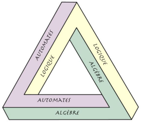
Separating languages with no ambiguity

Marc Zeitoun

Joint work with Thomas Place

LaBRI, U. Bordeaux

ANR Delta@IRIF, March 2018



Concatenation and Alternation Hierarchies

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

$$(ab)^+ =$$

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

$$(ab)^+ = a\overline{\emptyset}$$

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

$$(ab)^+ = a\overline{\emptyset} \cap \overline{\emptyset}b$$

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

$$(ab)^+ = a\overline{\emptyset} \cap \overline{\emptyset} b \cap \overline{\overline{\emptyset}(aa + bb)\overline{\emptyset}}$$

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

$$(ab)^+ = a\overline{\emptyset} \cap \overline{\emptyset}b \cap \overline{\overline{\emptyset}(aa+bb)\overline{\emptyset}} \cap \bigcap_{c \neq a,b} \overline{\overline{\emptyset}c\overline{\emptyset}}$$

The membership problem for classes of languages

Membership problem for a class $\mathcal{C}$

- **INPUT** A (regular) language L.
- **QUESTION** Does L belong to $\mathcal{C}$?

Examples of classes $\mathcal{C}$:

- Languages definable in **FO**.
- Languages of GSH $k \geq 1$.
- Languages of GSH 0 (called star-free, denoted SF).

Schützenberger '65

For L a regular language, the following are equivalent:

- L is star-free. **semantic**
- The minimal automaton of L is counter-free. **syntactic**

Why is Schützenberger's theorem interesting?

1. Provides an effective characterization of **SF**.

Why is Schützenberger's theorem interesting?

1. Provides an effective characterization of **SF**.

2. Link with first-order logic **FO**.

Schützenberger '65, McNaughton, Papert '71

For L a regular language, the following are equivalent:

- L is star-free. **semantic**
- L is **FO**-definable. **semantic**
- The minimal automaton of L is counter-free. **syntactic**

Why is Schützenberger's theorem interesting?

1. Provides an effective characterization of **SF**.

2. Link with first-order logic **FO**.

Schützenberger '65, McNaughton, Papert '71

For L a regular language, the following are equivalent:

- L is star-free. **semantic**
- L is **FO**-definable. **semantic**
- The minimal automaton of L is counter-free. **syntactic**

SF	**FO**
$A^*, \emptyset$	True, False
$\cup, \ A^* \setminus$	$\vee, \ \neg$
KaL	$\exists x \ a(x) \wedge \varphi_K^{<x}(x) \wedge \varphi_L^{>x}(x)$

Why is Schützenberger's theorem interesting?

1. Provides an effective characterization of **SF**.

2. Link with first-order logic **FO**.

Schützenberger '65, McNaughton, Papert '71

For L a regular language, the following are equivalent:

- L is star-free. **semantic**
- L is **FO**-definable. **semantic**
- The minimal automaton of L is counter-free. **syntactic**

SF	**FO**
$A^*, \emptyset$	True, False
$\cup,\ A^* \setminus$	$\vee,\ \neg$
KaL	$\exists x\ a(x) \wedge \varphi_K^{<x}(x) \wedge \varphi_L^{>x}(x)$

3. Constructive proof $\Rightarrow$ normal forms for star-free expressions, **FO** sentences.

Concatenation hierarchies: Motivation

Definition of SF alternate Boolean operations and concatenation

- **SF** = smallest class such that:
  - $\emptyset \in \mathsf{SF}$ and $A^* \in \mathsf{SF}$.
  - SF is closed under Boolean operations over A^*.
  - SF is closed under marked concatenation.

$$\begin{cases} K, L \in \mathsf{SF} \\ a \in A \end{cases} \implies KaL \in \mathsf{SF}$$

Concatenation hierarchies: Motivation

Definition of SF alternate Boolean operations and concatenation

- **SF** = smallest class such that:
  - $\emptyset \in \mathrm{SF}$ and $A^* \in \mathrm{SF}$.
  - SF is closed under Boolean operations over A^*.
  - SF is closed under marked concatenation.

$$\begin{cases} K, L \in \mathrm{SF} \\ a \in A \end{cases} \implies KaL \in \mathrm{SF}$$

Goal

- Finer understanding of FO / SF.
- Interplay between complement and concatenation.

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

How many needed alternations between Boolean operations and concatenations?

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

How many needed alternations between Boolean operations and concatenations?

Straubing-Thérien hierarchy '81

- $\text{ST}[0] = \{\emptyset, A^*\}$.

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

How many needed alternations between Boolean operations and concatenations?

Straubing-Thérien hierarchy '81

- $\mathrm{ST}[0] = \{\emptyset, A^*\}$.
- $\mathrm{ST}\left[n + \frac{1}{2}\right] = Pol(\mathrm{ST}[n])$.

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

How many needed alternations between Boolean operations and concatenations?

Straubing-Thérien hierarchy '81

- $\mathrm{ST}[0] = \{\emptyset, A^*\}$.
- $\mathrm{ST}\left[n+\frac{1}{2}\right] = Pol(\mathrm{ST}[n])$.
- $\mathrm{ST}[n+1] = Bool(\mathrm{ST}\left[n+\frac{1}{2}\right])$.

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

How many needed alternations between Boolean operations and concatenations?

Straubing-Thérien hierarchy '81

- $\text{ST}[0] = \{\emptyset, A^*\}$.
- $\text{ST}\left[n + \frac{1}{2}\right] = Pol(\text{ST}[n])$.
- $\text{ST}[n+1] = Bool(\text{ST}\left[n + \frac{1}{2}\right])$.

Brzozowski-Cohen hierarchy '71

- $\text{DD}[0] = \{\emptyset, \{\varepsilon\}, A^+, A^*\}$.
- $\text{DD}\left[n + \frac{1}{2}\right] = Pol(\text{DD}[n])$.
- $\text{DD}[n+1] = Bool(\text{DD}\left[n + \frac{1}{2}\right])$.

Classical hierarchies inside SF

Two classes built on top of $\mathcal{C}$

- Boolean closure $Bool(\mathcal{C})$.
- Polynomial closure $Pol(\mathcal{C})$ = closure under **marked concatenation** + **union**

How many needed alternations between Boolean operations and concatenations?

Straubing-Thérien hierarchy '81

- $\mathrm{ST}[0] = \{\emptyset, A^*\}$.
- $\mathrm{ST}\left[n+\frac{1}{2}\right] = Pol(\mathrm{ST}[n])$.
- $\mathrm{ST}[n+1] = Bool(\mathrm{ST}\left[n+\frac{1}{2}\right])$.

Brzozowski-Cohen hierarchy '71

- $\mathrm{DD}[0] = \{\emptyset, \{\varepsilon\}, A^+, A^*\}$.
- $\mathrm{DD}\left[n+\frac{1}{2}\right] = Pol(\mathrm{DD}[n])$.
- $\mathrm{DD}[n+1] = Bool(\mathrm{DD}\left[n+\frac{1}{2}\right])$.

$$0 \xrightarrow{Pol} \tfrac{1}{2} \xrightarrow[Bool]{} 1 \xrightarrow{Pol} \tfrac{3}{2} \xrightarrow[Bool]{} 2 \xrightarrow{Pol} \tfrac{5}{2} \cdots\cdots$$

Brzozowski-Cohen and Straubing-Thérien hierarchies

Natural questions

- Are the hierarchies strict?
- Logical description of each level?
- What is known about membership?

Brzozowski-Cohen and Straubing-Thérien hierarchies

Natural questions

- Are the hierarchies strict?
- Logical description of each level?
- What is known about membership?

What is known

1. Both hierarchies are strict (Brzozowski-Knast 1978 + interleaving),

$$(a \cdots (a(ab)^* b)^* \cdots b)^*$$

2. Natural logical description wihin FO.
3. Membership for BC reduces to membership for ST.
4. Membership solved for only few levels.

Logical counterpart: quantifier alternation hierarchies

Intuition: marked concatenation corresponds to existential quantification.

- $\Sigma_i = \underbrace{\exists^*\forall^*\exists^*\forall^*\exists^*\cdots}_{\text{at most } i \text{ blocks } \exists^* \text{ or } \forall^*} \quad \varphi, \qquad (\varphi \text{ quantifier free}).$
- $\mathcal{B}\Sigma_i$ = Finite Boolean combinations of Σ_i.

Logical counterpart: quantifier alternation hierarchies

Intuition: marked concatenation corresponds to existential quantification.

- $\Sigma_i = \underbrace{\exists^*\forall^*\exists^*\forall^*\exists^*\cdots}_{\text{at most } i \text{ blocks } \exists^* \text{ or } \forall^*} \quad \varphi, \qquad (\varphi \text{ quantifier free}).$
- $\mathcal{B}\Sigma_i$ = Finite Boolean combinations of Σ_i.

Quantifier Alternation Hierarchies

$\Sigma_1 \subsetneq \mathcal{B}\Sigma_1 \subsetneq \Sigma_2 \subsetneq \mathcal{B}\Sigma_2 \subsetneq \Sigma_3 \subsetneq \mathcal{B}\Sigma_3 \subsetneq \Sigma_4 \cdots \subsetneq \cdots \text{FO}$

Logical counterpart: quantifier alternation hierarchies

Intuition: marked concatenation corresponds to existential quantification.

- $\Sigma_i = \underbrace{\exists^*\forall^*\exists^*\forall^*\exists^*\cdots}_{\text{at most } i \text{ blocks } \exists^* \text{ or } \forall^*} \quad \varphi, \qquad (\varphi \text{ quantifier free}).$
- $\mathcal{B}\Sigma_i$ = Finite Boolean combinations of Σ_i.

Quantifier Alternation Hierarchies

$\Sigma_1 \subsetneq \mathcal{B}\Sigma_1 \subsetneq \Sigma_2 \subsetneq \mathcal{B}\Sigma_2 \subsetneq \Sigma_3 \subsetneq \mathcal{B}\Sigma_3 \subsetneq \Sigma_4 \cdots\cdots \subsetneq \cdots\cdots \mathrm{FO}$

Two versions

- Order signature: $<$ and $a()$.
- Enriched signature: $<$, $a()$, $+1$, $\min()$, $\max()$ and ε.

Logical counterpart: quantifier alternation hierarchies

Intuition: marked concatenation corresponds to existential quantification.

- $\Sigma_i = \underbrace{\exists^*\forall^*\exists^*\forall^*\exists^* \cdots}_{\text{at most } i \text{ blocks } \exists^* \text{ or } \forall^*} \quad \varphi,$ (φ quantifier free).
- $\mathcal{B}\Sigma_i$ = Finite Boolean combinations of Σ_i.

Quantifier Alternation Hierarchies

$$\Sigma_1 \subsetneq \mathcal{B}\Sigma_1 \subsetneq \Sigma_2 \subsetneq \mathcal{B}\Sigma_2 \subsetneq \Sigma_3 \subsetneq \mathcal{B}\Sigma_3 \subsetneq \Sigma_4 \cdots \subsetneq \cdots \mathrm{FO}$$

Σ_1	$\mathcal{B}\Sigma_1$	Σ_2	$\mathcal{B}\Sigma_2$	Σ_3	$\mathcal{B}\Sigma_3$	Σ_4
$\frac{1}{2}$	1	$\frac{3}{2}$	2	$\frac{5}{2}$	3	$\frac{7}{2}$

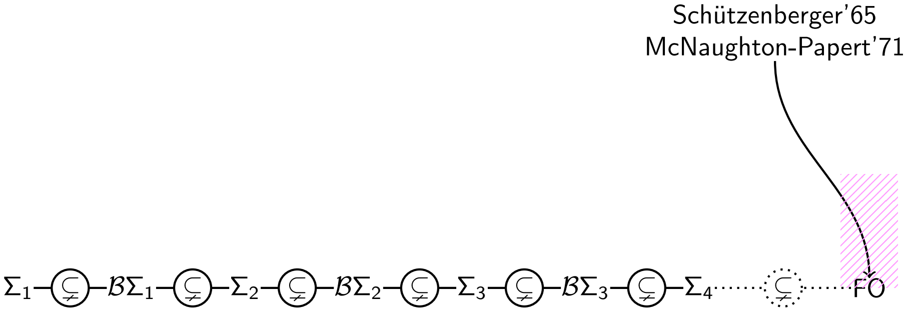
Two versions

- Order signature: $<$ and $a()$.
- Enriched signature: $<$, $a()$, $+1$, $\min()$, $\max()$ and ε.

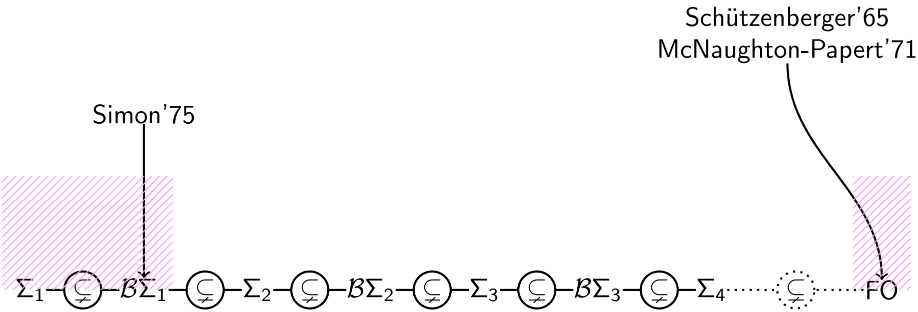
Logical Correspondence Theorem (Thomas '82, Perrin-Pin '86)

- Brzozowski-Cohen hierarchy = enriched quantifier alternation hierarchy.
- Straubing-Thérien hierarchy = order quantifier alternation hierarchy.

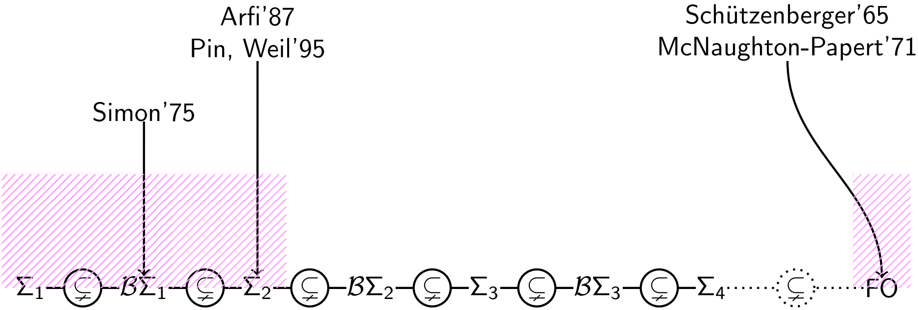
The membership problem for BC and ST hierarchies



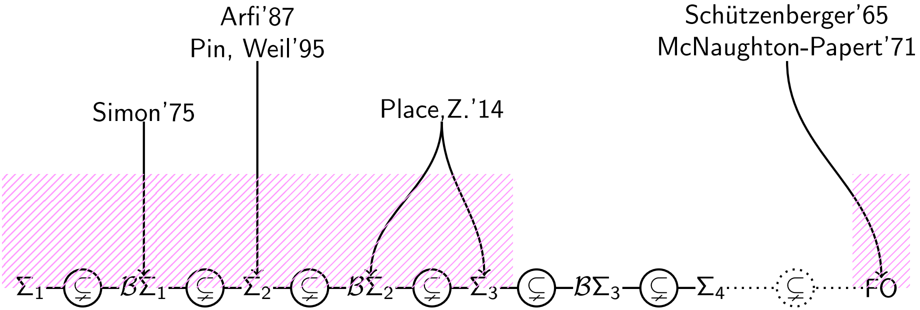
The membership problem for BC and ST hierarchies



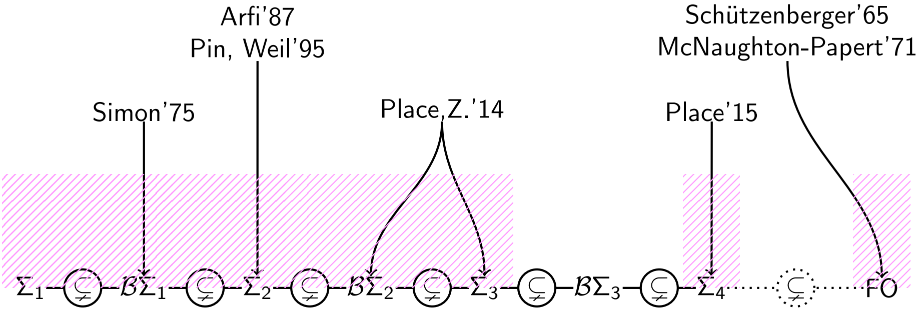
The membership problem for BC and ST hierarchies



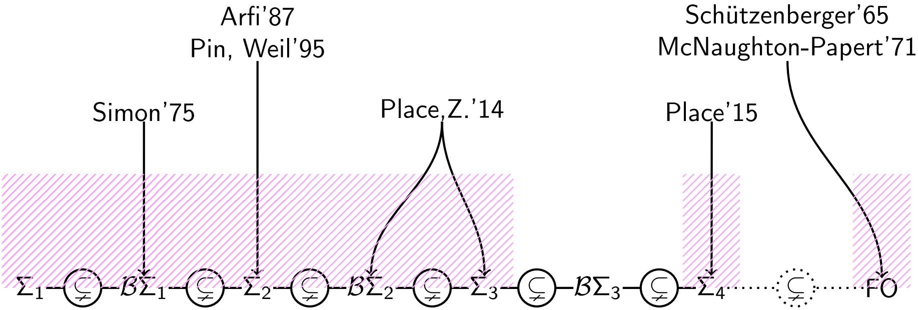
The membership problem for BC and ST hierarchies



The membership problem for BC and ST hierarchies



The membership problem for BC and ST hierarchies



Enrichment Theorem for membership (Straubing '85 – Pin, Weil '97)

Membership for a level in the enriched hierarchy (ie, BC)
reduces to
Membership for the same level in the order hierarchy (ie, ST).

Generalizations in two directions

1. Proofs are ad hoc for DD and ST: obtain generic theorems.
   For given $\mathcal{C}$, what about $Pol(\mathcal{C})$, $Bool(Pol(\mathcal{C}))$,...

2. Recent results via generalizations of membership: separation and covering.

Generic concatenation hierarchies

Generic pattern parametrized by the basis

- $\mathcal{C}[0]$ (basis) **Boolean algebra in REG closed under left/right quotients**

Generic concatenation hierarchies

Generic pattern parametrized by the basis

- $\mathcal{C}[0]$ (basis) **Boolean algebra in REG closed under left/right quotients**

  $\mathcal{C}[n+\frac{1}{2}]$: close $\mathcal{C}[n]$ under $K, L \mapsto KaL$ and $\cup$.

  $\mathcal{C}[n+1]$: close $\mathcal{C}[n+\frac{1}{2}]$ under Boolean operations.

$$0 \xrightarrow{Pol} \frac{1}{2} \xrightarrow[Bool]{} 1 \xrightarrow{Pol} \frac{3}{2} \xrightarrow[Bool]{} 2 \xrightarrow{Pol} \frac{5}{2} \cdots\cdots$$

Generic concatenation hierarchies

Generic pattern parametrized by the basis

- $\mathcal{C}[0]$ (basis) **Boolean algebra in REG closed under left/right quotients**
  $\mathcal{C}[n+\frac{1}{2}]$: close $\mathcal{C}[n]$ under $K, L \mapsto KaL$ and $\cup$.
  $\mathcal{C}[n+1]$: close $\mathcal{C}[n+\frac{1}{2}]$ under Boolean operations.

$$0 \xrightarrow{Pol} \tfrac{1}{2} \xrightarrow[Bool]{} 1 \xrightarrow{Pol} \tfrac{3}{2} \xrightarrow[Bool]{} 2 \xrightarrow{Pol} \tfrac{5}{2} \cdots\cdots$$

Examples

- Straubing-Thérien: $\mathcal{C}[0] = \{\emptyset, A^*\}$.
- Brzozowski-Cohen: $\mathcal{C}[0] = \{\emptyset, \{\varepsilon\}, A^*, A^+\}$.
- Pin-Margolis: $\mathcal{C}[0] =$ group languages.

Generic Hierarchies

Natural questions

- Are the hierarchies strict?
- Logical description of each level?
- What is known about membership?

Basic properties of generic hierarchies

Basic Structure (Place, Z. '17)

For any hierarchy:

- All levels are closed under quotient and in REG.
- Full levels are closed under Bool.
- Half levels are closed under union, intersection (and concatenation).

Quotienting: closed under $u^{-1}L = \{x \mid ux \in L\}$ and Lu^{-1}.

Basic properties of generic hierarchies

Basic Structure (Place, Z. '17)

For any hierarchy:

- All levels are closed under quotient and in REG.
- Full levels are closed under Bool.
- Half levels are closed under union, intersection (and concatenation).

Quotienting: closed under $u^{-1}L = \{x \mid ux \in L\}$ and Lu^{-1}.

Strictness Theorem (Place, Z. '17)

Any hierarchy whose basis is finite is **strict**.

Generic logical correspondence

Logical Correspondence Theorem (Place, Z. '17)

For any basis $\mathcal{C}$, there is a natural set $\mathcal{S}$ of first order predicates, st.

Concatenation hierarchy of basis $\mathcal{C}$

=

Quantifier alternation hierarchy over signature $\mathcal{S}$

Generalizes the correspondences discovered for BC and ST hierarchies.

Generic logical correspondence

Logical Correspondence Theorem (Place, Z. '17)

For any basis $\mathcal{C}$, there is a natural set $\mathcal{S}$ of first order predicates, st.

Concatenation hierarchy of basis $\mathcal{C}$
=
Quantifier alternation hierarchy over signature $\mathcal{S}$

Generalizes the correspondences discovered for BC and ST hierarchies.

Intuition

For each $L \in \mathcal{C}$, add 4 predicates in addition to $<$ and $a(), b(), \dots$

- $w \models I_L(x, y)$ when $x < y$ and $w]x, y[\in L$ (*Infix*).

Generic logical correspondence

Logical Correspondence Theorem (Place, Z. '17)

For any basis $\mathcal{C}$, there is a natural set $\mathcal{S}$ of first order predicates, st.

Concatenation hierarchy of basis $\mathcal{C}$
=
Quantifier alternation hierarchy over signature $\mathcal{S}$

Generalizes the correspondences discovered for BC and ST hierarchies.

Intuition

For each $L \in \mathcal{C}$, add 4 predicates in addition to $<$ and $a(), b(), \dots$

- $w \models I_L(x, y)$ when $x < y$ and $w]x, y[\in L$ (*Infix*).
- $w \models P_L(y)$ when $w[1, y[\in L$ (*Prefix*).
- $w \models S_L(x)$ when $w]x, n] \in L$ (*Suffix*).

Generic logical correspondence

Logical Correspondence Theorem (Place, Z. '17)

For any basis $\mathcal{C}$, there is a natural set $\mathcal{S}$ of first order predicates, st.

Concatenation hierarchy of basis $\mathcal{C}$
=
Quantifier alternation hierarchy over signature $\mathcal{S}$

Generalizes the correspondences discovered for BC and ST hierarchies.

Intuition

For each $L \in \mathcal{C}$, add 4 predicates in addition to $<$ and $a(), b(), \dots$

- $w \models I_L(x, y)$ when $x < y$ and $w]x, y[\in L$ (*Infix*).
- $w \models P_L(y)$ when $w[1, y[\in L$ (*Prefix*).
- $w \models S_L(x)$ when $w]x, n] \in L$ (*Suffix*).
- $w \models W_L$ when $w \in L$ (*Whole word*).

Generic membership theorem

Generic membership Theorem (Place, Z. '17, Place '15)

For any finite basis $\mathcal{C}$, levels $\frac{1}{2}, 1, \frac{3}{2}, \quad \frac{5}{2}$ have decidable membership.

Generic membership theorem

Generic membership Theorem (Place, Z. '17, Place '15)

For any finite basis $\mathcal{C}$, levels $\frac{1}{2}, 1, \frac{3}{2}, \quad \frac{5}{2}$ have decidable membership.

Remember: decidability known up to level $\frac{7}{2}$ for ST and BC hierarchies.

Generic membership theorem

Generic membership Theorem (Place, Z. '17, Place '15)

For any finite basis $\mathcal{C}$, levels $\frac{1}{2}, 1, \frac{3}{2}, \quad \frac{5}{2}$ have decidable membership.

Remember: decidability known up to level $\frac{7}{2}$ for ST and BC hierarchies.

The alphabet trick

Languages in ST $\left[\frac{3}{2}\right]$ (Pin and Straubing '85)

Languages of level ST $\left[\frac{3}{2}\right]$ are unions of languages of the form $B_0^* a_1 B_1^* \cdots a_n B_n^*$

Generic membership theorem

Generic membership Theorem (Place, Z. '17, Place '15)

For any finite basis $\mathcal{C}$, levels $\frac{1}{2}, 1, \frac{3}{2}, \quad \frac{5}{2}$ have decidable membership.

Remember: decidability known up to level $\frac{7}{2}$ for ST and BC hierarchies.

The alphabet trick

Languages in ST $\left[\frac{3}{2}\right]$ (Pin and Straubing '85)

Languages of level ST $\left[\frac{3}{2}\right]$ are unions of languages of the form $B_0^* a_1 B_1^* \cdots a_n B_n^*$

$$\text{ST}\left[\frac{3}{2}\right] = \text{level } \frac{1}{2} \text{ with basis AT} = \{B^* \mid B \subseteq A\}.$$

ST[q] is also level $(q-1)$ in another hierarchy with finite basis.

$$\text{ST}[q] = \text{AT}[q-1].$$

Generic membership theorem

Generic membership Theorem (Place, Z. '17, Place '15)

For any finite basis $\mathcal{C}$, levels $\frac{1}{2}, 1, \frac{3}{2}$, $\frac{5}{2}$ have decidable membership.

Remember: decidability known up to level $\frac{7}{2}$ for ST and BC hierarchies.

The alphabet trick

Languages in ST $\left[\frac{3}{2}\right]$ (Pin and Straubing '85)

Languages of level ST $\left[\frac{3}{2}\right]$ are unions of languages of the form $B_0^* a_1 B_1^* \cdots a_n B_n^*$

$$\text{ST}\left[\tfrac{3}{2}\right] = \text{level } \tfrac{1}{2} \text{ with basis AT} = \{B^* \mid B \subseteq A\}.$$

ST[q] is also level $(q-1)$ in another hierarchy with finite basis.

$$\text{ST}[q] = \text{AT}[q-1].$$

Corollary (by **Alphabet trick**)

In ST hierarchy, levels $\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}$, $\frac{7}{2}$ have decidable membership.

Generic membership theorem

Generic membership Theorem (Place, Z. '17, Place '15)

For any finite basis $\mathcal{C}$, levels $\frac{1}{2}, 1, \frac{3}{2}, \quad \frac{5}{2}$ have decidable membership.

Remember: decidability known up to level $\frac{7}{2}$ for ST and BC hierarchies.

The alphabet trick

Languages in ST $\left[\frac{3}{2}\right]$ (Pin and Straubing '85)

Languages of level ST $\left[\frac{3}{2}\right]$ are unions of languages of the form $B_0^* a_1 B_1^* \cdots a_n B_n^*$

$$\text{ST}\left[\tfrac{3}{2}\right] = \text{level } \tfrac{1}{2} \text{ with basis AT} = \{B^* \mid B \subseteq A\}.$$

ST[q] is also level $(q-1)$ in another hierarchy with finite basis.

$$\text{ST}[q] = \text{AT}[q-1].$$

Corollary (by **Alphabet trick**)

In ST and BC hierarchy, levels $\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, \quad \frac{7}{2}$ have decidable membership.

Recap

- Generic construction process for concatenation hierarchies.
- Generic logical correspondence.
- Generic strictness theorem.
- Generic membership theorem.

Recap

- Generic construction process for concatenation hierarchies.
- Generic logical correspondence.
- Generic strictness theorem.
- Generic membership theorem.

Recent results required solving **harder problems** than membership.

Beyond Membership: Separation Problems

Beyond membership: Separation

Recent results required solving **harder problems** than membership.

Motivation:

- Classe $\mathcal{C}$ with decidable membership.
- Class $\mathrm{Op}(\mathcal{C})$ built on top of $\mathcal{C}$ with undecidable membership.

Beyond membership: Separation

Recent results required solving **harder problems** than membership.

Motivation:

- Classe $\mathcal{C}$ with decidable membership.
- Class $\mathrm{Op}(\mathcal{C})$ built on top of $\mathcal{C}$ with undecidable membership.

Nice idea, Henckell and Rhodes '88

Prove more on $\mathcal{C}$ to recover membership decidability for $\mathrm{Op}(\mathcal{C})$.

Nice statement, Almeida '96

Almeida'96: a problem introduced by Henckell can be formulated as **separation**.

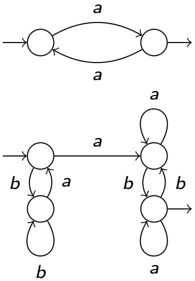
Beyond membership: Separation

Decide the following problem:

Take **2** regular languages L_1, L_2

L_1

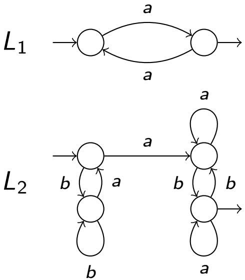
L_2



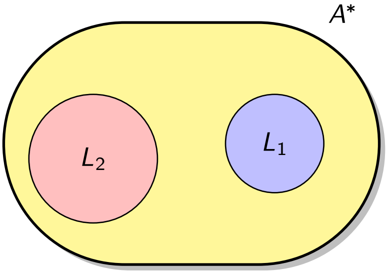
Beyond membership: Separation

Decide the following problem:

Take **2** regular languages L_1, L_2



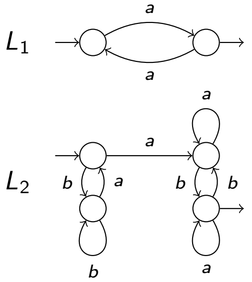
Can L_1 be separated from L_2 with a language from $\mathcal{C}$?



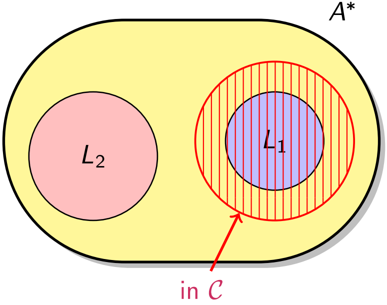
Beyond membership: Separation

Decide the following problem:

Take **2** regular languages L_1, L_2



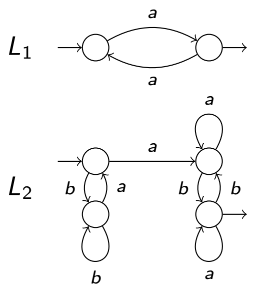
Can L_1 be separated from L_2 with a language from $\mathcal{C}$?



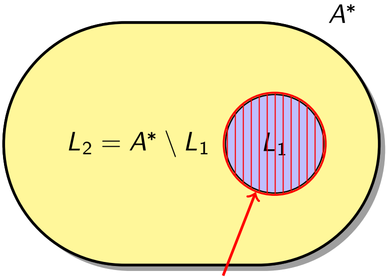
Beyond membership: Separation

Decide the following problem:

Take **2** regular languages L_1, L_2



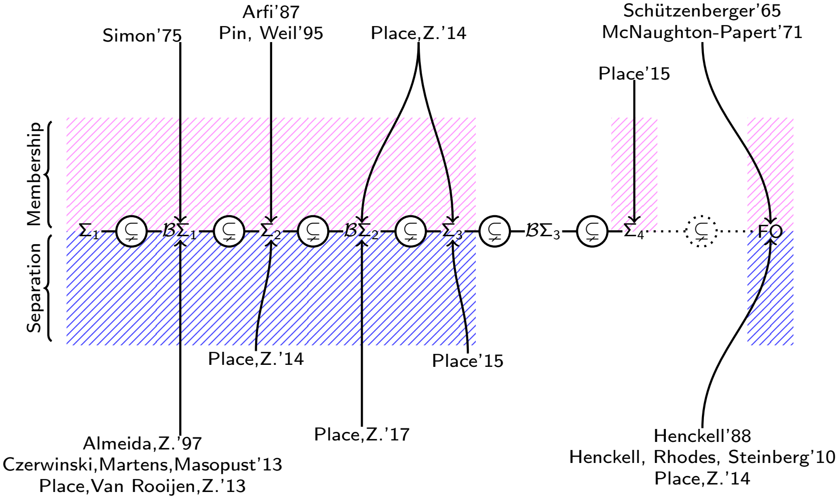
Can L_1 be separated from L_2 with a language from $\mathcal{C}$?



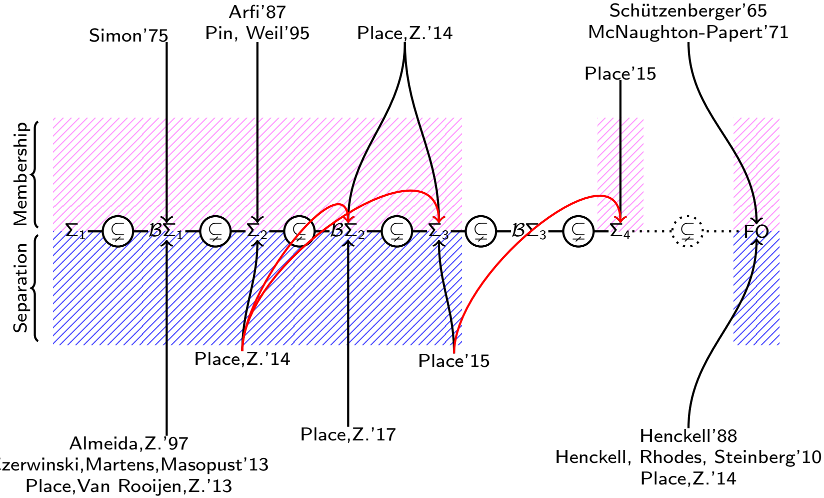
$\mathcal{C}$-separable from complement
$\Leftrightarrow$
in $\mathcal{C}$

Membership can be formally reduced to separation

Separation for classical hierarchies



Separation for classical hierarchies



Some membership algorithms come from separation algorithms for simpler levels

Separation results for generic hierarchies

All what we know for ST and BC follow from:

Generic Separation Theorem (Place, Z. '17, Place '15)

Levels $\frac{1}{2}, 1$ and $\frac{3}{2}$ have decidable **separation** for any hierarchy whose basis is finite.

Separation results for generic hierarchies

All what we know for ST and BC follow from:

Generic Separation Theorem (Place, Z. '17, Place '15)

Levels $\frac{1}{2}, 1$ and $\frac{3}{2}$ have decidable **separation** for any hierarchy whose basis is finite.

Corollary (by **Alphabet trick + Enrichment**)

Levels $\frac{1}{2}, 1$, $\frac{3}{2}$, 2 and $\frac{5}{2}$ have decidable **separation** in ST and BC hierarchies.

Separation results for generic hierarchies

All what we know for ST and BC follow from:

Generic Separation Theorem (Place, Z. '17, Place '15)

Levels $\frac{1}{2}, 1$ and $\frac{3}{2}$ have decidable **separation** for any hierarchy whose basis is finite.

Corollary (by **Alphabet trick** + **Enrichment**)

Levels $\frac{1}{2}, 1$, $\frac{3}{2}$, 2 and $\frac{5}{2}$ have decidable **separation** in ST and BC hierarchies.

Jump Theorem for quotienting lattices (Place, Z. '15)

For any basis, membership for level $n + \frac{1}{2}$ reduces to separation for level $n - \frac{1}{2}$.

Separation results for generic hierarchies

All what we know for ST and BC follow from:

Generic Separation Theorem (Place, Z. '17, Place '15)

Levels $\frac{1}{2}, 1$ and $\frac{3}{2}$ have decidable **separation** for any hierarchy whose basis is finite.

Corollary (by **Alphabet trick + Enrichment**)

Levels $\frac{1}{2}, 1, \frac{3}{2}$, 2 and $\frac{5}{2}$ have decidable **separation** in ST and BC hierarchies.

Jump Theorem for quotienting lattices (Place, Z. '15)

For any basis, membership for level $n + \frac{1}{2}$ reduces to separation for level $n - \frac{1}{2}$.

Enrichment Theorem (Place, Z. '15)

Separation for a level in the enriched hierarchy (ie, BC)
reduces to
Separation for the same level in the order hierarchy (ie, ST).

The Jump Theorem

Jump Theorem for quotienting lattices (Place, Z. '15)

For any basis, membership for level $n + \frac{1}{2}$ reduces to separation for level $n - \frac{1}{2}$.

The Jump Theorem

Jump Theorem for quotienting lattices (Place, Z. '15)

For any basis, membership for level $n + \frac{1}{2}$ reduces to separation for level $n - \frac{1}{2}$.

The Jump Theorem on Automata

A regular language **is in $\mathcal{C}\left[\mathbf{n} + \frac{1}{2}\right]$** iff its minimal automaton has **no pattern**:

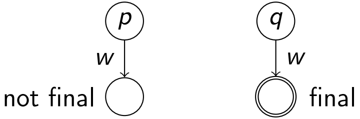
The Jump Theorem

Jump Theorem for quotienting lattices (Place, Z. '15)

For any basis, membership for level $n + \frac{1}{2}$ reduces to separation for level $n - \frac{1}{2}$.

The Jump Theorem on Automata

A regular language **is in** $\mathcal{C}\left[\mathbf{n} + \frac{\mathbf{1}}{\mathbf{2}}\right]$ iff its minimal automaton has **no pattern**:



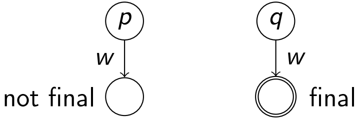
The Jump Theorem

Jump Theorem for quotienting lattices (Place, Z. '15)

For any basis, membership for level $n + \frac{1}{2}$ reduces to separation for level $n - \frac{1}{2}$.

The Jump Theorem on Automata

A regular language **is in** $\mathcal{C}\left[\mathbf{n} + \frac{\mathbf{1}}{\mathbf{2}}\right]$ iff its minimal automaton has **no pattern**:



where $L_{p,q}$ is **not** $\mathcal{C}\left[\mathbf{n} - \frac{\mathbf{1}}{\mathbf{2}}\right]$**-separable** from $L_{p,p} \cap L_{q,q}$

$$L_{p,q} = \{w \mid p \xrightarrow{w} q\}$$

Recap

Current knowledge is captured by few generic results:

1. **Separation theorem**
   $\mathcal{C}$ finite $\Rightarrow$ separation decidable for $Pol(\mathcal{C})$, $BPol(\mathcal{C})$ and $Pol(BPol(\mathcal{C}))$.
   In particular, cannot yet deal with 2 levels of complement.

2. **Jump theorem**
   $\mathcal{C}$-separation decidable $\Rightarrow$ $Pol(\mathcal{C})$-membership decidable.

3. **Enrichment theorem** (see also Varun's talk).

Unambiguous Polynomial Closure

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\mathsf{AT} = \{B^* \mid B \subseteq A\}.$$

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\mathsf{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\mathsf{FO}^2(<)$, $\Delta_2(<)$,

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\mathrm{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\mathrm{FO}^2(<)$, $\Delta_2(<)$, UTL, TL(rankers),

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\mathrm{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\mathrm{FO}^2(<)$, $\Delta_2(<)$, UTL, TL(rankers), 2way turtle DFAs,

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\text{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\text{FO}^2(<)$, $\Delta_2(<)$, UTL, TL(rankers), 2way turtle DFAs, **DA**...

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\mathrm{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\mathrm{FO}^2(<)$, $\Delta_2(<)$, UTL, TL(rankers), 2way turtle DFAs, **DA**...

- **Pin, 1980**: **Generic** investigation of unambiguous polynomial closure

Nous dirons avec Schützenberger [7] que le produit AB est déterministe si A et B sont vides et vérifient l'une des conditions suivantes (a) A est préfixe ; (b) AB est préfixe et B est une partie de l'alphabet X

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\mathsf{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\mathsf{FO}^2(<)$, $\Delta_2(<)$, UTL, TL(rankers), 2way turtle DFAs, **DA**...

- **Pin, 1980**: **Generic** investigation of unambiguous polynomial closure

Nous dirons avec Schützenberger [7] que le produit AB est déterministe si A et B sont vides et vérifient l'une des conditions suivantes (a) A est préfixe ; (b) AB est préfixe et B est une partie de l'alphabet X

History and motivations

- **Schützenberger, 1976**: UL = **Unambiguous** Polynomial closure(AT).

$$\text{AT} = \{B^* \mid B \subseteq A\}.$$

- **Robust**: $\text{FO}^2(<)$, $\Delta_2(<)$, UTL, TL(rankers), 2way turtle DFAs, **DA**...

- **Pin, 1980**: **Generic** investigation of unambiguous polynomial closure

Nous dirons avec Schützenberger [7] que le produit AB est déterministe si
A et B sont vides et vérifient l'une des conditions suivantes
(a) A est préfixe ; (b) AB est préfixe et B est une partie de l'alphabet X

- **Margolis Pin Thérien, 1988** *UPol*($\mathcal{C}$) membership reduces to $\mathcal{C}$-membership
  Algebraic proofs (relational morphisms, categories, bilateral kernels, etc.)

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$. Generalizes to n languages (and so to).

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$.
  Generalizes to n languages (and so to).
- A^*aA^*?

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$.
  Generalizes to n languages (and so to).
- A^*aA^*? $(A \setminus \{a\})^*aA^*$?

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$.
  Generalizes to n languages (and so to).
- A^*aA^*? $(A \setminus \{a\})^*aA^*$?

- $UPol(\mathcal{C})$ = closure of $\mathcal{C}$ under
  - **unambiguous** concatenation,
  - **disjoint** union.

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$. Generalizes to n languages (and so to).
- A^*aA^*? $(A \setminus \{a\})^*aA^*$?

- $UPol(\mathcal{C})$ = closure of $\mathcal{C}$ under
  - **unambiguous** concatenation,
  - **disjoint** union.

- **Examples**
  - $UPol(\{\emptyset, A^*\})$?

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$. Generalizes to n languages (and so to).
- A^*aA^*? $\quad (A \setminus \{a\})^*aA^*$?

- $UPol(\mathcal{C})$ = closure of $\mathcal{C}$ under
  - **unambiguous** concatenation,
  - **disjoint** union.

- **Examples**
  - $UPol(\{\emptyset, A^*\})$?
  - $UPol(\mathrm{AT})$ = UL from Schützenberger.

Unambiguous polynomial closure

- KL is **unambiguous** if each $w \in KL$ has a **single** factorization in $K \times L$.
  Generalizes to n languages (and so to).
- A^*aA^*? $(A \setminus \{a\})^*aA^*$?

- $UPol(\mathcal{C})$ = closure of $\mathcal{C}$ under
  - **unambiguous** concatenation,
  - **disjoint** union.

- **Examples**
  - $UPol(\{\emptyset, A^*\})$?
  - $UPol(\mathrm{AT})$ = UL from Schützenberger.

- Not even clear whether $UPol(\mathcal{C})$ is closed under union.

Alternating polynomial closure

- KaL is **left deterministic** if $KaA^* \cap K = \emptyset$.
  Note. Does not depend on L.

Alternating polynomial closure

- KaL is **left deterministic** if $KaA^* \cap K = \emptyset$.
  Note. Does not depend on L.
- Right deterministic: symmetric

Alternating polynomial closure

- KaL is **left deterministic** if $KaA^* \cap K = \emptyset$.
  Note. Does not depend on L.
- Right deterministic: symmetric
- Unambiguous concatenation which is not left neither right deterministic?

Alternating polynomial closure

- KaL is **left deterministic** if $KaA^* \cap K = \emptyset$.
  Note. Does not depend on L.
- Right deterministic: symmetric
- Unambiguous concatenation which is not left neither right deterministic?
  $(A \setminus \{a\})^* aA^* b(A \setminus \{b\})^*$.

Alternating polynomial closure

- KaL is **left deterministic** if $KaA^* \cap K = \emptyset$.
  Note. Does not depend on L.

- Right deterministic: symmetric

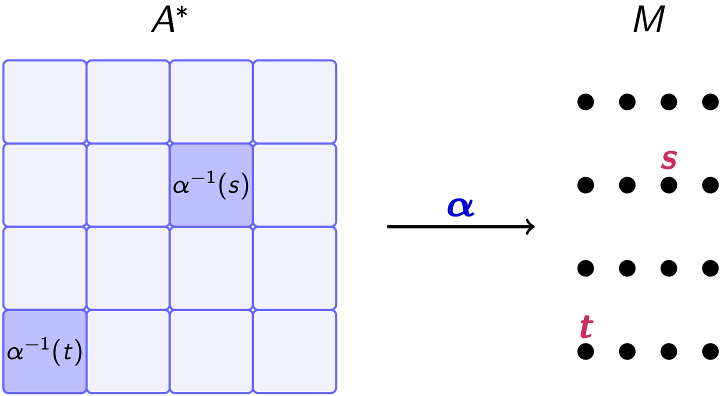
- Unambiguous concatenation which is not left neither right deterministic?
  $(A \setminus \{a\})^* a A^* b (A \setminus \{b\})^*$.

- $ADet(\mathcal{C})$ = closure of $\mathcal{C}$ under left+right deterministic marked concatenation

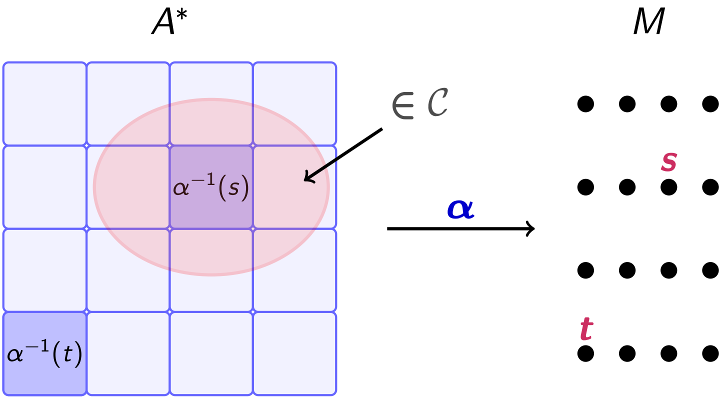
Easy remark

$$ADet(\mathcal{C}) \subseteq UPol(\mathcal{C})$$

Quasi separation wrt. a surjective morphism



Quasi separation wrt. a surjective morphism

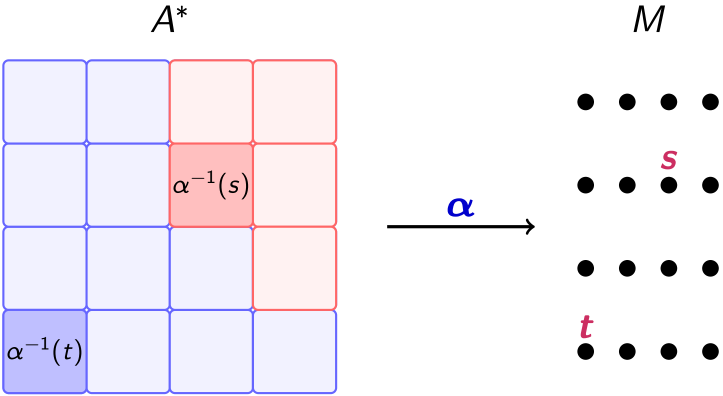


$\alpha^{-1}(s), \alpha^{-1}(t)$ are $\mathcal{C}$-separable

$\Leftrightarrow$

(s, t) is **not** a $\mathcal{C}$-pair

Quasi separation wrt. a surjective morphism



$\alpha^{-1}(s), \alpha^{-1}(t)$ are $\mathcal{C}$-separable by some $\alpha^{-1}(P)$

$\Leftrightarrow$

(s, t) is **not** a **quasi** $\mathcal{C}$-pair

Quasi $\mathcal{C}$-pairs

(s, t) quasi $\mathcal{C}$-pair = $\alpha^{-1}(s), \alpha^{-1}(t)$ **not** $\mathcal{C}$-separable by language recognized by α

- $\mathcal{C}$-membership decidable $\Longrightarrow$ quasi $\mathcal{C}$-pairs computable.

Quasi $\mathcal{C}$-pairs

(s, t) quasi $\mathcal{C}$-pair = $\alpha^{-1}(s), \alpha^{-1}(t)$ **not** $\mathcal{C}$-separable by language recognized by α

- $\mathcal{C}$-membership decidable $\Longrightarrow$ quasi $\mathcal{C}$-pairs computable.
- $\mathcal{C}$-pair $\Rightarrow$ quasi $\mathcal{C}$-pair.

Quasi $\mathcal{C}$-pairs

(s, t) quasi $\mathcal{C}$-pair $= \alpha^{-1}(s), \alpha^{-1}(t)$ **not** $\mathcal{C}$-separable by language recognized by α

- $\mathcal{C}$-membership decidable $\Longrightarrow$ quasi $\mathcal{C}$-pairs computable.
- $\mathcal{C}$-pair $\Rightarrow$ quasi $\mathcal{C}$-pair.
- The "$\mathcal{C}$-pair" relation is **not** transitive.

  Ex. for $\mathcal{C} =$AT: $L_1 = \{ab\}$, $L_2 = \{ba, cd\}$, $L_3 = \{dc\}$.

Quasi $\mathcal{C}$-pairs

(s, t) quasi $\mathcal{C}$-pair = $\alpha^{-1}(s), \alpha^{-1}(t)$ **not** $\mathcal{C}$-separable by language recognized by α

- $\mathcal{C}$-membership decidable $\implies$ quasi $\mathcal{C}$-pairs computable.
- $\mathcal{C}$-pair $\Rightarrow$ quasi $\mathcal{C}$-pair.
- The "$\mathcal{C}$-pair" relation is **not** transitive.
  Ex. for $\mathcal{C}$ =AT: $L_1 = \{ab\}$, $L_2 = \{ba, cd\}$, $L_3 = \{dc\}$.
- The "quasi $\mathcal{C}$-pair" relation is a **congruence**.

Quasi $\mathcal{C}$-pairs

(s, t) quasi $\mathcal{C}$-pair = $\alpha^{-1}(s), \alpha^{-1}(t)$ **not** $\mathcal{C}$-separable by language recognized by α

- $\mathcal{C}$-membership decidable $\Longrightarrow$ quasi $\mathcal{C}$-pairs computable.
- $\mathcal{C}$-pair $\Rightarrow$ quasi $\mathcal{C}$-pair.
- The "$\mathcal{C}$-pair" relation is **not** transitive.

  Ex. for $\mathcal{C}$ =AT: $L_1 = \{ab\}$, $L_2 = \{ba, cd\}$, $L_3 = \{dc\}$.
- The "quasi $\mathcal{C}$-pair" relation is a **congruence**.
- Quasi $\mathcal{C}$-pair relation = transitive closure of $\mathcal{C}$-pair relation.

Membership results

Generic Separation Theorem (Place, Z. '18)

Let $\mathcal{C}$ be a quotienting Boolean algebra of regular languages.
Let $L \subseteq A^*$ be regular and $\alpha : A^* \to M$ be its syntactic morphism. TFAE:

1. $L \in UPol(\mathcal{C})$.
2. $L \in ADet(\mathcal{C})$.
3. $L \in Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$.
4. For all $\mathcal{C}$-pairs $(s, t) \in M^2$, we have $s^{\omega+1} = s^\omega t s^\omega$.
5. For all quasi $\mathcal{C}$-pairs $(s, t) \in M^2$, we have $s^{\omega+1} = s^\omega t s^\omega$.

Membership results

Generic Separation Theorem (Place, Z. '18)

Let $\mathcal{C}$ be a quotienting Boolean algebra of regular languages.
Let $L \subseteq A^*$ be regular and $\alpha : A^* \to M$ be its syntactic morphism. TFAE:

1. $L \in UPol(\mathcal{C})$.
2. $L \in ADet(\mathcal{C})$.
3. $L \in Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$.
4. For all $\mathcal{C}$-pairs $(s, t) \in M^2$, we have $s^{\omega+1} = s^{\omega} t s^{\omega}$.
5. For all quasi $\mathcal{C}$-pairs $(s, t) \in M^2$, we have $s^{\omega+1} = s^{\omega} t s^{\omega}$.

$$2 \Longrightarrow 1 \Longrightarrow 4 \Longleftrightarrow 3 \Longrightarrow 5 \Longrightarrow 2$$

Membership results

Generic Separation Theorem (Place, Z. '18)

Let $\mathcal{C}$ be a quotienting Boolean algebra of regular languages.
Let $L \subseteq A^*$ be regular and $\alpha : A^* \to M$ be its syntactic morphism. TFAE:

1. $L \in UPol(\mathcal{C})$.
2. $L \in ADet(\mathcal{C})$.
3. $L \in Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$.
4. For all $\mathcal{C}$-pairs $(s, t) \in M^2$, we have $s^{\omega+1} = s^{\omega} t s^{\omega}$.
5. For all quasi $\mathcal{C}$-pairs $(s, t) \in M^2$, we have $s^{\omega+1} = s^{\omega} t s^{\omega}$.

$$2 \implies 1 \implies 4 \iff 3 \implies 5 \implies 2$$

Corollary

1. *If $\mathcal{C}$ is a quotienting Boolean algebra, so is $UPol(\mathcal{C})$.*
2. *$UPol(\mathcal{C})$ membership reduces to $\mathcal{C}$ membership.*

Separation result

Generic Separation Theorem (Place, Z. '18)

If $\mathcal{C}$ is a **finite** quotienting Boolean algebra, separation for $UPol(\mathcal{C})$ is decidable.

Separation result

Generic Separation Theorem (Place, Z. '18)

If $\mathcal{C}$ is a **finite** quotienting Boolean algebra, separation for $UPol(\mathcal{C})$ is decidable.

Notes

1. This result is **stronger** than the one for membership!
2. Solved by considering a more general problem: covering.

Summary of results

- $\mathcal{C}$ finite+...: separation OK for $Pol(\mathcal{C})$, $BPol(\mathcal{C})$, $Pol(BPol(\mathcal{C}))$, $UPol(\mathcal{C})$.
- Via transfer results, imply all known algorithms.

Summary of results

- $\mathcal{C}$ finite+...: separation OK for $Pol(\mathcal{C})$, $BPol(\mathcal{C})$, $Pol(BPol(\mathcal{C}))$, $UPol(\mathcal{C})$.
- Via transfer results, imply all known algorithms.

Thanks!