Separation-like problems for regular languages

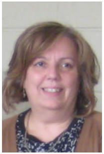
Marc Zeitoun

Joint work with Thomas Place

LaBRI, Univ. Bordeaux

International Conference on Semigroups and Automata

June 20, 2016



Happy 00111100-th birthday!

Motivation

Structures

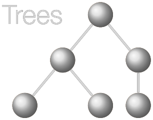
Descriptive Formalism

Express Properties

Words

ababcbaa

Trees



First-Order Logic (**FO**)

Fragments Σ_i, $\mathcal{B}\Sigma_i$

Piecewise Testable ($\mathcal{B}\Sigma_1$)

2-Variables **FO** (**FO**$_2$)

Locally Threshold Testable (**LTT**)

For this talk

Main problem: **Decide Membership**

Message: solving it requires focusing on **other problems**

Key Example: First-order Logic on Words

A way to define languages: first-order logic, with predicates '$<$' and $a(x)$.

$$\begin{array}{cccccccccc} a & b & b & b & c & a & a & a & c & a \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}$$

- A word is a sequence of labeled positions.
- Positions can be quantified: $\exists x \varphi$.
- Unary predicates $a(x), b(x), c(x)$ testing the label of position x.
- One binary predicate: the linear-order $x < y$.

Example: every a comes after some b

$$\forall x \; a(x) \Rightarrow \exists y \; (b(y) \wedge (y < x))$$

Quantifier alternation

Level i: Σ_i

For all i, a Σ_i formula is

$$\underbrace{\exists x_1, \ldots, x_{n_1} \forall y_1, \ldots, y_{n_2} \cdots\cdots}_{i \text{ blocks (starting with } \exists)} \quad \underbrace{\varphi(\bar{x}, \bar{y}, \ldots)}_{\text{quantifier-free}}$$

Σ_i is not closed under complement $\Rightarrow$ we get two other classes:

Level i: Π_i

Negation of a Σ_i formula:

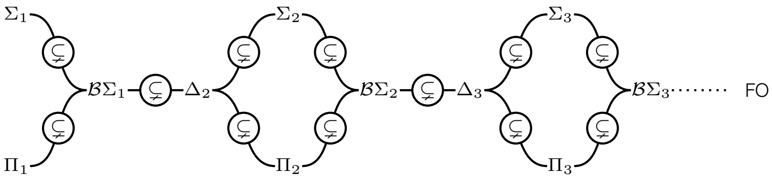
$$\underbrace{\forall x_1, \ldots, x_{n_1} \exists y_1, \ldots, y_{n_2} \cdots}_{i \text{ blocks (starting with } \forall)} \quad \varphi$$

Level i: $\mathcal{B}\Sigma_i$

Boolean combinations of Σ_i (and Π_i) formulas.

Recall goal: **Decide Membership**

FO Quantifier alternation hierarchy



- Corresponds to Straubing-Thérien hierarchy.
- Adding $+1$ to all fragments: Brzozowski-Cohen hierarchy (= dot-depth).

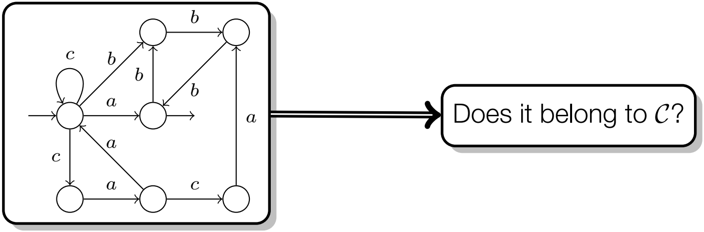
3 Major Milestones

- Syntactic approach: Schützenberger, Simon, Myhill, Nerode,…
- Classes not complement-closed: Ordered Monoids. Pin, Weil.
- **Separation:** Henckell, Rhodes, Steinberg, Auinger,
  Almeida, J.C. Costa, Pin, Reutenauer,…

Milestone 1: Syntactic Approach

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

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



Schützenberger '65, McNaughton and Papert '71

For L a regular language, the following are equivalent:

- L is **FO**-definable.
- The syntactic monoid of L is aperiodic, i.e., it satisfies $u^{\omega+1} = u^{\omega}$.

Milestone 2: Classes not complement-closed

- A language and its complement have the same syntactic monoid.
  $\Rightarrow$ cannot characterize classes not closed under complement (Σ_n).

Pin's Solution: recognition by ordered monoids.

- Myhill-Nerode: $L \in \mathcal{C}$ iff so are all languages recognized by $M(L)$.
- Pin's idea: relax this "all languages" condition.
  Accepting sets F constrained to be upwards-closed.

Pin, Weil '95

For L a regular language, the following are equivalent:

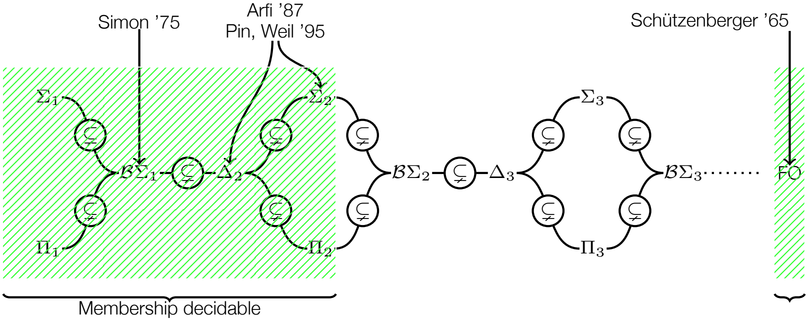
- L is Σ_2-definable.
- The ordered syntactic monoid of L satisfies

$$s^\omega \leqslant s^\omega t s^\omega$$

  when $\text{alph}(t) \subseteq \text{alph}(s)$.

FO Quantifier alternation hierarchy

State of the art using syntactic approach + ordered



Milestone 3: Beyond Membership

- Next interesting classes: $\mathcal{B}\Sigma_2$ and Σ_3. What is the difficulty?

- Approach by ordered monoids $\hookrightarrow$ build inductively a Σ_3- formula.
- Σ_3 sentences are layered: a Σ_3-layer, a Π_2 layer, a Σ_1 layer.

$$\exists^* x_i \quad \forall^* y_i \exists^* z_i \varphi$$

- Induction should decompose the input language and at some point, build Π_2 formulas.
- **But** there is no reason for these sublanguages to be Π_2-definable.

$\Rightarrow$ One must investigate properties that are
more demanding than membership decidability

There already exist such properties in the literature.

Milestone 3: Beyond Membership

- Other fundamental hierarchy of regular languages: complexity hierarchy.
- Counts "alternating cascade products" btw. aperiodic sgps and groups.

- Idea (Henckell, Rhodes): **strengthen** "having decidable membership".
- Problem called "computation of pointlike sets".

- Connected to profinite theory and investigated by Henckell, Rhodes, Steinberg, Auinger, Almeida, Pin, Reutenauer, J.C. Costa and others.

Rest of this talk: the original view and a new view of pointlike sets.

Pointlike Sets: definition

Fix V a pseudovariety of finite semigroups.

- Relational morphism $\mu : S \to T \stackrel{\text{def}}{=}$ subsemigroup of $S \times T$ whose projection on S is onto.
- $X \subseteq S$ is μ-pointlike if $\bigcap_{x \in X} \mu(x) \neq \emptyset$ where $\mu(x) = \{t \mid (x, t \in \mu)\}$.
- V-pointlike $\stackrel{\text{def}}{=}$ μ-pointlike for all relational morphisms $\mu : S \to T \in$ V.
- V-pointlike set problem:
  - **Input** Finite semigroup S and $X \subseteq S$.
  - **Question** Is X V-pointlike?

Fact

The V-membership problem reduces to the V-pointlike set problem (even for $|X| = 2$).

Beyond Membership: Pointlike Sets

Henckell '88

One can decide whether a subset of a finite semigroup is aperiodic-pointlike.

- Much harder than Schützenberger's result.
- Shorter proof of more general result by Henckell, Rhodes, Steinberg 2010.
- Membership can be formulated both on languages and on semigroups.

Is it the same for the pointlike set problem?

The Separation Problem

Almeida '96

Let V = semigroup pseudovariety, $\mathcal{V}$ = corresponding variety of languages.

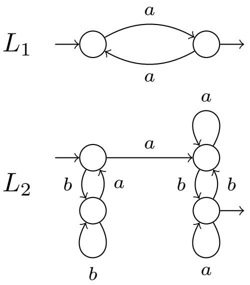
- The V-pointlike set problem for sets of size 2 is equivalent to the $\mathcal{V}$-**separation problem**.
- Similar interpretation for pointlike sets of arbitrary size.

Several approaches: (profinite) semigroup theory / formal language theory.

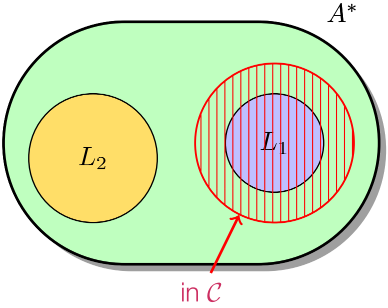
Beyond membership: Separation

Decide the following problem:

Take **2** regular languages L_1, L_2



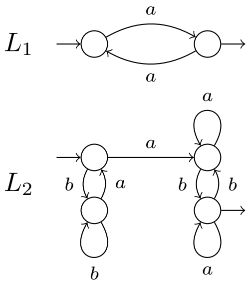
Can L_1 be separated from L_2 with a language from $\mathcal{C}$?



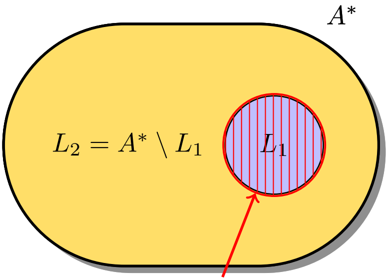
Beyond membership: Separation

Membership can be formally reduced to separation

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 $\mathcal{C}$

Separation and Pointlike Sets

Henckell '88, Henckell Rhodes Steinberg '10

The aperiodic pointlike sets of a finite monoid are computable.

Corollary

The separation problem by first-order languages is decidable.

Payoff of Separation? Transfer Results!

Place,Z.14 — Σ_{n+1}-membership reduces to Σ_n-separation

Let L be a regular language and $i \geqslant 2$. Then TFAE:

1. L is definable in Σ_{n+1}.
2. $\forall s, t \in M_L$: $\alpha_L^{-1}(s)$ not Π_n-separable from $\alpha_L^{-1}(t) \implies s^\omega \leqslant s^\omega t s^\omega$

- Note: we use here an asymetric version of separation.

Place,Z.14 — $\mathcal{B}\Sigma_n$-separation reduces to Σ_n-generalized separation

Let L_1, L_2 be languages and $\mathcal{C}$ a class closed under $\cap$ and $\cup$. Then TFAE:

1. No sequence $(L_1, L_2, L_1, L_2, \ldots)$ is $\mathcal{C}$-separable.
2. L_1, L_2 is not $\mathcal{BC}$-separable.

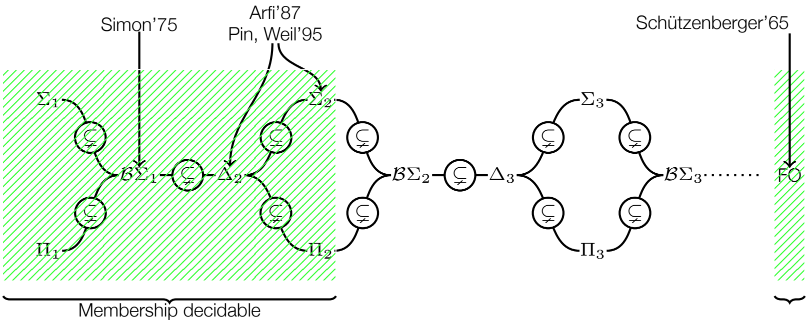
- Leads to a decision procedure $\mathcal{B}\Sigma_2$.

Steinberg '01, Place, Z.15 — Enriching the fragment

Separation transfers when enriched formalism, adding predicate $+1$.

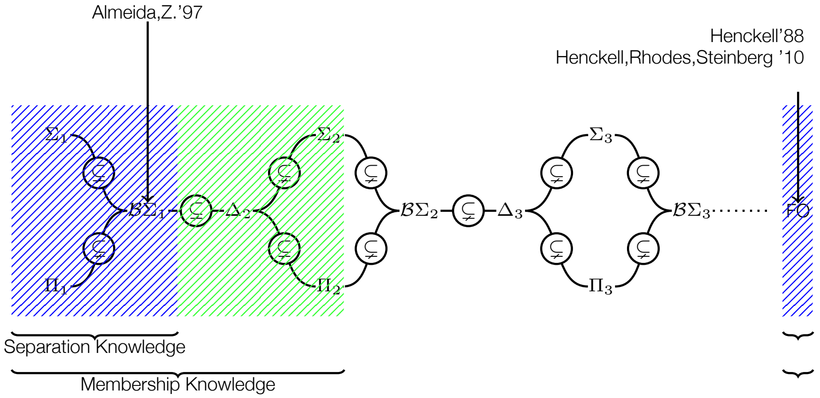
FO Quantifier alternation hierarchy

State of the art in 2013

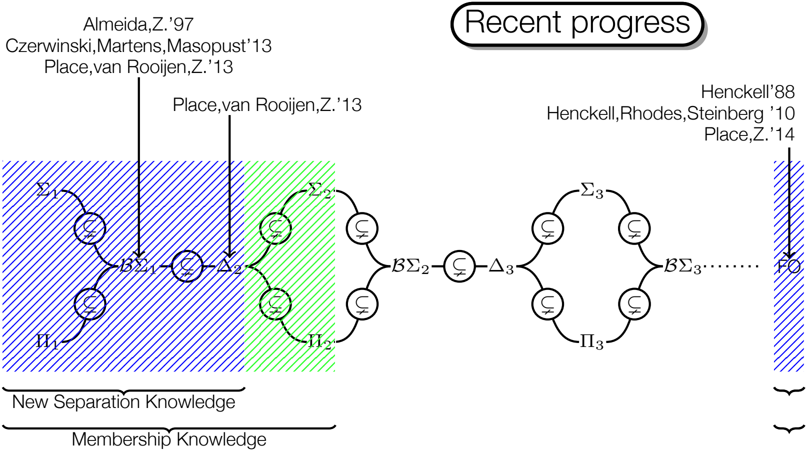


FO Quantifier alternation hierarchy

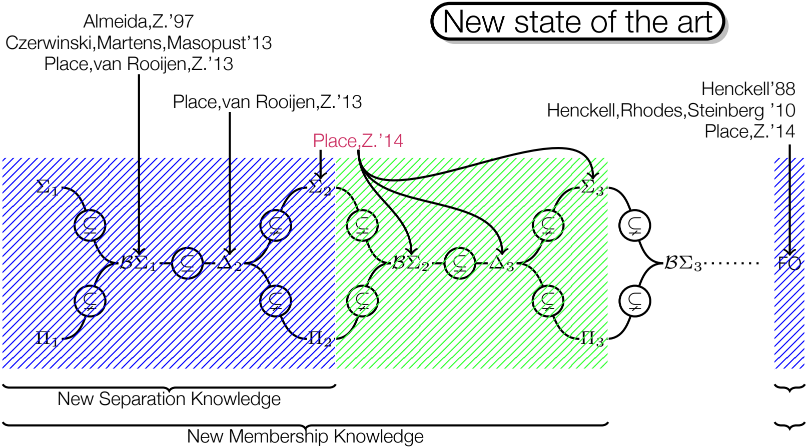
State of the art in 2013



FO Quantifier alternation hierarchy

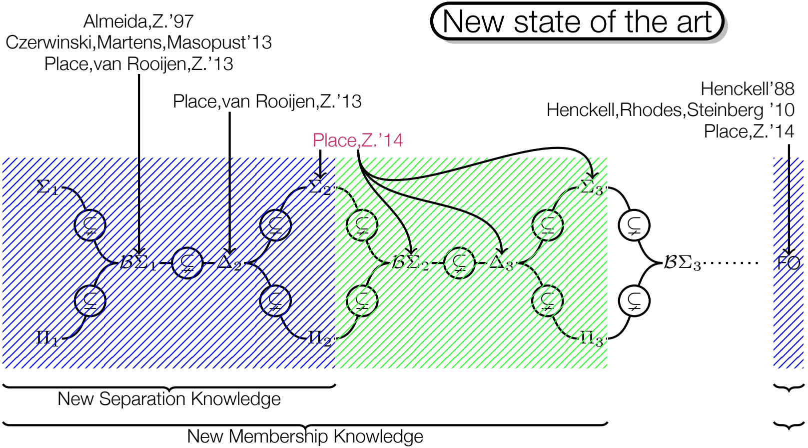


FO Quantifier alternation hierarchy



Analyzing Σ_2-separation algorithm yields membership for $\mathcal{B}\Sigma_2, \Delta_3, \Sigma_3$ and Π_3.

FO Quantifier alternation hierarchy



Place '15 Separation for Σ_3 (hard) $\Longrightarrow$ Membership for $\Delta_4, \Sigma_4, \Pi_4$.

Almeida,Bartonova,Klíma,Kunc '15 Δ_n-membership $\leqslant$ Σ_{n-1}-membership $\Longrightarrow$ Membership for Δ_5.

Membership open for $\mathcal{B}\Sigma_3$, Separation open for Δ_3.

The Covering Problem

- Generalizes separation.
- Corresponds to pointlike sets for (pseudo)varieties.
- But only requires mild hypotheses on the class $\mathcal{C}$ of languages.

- This talk: $\mathcal{C}$ Boolean algebra closed under $L \mapsto a^{-1}L$ and $L \mapsto La^{-1}$.

- Closure under inverse morphisms not required.
- Can be generalized to lattices.

The Covering Problem: Definition

- **L** $= \{L_1, \ldots, L_n\}$ = set of languages.
- A cover of **L** is a finite set of languages **K** $= \{K_1, \ldots, K_m\}$ st

$$L_1 \cup \cdots \cup L_n \subseteq K_1 \cup \cdots \cup K_m.$$

- **Note**: If K separates L_1 from L_2, then $\{K, A^* \setminus K\}$ is a cover of $\{L_1, L_2\}$.

Quality of a cover

- $\{L_1, L_2\}$ is always a cover of $\{L_1, L_2\}$.
- A^* is always a cover of $\{L_1, L_2\}$.

- Goal: Measure how good a cover is at "separating" an input set $\mathbf{L}$.

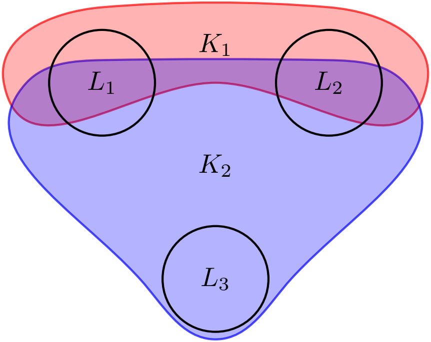
- **Hitting set** of a language K on $\mathbf{L}$:

$$\langle \mathbf{L} | K \rangle = \{ L \in \mathbf{L} \mid L \cap K \neq \emptyset \}$$

- **Imprint** of $\mathbf{K}$ on $\mathbf{L}$ $\overset{\text{def}}{=}$ set of all filterings $\langle \mathbf{L} | K \rangle$ for $K \in \mathbf{K}$.

$$\mathcal{I}[\mathbf{L}](\mathbf{K}) \quad = \quad \downarrow \{ \langle \mathbf{L} | K \rangle \mid K \in \mathbf{K} \} \subseteq 2^{\mathbf{L}}$$

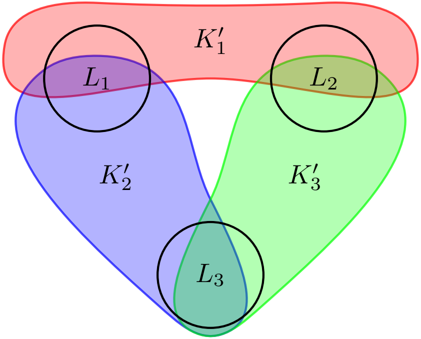
Covers: Example 1



Cover $\mathbf{K} = \{K_1, K_2\}$

$$\mathcal{I}[\mathbf{L}](\mathbf{K}) = \left\{ \begin{array}{l} \{L_1, L_2, L_3\}, \\ \{L_1, L_2\}, \{L_1, L_3\}, \{L_2, L_3\}, \\ \{L_1\}, \{L_2\}, \{L_3\}, \emptyset \end{array} \right\}$$

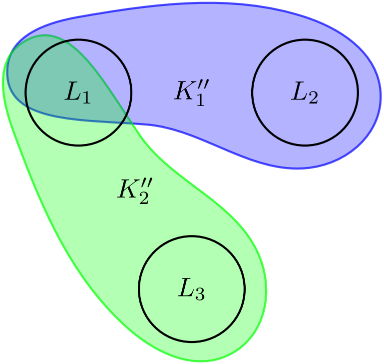
Covers: Example 2 (better than example 1)



Cover $\mathbf{K}' = \{K_1', K_2', K_3'\}$

$$\mathcal{I}[\mathbf{L}](\mathbf{K}') = \left\{ \begin{array}{l} \{L_1, L_2\}, \{L_1, L_3\}, \{L_2, L_3\}, \\ \{L_1\}, \{L_2\}, \{L_3\}, \emptyset \end{array} \right\}$$

Covers: Example 3 (even better than example 2)



Cover $\mathbf{K}'' = \{K_1'', K_2''\}$

$$\mathcal{I}[\mathbf{L}](\mathbf{K}'') = \left\{ \begin{array}{l} \{L_1, L_2\}, \{L_1, L_3\}, \\ \{L_1\}, \{L_2\}, \{L_3\}, \emptyset \end{array} \right\}$$

Connection with Separation

Easy fact: Imprints vs. separation

Let **L** be a finite set of languages. Let **K** cover **L**. For all $L_1, L_2 \in$ **L**:

$\{L_1, L_2\} \notin \mathcal{I}[\mathbf{L}](\mathbf{K}) \implies L_1$ is separated from L_2 by a union of languages in **K**.

- **Note.** The converse does not hold: take **K** $= \{L_1, L_2, L_1 \cup L_2\}$.
- Covers, like pointlikes, capture more information than separation.
- Covers with **smaller imprints** are better at separating **L**.

Optimal $\mathcal{C}$-covers

- A $\mathcal{C}$-cover is a cover whose elements belong to $\mathcal{C}$.
- Since $\mathcal{C}$ is a Boolean algebra, $\{A^*\}$ is a $\mathcal{C}$-cover of $\{L_1, \dots, L_n\}$…
- …**but** the cover $\{L_1, \dots, L_n\}$ of $\{L_1, \dots, L_n\}$ may not be a $\mathcal{C}$-cover.
- A $\mathcal{C}$-cover **K** is optimal if

$$\mathcal{I}[\mathbf{L}](\mathbf{K}) \subseteq \mathcal{I}[\mathbf{L}](\mathbf{H}) \quad \text{for any } \mathcal{C}\text{-cover } \mathbf{H} \text{ of } \mathbf{L}$$

Example

- $\mathcal{C}$ = Boolean algebra generated by languages A^*aA^* for $a \in A$.
- What is an optimal $\mathcal{C}$-cover of $\mathbf{L} = \{(ab)^+, (ba)^+, (ac)^+\}$?

Existence Lemma

As soon as $\mathcal{C}$ is closed under intersection, there exists an optimal cover.

- Trivial, but non-constructive proof.

The $\mathcal{C}$-Covering Problem

Optimal imprint: $\mathcal{I}_{\mathcal{C}}[\mathbf{L}] \stackrel{\text{def}}{=} \mathcal{I}[\mathbf{L}](\mathbf{K})$ for any optimal $\mathcal{C}$-cover $\mathbf{K}$ of $\mathbf{L}$.

Definition of the $\mathcal{C}$-covering problem

INPUT:	A finite set $\mathbf{L}$ of names of regular languages.
QUESTION:	Compute $\mathcal{I}_{\mathcal{C}}[\mathbf{L}]$.

- Bonus question: compute an actual $\mathcal{C}$-cover of $\mathbf{L}$.

$\mathcal{C}$-cover vs. $\mathcal{C}$-separation

Optimal imprint: $\mathcal{I}_{\mathcal{C}}[\mathbf{L}] \stackrel{\text{def}}{=} \mathcal{I}[\mathbf{L}](\mathbf{K})$ for any optimal $\mathcal{C}$-cover **K** of **L**.

Proposition (Place, Z. '16)

Let $\mathcal{C}$ be a Boolean algebra and **L** be a finite set of languages.
Given $L_1, L_2 \in \mathbf{L}$, **TFAE**:

1. L_1 and L_2 are $\mathcal{C}$-separable.
2. $\{L_1, L_2\} \notin \mathcal{I}_{\mathcal{C}}[\mathbf{L}]$.
3. For any optimal $\mathcal{C}$-cover **K** of **L**, L_1 and L_2 are separable by a union of languages in **K**.

Computing Optimal Imprints

Optimal imprint: $\mathcal{I}_{\mathcal{C}}[\mathbf{L}] \stackrel{\text{def}}{=} \mathcal{I}[\mathbf{L}](\mathbf{K})$ for any optimal $\mathcal{C}$-cover $\mathbf{K}$ of $\mathbf{L}$.

- The minimal automaton is a **canonical** object associated to a language.
  - Useful for membership,
  - Useless for covering or separation.
- **Canonical** object associated to $\mathcal{C}$ and $\mathbf{L}$: optimal imprint $\mathcal{I}_{\mathcal{C}}[\mathbf{L}]$.
- When $\mathcal{C}$ is a variety of languages and languages of $\mathbf{L}$ are disjoint:

The optimal imprint is exactly the set of pointlike sets.

Decomposition-closed Inputs

- Assume $\mathbf{L}$ equipped with a partial multiplication $\odot$ w/ mild properties.
- Hold when $\mathbf{L}$ consists of languages of the form $\alpha^{-1}(F)$ for $\alpha : A^* \to S$.
- Can be assumed for any input via a reduction.

Proposition (Place, Z. '16): $\mathcal{I}_{\mathcal{C}}[\mathbf{L}]$ is a semigroup

Under these conditions,

- $2^{\mathbf{L}}$ is a semigroup for the usual powerset multiplication inherited from $\odot$.
- If $\mathcal{C}$ closed under $L \mapsto a^{-1}L$ and $L \mapsto La^{-1}$, then $\mathcal{I}_{\mathcal{C}}[\mathbf{L}]$ is a subsetmigroup of $2^{\mathbf{L}}$

For all $\mathbf{L}_1$ and $\mathbf{L}_2$ in $\mathcal{I}_{\mathcal{C}}[\mathbf{L}]$, $\mathbf{L}_1 \odot \mathbf{L}_2 \in \mathcal{I}_{\mathcal{C}}[\mathbf{L}]$.

Computing optimal imprints

$\mathcal{I}_{\mathcal{C}}[\mathbf{L}]$ being a semigroup validates the following algorithm pattern:

Generic algorithm (Place, Z. '16)

$\text{Sat}_{\mathcal{C}}(\mathbf{L}) \stackrel{\text{def}}{=}$ smallest subset of $2^{\mathbf{L}}$ containing $\mathcal{I}_{triv}[\mathbf{L}]$ and is closed under:

1. Downset.
2. Product.
3. $\cdots$ (additional operation(s) specific to $\mathcal{C}$)

Recover the separation results in a **constructive** way.

Conclusion

- Language-theoretic view of pointlike sets.
- Definition and link with separation for quotienting Boolean algebras.
- Extends well to quotienting lattices.
- Can be parametrized by restricting the “hitting set” definition.
- Constructive separators when separation known decidable.
- Backbone for computation algorithms.

Further Work

- Adapt covering to go up in the quantifier alternation hierarchy.
- Interpret the results back in terms of (pro)finite semigroups.
- In particular, use the work of Grigorieff, Gehrke, Pin on lattices.