On the Structure of Sets Testable in the Strict Sense

Dakotah Lambert

Department of Mathematics and Computer Science
Lake Forest College

Mathematics of Language 2025
16 August 2025

Strictly local languages

Forbidden k-factors

Alternatively: Permissible k-factors

Famous from McNaughton and Papert (1971)

Characterizations

- **Language-theoretic:** suffix-substitution closure

$$\frac{\begin{array}{c} a\overbrace{x}^{k-1}b \\ cxd \end{array}}{axd}$$

- **Logical:** conjunction of negated "w is a substring" literals
- **Automata-theoretic:** acyclic pair-graph
- **Algebraic:** ???

Why algebra?

Language and Logic require thought

Automata-based methods differ wildly by class

Algebra, for subregular things, is **Just Look At It**

Monoids

Monoid: $\langle M, \cdot, 1\rangle$ where $u \cdot (v \cdot w) = (u \cdot v) \cdot w$; $1 \cdot u = u = u \cdot 1$

Free monoid: Σ^* — words over alphabet Σ

Syntactic monoid (from automaton)
treat letters x as functions $[x] \colon Q \to Q$
include $1 = [\lambda]$ where $[\lambda](q) = q$

$$[u] \cdot [v] = [uv] = [v] \circ [u]$$

Recognition

Homomorphism $h\colon \langle A, \cdot\rangle \to \langle B, \odot\rangle$

where $\quad h(u \cdot v) = h(u) \odot h(v); \quad h(1_A) = 1_B$

Recognition

there is $h\colon \Sigma^* \to M$ and $X \subseteq M$ where $L = h^{-1}(X)$

"elements represent only accepted words or only rejected words"

Complementation

Swap which elements are accepting / rejecting

But the complement of "forbid ab" (SL)
is "require ab" (not SL)

$$\begin{array}{rl} b\overbrace{b^{k-1}}^{k-1}ab & \in \\ abb^{k-1}b & \in \\ \hline bb^{k-1}b & \notin \end{array}$$

Valid extensions ("quotients")

$w^{-1}L = \{x : wx \in L\}$
make accepting (only) those x where wx was accepting

$Lw^{-1} = \{x : xw \in L\}$
make accepting (only) those x where xw was accepting

SL is closed under these operations!

Inverse images of homomorphisms

$$
\begin{tikzcd}
\Gamma^* \arrow[r, "\phi"] \arrow[dr, dashed, "\psi \circ \phi"'] & \Sigma^* \arrow[d, "\psi"] \\
& M
\end{tikzcd}
$$

But with $\Gamma = \{a, b, c\}$, $\Sigma = \{a, b\}$, $\phi(a) = a$, $\phi(b) = b$, $\phi(c) = \lambda$
L as "forbid ab" on Σ becomes $\phi^{-1}(L)$ where

$$
\begin{array}{cc}
a\overbrace{c^{k-1}}^{k-1}c & \in \\
cc^{k-1}b & \in \\
\hline
ac^{k-1}b & \notin
\end{array}
$$

Okay not monoids. Semigroups?

Semigroup: like a monoid, but 1 not required

Free semigroup: Σ^+ — nonempty words over alphabet Σ

Syntactic semigroup (from automaton)
like syntactic monoid, but include $1 = [\lambda]$ only if needed

Homomorphisms only require preserving concatenation

Closures

Complementation relative to Σ^+ (still bad!)

Nonempty valid extensions (still fine!)

Inverse images of nonerasing homomorphisms
because there isn't necessarily a unit to erase into
(SL is closed under this!)

Okay not semigroups. Ordered semigroups?

Avoid closure under complementation
by adding an order: $x \leqslant y$ means $uxv \in L \Leftarrow uyv \in L$
(Pin, 1997)

If $x \leqslant y$ in L, then $y \leqslant x$ in $\overline{L}$

This is almost what we need, BUT...

Multiple structures: pseudovarieties

Pseudovarieties capture all objects
with a given kind of structure, given by equations
(or, for “ordered” objects, inequalities)

These are preserved by finite products too
(and taking of subalgebras and homomorphic images)

So get union (bad!) and intersection (fine!)

Two operations?

Need an operation that induces
something like an order (to avoid complementation)
and that makes unions not be a direct product

Equations and their meaning

In the syntactic semigroup, $[x] = [y]$ means
$$uxv \in L \Leftrightarrow uyv \in L$$

This is why ordered structures have $[x] \leqslant y$ means
$$uxv \in L \Leftarrow uyv \in L$$
"Having this word implies having this one"

What if we had sets?
"Having these words implies having this one"
Just like "having axb and cxd implies having axd"

The solution

Join-semigroup: $\langle S, \cdot, \vee \rangle$ where
$\langle S, \cdot \rangle$ is a semigroup,
$\langle S, \vee \rangle$ is a semilattice
($x \vee x = x$ and $x \vee y = y \vee x$ for all x, y)

$$x \cdot (y \vee z) = x \cdot y \vee x \cdot z \qquad (x \vee y) \cdot z = x \cdot z \vee y \cdot z$$

Not a semiring, though that is the terminology of Polák (2001).

Free join-semigroup: $\Sigma^{\boxplus}$ — nonempty subsets of Σ^{+}

$$A \cdot B = \{a \cdot b : a \in A, b \in B\} \qquad A \vee B = A \cup B$$

The syntactic join-semigroup

For T a set of states, let $\mathcal{L}(T)$ be the intersection of the languages of those states.

- Build the syntactic semigroup as usual and lift $\cdot$ to sets
- Say two sets A and B are equivalent if for all sets T of states,

  $$\bigcap_{a\in A} a^{-1}\mathcal{L}(T) = \bigcap_{b\in B} b^{-1}\mathcal{L}(T)$$

  This means for any U, the valid continuations of all UA are the same as those of all UB. That is, $UAV \subseteq L \Leftrightarrow UBV \subseteq L$.
- Join induces an order: $A \leqslant B$ means $A \vee B = B$ as above $U(A \cup B)V \subseteq L \Leftrightarrow UBV \subseteq L$

Strictly local languages

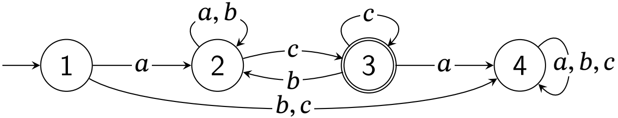
"For sufficiently long x, having axb and cxd implies having axd"
still valid if wrapped in $u\_\_v$

$$U\{ax^{\omega}d, ax^{\omega}b, cx^{\omega}d\}V \subseteq L \Leftrightarrow U\{ax^{\omega}b, cx^{\omega}d\}V$$

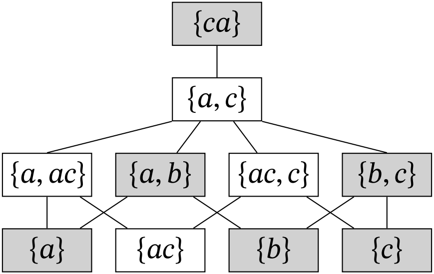
$$[\![ax^{\omega}d \leqslant ax^{\omega}b \vee cx^{\omega}d]\!]$$

Example

start with a, end with c, forbid ca



$$
\begin{array}{rcccc}
 & 1 & 2 & 3 & 4 \\
[\mathrm{a}] = & \langle 2 & 2 & 4 & 4 \rangle \\
[\mathrm{b}] = & \langle 4 & 2 & 2 & 4 \rangle \\
[\mathrm{c}] = & \langle 4 & 3 & 3 & 4 \rangle \\
[\mathrm{ac}] = & \langle 3 & 3 & 4 & 4 \rangle \\
[\mathrm{ca}] = & \langle 4 & 4 & 4 & 4 \rangle
\end{array}
$$



Strictly piecewise languages

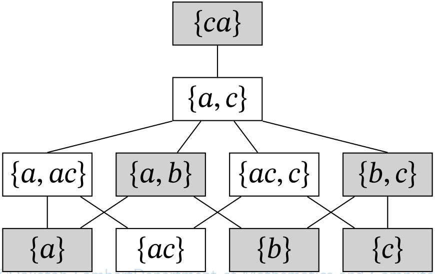
Closure under subsequence (Rogers et al., 2010):

$$uav \in L \Rightarrow uv \in L$$

Equivalently: $U(\{\lambda\} \cup A)V \subseteq L \;\Leftrightarrow\; UAV \subseteq L$

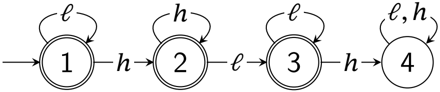
$$[\![1 \leqslant a]\!]$$

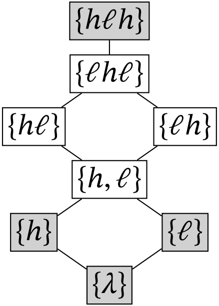
Not strictly piecewise because, *e.g.*, $\{a\} \nleqslant \{ac\}$



Example 2

No $h \ldots \ell \ldots h$





Not strictly local:
$h\ell^\omega h \not\leq h\ell^\omega\ell \vee \ell\ell^\omega h$

Is strictly piecewise

Summary and Prospects

- Join-semigroups (/monoids) for
  "if I have this and this and...then I have this"
- $\leqslant$ is an equality in disguise
- Strictly local: $[\![ax^{\omega}d \leqslant ax^{\omega}b \vee cx^{\omega}d]\!]$
- Strictly piecewise: $[\![1 \leqslant x]\!]$

- There's an analogous class of functions in there.
- Multitier SL: "what stays true when λ is allowed?"

Make the computer do it

install Haskell (GHC), then

```
$ cabal update && cabal install language-toolkit
```

```
$ plebby
plebby, version 1.3: https://hackage.haskell.org/package/language-toolkit
:help for help
> =a{/a} =b{/b} =c{/c}
> :isJVarietyS [ax*d<ax*b+cx*d] !<a,b,a>
False
> :isJVarietyS [ax*d<ax*b+cx*d] /\{%|<a>,|%<c>,!<c a>}
True
```

References

Robert McNaughton and Seymour Aubrey Papert. 1971. Counter-Free Automata. MIT Press.

Jean-Éric Pin. 1997. Syntactic semigroups. In Grzegorz Rozenberg and Arto Salomaa, editors, Handbook of Formal Languages: Volume 1 Word, Language, Grammar, pages 679–746. Springer-Verlag, Berlin.

Libor Polák. 2001. Syntactic semiring of a language. In Mathematical Foundations of Computer Science 2001, volume 2136 of Lecture Notes in Computer Science, pages 611–620.

James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, The Mathematics of Language: Revised Selected Papers from the 10th and 11th Biennial Conference on the Mathematics of Language, volume 6149 of LNCS/LNAI, pages 255–265. FoLLI/Springer.