

Subregular Expressions and Their Expressive Power

Martin Kutrib & Matthias Wendlandt

Institut für Informatik, Justus-Liebig-Universität Gießen
FLAT 2026

Exploring the Landscape of Subregular Expressions

How far do two or three operations take us?

Martin Kutrib & Matthias Wendlandt

Institut für Informatik, Justus-Liebig-Universität Gießen
FLAT 2026

REG

Regular expressions give us all regular languages.

But how much machinery do we actually need?

- ▶ By restricting the available operators, we reveal how different combinations shape the expressive power of regular expressions.

Five tools define the terrain

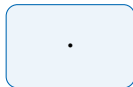


union

Five tools define the terrain

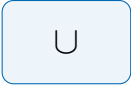


union



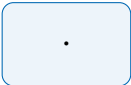
concatenation

Five tools define the terrain



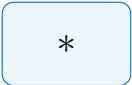
$\cup$

union



.

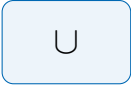
concatenation



*

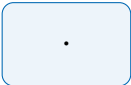
iteration

Five tools define the terrain



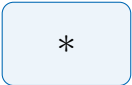
$\cup$

union



$\cdot$

concatenation



$*$

iteration



$\cap$

intersection

Five tools define the terrain

 $\cup$

union

 $\cdot$

concatenation

 $*$

iteration

 $\cap$

intersection

 $-$

complement

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

$$a^* \cup b^*$$

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

$$a^* \cup b^*$$

$$L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$$

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

$$a^* \cup b^*$$

$$L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$$

$$L = ((aa)^* \cup ba(aa)^*aac)^*$$

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

$$a^* \cup b^*$$

$$L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$$

$$L = ((aa)^* \cup ba(aa)^*aac)^*$$

$$\overline{\emptyset \cdot aa \cdot \emptyset} \cap \overline{\emptyset \cdot bb \cdot \emptyset}$$

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

$$a^* \cup b^*$$

$$L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$$

$$L = ((aa)^* \cup ba(aa)^*aac)^*$$

$$\overline{\emptyset \cdot aa \cdot \emptyset} \cap \overline{\emptyset \cdot bb \cdot \emptyset}$$

$$(ab)^*(\varepsilon \cup a) \cup (ba)^*(\varepsilon \cup b)$$

Equivalent expressions can use different operators

Over $\Sigma = \{a, b\}$, the same language can expose very different operators.

$$a^*b^* \cap b^*a^*$$

$$a^* \cup b^*$$

$$L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$$

$$L = ((aa)^* \cup ba(aa)^*aac)^*$$

$$\overline{\emptyset \cdot aa \cdot \emptyset} \cap \overline{\emptyset \cdot bb \cdot \emptyset}$$

$$(ab)^*(\varepsilon \cup a) \cup (ba)^*(\varepsilon \cup b)$$

Operators that are syntactically absent may still emerge semantically.

Five tools define the terrain

 $\cup$

union

 $\cdot$

concatenation

 $*$

iteration

 $\cap$

intersection

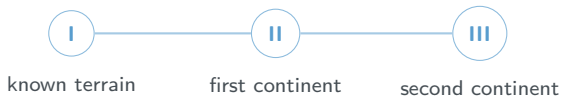
 $-$

complement

The expedition question

What happens if we may choose only **two** or **three** of them?

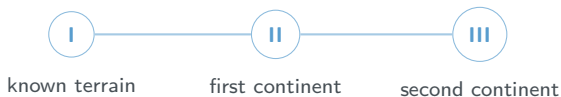
Two continents beyond the familiar terrain



I. Familiar terrain

Clear the routine
consequences before setting
out.

Two continents beyond the familiar terrain



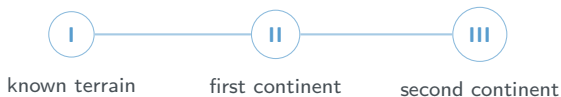
I. Familiar terrain

Clear the routine consequences before setting out.

II. The first continent

Chart every two-tool region, mostly over arbitrary alphabets.

Two continents beyond the familiar terrain



I. Familiar terrain

Clear the routine consequences before setting out.

II. The first continent

Chart every two-tool region, mostly over arbitrary alphabets.

III. The second continent

Chart every three-tool region, mostly in the unary world.

Two continents, every region, and a few revealing landmarks.

A uniform coordinate system for every region

Let Σ be a finite alphabet, $\Lambda \subseteq \Sigma^*$ a finite set of initial words, and Φ a set of regularity-preserving operations.

Restricted regular-like expressions

$$\text{RE}(\Sigma, \Lambda, \Phi) \quad \rightsquigarrow \quad \mathcal{L}_\Phi = \{L(\alpha) \mid \alpha \in \text{RE}(\Sigma, \Lambda, \Phi)\}.$$

- ▶ The same finite basis is used in every region.
- ▶ Only the available operations change.

Why finite literals are part of the equipment

With concatenation

Letters are enough:

$$\{a\} \cdot \{b\} = \{ab\}.$$

Why finite literals are part of the equipment

With concatenation

Letters are enough:

$$\{a\} \cdot \{b\} = \{ab\}.$$

Without concatenation

The word ab cannot be assembled from a and b . A finite literal such as $\{ab, ba\}$ must be available directly.

Map note. The finite basis keeps all operator systems non-trivial and comparable. It also makes our $\mathcal{L}_{(.,*)}$ slightly more liberal than classical union-free expressions.

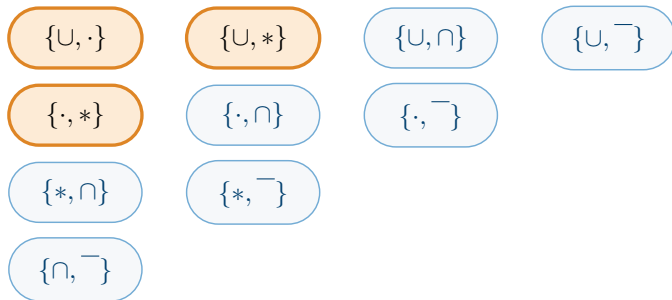
Ten pairings, ten open questions

$\{\cup, \cdot\}$

$\{\cup, *\}$

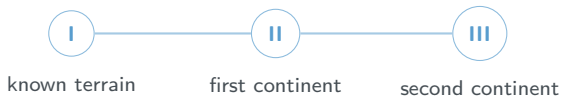
$\{\cdot, *\}$

Ten pairings, ten open questions



Map note. At base camp we know the toolkits, but none of their collapses, borders, or passages.

Expedition I: The familiar territory



Start where closure already tells us the answer.

Different toolkits can lead to exactly the same region.

No generator of infinity means finite languages

Theorem

$$\mathcal{L}_{(U, \cdot)} = \mathcal{L}_{(U, \cap)} = \mathcal{L}_{(\cdot, \cap)} = \text{FIN}.$$

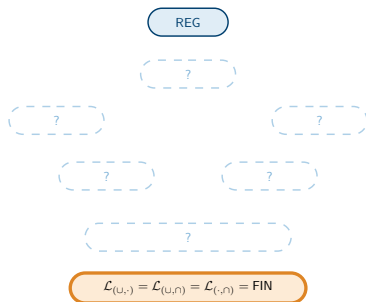
- ▶ We start finite.
- ▶ Union, intersection, and concatenation preserve finiteness.
- ▶ Three syntaxes collapse to one semantic region.

No generator of infinity means finite languages

Theorem

$$\mathcal{L}_{(U, \cdot)} = \mathcal{L}_{(U, \cap)} = \mathcal{L}_{(\cdot, \cap)} = \text{FIN}.$$

- ▶ We start finite.
- ▶ Union, intersection, and concatenation preserve finiteness.
- ▶ Three syntaxes collapse to one semantic region.



Complement adds exactly one new shoreline

De Morgan collapse

$$\mathcal{L}_{(\cup, -)} = \mathcal{L}_{(\cap, -)} = \text{FIN} \cup \text{COFIN}.$$

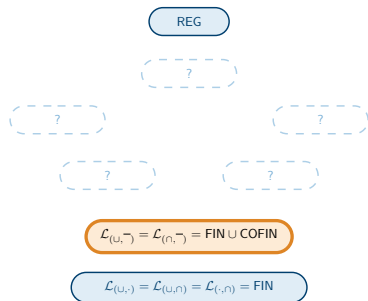
Complement adds exactly one new shoreline

De Morgan collapse

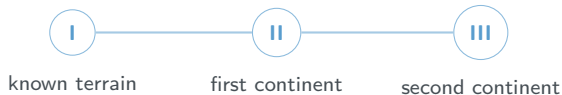
$$\mathcal{L}_{(u,-)} = \mathcal{L}_{(n,-)} = \text{FIN} \cup \text{COFIN}.$$

Complement swaps finite and cofinite sets; union or intersection cannot create any other infinite structure.

Map note. A new operator can enlarge the class without creating periodicity.



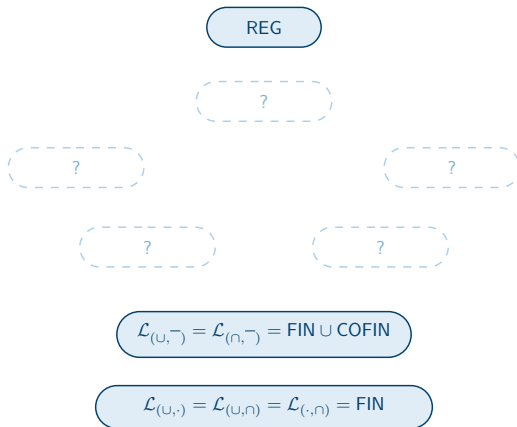
Expedition II: The two-tool continent



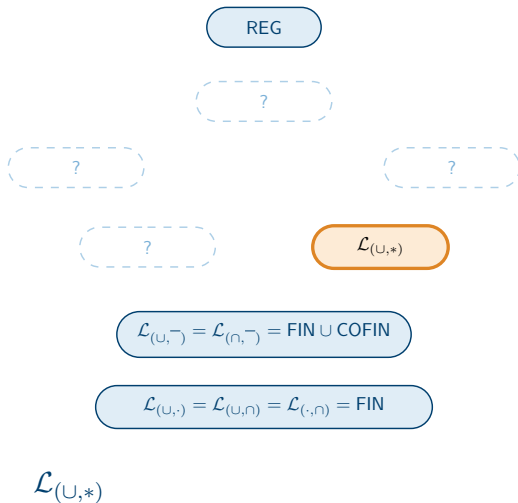
Our first continent contains every two-tool region.

We travel mostly over arbitrary alphabets and stop at revealing landmarks.

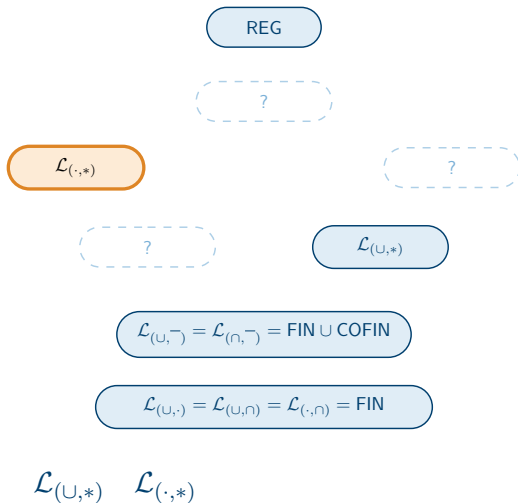
Five regions survive the easy collapses



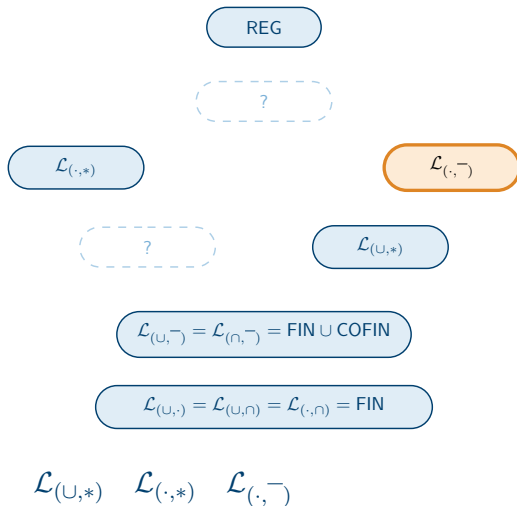
Five regions survive the easy collapses



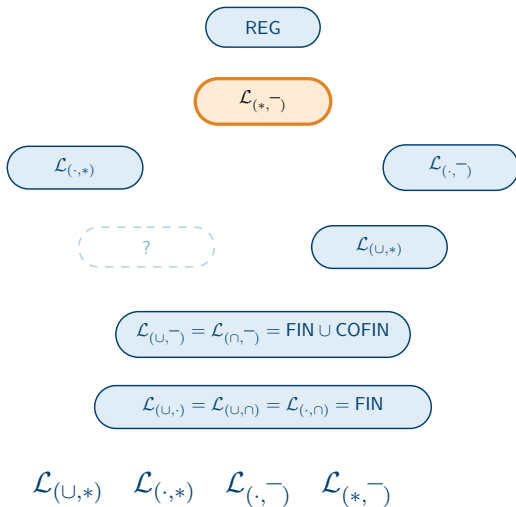
Five regions survive the easy collapses



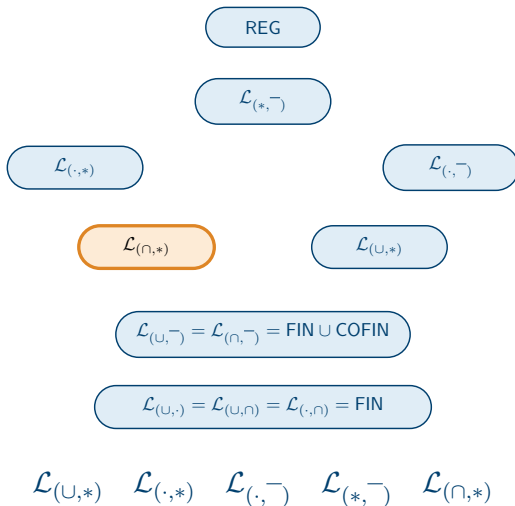
Five regions survive the easy collapses



Five regions survive the easy collapses



Five regions survive the easy collapses



A fixed prefix cannot precede an independent loop

The class $\mathcal{L}_{(\cup,*)}$ is generated from finite languages using only union and Kleene star.

Let $u, v \in \Sigma^+$ and suppose that $u \neq v^\ell$ for every $\ell \geq 1$. Then

$$uv^* \notin \mathcal{L}_{(\cup,*)} \quad \text{and, symmetrically,} \quad v^*u \notin \mathcal{L}_{(\cup,*)}.$$

A fixed prefix cannot precede an independent loop

The class $\mathcal{L}_{(\cup,*)}$ is generated from finite languages using only union and Kleene star.

Let $u, v \in \Sigma^+$ and suppose that $u \neq v^\ell$ for every $\ell \geq 1$. Then

$$uv^* \notin \mathcal{L}_{(\cup,*)} \quad \text{and, symmetrically,} \quad v^*u \notin \mathcal{L}_{(\cup,*)}.$$

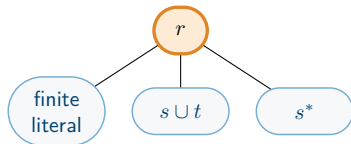
Concrete witness

$$L = \{ab^n \mid n \geq 1\} \notin \mathcal{L}_{(\cup,*)}$$

Why union and star cannot generate the witness

The claim

$$L = \{ab^n \mid n \geq 1\} = abb^* \notin \mathcal{L}_{(\cup,*)}.$$

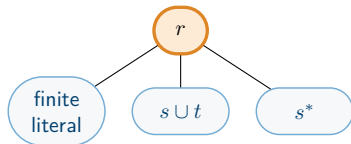


- 1 A finite literal cannot describe the infinite language L .

Why union and star cannot generate the witness

The claim

$$L = \{ab^n \mid n \geq 1\} = abb^* \notin \mathcal{L}_{(\cup,*)}.$$

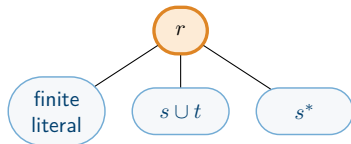


- 1 A finite literal cannot describe the infinite language L .
- 2 If $ab^n \in L(s^*)$, then $(ab^n)^2 \in L(s^*)$, but this word has two occurrences of a and is not in L .

Why union and star cannot generate the witness

The claim

$$L = \{ab^n \mid n \geq 1\} = abb^* \notin \mathcal{L}_{(\cup,*)}.$$



- 1 A finite literal cannot describe the infinite language L .
- 2 If $ab^n \in L(s^*)$, then $(ab^n)^2 \in L(s^*)$, but this word has two occurrences of a and is not in L .
- 3 If $r = s \cup t$, one child contains infinitely many words of L . Follow that child. A finite tree must eventually reach either a star or a finite literal, both already excluded.

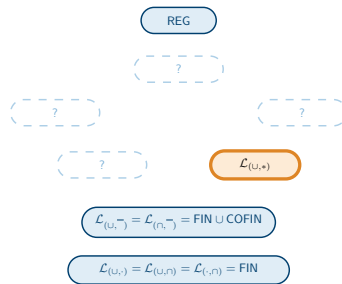
Map note. Every possible root along an infinite branch of the operator tree leads to a contradiction.

Region $\mathcal{L}_{(\cup,*)}$: star repeats the whole expression

Every language has the normal form

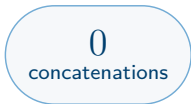
$$L_0 \cup L_1^* \cup \dots \cup L_k^*, \quad L_0, \dots, L_k \in \text{FIN}.$$

- ▶ Nested stars flatten.
- ▶ Union combines complete iterative components.
- ▶ No fixed prefix can be attached to an independent loop.



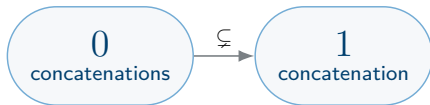
Each concatenation opens a new level

Start in $\mathcal{L}_{(\cup,*)}$ and allow only a bounded number of concatenations.



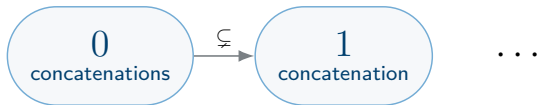
Each concatenation opens a new level

Start in $\mathcal{L}_{(\cup,*)}$ and allow only a bounded number of concatenations.



Each concatenation opens a new level

Start in $\mathcal{L}_{(\cup,*)}$ and allow only a bounded number of concatenations.



Each concatenation opens a new level

Start in $\mathcal{L}_{(\cup,*)}$ and allow only a bounded number of concatenations.



Each concatenation opens a new level

Start in $\mathcal{L}_{(\cup,*)}$ and allow only a bounded number of concatenations.



A language at level k

$$L_k = u_0 v_1^* u_2 v_3^* \cdots u_{k-1} v_k^*.$$

Each concatenation opens a new level

Start in $\mathcal{L}_{(\cup,*)}$ and allow only a bounded number of concatenations.



A language at level k

$$L_k = u_0 v_1^* u_2 v_3^* \cdots u_{k-1} v_k^*.$$

Map note. Each additional concatenation can separate one more fixed component from an independently repeated one; every level is strict.

Union-free languages remain surprisingly expressive

A proper subclass

Union-free expressions use concatenation and iteration, but no union.

$$UF \subsetneq REG.$$

Source: Nagy, 2004, 2006, 2019, 2021;
Jirásková and Masopust, 2011; Jirásková and Nagy, 2012;
Brzozowski and Davies, 2018.

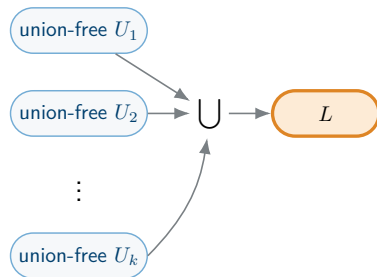
Union-free languages remain surprisingly expressive

A proper subclass

Union-free expressions use concatenation and iteration, but no union.

$$UF \subsetneq REG.$$

Every regular language is a finite union of union-free languages.



Source: Nagy, 2004, 2006, 2019, 2021;
Jirásková and Masopust, 2011; Jirásková and Nagy, 2012;
Brzozowski and Davies, 2018.

Union-free languages remain surprisingly expressive

A proper subclass

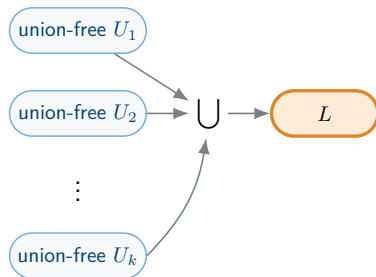
Union-free expressions use concatenation and iteration, but no union.

$$UF \subsetneq REG.$$

Every regular language is a finite union of union-free languages.

Union-complexity

The minimum number of union-free components in such a decomposition is the language's *union-complexity*.



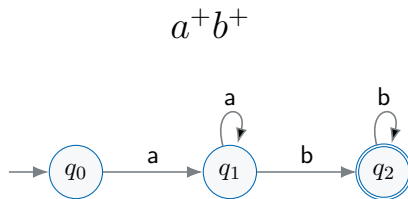
Source: Nagy, 2004, 2006, 2019, 2021;
Jirásková and Masopust, 2011; Jirásková and Nagy, 2012;
Brzozowski and Davies, 2018.

The automaton knows when union is absent

One-cycle-free-path automata

A 1CFPA has exactly one cycle-free path from every state to a final state.

Union-free languages are exactly the languages accepted by 1CFPAs.



Our map and the classical landmark are not identical

Classical union-free

Atomic expressions are single letters.

Every nonempty language has a unique shortest word.

Our map and the classical landmark are not identical

Classical union-free

Atomic expressions are single letters.
Every nonempty language has a unique shortest word.

Our region $\mathcal{L}_{(\cdot,*)}$

Atomic expressions may be arbitrary finite languages. Thus, a literal can already contain several shortest words.

Our map and the classical landmark are not identical

Classical union-free

Atomic expressions are single letters.
Every nonempty language has a unique shortest word.

Our region $\mathcal{L}_{(\cdot,*)}$

Atomic expressions may be arbitrary finite languages. Thus, a literal can already contain several shortest words.

$\{b, c\}\{ac\}^*\{a, b\} \in \mathcal{L}_{(\cdot,*)}$, but the language is not classically union-free.

The separation comes from unstarred finite choice

If a finite literal occurs directly under star, its hidden choice can be reproduced without union:

Union-free simulation

$$\{u_0, u_1, \dots, u_k\}^* = (u_0^* u_1^* \cdots u_k^*)^*.$$

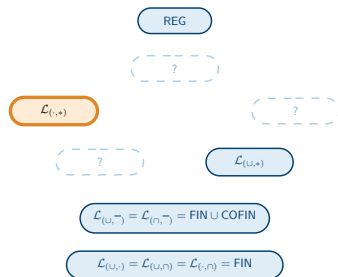
The separation comes from unstarred finite choice

If a finite literal occurs directly under star, its hidden choice can be reproduced without union:

Union-free simulation

$$\{u_0, u_1, \dots, u_k\}^* = (u_0^* u_1^* \cdots u_k^*)^*.$$

The extra expressive power therefore comes from finite literals that occur *outside* a star.



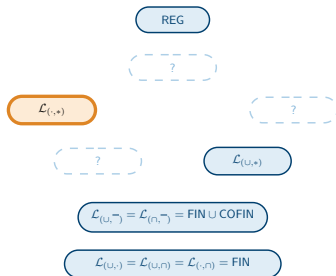
The separation comes from unstarred finite choice

If a finite literal occurs directly under star, its hidden choice can be reproduced without union:

Union-free simulation

$$\{u_0, u_1, \dots, u_k\}^* = (u_0^* u_1^* \cdots u_k^*)^*.$$

The extra expressive power therefore comes from finite literals that occur *outside* a star.



The real obstruction is a non-starred finite union of independently iterated branches.

Missing inhabitant

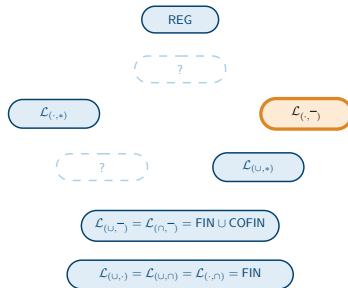
$$a^* \cup b^* \notin \mathcal{L}_{(\cdot,*)}.$$

Region $\mathcal{L}_{(\cdot, \neg)}$ reaches star-free territory

Classical upper bound

$$\mathcal{L}_{(\cdot, \neg)} \subseteq \text{star-free} \subsetneq \text{REG}$$

- ▶ Complement creates infinite sets without star.
- ▶ Concatenation creates ordered composition without union.
- ▶ The missing union cannot always be simulated.



Map note. This region lies inside a classical continent, yet has its own border.

Source: Schützenberger (1965); Cohen–Brzozowski (1971); McNaughton–Papert (1971); Brzozowski–Knast (1978).

Matching endpoints require union

Over $\Sigma = \{a, b\}$, consider

$$L = a\Sigma^*a \cup b\Sigma^*b$$

starts and ends in a

$\cup$

starts and ends in b

Border witness

L is star-free, but $L \notin \mathcal{L}_{(\cdot, -)}$.

Region $\mathcal{L}_{(*, \neg)}$ has no branching at all

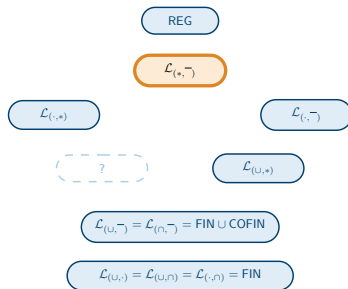
Both operations are unary, so every expression tree degenerates into a path:

$$F \mapsto F^* \mapsto \overline{F^*} \mapsto (\overline{F^*})^* \mapsto \dots$$

The natural expectation is an infinite alternation hierarchy.

Surprise

The hierarchy collapses.



Why the star-complement alternation closes after four steps

Start with a finite language L_0 and alternate the two operations:

$$L_{2i+1} = L_{2i}^*, \quad L_{2i+2} = \overline{L_{2i+1}} \quad (i \geq 0).$$

Why the star-complement alternation closes after four steps

Start with a finite language L_0 and alternate the two operations:

$$L_{2i+1} = L_{2i}^*, \quad L_{2i+2} = \overline{L_{2i+1}} \quad (i \geq 0).$$

Why $L_2 \subseteq L_6 \subseteq L_3$

- ▶ $L_4 \subseteq L_1$ and $L_1 = L_0^*$ is star-closed, hence $L_5 = L_4^* \subseteq L_1$.
- ▶ Complement reverses this inclusion:
 $L_2 = \overline{L_1} \subseteq \overline{L_5} = L_6$.
- ▶ If $x \in L_6$, then $x \notin L_4$; since $L_4 = L_1 \setminus L_3$ and $L_1 \cup L_3 = \Sigma^*$, we get $x \in L_3$.

Why the star-complement alternation closes after four steps

Start with a finite language L_0 and alternate the two operations:

$$L_{2i+1} = L_{2i}^*, \quad L_{2i+2} = \overline{L_{2i+1}} \quad (i \geq 0).$$

Why $L_2 \subseteq L_6 \subseteq L_3$

- ▶ $L_4 \subseteq L_1$ and $L_1 = L_0^*$ is star-closed, hence $L_5 = L_4^* \subseteq L_1$.
- ▶ Complement reverses this inclusion: $L_2 = \overline{L_1} \subseteq \overline{L_5} = L_6$.
- ▶ If $x \in L_6$, then $x \notin L_4$; since $L_4 = L_1 \setminus L_3$ and $L_1 \cup L_3 = \Sigma^*$, we get $x \in L_3$.

Star the entire sandwich

$$\underbrace{L_2^*}_{L_3} \subseteq \underbrace{L_6^*}_{L_7} \subseteq \underbrace{L_3^*}_{L_3}.$$

Hence $L_3 = L_7$.

$$\begin{array}{ccccc} L & L^* & \overline{L^*} & \overline{L^*}^* & \overline{\overline{L^*}^*} \\ L_0 & L_1 & L_2 & L_3 & L_4 \end{array}$$

Why the star-complement alternation closes after four steps

Start with a finite language L_0 and alternate the two operations:

$$L_{2i+1} = L_{2i}^*, \quad L_{2i+2} = \overline{L_{2i+1}} \quad (i \geq 0).$$

Why $L_2 \subseteq L_6 \subseteq L_3$

- ▶ $L_4 \subseteq L_1$ and $L_1 = L_0^*$ is star-closed, hence $L_5 = L_4^* \subseteq L_1$.
- ▶ Complement reverses this inclusion: $L_2 = \overline{L_1} \subseteq \overline{L_5} = L_6$.
- ▶ If $x \in L_6$, then $x \notin L_4$; since $L_4 = L_1 \setminus L_3$ and $L_1 \cup L_3 = \Sigma^*$, we get $x \in L_3$.

Star the entire sandwich

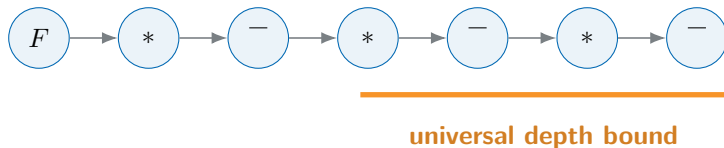
$$\underbrace{L_2^*}_{L_3} \subseteq \underbrace{L_6^*}_{L_7} \subseteq \underbrace{L_3^*}_{L_3}.$$

Hence $L_3 = L_7$.

$$\begin{array}{ccccc} L & L^* & \overline{L^*} & \overline{L^*}^* & \overline{\overline{L^*}^*} \\ L_0 & L_1 & L_2 & L_3 & L_4 \end{array}$$

Map note. Applying complement and star again preserves equality: $L_i = L_{i+4}$ for every $i \geq 3$.

Three stars and three complements are enough



Every language in $\mathcal{L}_{(*,-)}$ has an expression with at most **three applications of star** and **three of complement**.

Map note. Alternation enters a periodic cycle instead of producing a dot-depth-like hierarchy.

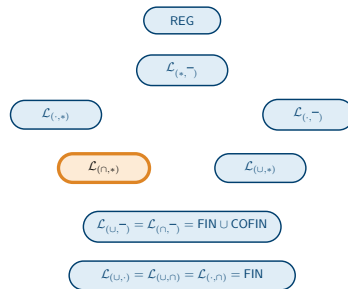
Region $\mathcal{L}_{(\cap,*)}$ only filters iterative monoids

Every language is finite or has the normal form

$$L_1^* \cap L_2^* \cap \dots \cap L_k^*, \quad L_i \in \text{FIN}.$$

Therefore every infinite language in the class satisfies

$$w \in L \implies w^n \in L \quad (n \geq 1).$$



On a unary alphabet, intersection adds nothing after star

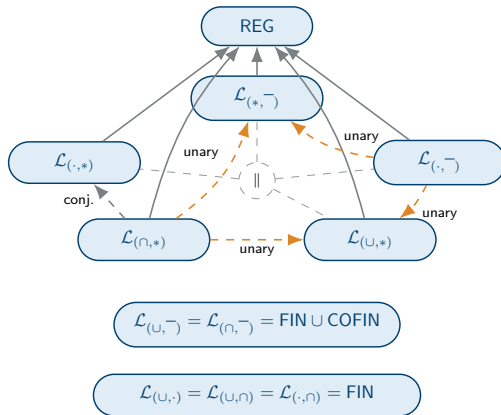
Unary normal form

Every unary language in $\mathcal{L}_{(\cap,*)}$ is finite or equals L^* for one finite language L .



Map note. The alphabet changes the geometry of the landscape.

The general two-tool landscape branches sharply



Map note. Several regions are pairwise incomparable over general alphabets.

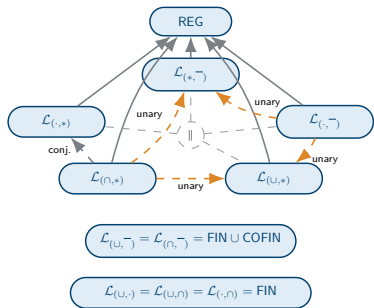
The original witnesses certify incomparability

$$\begin{array}{ll} \mathbf{A}: \mathcal{L}_{(\cup, *)} & \mathbf{B}: \mathcal{L}_{(\cdot, *)} \\ \mathbf{C}: \mathcal{L}_{(\cdot, -)} & \mathbf{E}: \mathcal{L}_{(*, -)} \end{array}$$

Class	Original witness language	Witness is outside
A	$L_k = \{a^{km} \mid m \geq 0\} \cup \{b^{kn} \mid n \geq 0\}$	B, C, E
B	$\{ab^n \mid n \geq 1\}$	A, C, E
C	$\overline{a\Sigma^*a}$, where $\Sigma = \{a, b\}$	A, B, E
E	$\overline{\{aa, bb\}^*}; \quad \overline{\{a^6, a^9\}^*}$	A; B, C

Map note. For every pair of classes, the table supplies a witness in each direction. Hence A, B, C, and E are pairwise incomparable.

Unary alphabets reveal strict inclusions



Strict inclusions for unary languages

For unary languages,

$$\mathcal{L}_{(n,*)} \subsetneq \mathcal{L}_{(u,*)},$$

$$\mathcal{L}_{(\cap,*)} \subsetneq \mathcal{L}_{(\cdot,*)},$$

$$\mathcal{L}_{(\cap, *)} \subsetneq \mathcal{L}_{(*, -)}.$$

Moreover, in the unary case,

$$\mathcal{L}_{(\cdot, -)} = \mathcal{L}_{(\cup, -)} = \text{FIN} \cup \text{COFIN}.$$

A third operation reaches a classical landmark

From two operations to three

Adding union to concatenation and complement gives

$$\mathcal{L}_{(\cup, \cdot, \overline{})} = \text{SF} \subsetneq \text{REG}.$$

- ▶ Infinite languages remain possible.
- ▶ Complement can replace star for many global conditions.
- ▶ Modular counting remains impossible.

Infinity without star

For finite $F \subseteq \Sigma^*$,

$$\overline{F} = \Sigma^* \setminus F$$

is cofinite and therefore infinite.

The remaining border

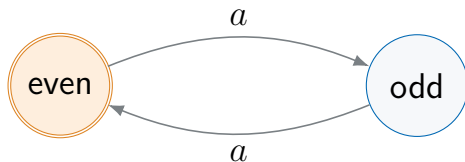
$$\{a^{2n} \mid n \geq 0\} \notin \text{SF}.$$

Parity requires modular counting.

Source: Schützenberger, 1965; McNaughton and Papert, 1971; Brzozowski and Knast, 1978.

Modular counting remains impossible

$$L_{\text{even}} = \{a^{2n} \mid n \geq 0\}.$$



The non-trivial two-cycle records genuine periodicity. Therefore

$$L_{\text{even}} \notin \text{SF}.$$

Two languages with star are nevertheless star-free

Throughout the slide, $\Sigma = \{a, b\}$ and $\Sigma^* = \overline{\emptyset}$.

Language $(ab)^*$

Words alternate, start in a , and end in b . Thus they

- ▶ contain neither aa nor bb ,
- ▶ do not start in b and do not end in a .

$$(ab)^* = \overline{\Sigma^*aa\Sigma^* \cup \Sigma^*bb\Sigma^* \cup b\Sigma^* \cup \Sigma^*a}.$$

Two languages with star are nevertheless star-free

Throughout the slide, $\Sigma = \{a, b\}$ and $\Sigma^* = \bar{\emptyset}$.

Language $(ab)^*$

Words alternate, start in a , and end in b . Thus they

- ▶ contain neither aa nor bb ,
- ▶ do not start in b and do not end in a .

$$(ab)^* = \overline{\Sigma^*aa\Sigma^* \cup \Sigma^*bb\Sigma^* \cup b\Sigma^* \cup \Sigma^*a}.$$

Language $(b \cup ab)^*$

Every occurrence of a is followed by b . Equivalently, a word

- ▶ contains no factor aa ,
- ▶ does not end in a .

$$(b \cup ab)^* = \overline{\Sigma^*aa\Sigma^* \cup \Sigma^*a}.$$

Two languages with star are nevertheless star-free

Throughout the slide, $\Sigma = \{a, b\}$ and $\Sigma^* = \overline{\emptyset}$.

Language $(ab)^*$

Words alternate, start in a , and end in b . Thus they

- ▶ contain neither aa nor bb ,
- ▶ do not start in b and do not end in a .

$$(ab)^* = \overline{\Sigma^*aa\Sigma^* \cup \Sigma^*bb\Sigma^* \cup b\Sigma^* \cup \Sigma^*a}.$$

Language $(b \cup ab)^*$

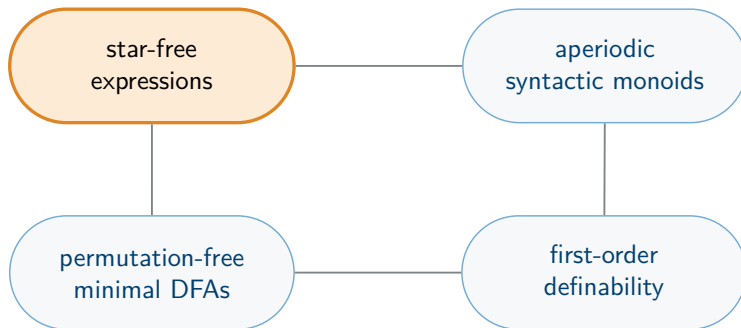
Every occurrence of a is followed by b .
Equivalently, a word

- ▶ contains no factor aa ,
- ▶ does not end in a .

$$(b \cup ab)^* = \overline{\Sigma^*aa\Sigma^* \cup \Sigma^*a}.$$

Map note. The occurrence of star in one expression does not prevent the language from having a star-free expression.

Four views locate exactly the same star-free continent



Aperiodicity

There exists n such that $m^n = m^{n+1}$ for every element m of the syntactic monoid.

Source: Schützenberger (1965); McNaughton–Papert (1971).

Two classical restrictions suggest a broader landscape

Union-free

Start with the classical operations and remove union:

$$\{\cup, \cdot, *\} \setminus \{\cup\} = \{\cdot, *\}.$$

characterized by 1CFPAs

Two classical restrictions suggest a broader landscape

Union-free

Start with the classical operations and remove union:

$$\{\cup, \cdot, *\} \setminus \{\cup\} = \{\cdot, *\}.$$

characterized by 1CFPAs

Star-free

Remove star and add complement:

$$(\{\cup, \cdot, *\} \setminus \{*\}) \cup \{\bar{}\} = \{\cup, \cdot, \bar{}\}.$$

characterized by permutation-free DFAs

Two classical restrictions suggest a broader landscape

Union-free

Start with the classical operations and remove union:

$$\{\cup, \cdot, *\} \setminus \{\cup\} = \{\cdot, *\}.$$

characterized by 1CFPAs

Star-free

Remove star and add complement:

$$(\{\cup, \cdot, *\} \setminus \{*\}) \cup \{\bar{}\} = \{\cup, \cdot, \bar{}\}.$$

characterized by permutation-free DFAs

The next question

What happens if we **remove one** of the three basic operations and **add complement** instead?

Expedition III: The three-tool continent



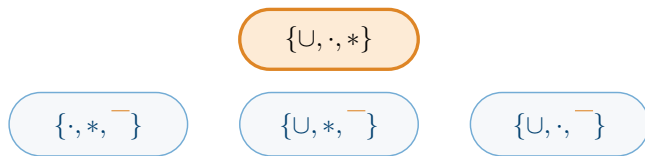
Our second continent contains every three-tool region.

The detailed routes are mostly unary, with a few open borders.

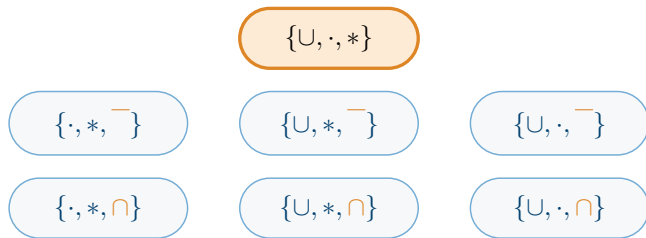
Ten triples, ten new questions

$\{\cup, \cdot, *\}$

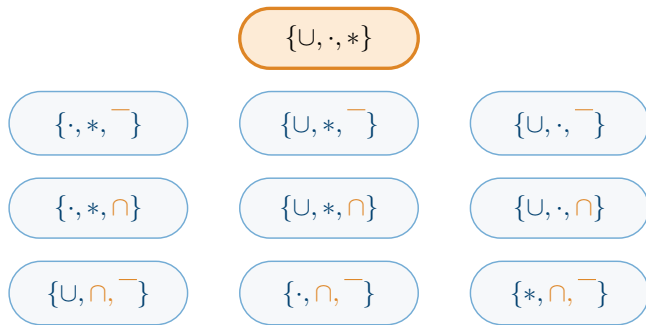
Ten triples, ten new questions



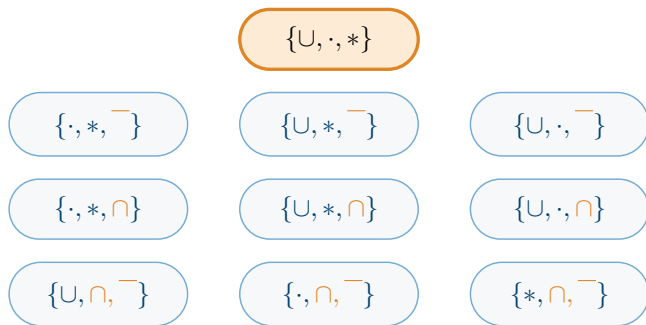
Ten triples, ten new questions



Ten triples, ten new questions



Ten triples, ten new questions



Map note. We now revisit the same question with one additional tool, without assuming which triples collapse.

The familiar destinations reappear immediately

Four triples lead to familiar language families:

- ▶ $\mathcal{L}_{(\cup, \cdot, *)} = \text{REG}$

The familiar destinations reappear immediately

Four triples lead to familiar language families:

- ▶ $\mathcal{L}_{(\cup, \cdot, *)} = \text{REG}$
- ▶ $\mathcal{L}_{(\cup, \cdot, -)} = \text{SF}$

The familiar destinations reappear immediately

Four triples lead to familiar language families:

- ▶ $\mathcal{L}_{(\cup, \cdot, *)} = \text{REG}$
- ▶ $\mathcal{L}_{(\cup, \cdot, -)} = \text{SF}$
- ▶ $\mathcal{L}_{(\cup, \cap, -)} = \text{FIN} \cup \text{COFIN}$

The familiar destinations reappear immediately

Four triples lead to familiar language families:

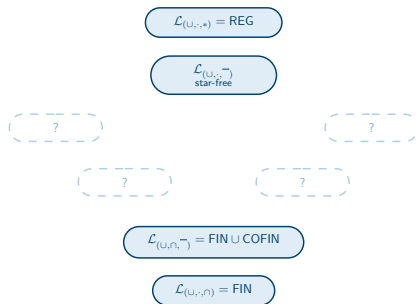
- ▶ $\mathcal{L}_{(\cup, \cdot, *)} = \text{REG}$
- ▶ $\mathcal{L}_{(\cup, \cdot, -)} = \text{SF}$
- ▶ $\mathcal{L}_{(\cup, \cap, -)} = \text{FIN} \cup \text{COFIN}$
- ▶ $\mathcal{L}_{(\cup, \cdot, \cap)} = \text{FIN}$

The familiar destinations reappear immediately

Four triples lead to familiar language families:

- ▶ $\mathcal{L}_{(\cup, \cdot, *)} = \text{REG}$
- ▶ $\mathcal{L}_{(\cup, \cdot, -)} = \text{SF}$
- ▶ $\mathcal{L}_{(\cup, \cap, -)} = \text{FIN} \cup \text{COFIN}$
- ▶ $\mathcal{L}_{(\cup, \cdot, \cap)} = \text{FIN}$

These four require no structural argument.



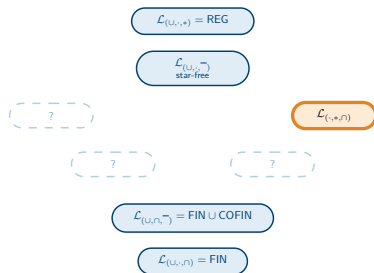
Replacing union by intersection still misses a simple exception

In $\mathcal{L}_{(\cdot, *, \cap)}$, star creates periodic structure and intersection can only remove words already present.

Unary border witness

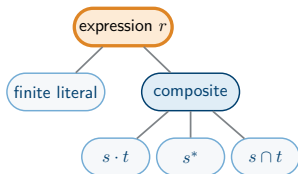
For every $x \geq 2$,

$$W_x = \{a\} \cup \{a^n \mid n \equiv 0 \pmod{x}\} \notin \mathcal{L}_{(\cdot, *, \cap)}.$$



Why the isolated word cannot be manufactured

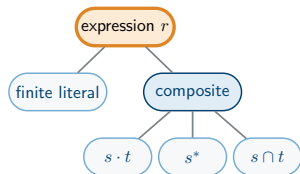
Fix $W_x = \{a\} \cup (a^x)^*$ with $x \geq 2$ and follow the expression tree that is supposed to generate it.



- 1 A finite literal cannot supply the infinite set $(a^x)^*$.

Why the isolated word cannot be manufactured

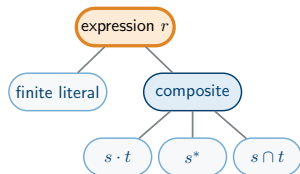
Fix $W_x = \{a\} \cup (a^x)^*$ with $x \geq 2$ and follow the expression tree that is supposed to generate it.



- 1 A finite literal cannot supply the infinite set $(a^x)^*$.
- 2 Star supplies infinitely many multiples, but any component that supplies a also supplies $a^2, a^3, \dots$

Why the isolated word cannot be manufactured

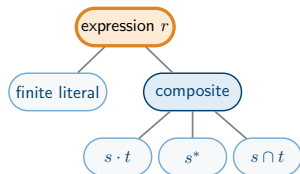
Fix $W_x = \{a\} \cup (a^x)^*$ with $x \geq 2$ and follow the expression tree that is supposed to generate it.



- 1 A finite literal cannot supply the infinite set $(a^x)^*$.
- 2 Star supplies infinitely many multiples, but any component that supplies a also supplies $a^2, a^3, \dots$
- 3 Concatenation combines existing length patterns; it does not detach a from its repetitions.

Why the isolated word cannot be manufactured

Fix $W_x = \{a\} \cup (a^x)^*$ with $x \geq 2$ and follow the expression tree that is supposed to generate it.



- 1 A finite literal cannot supply the infinite set $(a^x)^*$.
- 2 Star supplies infinitely many multiples, but any component that supplies a also supplies $a^2, a^3, \dots$
- 3 Concatenation combines existing length patterns; it does not detach a from its repetitions.
- 4 Intersection can only retain a word already present in every operand. It cannot add a independently of the iterative part.

Map note. The exceptional length 1 and the periodic tail cannot originate from one intersection-only branch structure.

Complement is strictly stronger on unary alphabets

Every unary regular language belongs to

$$\mathcal{L}_{(\cdot, *, \bar{\cdot})}.$$

Complement can describe the missing lengths of an ultimately periodic set; concatenation and star assemble its finite prefix and periodic tail.

Source: Werner, 2013; K. W., 2018.

Complement is strictly stronger on unary alphabets

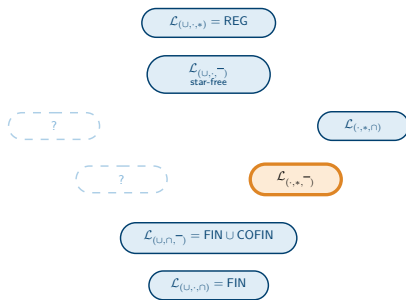
Every unary regular language belongs to

$$\mathcal{L}_{(\cdot, *, \neg)}.$$

Complement can describe the missing lengths of an ultimately periodic set; concatenation and star assemble its finite prefix and periodic tail.

Strict unary inclusion

$$\mathcal{L}_{(\cdot, *, \cap)} \subsetneq \mathcal{L}_{(\cdot, *, \neg)} = \text{REG} \quad \text{over } \{a\}.$$



Source: Werner, 2013; K. W., 2018.

Complement is strictly stronger on unary alphabets

Every unary regular language belongs to

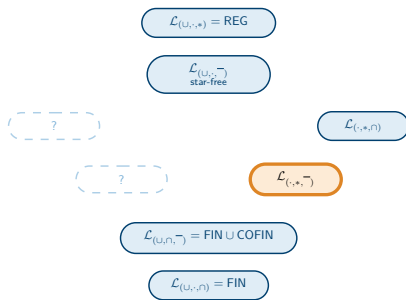
$$\mathcal{L}_{(\cdot, *, \bar{\cdot})}.$$

Complement can describe the missing lengths of an ultimately periodic set; concatenation and star assemble its finite prefix and periodic tail.

Strict unary inclusion

$$\mathcal{L}_{(\cdot, *, \cap)} \subsetneq \mathcal{L}_{(\cdot, *, \bar{\cdot})} = \text{REG} \quad \text{over } \{a\}.$$

For arbitrary alphabets, whether this inclusion remains proper is open.



Source: Werner, 2013; K. W., 2018.

Complement isolates one exceptional length

Unary language and expression

$aaaaa$ 0000010000...

$\overline{aaaaa}$ 1111101111...

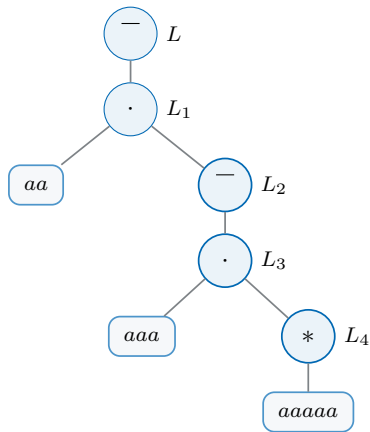
$a\overline{aaaaa}$ 01111101111...

$\overline{a\overline{aaaaa}}$ 10000010000...

Complement isolates one exceptional length

Unary language and expression

$$L = \{a\} \cup \{a^{5n} \mid n \geq 0\},$$
$$r = \overline{\{aa\} \cdot \overline{\{aaa\} \cdot \{aaaaa\}^*}}.$$



Complement isolates one exceptional length

Unary language and expression

$$L = \{a\} \cup \{a^{5n} \mid n \geq 0\},$$
$$r = \overline{\{aa\} \cdot \overline{\{aaa\} \cdot \{aaaaa\}^*}}.$$

Bottom-up evaluation

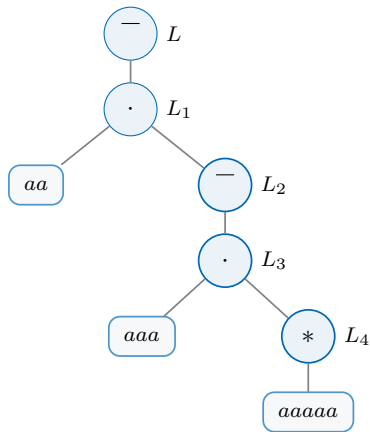
L_4 = lengths 0 (mod 5)

L_3 = lengths 3 (mod 5)

L_2 = lengths not 3 (mod 5)

L_1 = $n \geq 2$ and $n \not\equiv 0 \pmod{5}$

L = $\{a\}$ plus lengths 0 (mod 5)



An open inclusion problem over arbitrary alphabets

Known over $\{a\}$

$$\mathcal{L}_{(\cdot, *, \cap)} \subsetneq \mathcal{L}_{(\cdot, *, \bar{\cdot})}.$$

The left class misses $\{a\} \cup (a^x)^*$; the right class contains every unary regular language.

An open inclusion problem over arbitrary alphabets

Known over $\{a\}$

$$\mathcal{L}_{(\cdot, *, \cap)} \subsetneq \mathcal{L}_{(\cdot, *, \bar{\cdot})}.$$

The left class misses $\{a\} \cup (a^x)^*$; the right class contains every unary regular language.

Open over arbitrary alphabets

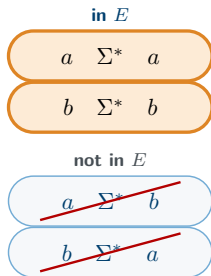
Does the inclusion still hold, or are the two classes incomparable?

$$\mathcal{L}_{(\cdot, *, \cap)} \dashrightarrow \mathcal{L}_{(\cdot, *, \bar{\cdot})}$$

A binary witness marks the general border

Over $\Sigma = \{a, b\}$, consider the regular language

$$E = \{\lambda\} \cup \{awa \mid w \in \Sigma^*\} \cup \{bwb \mid w \in \Sigma^*\}.$$

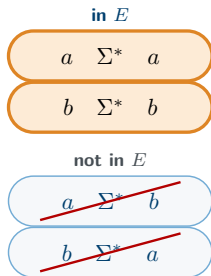


- ▶ A finite automaton only has to remember the first symbol and compare it with the last one.
- ▶ Since $a^n, b^n \in E$ for every $n \geq 2$, the expression must generate two unbounded families through starred subexpressions.
- ▶ Their iterations cannot maintain this dependency between two arbitrarily distant endpoints.

A binary witness marks the general border

Over $\Sigma = \{a, b\}$, consider the regular language

$$E = \{\lambda\} \cup \{awa \mid w \in \Sigma^*\} \cup \{bwb \mid w \in \Sigma^*\}.$$



- ▶ A finite automaton only has to remember the first symbol and compare it with the last one.
- ▶ Since $a^n, b^n \in E$ for every $n \geq 2$, the expression must generate two unbounded families through starred subexpressions.
- ▶ Their iterations cannot maintain this dependency between two arbitrarily distant endpoints.

$E \notin \mathcal{L}_{(\cdot, *, -)}$, hence $\mathcal{L}_{(\cdot, *, -)} \subsetneq \text{REG}$.

The three remaining concatenation-free families

The three remaining operation sets define

$$\mathcal{L}_{(\cup, *, \cap)} \quad \text{and} \quad \mathcal{L}_{(\cup, *, -)} = \mathcal{L}_{(\cap, *, -)}.$$

Stretching

In the unary case, let $L \subseteq \{a\}^*$ and $m \geq 1$. The *stretching of L by m* is

$$L_{(m)} = \{a^{mn} \mid a^n \in L\}.$$

Thus, stretching multiplies every word length by the same factor m .

Unary concatenation-free languages are stretched patterns

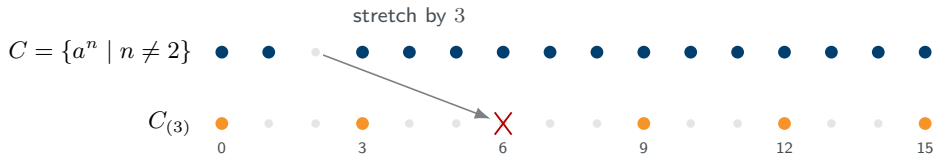
For $L \subseteq \{a\}^*$ and $m \geq 1$, define

$$L_{(m)} = \{a^{mn} \mid a^n \in L\}.$$

Unary concatenation-free languages are stretched patterns

For $L \subseteq \{a\}^*$ and $m \geq 1$, define

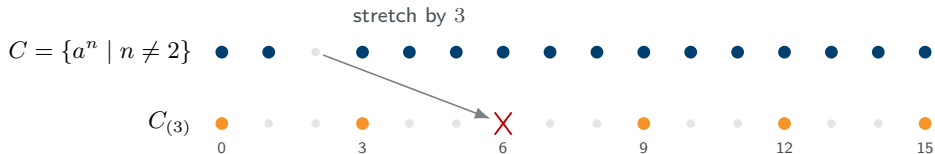
$$L_{(m)} = \{a^{mn} \mid a^n \in L\}.$$



Unary concatenation-free languages are stretched patterns

For $L \subseteq \{a\}^*$ and $m \geq 1$, define

$$L_{(m)} = \{a^{mn} \mid a^n \in L\}.$$



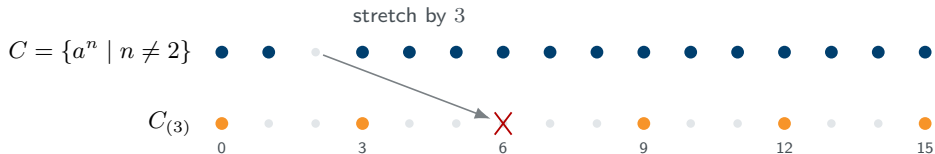
What stretching does

It multiplies every accepted length by m ; it cannot shift the pattern to another residue class.

Unary concatenation-free languages are stretched patterns

For $L \subseteq \{a\}^*$ and $m \geq 1$, define

$$L_{(m)} = \{a^{mn} \mid a^n \in L\}.$$



What stretching does

It multiplies every accepted length by m ; it cannot shift the pattern to another residue class.

Why cofinite bases matter

If C is cofinite, then $C_{(m)}$ contains all sufficiently large multiples of m , with only finitely many holes.

Map note. $\mathcal{U}$ consists of finite unary languages and stretched cofinite unary languages containing λ .

Union, star, and intersection give the union closure of $\mathcal{U}$

Here

$$\mathcal{U} = \text{FIN}_{\{a\}} \cup \{C_{(m)} \mid C \subseteq \{a\}^* \text{ cofinite}, \lambda \in C, m \geq 1\}.$$

Thus, $\mathcal{U}$ consists of all finite unary languages and the stretchings of cofinite unary languages containing λ .

Union, star, and intersection give the union closure of $\mathcal{U}$

Here

$$\mathcal{U} = \text{FIN}_{\{a\}} \cup \{C_{(m)} \mid C \subseteq \{a\}^* \text{ cofinite}, \lambda \in C, m \geq 1\}.$$

Thus, $\mathcal{U}$ consists of all finite unary languages and the stretchings of cofinite unary languages containing λ .

Exact unary characterization

$$\mathcal{L}_{(\cup, *, \cap)} = \Gamma_{\cup}(\mathcal{U}) = \left\{ \bigcup_{i=1}^k U_i \mid k \geq 0, U_i \in \mathcal{U} \right\}.$$

Union, star, and intersection give the union closure of $\mathcal{U}$

Here

$$\mathcal{U} = \text{FIN}_{\{a\}} \cup \{C_{(m)} \mid C \subseteq \{a\}^* \text{ cofinite, } \lambda \in C, m \geq 1\}.$$

Thus, $\mathcal{U}$ consists of all finite unary languages and the stretchings of cofinite unary languages containing λ .

Exact unary characterization

$$\mathcal{L}_{(\cup, *, \cap)} = \Gamma_{\cup}(\mathcal{U}) = \left\{ \bigcup_{i=1}^k U_i \mid k \geq 0, U_i \in \mathcal{U} \right\}.$$

This means that a unary language belongs to $\mathcal{L}_{(\cup, *, \cap)}$ exactly when it is a finite union

$$L = U_1 \cup U_2 \cup \cdots \cup U_k \quad \text{with each } U_i \in \mathcal{U}.$$

Union, star, and intersection give the union closure of $\mathcal{U}$

Here

$$\mathcal{U} = \text{FIN}_{\{a\}} \cup \{C_{(m)} \mid C \subseteq \{a\}^* \text{ cofinite, } \lambda \in C, m \geq 1\}.$$

Thus, $\mathcal{U}$ consists of all finite unary languages and the stretchings of cofinite unary languages containing λ .

Exact unary characterization

$$\mathcal{L}_{(\cup, *, \cap)} = \Gamma_{\cup}(\mathcal{U}) = \left\{ \bigcup_{i=1}^k U_i \mid k \geq 0, U_i \in \mathcal{U} \right\}.$$

This means that a unary language belongs to $\mathcal{L}_{(\cup, *, \cap)}$ exactly when it is a finite union

$$L = U_1 \cup U_2 \cup \cdots \cup U_k \quad \text{with each } U_i \in \mathcal{U}.$$

Hence every infinite component is eventually concentrated on multiples of some fixed modulus.

A shifted residue class lies beyond that coast

Separation theorem

For $1 \leq x < y$,

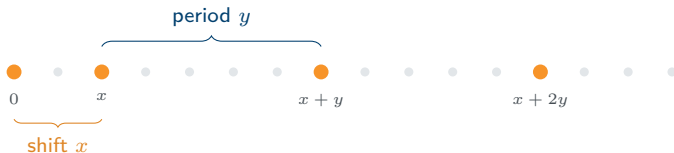
$$R_{x,y} = \{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\} \notin \mathcal{L}_{(\cup, *, \cap)}.$$

A shifted residue class lies beyond that coast

Separation theorem

For $1 \leq x < y$,

$$R_{x,y} = \{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\} \notin \mathcal{L}_{(\cup, *, \cap)}.$$

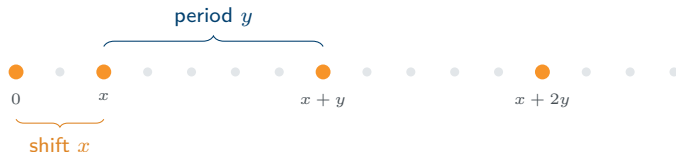


A shifted residue class lies beyond that coast

Separation theorem

For $1 \leq x < y$,

$$R_{x,y} = \{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\} \notin \mathcal{L}_{(\cup, *, \cap)}.$$



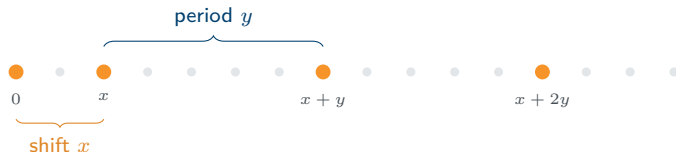
Every infinite member of $\mathcal{U}$ is eventually supported on multiples of a modulus. Finite unions preserve this zero-based support; $R_{x,y}$ occupies a genuinely shifted residue class.

A shifted residue class lies beyond that coast

Separation theorem

For $1 \leq x < y$,

$$R_{x,y} = \{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\} \notin \mathcal{L}_{(\cup, *, \cap)}.$$



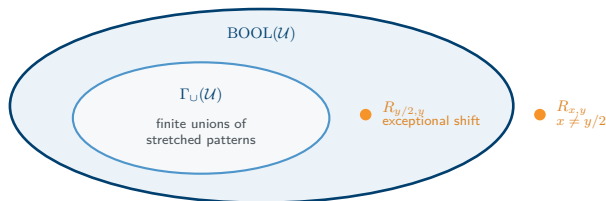
Every infinite member of $\mathcal{U}$ is eventually supported on multiples of a modulus. Finite unions preserve this zero-based support; $R_{x,y}$ occupies a genuinely shifted residue class.

Map note. The empty word can be added finitely, but the infinite shifted tail cannot be generated.

Complement strictly enlarges the unary concatenation-free region

In the unary case,

$$\mathcal{L}_{(\cup, *, \cap)} = \Gamma_{\cup}(\mathcal{U}) \subsetneq \text{BOOL}(\mathcal{U}) = \mathcal{L}_{(\cup, *, -)} = \mathcal{L}_{(\cap, *, -)}.$$



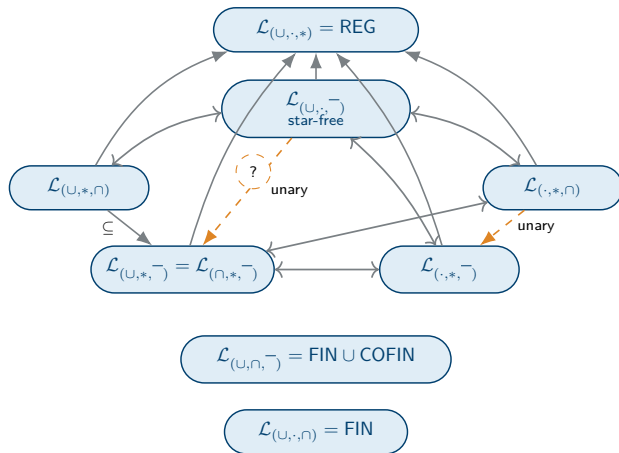
The exceptional case $x = y/2$

For $x = y/2$,

$$r = \overline{(a^x)^* \cup (a^y)^*}, \quad L(r) = \{a^n \mid n \equiv x \pmod{y}\}.$$

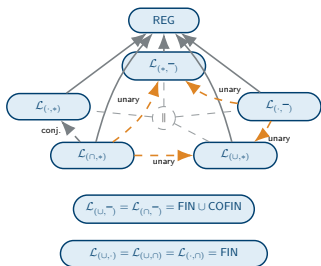
Hence $R_{y/2, y} = L(r) \cup \{\lambda\}$ belongs to $\text{BOOL}(\mathcal{U})$.

The three-tool map contains facts and blank borders

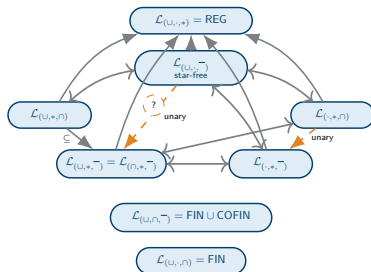


Two tools create regions; three tools reveal passages

Two tools



Three tools



Four routes beyond the current map

Classification

lift or refute unary relations

Structure

automata, monoids, normal forms

Effective recognition

decide $L \in \mathcal{L}_\Phi$ and construct expressions

Complexity

measure the cost of a missing operator

Map note. A structural characterization can turn a blank border into both a theorem and an algorithm.

Three open borders stand out

Border	Unary alphabets	Arbitrary alphabets
$\mathcal{L}_{(\cap,*)}$ versus $\mathcal{L}_{(.,*)}$	$\mathcal{L}_{(\cap,*)} \subsetneq \mathcal{L}_{(.,*)}$	strict inclusion conjectured
$\mathcal{L}_{(.,*,\cap)}$ versus $\mathcal{L}_{(.,*,\bar{-})}$	$\mathcal{L}_{(.,*,\cap)} \subsetneq \mathcal{L}_{(.,*,\bar{-})}$	inclusion or incomparability?
$\mathcal{L}_{(\cup,.,\bar{-})}$ versus $\mathcal{L}_{(\cup,*,\bar{-})}$	$\mathcal{L}_{(\cup,.,\bar{-})} \subsetneq \mathcal{L}_{(\cup,*,\bar{-})}$	exact relation unknown

Map note. Which arithmetic inclusions over unary alphabets survive when symbol order becomes available?

Thank you for your attention!
