An Algebraic Characterization of Total Input Strictly Local Functions

Dakotah Lambert* and Jeffrey Heinz**

*Université Jean Monnet Saint-Étienne, CNRS
Institut d Optique Graduate School
Laboratoire Hubert Curien UMR 5516

**Department of Linguistics
Institute of Advanced Computational Science
Stony Brook University

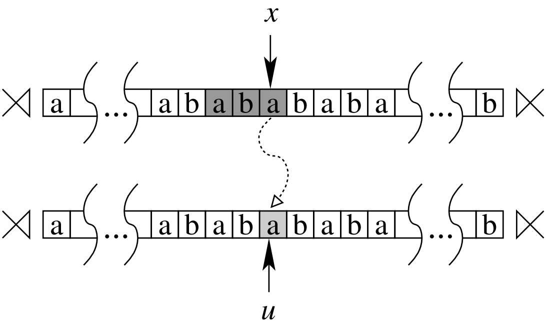
SCiL
UMass Amherst
June 15, 2023

Input Strictly Local Functions

- They are **Definite** functions.
- Definite structures form an algebraic variety.
- Definite functions are decidable.
- There is a rich class of subregular algebraic structures unifying functions and stringsets.
- Algebraic tools like direct and semi-direct products become available for linguistic analysis and synthesis.

History

Input Strictly Local Functions (Chandlee 2014)



The output at position i only depends on the ith symbol and the previous $k-1$ symbols.

Input Strictly Local Functions (Chandlee 2014)

- Have good empirical coverage of local phonological and morphological processes.
  (Chandlee 2017, Chandlee and Heinz 2018)
- Have efficient, interpretable and provably correct learning algorithms (given k).
  (Chandlee et al. 2014, Jardine et al. 2014)
- Have a grammar-independent characterization in terms of their residual functions.
  (Chandlee 2014)
- Are logically characterized with quantifier-free logical transductions.
  (Lindell and Chandlee 2016)
- Are directly related to the Strictly Local stringsets.
  (Rogers and Pullum 2012)

Open Questions

- What else is k-ISL? (tokenization, g2p, p2g, etc)
- How does one decide if a given transducer is k-ISL?
- Are k-ISL functions closed under composition?

Open Questions

- What else is k-ISL? (tokenization, g2p, p2g, etc)

  In progress, stay tuned

- How does one decide if a given transducer is k-ISL?

  Use Definiteness

- Are k-ISL functions closed under composition?

  Yes, provided the function applied first outputs non-empty strings in every k-span, otherwise No

Local Functions (Sakarovitch 2009)

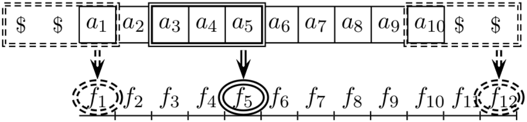


Figure 1.9: Sliding window of length 3

"An even more restricted family of functions [than rational functions] is that where the only memory is a fixed-size part of the input."

p-Local Functions (Vaysse 1986)

Dans cet article, nous introduisons les fonctions p-locales et les fonctions p-sous locales (où p est un entier strictement positif) et nous les caractérisons par une propriété simple de leur semigroupe syntactique : ce semigroupe doit satisfaire l'équation $yx_1 \ldots x_p = x_1 \ldots x_p$. Nous en déduisons quelques propriétés des fonctions p- locales.

Definite Automata (Perles et al. 1963)

"A definite automaton is, roughly speaking, an automaton (sequential circuit) with the property that for some fixed integer k its action depends only on the last k inputs."

Algebraic Theory of Formal Languages

Nerode and Myhill Equivalence

Nerode: $x \overset{N}{\sim} y$ iff for all $v \in \Sigma^*$ it holds that
$xv \in L \Leftrightarrow yv \in L$

Myhill: $x \overset{M}{\sim} y$ iff for all $u, v \in \Sigma^*$ it holds that
$uxv \in L \Leftrightarrow uyv \in L$

Nerode is a right congruence and Myhill is a congruence.

Automata

- The Nerode equivalence corresponds to the minimal deterministic DFA accepting L.

- The Myhill equivalence corresponds with the DFA called the syntactic monoid for L.

Given any automaton for a regular language, there are algorithms to construct its syntactic monoid.

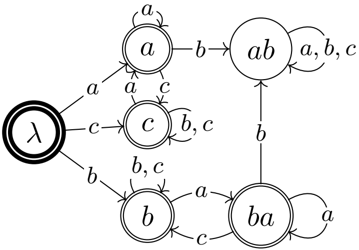
Syntactic Monoids and Semigroups

- The states of the syntactic monoid are elements of a monoid.
  - A semigroup is a set closed under a binary operation $(S, \times)$.
  - A monoid is a semigroup with an identity element $(S, \times, 1)$.

- The product of two elements x, y in the semigroup is determined by the state reached by taking the path labeled y from state x in the syntactic semigroup for L.

$$xy = z \text{ iff } x \overset{y}{\hookrightarrow} z$$

Example: $*ab$ with $\Sigma = \{a, b, c\}$



	a	b	c	ab	ba
a	a	ab	c	ab	ab
b	ba	b	b	ab	ba
c	a	c	c	ab	a
ab	ab	ab	ab	ab	ab
ba	ba	ab	b	ab	ab

A DFA forbidding *ab* substrings induced by the Myhill relation and its corresponding Cayley table.

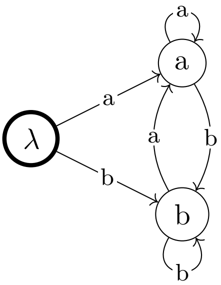
Doubly circled states are accepting and extra thick borders designate initial states.

Definite Languages (Kleene 1956)

A set of strings L is k-definite if and only if there exists $k > 0$ such that for all strings u, v, it holds that if the last k symbols of u and v coincide then either both belong to L or neither belong to L.

$\delta(q, a) = \text{Suff}^{k-1}(qa)$

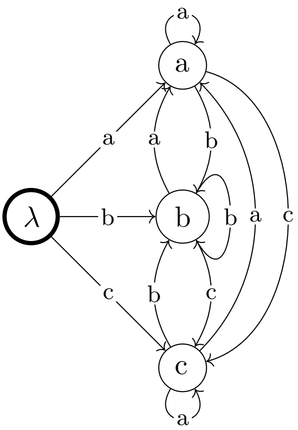
$|\Sigma| = 2, k = 2$



DEFINITE STRUCTURE: A VISUALIZATION

$\delta(q, a) = \mathrm{Suff}^{k-1}(qa)$

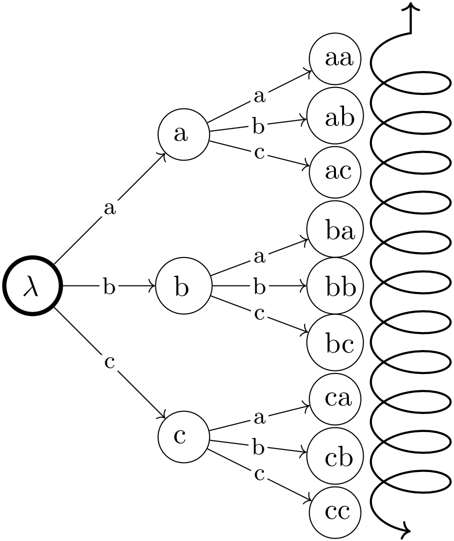
$|\Sigma| = 3, k = 2$



Definite Structure: a Visualization

$\delta(q, a) = \mathrm{Suff}^{k-1}(qa)$

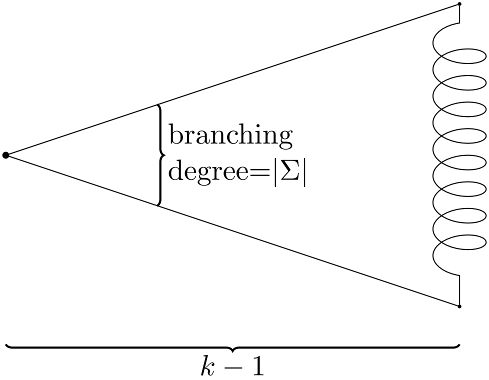
$|\Sigma| = 3, k = 3$



"the loopy part"

Definite Structure: a Visualization

$\delta(q, a) = \text{Suff}^{k-1}(qa)$



Definite Structure: $Se = e$

- An idempotent is an element e in a semigroup S such that $ee = e$.
- Theorem (Brzozowski and Simon, 1973): Syntactic semigroups of definite languages have the property that for all idempotents $e \in S$ and for all $x \in S$, it holds that $xe = e$.
- This is often written $Se = e$ with universal quantification left implicit.

	a	b	c	ab	ba
a	a	ab	c	ab	ab
b	ba	b	b	ab	ba
c	a	c	c	ab	a
ab	ab	ab	ab	ab	ab
ba	ba	ab	b	ab	ab

	T	V	D	VT
T	D	V	D	VT
V	VT	V	D	VT
D	D	V	D	VT
VT	D	V	D	VT

Deciding Definiteness

Input: a finite-state automaton

1. Construct the syntactic monoid.
2. Construct the Cayley Table.
3. Identify the idempotents.
4. Return the answer to this question:
   For all idempotents e, does $Se = e$?

Summary

- The algebraic structure identifies the primitive elements of the system (elements of the semigroup) by distinguishing them in terms of their most basic behaviours, as realized by how multiplication ($\cdot$) works.

- For the definite languages, which are decided by suffixes, any idempotent saturates any suffix, providing the correspondence between the language characterization and the algebraic analysis.

What about functions?

Lifting Nerode and Myhill to Functions

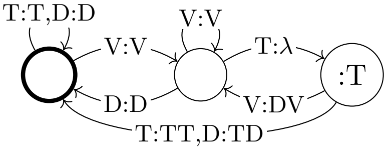
Nerode: $x \overset{N}{\sim} \bar{x}$ iff for all y, v it holds that
$$\langle\ xy,\ \ \text{lcp}(f(x\Sigma^*)v)\ \rangle \in f \Leftrightarrow$$
$$\langle\ \bar{x}y,\ \ \text{lcp}(f(\bar{x}\Sigma^*)v)\ \rangle \in f$$

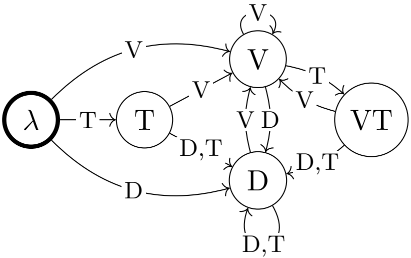
Myhill: $x \overset{M}{\sim} \bar{x}$ iff for all w, y, v it holds that
$$\langle\ wxy,\ \ \text{lcp}(f(wx\Sigma^*)v)\ \rangle \in f \Leftrightarrow$$
$$\langle\ w\bar{x}y,\ \ \text{lcp}(f(w\bar{x}\Sigma^*)v)\ \rangle \in f$$

The lcp is the longest common prefix operator.

Given any deterministic transducer for a sequential function, the same procedures and algorithms are used to construct its syntactic monoid.

Example: Intervocalic Voicing





	T	V	D	VT
T	D	V	D	VT
V	VT	V	D	VT
D	D	V	D	VT
VT	D	V	D	VT

Theorem

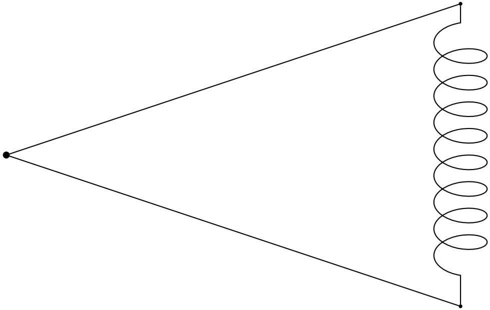
f is a total input strictly local function iff for all idempotents e in the syntactic semigroup S of f, it holds that $Se = e$.

Theorem

f is a ~~total~~ input strictly local function iff for all idempotents e in the syntactic semigroup S of f, it holds that $Se = e$.

Definiteness and Strict Locality

- Acceptors determine acceptance by the **final state**.
- Transducers determine output by the **path**.
- Transducers can define languages with outputs as Booleans values that are conjoined along the path.
  - Definite structure + state acceptance = Def lgs
  - Definite structure + Boolean paths = SL lgs



Other Algebraic Classes and Phonology Lambert 2022

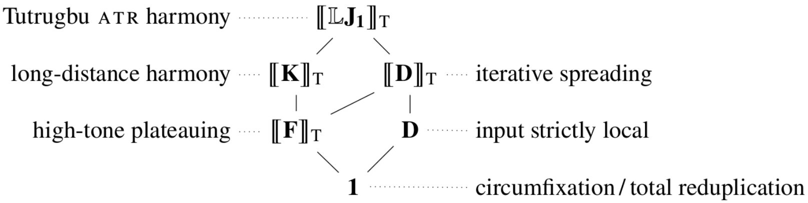


Figure 5.18: Some attested morphophonological functions.

- Both Tutrugbu ATR Harmony and High Tone Plateauing require non-deterministic or 2-way transducers.
- Their algebraic characterizations follow the methods of Carton and Dartois (2015).

Thank you

https://hackage.haskell.org/package/language-toolkit