\mathrm{Con}\,(\mathbb{A} \circ \mathbb{B})$ *is isomorphic to a copy of* $\mathrm{Con}\,(\mathbb{B})$ *glued on top of a copy of* $\mathrm{Con}\,(\mathbb{A})$*.*
- $\mathbb{A}$ *and* $\mathbb{B}$ *are subreducts of* $\mathbb{A} \circ \mathbb{B}$*.*

Wreath products

Remarks

- *In general the wreath product is not commutative.*
- $\mathbb{B}$ *is a homomorphic image of* $\mathbb{A} \circ \mathbb{B}$ *via the projection map onto the second coordinate.*
- $\mathrm{Con}\,(\mathbb{A} \circ \mathbb{B})$ *is isomorphic to a copy of* $\mathrm{Con}\,(\mathbb{B})$ *glued on top of a copy of* $\mathrm{Con}\,(\mathbb{A})$*.*
- $\mathbb{A}$ *and* $\mathbb{B}$ *are subreducts of* $\mathbb{A} \circ \mathbb{B}$*.*
- *If* $\mathbb{A}' \prec \mathbb{A}$ *and* $\mathbb{B}' \prec \mathbb{B}$ *then* $\mathbb{A}' \circ \mathbb{B}' \prec \mathbb{A} \circ \mathbb{B}$ *so* $\mathbf{WrD}(\mathcal{K}) \subseteq \mathbf{DWr}(\mathcal{K})$*.*

Wreath products

Remarks

- *In general the wreath product is not commutative.*
- $\mathbb{B}$ *is a homomorphic image of* $\mathbb{A} \circ \mathbb{B}$ *via the projection map onto the second coordinate.*
- $\mathrm{Con}\,(\mathbb{A} \circ \mathbb{B})$ *is isomorphic to a copy of* $\mathrm{Con}\,(\mathbb{B})$ *glued on top of a copy of* $\mathrm{Con}\,(\mathbb{A})$.
- $\mathbb{A}$ *and* $\mathbb{B}$ *are subreducts of* $\mathbb{A} \circ \mathbb{B}$.
- *If* $\mathbb{A}' \prec \mathbb{A}$ *and* $\mathbb{B}' \prec \mathbb{B}$ *then* $\mathbb{A}' \circ \mathbb{B}' \prec \mathbb{A} \circ \mathbb{B}$ *so* $\mathbf{WrD}(\mathcal{K}) \subseteq \mathbf{DWr}(\mathcal{K})$.
- *Matrix powers and wreath products commute:* $(\mathbb{A} \circ \mathbb{B})^{[k]}$ *is isomorphic to* $\mathbb{A}^{[k]} \circ \mathbb{B}^{[k]}$, *so* $\mathbf{WrM}(\mathcal{K}) = \mathbf{MWr}(\mathcal{K})$.

Wreath products

Remarks

- *In general the wreath product is not commutative.*
- $\mathbb{B}$ *is a homomorphic image of* $\mathbb{A} \circ \mathbb{B}$ *via the projection map onto the second coordinate.*
- $\mathrm{Con}\,(\mathbb{A} \circ \mathbb{B})$ *is isomorphic to a copy of* $\mathrm{Con}\,(\mathbb{B})$ *glued on top of a copy of* $\mathrm{Con}\,(\mathbb{A})$.
- $\mathbb{A}$ *and* $\mathbb{B}$ *are subreducts of* $\mathbb{A} \circ \mathbb{B}$.
- *If* $\mathbb{A}' \prec \mathbb{A}$ *and* $\mathbb{B}' \prec \mathbb{B}$ *then* $\mathbb{A}' \circ \mathbb{B}' \prec \mathbb{A} \circ \mathbb{B}$ *so* $\mathbf{WrD}(\mathcal{K}) \subseteq \mathbf{DWr}(\mathcal{K})$.
- *Matrix powers and wreath products commute:* $(\mathbb{A} \circ \mathbb{B})^{[k]}$ *is isomorphic to* $\mathbb{A}^{[k]} \circ \mathbb{B}^{[k]}$, *so* $\mathbf{WrM}(\mathcal{K}) = \mathbf{MWr}(\mathcal{K})$.
- *So, the smallest class of algebras that contains* $\mathcal{K}$ *and is closed under division, matrix powers, and wreath products is* $\mathbf{DWrM}(\mathcal{K})$, *i.e.,* $\mathbf{Sim}(\mathcal{K}) = \mathbf{DWrM}(\mathcal{K})$.

Chain Logic

Theorem (Bojańczyk)

The class of tree languages recognized by algebras in $\mathbf{Sim}(\langle\{0,1\},\vee\rangle)$ *is equal to the class of tree languages that are definable in Chain Logic.*

Chain Logic

Theorem (Bojańczyk)

The class of tree languages recognized by algebras in $\mathbf{Sim}(\langle\{0,1\},\vee\rangle)$ *is equal to the class of tree languages that are definable in Chain Logic.*

Corollary

The definability problem for Chain Logic is equivalent to the membership problem for the simulation class $\mathbf{Sim}(\langle\{0,1\},\vee\rangle)$.

Chain Logic

Theorem (Bojańczyk)

The class of tree languages recognized by algebras in $\mathbf{Sim}(\langle\{0,1\},\vee\rangle)$ *is equal to the class of tree languages that are definable in Chain Logic.*

Corollary

The definability problem for Chain Logic is equivalent to the membership problem for the simulation class $\mathbf{Sim}(\langle\{0,1\},\vee\rangle)$.

Remark

Using the work of VanderWerf, this problem reduces to the problem of determining if there is a procedure to decide if a given finite simple algebra of semilattice type is equal to a divisor of wreath products of matrix powers of $\langle\{0,1\},\vee\rangle$.

Simulation classes

- VanderWerf has established a number of interesting/surprising results about simulation classes.

Simulation classes

- VanderWerf has established a number of interesting/surprising results about simulation classes.
- For example, the class of finite strongly solvable algebras is equal to $\mathbf{Sim}(\langle\{0,1\}\rangle)$, the smallest non-trivial simulation class, and

Simulation classes

- VanderWerf has established a number of interesting/surprising results about simulation classes.
- For example, the class of finite strongly solvable algebras is equal to $\mathbf{Sim}(\langle\{0,1\}\rangle)$, the smallest non-trivial simulation class, and
- the class of finite solvable algebras is equal to $\mathbf{Sim}(\{\mathbb{Z}_p \mid p \text{ prime}\})$.

Simulation classes

- VanderWerf has established a number of interesting/surprising results about simulation classes.
- For example, the class of finite strongly solvable algebras is equal to $\mathbf{Sim}(\langle\{0,1\}\rangle)$, the smallest non-trivial simulation class, and
- the class of finite solvable algebras is equal to $\mathbf{Sim}(\{\mathbb{Z}_p \mid p \text{ prime}\})$.
- If $\mathbb{A}$ is a finite algebra that admits the lattice or boolean type then $\mathbf{Sim}(\mathbb{A})$ is the class of all finite algebras.