On the Structure of Sets Testable in the Strict Sense

Dakotah Lambert
Department of Mathematics and Computer Science
Lake Forest College
dakotahlambert@acm.org

Abstract

This paper presents purely algebraic characterizations of the languages locally testable in the strict sense (strictly local) and those piecewise testable in the strict sense (strictly piecewise). These characterizations provide polynomial time algorithms to effectively decide whether a given regular language, presented only as its algebraic structure, belongs to the class: linear time for strictly piecewise and quintic time for strictly local. The techniques described herein are broadly applicable across the subregular hierarchy, admitting a universal approach to deciding membership.

1 Introduction

A longstanding hypothesis in computational phonology is that all phonological surface constraints are regular, and indeed that they require significantly less computational power to recognize or to learn than arbitrary regular languages (Rogers et al., 2013; Heinz, 2018). Each of the many *subregular* language classes provides another perspective on this space; a language within a given class has some additional structure that can facilitate perception and learning (Heinz et al., 2012; Heinz and Rogers, 2013; Rogers et al., 2013; Lambert, 2021; Lambert et al., 2021; Johnson and De Santo, 2024), while languages not in the class do not have this additional structure. A common question regards which single class has enough structure to be effectively learnable but weak enough structure to contain every attested pattern, and much work has gone into the quest to find such a class or to find patterns witnessing that a given class is insufficient (Graf, 2017; Heinz, 2018; Mayer and Major, 2018; De Santo and Graf, 2019; Jardine, 2020). Recently, researchers have investigated the performance of neural networks on various subregular classes, showing a bias toward simpler classes (Bhattamishra et al., 2020; Torres and Futrell, 2023; van der Poel et al., 2024), lending support to the simplicity bias proposed by Lambert et al. (2021).

As abstract algebra is the study of structure, algebraic techniques have long been used by computer scientists to study subregular classes of languages (Straubing, 1985; Tilson, 1987; Almeida, 1995; Pin, 1997; Straubing and Weil, 2021), and these techniques have recently been applied to linguistic study by Lambert (2023) and by Lambert and Heinz (2023, 2024). A semigroup is a set alongside an associative binary operation under which it is closed; this kind of structure naturally represents a formal language when the set is the set of strings over some alphabet Σ and the operation is concatenation. By collapsing strings into equivalence classes that respect the language, each regular and subregular language is associated to a finite semigroup. Many subregular classes are varieties of languages in the sense of Eilenberg (1976), meaning they are completely characterized by equations that hold for every instantiation of variables for all and only the languages in the class (Reiterman, 1982). The Language Toolkit software of Lambert (2024) uses these equations to classify regular languages with respect to a large portion of the subregular hierarchy.

However, to be a variety, a class of languages must be closed under the Boolean operations of union, intersection and complement. Many classes important to the study of natural language patterns are not closed under all of these operations. In this work, we primarily consider the strictly local languages (which operate by forbidding particular substrings, blocks of adjacent symbols) and the strictly piecewise languages (which operate by forbidding particular subsequences, which may have gaps), neither of which is closed under complementation. In order to capture these less-well-behaved classes, one has traditionally turned to external information outside of the semigroup structure. For instance, De Luca and Restivo (1980) first query

whether the language accepts words of the form $\Sigma^* w \Sigma^*$ before applying algebraic techniques in characterizing the strictly local languages. Fu et al. (2011) similarly make membership queries as part of their characterization of the strictly piecewise languages.

We show that the strictly piecewise languages can be captured by imposing an order over the semigroup elements that is compatible with multiplication and captures some amount of membership information while not explicitly querying acceptability. The limited information added by this ordering is not, however, sufficient to capture the strictly local languages. For that, we add additional structure, looking not at individual words under concatenation, but instead at sets of words with both concatenation and union: a *join-semigroup* structure. This is a semigroup augmented with an extra operation that forms a semilattice; the resulting structure is akin to a *semiring* with extra requirements on its addition, except that it need not necessarily have neutral elements. Our primary results are that the strictly piecewise languages are characterized by an inequality over their associated monoid structure:

$$1 \leqslant x$$

where 1 is the neutral element of the monoid, and the strictly local languages are characterized by an inequality over an associated join-semigroup structure:

$$ax^\omega d \leqslant ax^\omega b \vee cx^\omega d.$$

In each of these inequalities, all variables (a, b, c, d, and x) are universally quantified over the elements of the algebraic structure in question. The notation x^ω refers to the unique element that both is a power of x and squares to itself; this is formally defined in §3. Note that $\vee$ here refers not to a logical disjunction but to the least upper bound in the ordering imposed by the semilattice in the join-semigroup.

2 Preliminaries

Given a finite alphabet Σ, the set Σ^* is the set of all finite strings over Σ. A **formal language** L is a subset of Σ^*. The unique empty string is denoted by λ and the set of nonempty strings is $\Sigma^+ = \Sigma^* - \{\lambda\}$. Henceforth, the word "language" is used unambiguously to mean "formal language". A **semigroup** is a set S alongside a binary operation, usually indicated by adjacency, under which the set is closed and associative: $s(tu) = (st)u$ for all s, t and u in S. A **monoid** is a semigroup with a neutral element 1, where $1s = s = s1$ for all elements s. The **free semigroup** (**free monoid**) over Σ is Σ^+ (resp. Σ^*) under concatenation. The neutral element of such a monoid is $1 = \lambda$.

Define the **Myhill relation** for a subset L of a semigroup S such that $x \sim_L y$ means that for every context $u\_\_v$ it holds that $uxv \in L$ if and only if $uyv \in L$, where u, v, x and y are elements in S. If there is any context that distinguishes x and y, then they are not related. The Myhill relation is an equivalence relation: it is reflexive ($x \sim_L x$), symmetric ($x \sim_L y$ implies $y \sim_L x$), and transitive ($x \sim_L y$ and $y \sim_L z$ together imply $x \sim_L z$).

A relation R on a semigroup S is **stable** if and only if for all elements x, y and z of S it holds that whenever $x \; R \; y$ it is also the case that $zx \; R \; zy$ and $xz \; R \; yz$. That is, a stable relation is a relation that is compatible with the operation of the semigroup.

The Myhill relation is a stable equivalence relation. Denote by $[x]_L$ the equivalence class of x under $\sim_L$. One can form a new semigroup $\Sigma^+/\sim_L$ (or monoid, $\Sigma^*/\sim_L$) whose elements are the $[x]_L$ and whose operation is $[x]_L[y]_L = [xy]_L$. This is the **syntactic semigroup**, $S(L)$ (resp. **syntactic monoid**, $M(L)$) of the language L. The syntactic monoid **recognizes** L, in the sense that L is a union of equivalence classes.

A deterministic finite-state automaton is a five-tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ where Q is a finite set of states, Σ is a finite alphabet, $\delta \colon \Sigma \to Q \to Q$ is a transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of accepting states. The automaton is in canonical form if for all states q_1 and q_2 there is some string w such that the path labeled w from q_1 leads to a state in F while that from q_2 does not, or vice versa. It is trimmed if the transition function is made partial by removing any states from which no path leads to a state in F.

3 Varieties of Sets and Structures

In this section we present the standard definitions of varieties of sets and pseudovarieties of algebraic structures, which we will refer to simply as varieties, and their importance in the study of formal languages. We briefly demonstrate that these notions are not equipped to handle the classes of languages under discussion in this work.

A class $\mathcal{C}$ of languages associates with each finite alphabet Σ the set $\Sigma\mathcal{C}$ of languages $L \subseteq \Sigma^*$

that are in the class. A $*$**-variety** of languages is a class $\mathcal{V}$ such that each $\Sigma\mathcal{V}$ is a Boolean algebra (*i.e.* it is closed under finite unions, finite intersections, and complementation) and is closed under both left and right quotients in the sense of Brzozowski (1964). Further, if $\varphi : \Gamma^* \to \Sigma^*$ is a homomorphism (a function such that $\varphi(xy) = \varphi(x)\varphi(y)$ for all x and y and where $\varphi(1) = 1$ where applicable) and L is a language in $\Sigma\mathcal{V}$, then $\varphi^{-1}(L)$ must be in $\Gamma\mathcal{V}$; the class is closed under inverse images of homomorphisms. A $+$**-variety** is exactly the same, except that φ is a nonerasing homomorphism from Γ^+ to Σ^+. One often uses $+$-varieties when dependencies are local, as, if φ is allowed to erase symbols, then φ^{-1} can freely insert those same symbols, destroying any semblance of locality.

A **pseudovariety of monoids** is a set $\mathbf{V}$ of finite monoids that is closed under finitary direct products, quotients, and inverse images of homomorphisms. A **pseudovariety of semigroups** is defined analogously. The "pseudo-" in pseudovariety regards the fact that we are considering only finite structures. Henceforth, the prefix shall be omitted and we will say simply "variety". The direct product of S and T, written $S \times T$ is a semigroup whose elements are of the form $\langle s, t\rangle$ for $s \in S$ and $t \in T$. Multiplication is pointwise, so $\langle s_1, t_1\rangle\langle s_2, t_2\rangle = \langle s_1 s_2, t_1 t_2\rangle$ If both S and T are monoids, then so too is $S \times T$. A semigroup $\langle T, \cdot\rangle$ is a **subsemigroup** of a semigroup $\langle S, \cdot\rangle$ with the same operation iff $T \subseteq S$ and T is closed under the semigroup operation. Finally, a semigroup T divides a semigroup S if there is some equivalence relation $\sim$ such that for some subsemigroup U of S it holds that $T = U/\sim$.

Eilenberg's theorem (1976, Theorems 3.2–3.4) states that there is a 1–1 correspondence between varieties of monoids and $*$-varieties of languages, and that there is a 1–1 correspondence between varieties of semigroups and $+$-varieties of languages. Reiterman's theorem (1982, Theorem 3.1) states that varieties of semigroups and monoids can be characterized by systems of universally-satisfied equations over profinite words $u = v$. This means that, given the syntactic semigroup or monoid of a language, one can determine whether it belongs to a given variety by merely checking that the equations are satisfied for all (finitely many) instantiations of its variables. If a system includes k variables, then this method decides membership in time $O(n^k)$ where n is the number of elements in the structure.

The variables in the equations can usually be treated as words, but in reality they represent a completion of this space. This work will not discuss topology in depth, but we must introduce one small notion: each element in the equations is the limit of a Cauchy sequence of words. A sequence is **Cauchy** if for any given positive distance, all but finitely many elements of the sequence are within that distance from one another. Following Almeida (1995), we consider the distance between two distinct words to be $2^{-r(w_1,w_2)}$, where $r(w_1, w_2)$ is the number of elements in the smallest monoid that separates w_1 and w_2, and the distance between a word and itself to be zero. This means that words that are only distinct in large monoids are "closer" than words that can be distinguished by small monoids. In the context of regular languages, whose syntactic monoids M are finite, this further means that Cauchy sequences of words map to sequences of Myhill classes (semigroup elements) that are eventually constant, as words with distance below $2^{-|M|}$ cannot be distinguished by M.

The simplest Cauchy sequence of words is x representing the limit of the constant sequence $a_n = x$, where all words are at distance 0 from each other. The sequence $a_n = x^{n!}$ is also Cauchy;[1] its limit is often denoted by x^ω. In any finite semigroup, x^ω is the unique element that is both a positive power of x and **idempotent** (it squares to itself). Oftentimes, the notion of "for all sufficiently long words" is useful; in these instances one turns to elements of the form x^ω.

For example, per Almeida (1989), the piecewise testable languages are all and only those whose syntactic monoids universally satisfy the equation $(xy)^\omega x = (xy)^\omega = y(xy)^\omega$ (or, alternatively both $x^\omega x = x^\omega$ and $(xy)^\omega = (yx)^\omega$). And the locally testable languages are all and only those whose syntactic semigroups universally satisfy both $x^\omega y x^\omega z x^\omega = x^\omega z x^\omega y x^\omega$ and $x^\omega y x^\omega = x^\omega y x^\omega y x^\omega$ (Straubing, 1985). The variety induced by a set of equations is denoted by the equations separated by semicolon (;) within double-brackets $[\![\ldots]\!]$. That is, we say that the variety corresponding to piecewise testable is

$$\mathbf{J} = [\![(xy)^\omega = (yx)^\omega; x^\omega x = x^\omega]\!]_{(*)}$$

[1] Note that $a_n = x^n$ is not Cauchy, as any infinite subsequence will contain words whose length modulo $2|x|$ is 0 and words whose length modulo $2|x|$ is $|x|$, with a distance of at least $2^{-2|x|}$.

and the variety corresponding to locally testable is

$$\mathbf{J_1} * \mathbf{D} = [\![x^\omega y x^\omega z x^\omega = x^\omega z x^\omega y x^\omega; \\ x^\omega y x^\omega = x^\omega y x^\omega y x^\omega]\!]_{(+)}$$

See Straubing (1985) for more information on varieties of the form $\mathbf{V} * \mathbf{D}$; this $*$ is a product of varieties, the details of which exceed the scope of the present work. The "$(*)$" and "$(+)$" subscripts are not standard notation but are used here to distinguish between monoid varieties and semigroup varieties.

However, none of these results can immediately apply in our situation. A language L and its complement $\overline{L}$ share the same syntactic semigroup (or monoid), which means that if $S(L) \in \mathbf{V}$ it must hold that $S(\overline{L}) \in \mathbf{V}$. But the language classes considered in this work are not closed under complementation, so no variety can possibly characterize them. In later sections, we discuss how imposing additional structure allows us to capture classes that do not have all of the closure properties required to be a variety.

4 Strictly Piecewise Languages

In order to reason about long-distance dependencies that arise in the study of natural language constraints, Rogers et al. (2010) examined the languages **piecewise testable in the strict sense**, also called **strictly piecewise**. Per Heinz (2007, 2010a), these languages capture aspects of long-distance harmony patterns that are attested in natural language, such as the sibilant harmony of Tsuut'ina. The following examples taken from Cook (1978) and Li (1930) demonstrate that, in Tsuut'ina, a [+anterior] sibilant like [s] cannot precede [−anterior] sibilants like [ʃ] at any distance, although the reverse is not true.

1.	/sì-tʃʊ́gɒ̀/	ʃìtʃʊ́gɒ̀	'my flank'
	/sì-tsáɣà/	sìtsáɣá	'my hair'
	/gi-si-tˀʃast/	giʃitˀʃast	'they are piled up'

The strictly piecewise languages are defined in terms of a grammar. Let Σ be a fixed finite alphabet. A (negative) **precedence grammar** G is a finite set of finite words over Σ. A word $x = x_1 x_2 \dots x_n$ is a **subsequence** of another word w, written $x \sqsubseteq w$, if and only if $w = u_0 x_1 u_1 \dots x_n u_n$ for strings u_i. The language of the grammar G is the set of words $w \in \Sigma^*$ such that $x \sqsubseteq w \Rightarrow x \notin G$. A language is **piecewise testable in the strict sense** (**strictly piecewise**) if and only if it is the language of a precedence grammar. If the longest word in the precedence grammar is at most length k, we say that its language is piecewise k-testable in the strict sense, or alternatively, strictly k-piecewise. Rogers et al. (2010) provide several other characterizations of the class, but the most useful to us here is that the strictly piecewise languages are all and only those that are closed under taking of subsequences.

4.1 Known Decision Procedures

Several algorithms have been proposed to determine whether a given regular language is strictly piecewise, and, perhaps, to obtain its grammar if it is.

Rogers et al. (2010) provide a simple grammar-based decision procedure. The input is presented in the form of a deterministic finite-state automaton in canonical form. First, they show that if a language is properly strictly k-piecewise, then it must contain at least k states. Then, by enumerating all possible grammars whose longest word is at most length k, each can be converted to a finite-state automaton and compared against the original language. If one is equal, then that grammar witnesses the fact that the language is strictly piecewise. Otherwise, the language is not strictly piecewise. This algorithm runs in time $\Theta(|\Sigma|^{|Q|})$, where Σ is the alphabet and Q is the set of states in the input automaton.

Rogers and Lambert (2019a) construct the smallest strictly piecewise superset of the given language by allowing edges in the automaton to be skipped, and test whether the resulting language is equal to the original. This too runs in exponential time, as the construction invokes nondeterminism and the result must be determinized before comparison.

Fu et al. (2011) use some tools from algebra and state that a language is strictly piecewise if and only if it is "wholly nonzero" and "right annihilating". A **zero** in a semigroup is an element 0 such that $0x = 0 = x0$ for all x. Let L be a language and let $\eta : \Sigma^* \to M(L)$ be the **syntactic morphism** mapping $w \mapsto [w]_L$. Fu et al. (2011, Definition 5) define that a language is "wholly nonzero" if and only if $\eta(\overline{L}) = 0$. However, the syntactic monoid for the strictly piecewise language Σ^* has $\eta(L) = 0$ while $\eta(\overline{L})$ does not exist.[2] Per-

[2] Fu et al. (2011, Corollary 2) also claim that strictly local languages are wholly nonzero. We describe strictly local languages in a later section, but at this point it is important to note that the language of words that begin with "a" is strictly

haps a better phrasing would be that L is wholly nonzero if and only if $w \in \overline{L} \Rightarrow \eta(w) = 0$. Fu et al. (2011, Definition 7) define that a language is "right annihilating" if and only if its monoid satisfies $xa = 0 \Rightarrow xya = 0$. Modulo the trivial languages Σ^* and $\varnothing$, this provides a mechanism by which one can decide in $O(n^3)$ time whether a given monoid represents a strictly piecewise language, assuming that the elements are marked for whether they contain accepted words or rejected words.

In the following sections, we present a different algebraic characterization that admits a linear-time decision procedure and more closely resembles how varieties of languages are defined.

4.2 Closure Properties

In this section we explore the closure properties of the class of strictly piecewise languages and see where they fail to be a variety. From there, we present a piece of additional structure that allows algebraic techniques to make the distinctions necessary in order to characterize the class. Finally, we present a characterization that admits a decision procedure that operates in linear time when given the appropriate kind of input. Recall that to be a variety, a class $\mathcal{C}$ must be such that each $\Sigma\mathcal{C}$ is a Boolean algebra closed under left- and right-quotients, and the class must be closed under taking inverse-images of homomorphisms.

First, the strictly piecewise languages over Σ are not a Boolean algebra. They are not closed under complementation. If Σ is nonempty, the language containing only the empty string $L = \{\lambda\}$ is strictly piecewise, but its complement is not, as $\lambda \notin \overline{L}$ despite the fact that $\lambda \sqsubseteq w$ for all w. They are, however, closed under both union and intersection. Consider the union. Let L_1 and L_2 be strictly piecewise languages and consider their union $L_1 \cup L_2$. Let $w \in L_1 \cup L_2$ and let $x \sqsubseteq w$. Then either $w \in L_1$ or $w \in L_2$. In the case that $w \in L_1$, because L_1 is strictly piecewise, it follows that $x \in L_1$ and therefore $x \in L_1 \cup L_2$. Alternatively, $w \in L_2$ and, because L_2 is strictly piecewise, it follows that $x \in L_2$ and therefore $x \in L_1 \cup L_2$. In either case, the union is closed under subsequence and is therefore strictly piecewise. Next consider the intersection $L_1 \cap L_2$. Let $w \in L_1 \cap L_2$ and let $x \sqsubseteq w$. Then $w \in L_1$ and $w \in L_2$, and as each is strictly piecewise, it follows that $x \in L_1$ and $x \in L_2$, so $x \in L_1 \cap L_2$. The intersection is closed under subsequence and is therefore strictly piecewise. As they are closed under both union and intersection, the strictly piecewise languages over Σ are a **positive Boolean algebra**.[3]

The strictly piecewise languages over Σ are closed under left- and right-quotients in the sense of Brzozowski (1964). The left quotient of a language L over Σ by a symbol $a \in \Sigma$ is

$$a^{-1}L = \{w \in \Sigma^* : aw \in L\}$$

and the right quotient is

$$La^{-1} = \{w \in \Sigma^* : wa \in L\}.$$

Let $a \in \Sigma$, let w be in $a^{-1}L_1$, and let $x \sqsubseteq w$. Then $aw \in L_1$ and because $x \sqsubseteq w$, and therefore $ax \sqsubseteq aw$, it follows that $ax \in L_1$ and $x \in a^{-1}L_1$. Similarly, if $w \in L_1a^{-1}$ then $x \in L_1a^{-1}$. Both quotients are closed under subsequence and thus strictly piecewise.

Finally, the strictly piecewise languages are closed under inverse images of homomorphisms. Let $\varphi\colon \Gamma^* \to \Sigma^*$ be a **homomorphism**, a function where $\varphi(xy) = \varphi(x)\varphi(y)$. Also let $L \subseteq \Gamma^*$ be the language

$$L = \varphi^{-1}(L_1) = \{w : \varphi(w) \in L\}.$$

Let $w \in L$ and let $x \sqsubseteq w$. This means that $x = x_1 \dots x_n$ for some n and there exist $u_0 \dots u_n$ such that $w = u_0x_1u_1 \dots x_nu_n$. We have that $w \in \varphi^{-1}(L_1)$ and so it follows that $\varphi(w) = \varphi(u_0)\varphi(x_1)\varphi(u_1) \dots \varphi(x_n)\varphi(u_n)$ is in L_1. As L_1 is closed under subsequence, the subsequence $\varphi(x_1)\varphi(x_2) \dots \varphi(x_n) = \varphi(x)$ of $\varphi(w)$ is also in L_1, so $x \in \varphi^{-1}(L_1)$. The inverse image of φ is closed under subsequence and therefore it is strictly piecewise.

Definition 1 (Pin, 1997)**.** A **positive $*$-variety of languages** is a class that is closed under taking inverse images of homomorphisms and assigns to each alphabet Σ a set of languages that is a positive Boolean algebra closed under the left- and right-quotient operations.

Lemma 1. *The strictly piecewise languages form a positive $*$-variety.*

local but is decidedly *not* wholly nonzero whenever $\{a\} \subsetneq \Sigma$, as its syntactic monoid does not even have a zero.

[3] The term "positive Boolean algebra" is common in French treatments of formal language theory. Another name for a Boolean algebra without complementation (a structure that is isomorphic to a class of sets that is closed under finitary union and intersection) is a "distributive lattice".

4.3 Ordered Semigroups and Monoids

A **preorder** is a relation $\leqslant$ that is both reflexive and transitive. Let S be a semigroup. An **ordered semigroup** is a semigroup equipped with a stable preorder. Every semigroup may be treated as an ordered semigroup under the trivial order where $x \leqslant y$ means $x = y$.

The **syntactic order** of a language L is the ordering defined such that $[s] \leqslant [t]$ if and only if $utv \in L \Rightarrow usv \in L$ for all strings u and v in Σ^* (Pin, 1997).[4] This is a stable preorder. The **syntactic ordered semigroup** of L is its syntactic semigroup endowed with the syntactic order.

Varieties of ordered semigroups and monoids are defined analogously to their unordered counterparts. Pin (1995, Theorems 5.7 and 5.8) citing to Bloom (1976) extends Eilenberg's theorem to provide a 1–1 correspondence between varieties of ordered monoids (ordered semigroups) and positive $*$-varieties (resp. positive $+$-varieties) of languages. Pin and Weil (1996, Theorem 3.3) extend Reiterman's theorem to show that varieties of ordered monoids or semigroups are characterized by universally satisfied sets of inequalities. In either case, a structure belongs to the variety if and only if the equations or inequalities are satisfied for every instantiation of the variables involved.

As we have shown that the strictly piecewise languages are a positive $*$-variety of languages, this means that there is a corresponding variety of ordered monoids.

Theorem 1. *Let L be a language. The following are equivalent.*

- *L is strictly piecewise*
- *$M(L)$ is a finite ordered monoid in the variety $[\![1 \leqslant x]\!]_{(*)}$*

Proof. First, we show that the syntactic ordered monoid of any strictly piecewise language is finite and satisfies the inequality $1 \leqslant x$ for all x. Finiteness is trivial, as strictly piecewise languages are regular and the defining characteristic of a regular language is that it has a finite syntactic monoid (Rabin and Scott, 1959). Let L be a strictly piecewise language over Σ whose syntactic ordered monoid is $M(L)$. Further, let $uxv \in L$ for strings u and v in Σ^*. Then, as $uv \sqsubseteq uxv$ we have that $uv \in L$ as well. By the definition of the syntactic order, we have that $1 \leqslant x$.

Next, we show that any language whose syntactic ordered monoid is finite and satisfies the inequality $1 \leqslant x$ for all x must be strictly piecewise. Let L be a language whose syntactic monoid $M(L)$ is finite and universally satisfies the inequality and let $\eta : \Sigma^* \to M(L)$ be the syntactic morphism. Let $x = x_1 \dots x_n$ and $w = u_0 x_1 u_1 \dots x_n u_n$ be strings such that $x \sqsubseteq w$. Because $\leqslant$ is stable we have the following.

$$\begin{aligned}\eta(x) &= \eta(x_1)\eta(x_2)\dots\eta(x_n) \\ &\leqslant \eta(u_0)\eta(x_1)\eta(u_1)\dots\eta(x_n)\eta(u_n) \\ &= \eta(w)\end{aligned}$$

By the definition of the syntactic order, then, we have that whenever $uwv \in L$ it is also the case that $uxv \in L$. By instantiating $u = v = \lambda$ we have $w \in L \to x \in L$. Thus, L is closed under subsequence and is strictly piecewise. □

Recall that two strings x and y map to the same equivalence class if and only if in all contexts $u\_\_v$ it holds that both $uxv \in L$ and $uyv \in L$, or neither is. This means that L and its complement, $\overline{L}$, share the same syntactic monoid. However, their syntactic *ordered* monoids are duals: whenever $[x] \leqslant [y]$ in $M(L)$, we have that for all u and v it holds that $uyv \in L \to uxv \in L$. By contrapositive, we have $uxv \notin L \to uyv \notin L$, so $[y] \leqslant [x]$ in $M(\overline{L})$.

Corollary. *A language L is the complement of a strictly piecewise language if and only if its ordered syntactic monoid is in the variety $[\![x \leqslant 1]\!]_{(*)}$.*

Corollary. *If a language and its complement are both strictly piecewise, then for all x, it holds that $1 \leqslant x \leqslant 1$, i.e. $x = 1$. The only such languages are the empty set and its complement.*

As a result of this characterization, there is an algorithm that decides in linear time whether a given regular language is strictly piecewise (or the complement of a strictly piecewise language), when the language is presented as its ordered syntactic monoid, or even as its syntactic order alone. For each element x, compare x with 1. If for all comparisons, $1 \leqslant x$, then the language is strictly piecewise. If for all comparisons, $x \leqslant 1$, then the complement of the language is strictly piecewise.

[4] Pin (2017) states that this definition was "regrettable" and that the inverse order should be used instead. However, the choice of orientation does not affect the results, and using $\leqslant$ in this sense connects more strongly to the additional structure imposed in the next section for strictly local languages. All this means is that one must be careful to consult the definitions used in any given work and to swap the roles of $\leqslant$ and $\geqslant$ where appropriate.

This class has been studied before under many names. The strictly piecewise languages that we know and love are known by Pin (1997) as "the complements of languages at the one-half level of the Straubing–Thérien hierarchy" and the order-theoretic characterization of this class is presented in that work. And before that, Haines (1969) also studied the languages closed under subsequence (and their complements: those closed under supersequence).

It is also worth noting that every variety of languages is a positive variety of languages, and every variety of semigroups (monoids) is a variety of ordered semigroups (resp. monoids) where identities of the form $x = y$ are replaced by the pair of inequalities $x \leqslant y$ and $y \leqslant x$. Therefore this structure provides sufficient information to capture varieties such as the piecewise testable and locally testable languages as well.

5 Strictly Local Languages

Just as the strictly piecewise languages are a restriction of the piecewise testable languages, the strictly local languages are a restriction of the locally testable languages. Introduced by McNaughton and Papert (1971) the **languages locally k-testable in the strict sense** (**strictly k-local**) are defined by a grammar $G \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$: a word w is accepted if and only if every sliding window of size k of $\rtimes w \ltimes^{k-1}$ is in G (Rogers and Pullum, 2011). A language is **strictly local** if and only if it is strictly k-local for some k. Per Edlefsen et al. (2008), approximately 75% of the more than one hundred distinct patterns governing stress assignment in natural language catalogued by Goedemans et al. (2015) are strictly local; Rogers and Lambert (2019a) found that 92.5% of these patterns were covered by a conjunction of a strictly piecewise constraint, a strictly local constraint, and a constraint whose complement is strictly local. The class handles restrictions on prefixes, on suffixes, and on what symbol clusters are permissible.

As reported by Edlefsen et al. (2008) and by Rogers and Pullum (2011), another characterization of the strictly local languages, adapted from McNaughton (1974), is **suffix substitution closure**: L is a strictly k-local language if and only if whenever $axb \in L$ and $cxd \in L$ are strings such that $|x| \geqslant k - 1$, it also holds that $axd \in L$.

5.1 Known Decision Procedures

The strictly local languages are all and only those in which all sufficiently long words w are **constant** in the canonical finite-state automaton: there exists a state q' such that for all q, w maps into the set $\{q', 0\}$, where 0 is the unique rejecting sink state, if any (De Luca and Restivo, 1980).

This means that, if the sink state is removed, then there will be no cycles involving states that correspond to sets of size greater than one in the automaton that results from applying the powerset construction used for determinization with an initial state corresponding to the set of all original states. This construction was used by Rogers and Lambert (2019a) to facilitate both the decision procedure and to recover the grammar, if one exists. This procedure runs in exponential time if presented with the trimmed canonical automaton, or in linear time if given the powerset graph.

Edlefsen et al. (2008) present a modification of this algorithm from Caron (2000) that runs in quadratic time when given the trimmed canonical automaton: consider only the graph generated by states that correspond to pairs in the original automaton. As it is deterministic, no larger sets will ever be generated.

De Luca and Restivo (1980) use the constancy property to show that a language that does not contain two-sided ideals $\Sigma^* w \Sigma^*$ is strictly local if and only if for any idempotent x^ω of its syntactic semigroup, $x^\omega y x^\omega \subseteq \{x^\omega, 0\}$. (If there is no 0, this becomes $x^\omega y x^\omega = x^\omega$.) As a two-sided ideal contains every possible length-k substring, the only strictly local language to contain such ideals is Σ^*, so this also results in an effective characterization. This subset property can be checked in quadratic time when presented with a syntactic semigroup.

All of these techniques employ information not contained in the algebraic structure of the language. In the next section, we demonstrate why even ordered semigroups cannot characterize the strictly local languages, then we present a related structure that does contain sufficient information to capture the class in a purely algebraic way. The resulting characterization strongly resembles the statement of the suffix substitution closure property.

5.2 Closure Properties

The strictly local languages over a fixed alphabet Σ are not a Boolean algebra nor even a positive Boolean algebra, as they are not closed under com-

plementation nor are they closed under union. To witness lack of closure under complementation, consider the set $L_1 = \Sigma^* - \Sigma^* ab \Sigma^*$, the strictly 2-local language in which the ab substring does not occur. Its complement then, is $\Sigma^* ab \Sigma^*$, but as discussed in the preceding section, the only strictly local language to contain a two-sided ideal is Σ^* itself. Every possible substring appears in $\overline{L_1}$, so every word must be accepted for this to be strictly local, but bb is not. The reversal of L_1 is $L_2 = \Sigma^* - \Sigma^* ba \Sigma^*$, another strictly 2-local language, in which the ba substring does not occur. To witness lack of closure under union, we consider $L_1 \cup L_2$. This is the language in which either there is no ab or there is no ba; accepted words are all and only those that do not contain both ab and ba. This does not satisfy suffix substitution closure, as $a(b^k)b \in L$ and $b(b^k)a \in L$ but $a(b^k)a \notin L$ for all $k \geqslant 1$.

This means that the strictly local languages cannot be captured by a variety of (ordered) semigroups. A variety of semigroups is insufficient because a language and its complement share the same semigroup. A variety of ordered semigroups is insufficient because, while a language and its complement can be distinguished, the inequalities used entail closure under union. Recall that $x \leqslant y$ means that $uyv \in L \Rightarrow uxv \in L$ for all $u, v \in \Sigma^*$. Let L_1 and L_2 be languages universally satisfying the inequality and let u, v, and y be strings such that $uyv \in L_1 \cup L_2$. Then either $uyv \in L_1$ and by the inequality we have $uxv \in L_1$, or $uyv \in L_2$ and by the inequality we have $uxv \in L_2$. In either case, $uxv \in L_1 \cup L_2$, and the inequality is satisfied.

However, the strictly local languages over Σ are closed under intersection. Let L_1 and L_2 be any strictly k-local languages over Σ, and let axb and cxd be words in $L_1 \cap L_2$ such that $|x| \geqslant k - 1$. Then, because each of axb and cxd are in each of L_1 and L_2, it follows by suffix substitution closure that axd is in both L_1 and L_2. That is, $axd \in L_1 \cap L_2$ and suffix substitution closure holds. The result is strictly k-local.

The class is also closed under left and right quotients. Let L be a strictly k-local language over Σ and let axb and cxd be words in $s^{-1}L$ for some $s \in \Sigma$ where $|x| \geqslant k - 1$. Then $saxb$ and $scxd$ are in L, which by suffix substitution closure means that $saxd \in L$ and $axd \in s^{-1}L$. Suffix substitution closure holds, so $s^{-1}L$ is strictly k-local. Similarly, Ls^{-1} is strictly k-local.

The class is not closed under inverse images of arbitrary homomorphisms; indeed the tier-based strictly local languages are defined as the inverse image of a strictly local language under a particular kind of erasing homomorphism (Heinz et al., 2011; Lambert, 2023). But the class is closed under inverse images of nonerasing homomorphisms. Let $\varphi\colon \Gamma^+ \to \Sigma^+$ be a (nonerasing) homomorphism Also let L be a strictly k-local language over Σ^*, and let axb and cxd be words in $\varphi^{-1}(L)$ such that $|x| \geqslant k - 1$. Then $\varphi(axb) = \varphi(a)\varphi(x)\varphi(b) \in L$ and $\varphi(cxd) = \varphi(c)\varphi(x)\varphi(d) \in L$, and because φ is nonerasing, $|\varphi(x)| \geqslant |x| \geqslant k - 1$. By suffix substitution closure, this means that $\varphi(axd) = \varphi(a)\varphi(x)\varphi(d) \in L$ and therefore $axd \in \varphi^{-1}(L)$. This demonstrates closure under inverse images of nonerasing homomorphisms.

Indeed, the strictly local languages satisfy a stronger condition. The following is adapted from Polák (2001). For a set S define $S^{\square}$ to be the set of all nonempty finite subsets of S^*, and $S^{\boxplus}$ to be the set of all nonempty finite subsets of S^+. Let $\psi\colon \Gamma^{\boxplus} \to \Sigma^{\boxplus}$ be a (nonerasing) homomorphism where the operation is concatenation lifted to sets, and define for $L \subseteq \Sigma^*$ an inverse:

$$\psi^{[-1]}(L) = \{w \in \Gamma^+ : \psi(\{w\}) \subseteq L\}$$

Let $L \subseteq \Sigma^*$ be strictly k-local, and let axb and cxd be words in $\psi^{-1}(L)$ such that $|x| \geqslant k-1$. Observe the following (omitting $\psi(\{w\})$ if $w = \lambda$):

$$\psi(\{axb\}) = \psi(\{a\})\psi(\{x\})\psi(\{b\}) \subseteq L$$
$$\psi(\{cxd\}) = \psi(\{c\})\psi(\{x\})\psi(\{d\}) \subseteq L$$

As every word in $\psi(\{x\})$ is at least as long as x, it holds that by suffix substitution $\psi(\{a\})\psi(\{x\})\psi(\{d\}) \subseteq L$ and therefore it holds that $axd \in \psi^{[-1]}(L)$. We have that $\psi^{[-1]}$ is strictly k-local.

Definition 2 (Polák, 2001)**.** A **conjunctive $+$-variety of languages** is a class that is closed under $\psi^{[-1]}$ for homomorphisms of the form $\psi\colon \Gamma^{\boxplus} \to \Sigma^{\boxplus}$ and that assigns to each alphabet Σ a set of languages that is closed under intersection and the left- and right-quotient operations.

Lemma 2. *The strictly local languages form a conjunctive $+$-variety.*

A **conjunctive $*$-variety** is defined analogously, using $^{\square}$ in place of $^{\boxplus}$.

5.3 Join-Semigroups

The following concepts are adapted from Polák (2001) and Klíma and Polák (2019). A **join-**

semigroup is a structure $\langle S, \cdot, \vee \rangle$ such that $\langle S, \cdot \rangle$ is a semigroup and $\langle S, \vee \rangle$ is a join-semilattice (a commutative semigroup where all elements are idempotent) alongside a distributive property: for all x, y, and z in S it holds that $x(y \vee z) = xy \vee xz$ and $(x \vee y)z = xz \vee yz$.[5] A join-monoid is a join-semigroup where $\langle S, \cdot \rangle$ is a monoid.

The usual ordering relation for a join-semilattice induces an order on the structure: $x \leqslant y$ means $x \vee y = y$. This is stable under multiplication. Let $x \leqslant y$ and let a be an element. Then $x \vee y = y$ and so $a(x \vee y) = ay$. By the distributive property, we have $ax \vee ay = ay$ and thus $ax \leqslant ay$. Similarly, $xa \leqslant ya$. The order is clearly also stable under join, as $a \vee (x \vee y) = a \vee y$ and $(x \vee y) \vee a = y \vee a$.

The **syntactic join-semigroup** of a language $L \subseteq \Sigma^*$, is the join-semigroup $\langle \Sigma^{\boxplus}, \cdot, \cup \rangle / \approx_L$, where $A \approx_L B$ means that for all sets U and V in $\Sigma^{\square}$ it holds that $UAV \subseteq L$ if and only if $UBV \subseteq L$. This is a set-based analogy to the Myhill relation. As before, $\Sigma^{\square}$ and $\Sigma^{\boxplus}$ are defined to be the set of nonempty finite subsets of Σ^* and Σ^+, respectively. The **syntactic join-monoid** is defined analogously. Note that for a regular language, this structure is finite, as one can consider the elements $\eta(w)$ of the syntactic semigroup or monoid rather than the strings w themselves.

Polák (2001) extends Eilenberg's and Reiterman's theorems to provide a 1–1 correspondence between varieties of join-monoids (join-semigroups) and conjunctive $*$-varieties (resp. conjunctive $+$-varieties) of languages and to provide a means for characterization by a system of inequalities. The inequalities are of the form $x \leqslant y_1 \vee y_2 \vee \cdots \vee y_n$ for $n \geqslant 1$, and can be interpreted as requiring that whenever it holds for all strings u and v that $uy_iv \in L$ for all $1 \leqslant i \leqslant n$, it also holds that $uxv \in L$.

As we have shown that the strictly local languages are an conjunctive $+$-variety of languages, this means that there is a corresponding variety of join-semigroups. To get there, we first must show that all sufficiently long words can be written in terms of an idempotent.

Lemma 3. *Let S be a finite semigroup and let x be a sequence of elements of S of length $k > |S|$. Then $x = ae^{\omega}b$ for some a, b and c in S.*

Proof. If any individual x_i is idempotent, then $x = (x_1 \dots x_{i-1})(x_i)(x_{i+1} \dots x_k)$ and we are done. Otherwise, consider the prefixes $x_{[i]} = x_1 \dots x_i$. By the pigeonhole principle, there exist $i < j$ such that $x_{[i]} = x_{[j]} = x_{[i]}(x_{i+1} \dots x_j)$. Let $a = x_{[i]}$ and $e = x_{i+1} \dots x_j$. By iteratively expanding, we have $a = ae = aee = \cdots = ae^{\omega}$. Finally, let $b = x_{j+1} \dots x_k$ if $j < k$ else e^{ω}. We then have $x = ae^{\omega}b$. □

Theorem 2. *Let L be a language. The following are equivalent.*

- *L is strictly local*
- *The syntactic join-semigroup of L is in the variety $[\![ax^{\omega}d \leqslant ax^{\omega}b \vee cx^{\omega}d]\!]_{(+)}$*

Proof. First, we show that the syntactic join-semigroup of any strictly local language is finite and satisfies the inequality $ax^{\omega}d \leqslant ax^{\omega}b \vee cx^{\omega}d$. Finiteness is trivial, as strictly local languages are regular. Let L be a strictly local language over Σ. Then there is some k for which L is strictly k-local. Further, let $uaebv$ and $ucedv$ belong to L for strings a, b, e, u, and v where $\eta(e) = \eta(e)\eta(e)$. That is, $\eta(e)^{\omega} = \eta(e)$. Then $ua(e^k)bv$ and $uc(e^k)dv$ are in L, as they are Myhill-equivalent to $uaebv$ and $ucedv$, respectively. By suffix-substitution, it follows that $ua(e^k)dv \in L$, that $uaedv \in L$. This demonstrates satisfaction of the inequality.

Next, we show that any language whose syntactic join-semigroup, Z, is finite and satisfies the inequality must be strictly local. Specifically, we show that it is strictly k-local for $k = |Z| + 2$. Let L be a regular language with finite syntactic join-semigroup Z satisfying the inequality, and let axb and cxd be words in L such that $|x| \geqslant k - 1$. In the case that $\eta(x) = \eta(x)^{\omega}$, the inequality directly guarantees that $axd \in L$. It is not necessarily the case that all long words are themselves idempotent, but, by Lemma 3, we can rewrite x as x_1ex_2 where $\eta(e) = \eta(e)^{\omega}$ to find that $ax_1ex_2d \in L$. This means that $axd \in L$, that suffix substitution closure is satisfied, and hence L is strictly k-local. □

A consequence of this is an effective algorithm for deciding whether a language is strictly local: for all instantiations of the five variables a, b, c, d and x^{ω}, where x^{ω} is restricted to idempotents, verify that $ax^{\omega}d \leqslant ax^{\omega}b \vee cx^{\omega}d$. This is not a particularly efficient algorithm; naïvely it runs in quintic

[5] Polák (2001) used the term *idempotent semiring*, with $\vee$ being $+$, but this is an abuse of terminology, as, properly, the definition of a semiring requires that both operations impose a monoid structure, with the neutral element of addition being 0 such that $0x = 0 = x0$ for all x. However, there appears to be no standard terminology for two interacting *semigroups* rather than two interacting *monoids*.

time in the size of the syntactic join-semigroup, as there are five variables. However, it does provide a purely algebraic characterization of the strictly local languages, unifying their analysis with that of other classes in the piecewise-local subregular hierarchy.

It also allows for a new proof of the obvious fact that strictly local languages are locally testable.

Proposition 1. *Let L be a language whose syntactic join-semigroup Z belongs to*

$$[\![ax^\omega d \leqslant ax^\omega b \vee cx^\omega d]\!]_{(+)},$$

i.e. let L be strictly local. Then L is locally testable, i.e. Z belongs to

$$\mathbf{J_1 * D} = [\![x^\omega yx^\omega zx^\omega = x^\omega zx^\omega yx^\omega; \\ x^\omega yx^\omega = x^\omega yx^\omega yx^\omega]\!]_{(+)}.$$

Proof. Let L be a strictly local language and let Z be its syntactic join-semigroup. In the following analyses, for clarity, we will parenthesize the shared idempotent that enables substitution of suffixes. Consider the element $x^\omega yx^\omega$ for all x and y. We have by strict locality and by the idempotence of both x^ω and $\vee$ that

$$\begin{aligned} x^\omega yx^\omega yx^\omega &\leqslant x^\omega y(x^\omega)x^\omega \vee x^\omega(x^\omega)yx^\omega \\ &= x^\omega yx^\omega. \end{aligned}$$

And we also have that

$$\begin{aligned} x^\omega yx^\omega &\leqslant x^\omega y(x^\omega)yx^\omega \vee x^\omega yx^\omega y(x^\omega)x^\omega \\ &= x^\omega yx^\omega yx^\omega. \end{aligned}$$

As each is less than or equal to the other,

$$x^\omega yx^\omega = x^\omega yx^\omega yx^\omega$$

Next, consider an element of the form $x^\omega yx^\omega zx^\omega$. Let $e = x^\omega$. We have by strict locality and by the idempotence of both $e = x^\omega$ and $\vee$ that

$$\begin{aligned} eyeze &\leqslant e(e)zeyeze \vee ez(e)yeze \\ &= ezeyeze \\ &\leqslant ezeyez(e)ye \vee ezeyezey(e)e \\ &= ezeyezeye. \end{aligned}$$

By the preceding argument this final form reduces:

$$x^\omega(zx^\omega y)x^\omega(zx^\omega y)x^\omega = x^\omega zx^\omega yx^\omega$$

So, $x^\omega yx^\omega zx^\omega \leqslant x^\omega zx^\omega yx^\omega$ by transitivity. By a similar argument, the reverse holds as well and the two are equal. Both equations that characterize local testability are universally satisfied. □

Without using any external knowledge about how the classes are defined or how they interact, we showed purely by symbolic manipulation that every language locally testable in the strict sense is locally testable.

5.4 Positive Varieties as Conjunctive Varieties

Recall that a conjunctive variety is like a positive variety except that closure under union is not required but additionally the class must be closed under inverse images homomorphisms of the form $\psi\colon A^\square \to B^\square$ (or $\psi\colon A^\boxplus \to B^\boxplus$). It turns out that every positive variety is a conjunctive variety of the same type.

Theorem 3. *Let $\mathcal{V}$ be a positive $*$-variety ($+$-variety). Then $\mathcal{V}$ is also a conjunctive $*$-variety (resp. $+$-variety). Moreover, the characteristic inequalities are identical.*

Proof. Let $\mathcal{V}$ be a positive variety and let ψ be a homomorphism from $A^\square$ or $A^\boxplus$ to $B^\square$ or $B^\boxplus$, as appropriate. Also let $L \subseteq B\mathcal{V}$. For all words $w \in A^*$, the output $\psi(\{w\})$ is completely characterized by the output $\psi(\{a\})$ for $a \in A$. Because $B^\square$ and $B^\boxplus$ contain only *finite* sets, it is the case that $\psi(\{a\})$ is finite for all $a \in A$. Construct $\widehat{A}$ by creating a symbol a_i for each $a \in A$ and each $1 \leqslant i \leqslant |\psi(\{a\})|$.

As there are finitely many elements of A and each maps to finitely many objects, there are finitely many functions f that map each pair $\langle a, i\rangle$ (with $a \in \widehat{A}$ and $1 \leqslant i \leqslant |\psi(\{a\})|$) to an element of $\psi(\{a\})$. Let F be the collection of these functions. Construct $\widehat{\psi}_f\colon \widehat{A}^* \to B^*$ such that $\widehat{\psi}_f(a_i) = f(a, i)$. As $\mathcal{V}$ is a positive variety, we have that each $\widehat{\psi}_f^{-1}(L) \in \widehat{A}\mathcal{V}$. The intersection of this family, $X = \bigcap_{f\in F} \widehat{\psi}_f^{-1}(L)$, is also in $\widehat{A}\mathcal{V}$, as positive varieties are closed under finitary intersections. The intersection accounts for the restriction that $\psi(\{w\}) \subseteq L$, rather than capturing only nondisjointness.

Moreover, as every permutation of the indices is handled by some $f \in F$, it is the case that $a_i \sim_X a_j$ for all appropriate i and j. Let $\#$ be the stable subrelation of the Myhill relation where $a_i \# a_j$ for all i and j but no other elements are related. Let S be the syntactic ordered semigroup (ordered monoid) of X; then $S/\#$ is a quotient of S and therefore the corresponding language is in $A\mathcal{V}$.

Finally, that the characteristic inequalities are identical is from the definition. An inequality for

an ordered variety of the form $x \leqslant y$ means that $uyv \in L \Rightarrow uxv \in L$, which is exactly what $x \leqslant y$ means under the semilattice ordering. □

Corollary. *The strictly piecewise languages are the conjunctive $*$-variety that corresponds to the variety of join-monoids* $\mathbf{J}^- = [\![1 \leqslant x]\!]_{(*)}$.

It is for this reason that we chose to use the syntactic order as presented by Pin (1997) rather than the inverse order as used by Pin (2017).

This means that the syntactic join-monoid or join-semigroup provides sufficient information to capture varieties such as the piecewise testable and locally testable languages as well as positive varieties such as the strictly piecewise languages.

6 Constructing Join-Semigroups

For completeness, this section briefly describes the construction of the syntactic join-semigroup or join-monoid detailed by Polák (2001). Let L be a language over Σ recognized by a deterministic finite-state automaton $\mathcal{A}$ in canonical form, whose state set is Q and whose transition function is $\delta\colon \Sigma \to Q \to Q$. Define $\mathcal{L}(q)$ to be the language of the state q, the set of permissible continuations from that state. Fix for each equivalence class $[w]$ in Σ^+ (Σ^*) under $\sim_L$ a shortest representative w such that $\eta(w) = [w]$ and let W be this set of representatives. Construct a modified automaton with state set

$$\widehat{Q} = \left\{ \bigcap_{q \in S} \mathcal{L}(q) : S \subseteq Q \right\}$$

and transition function

$$\widehat{\delta}\colon \mathcal{P}(W) - \{\varnothing\} \to \widehat{Q} \to \widehat{Q}$$

where $\widehat{\delta}(\{a\})(\bigcap_{q\in S} \mathcal{L}(q)) = \bigcap_{q\in S} \mathcal{L}(\delta(a)(q))$ for $S \subseteq Q$. This transition function is implicitly extended to (singleton sets of) finite words in the typical way, and is then extended again to arbitrary sets by $\widehat{\delta}^*(T)(q) = \bigcap_{w\in T} \widehat{\delta}(\{w\})(q)$. The syntactic join-semigroup (join-monoid) of the language is the transition semigroup of this automaton (Polák, 2001), *i.e* the structure

$$\langle \{\widehat{\delta}^*(T) : T \in \mathcal{P}(W) - \{\varnothing\}\}, \cdot, + \rangle$$

where $\cdot$ is $\widehat{\delta}^*(T_1)\widehat{\delta}^*(T_2) = \widehat{\delta}^*(T_2) \circ \widehat{\delta}^*(T_1)$ and $+$ is $\widehat{\delta}^*(T_1) + \widehat{\delta}^*(T_2) = \widehat{\delta}^*(T_1 \cup T_2)$. Whether it is the join-semigroup or join-monoid depends on whether $[\lambda]$ is included in W.

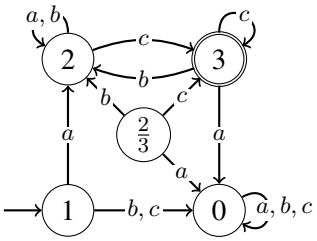

Figure 1: A strictly local language L, states extended.

6.1 Example Computation

Consider for example the strictly 2-local language L over $\Sigma = \{a, b, c\}$ consisting of all and only those words which begin with 'a', end with 'c', and do not contain "ca" as a substring. Its canonical finite-state automaton is shown in Figure 1 restricted to states $\{0, 1, 2, 3\}$. State 1 corresponds to the language itself. In state 2, the initial a has been accounted for, so its language is simply the set of continuations which end with 'c' and do not contain "ca" as a substring. In state 3, the initial 'a' has been accounted for and a 'c' has just been read: its language is the set of continuations which end with neither 'a' nor 'b' and do not contain "ca" as a substring (much like state 2, modulo λ), with the added restriction that the continuation not start with 'a', as, if it does, this creates a "ca" substring in the broader word. Finally state 0 is the rejecting sink.

The only nonempty subset of states whose languages intersect to yield a language not already accounted for is $\{2, 3\}$, whose intersection is the language of state 3 minus the empty string. The state $\frac{2}{3}$ represents this state in $\overline{Q}$. The empty intersection is Σ^* and is unaffected by transformations and therefore omitted.

By calculating the transition semigroup in the usual way, one finds that there are five elements in the syntactic semigroup of L, with representatives "a", "b", "c", "ac", and "ca". These five words represent every possible function over the original state set $\{0, 1, 2, 3\}$ that arise from any word in Σ^+. These elements multiply as shown in Table 1.

For each state q in $\overline{Q}$ and for each nonempty subset of semigroup elements, follow the paths labeled by each semigroup element in the subset, and construct the intersection of the resulting states. For example, $\{a, c\}$ acts on state 2 by having 'a' go to state 2 while 'c' goes to state 3; to simulate having done both of these at once, the resulting state is $\{2, 3\}$ (represented in Figure 1 as $\frac{2}{3}$). In

Table 1: Multiplication of semigroup elements of L, idempotents highlighted.

	a	b	c	ac	ca
a	a	a	ac	ac	ca
b	b	b	c	c	ca
c	ca	b	c	ca	ca
ac	ca	a	ac	ca	ca
ca	ca	ca	ca	ca	ca

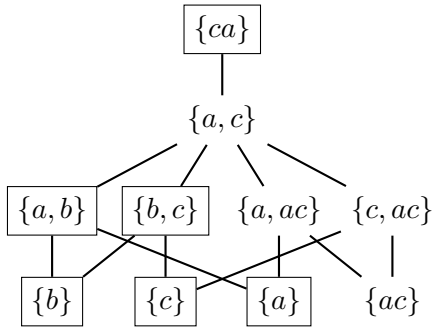

Figure 2: The join-semilattice structure that orders L. Multiplicative idempotents boxed.

the end, there are ten distinct actions arising from nonempty sets of semigroup elements, arranged into the semilattice shown in Figure 2. The join operation $x \vee y$ finds the least upper bound of x and y; *e.g.* $\{ac\} \vee \{b\} = \{a, c\}$.

Finally, one can verify that for any for sets A, B, C, and D and for any idempotent set X^ω, it holds that

$$AX^\omega D \vee AX^\omega B \vee CX^\omega D = AX^\omega B \vee CX^\omega D.$$

For instance, consider $A = \{ac\}$, $B = \{c\}$, $C = D = \{b\}$, and $X^\omega = \{b, c\}$:

$$AX^\omega B = \{ac\}\{b, c\}\{c\} = \{acbc, accc\} = \{ac\}$$
$$CX^\omega D = \{b\}\{b, c\}\{b\} = \{bbb, bcb\} = \{b\}$$
$$AX^\omega D = \{ac\}\{b, c\}\{b\} = \{acbb, accb\} = \{a\}$$

We verify that $\{ac\} \vee \{b\} = \{a, c\}$ and that $\{a\} \vee (\{ac\} \vee \{b\}) = \{a\} \vee \{a, c\} = \{a, c\}$. In other words, $AX^\omega D \leqslant AX^\omega B \vee CX^\omega D$ as desired. The same holds when considering any other valid instantiation of the variables, so the language is strictly local.

7 Conclusions and Prospects

We presented augmentations of the syntactic semigroup (monoid) structure to accommodate the additional information that is needed in order to capture two fundamental classes in the subregular hierarchy that are not closed under the Boolean operations. The syntactic join-semigroup (join-monoid) has enough information to capture *conjunctive* varieties of languages, which may not be closed under union or complementation. As computational linguistics research emphasizes intersection-closed classes without regard to whether they are also closed under the other Boolean operations, these techniques may prove more valuable in this space. Specifically, we provided purely algebraic characterizations for the strictly piecewise languages:

$$[\![1 \leqslant x]\!]_{(*)}$$

and the strictly local languages:

$$[\![ax^\omega d \leqslant ax^\omega b \vee cx^\omega d]\!]_{(+)}.$$

Recall that inequalities in the join-semigroup or join-monoid structure are notational shorthand for equalities, so one could also say that the strictly piecewise languages are characterized by

$$[\![1 \vee x = x]\!]_{(*)}$$

and the strictly local languages by

$$[\![ax^\omega d \vee ax^\omega b \vee cx^\omega d = ax^\omega b \vee cx^\omega d]\!]_{(+)}.$$

The latter perspective is preferred from the perspective of universal algebra, but the former offers a simpler statement of the characterizations.

Following Lambert (2023), the characterization of the strictly local languages extends to a characterization for tier-based strictly local languages of Heinz et al. (2011), by omitting $[\lambda]$ where possible in the construction of the syntactic join-semigroup, even when that class contains some *neutral letters*.

This also paves the way toward characterization and analysis of several other classes introduced in the computational linguistics literature. Using the techniques introduced herein alongside those of Lambert (in press), one might capture the multiple-tier-based strictly local languages of De Santo and Graf (2019). Join-semigroups may also help to capture the interval-based strictly piecewise languages of Graf (2017) or the melody-local languages of Jardine (2020). The decision procedures derived from the algebraic characterization could then be used to efficiently catalogue attested language patterns. For now, we leave open the question of whether these classes are conjunctive varieties, and, if so, what their characterizing inequalities are.

Known classes that are too restrictive can be extended to a more general class by removing one or more equations, perhaps after inserting redundant ones. For instance, piecewise testable languages satisfy both $(xy)^\omega x = (xy)^\omega$ *and* $y(xy)^\omega = (xy)^\omega$, while the broader class of $\mathcal{R}$-trivial languages (see Brzozowski and Fich, 1984) requires only the former. One can also extend a class, *e.g.* $\mathbf{J}^-$, by forming a product variety, *e.g.* $\mathbf{J}^- * \mathbf{D}$, which characterizes the "strictly piecewise-local" languages of Rogers and Lambert (2019b), also known as "generalized subsequence languages" (Heinz, 2010b) or "languages of dot-depth at most one-half". The resulting structures can instantiate learning algorithms using ideas from García and Ruiz (2006) and Heinz et al. (2012).

Another area of future work concerns the extension of these techniques to transducers. Lambert and Heinz (2023, 2024) explored the algebraic structure of phonological maps and some of the classes used to classify them. For each subregular $*$-variety ($+$-variety), one can say that a finite-state transducer belongs to an analogous class of functions if and only if its syntactic monoid (resp. semigroup) belongs to the corresponding variety of algebraic structures. Lambert and Heinz (2023) demonstrated that the "input strictly local" functions of Chandlee et al. (2014) have definite structure, a proper subclass of strictly local. Understanding how to derive a join-semigroup structure from a transducer will allow us to understand the class of functions that has the richer structure of the strictly local languages. It will also show whether the function class that Burness and McMullin (2020) refer to as strictly piecewise captures all and only those processes that are structurally strictly piecewise, or if there is another related class that does so.

Acknowledgments

The author thanks three anonymous reviewers for their contributions to the clarity of this document.

References

Jorge Almeida. 1989. Equations for pseudovarieties. In *Formal Properties of Finite Automata and Applications: Proceedings of the LITP Spring School on Theoretical Computer Science*, volume 386 of *Lecture Notes in Computer Science*, pages 148–164.

Jorge Almeida. 1995. *Finite Semigroups and Universal Algebra*, volume 3 of *Series in Algebra*. World Scientific, Singapore.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Proceessing (EMNLP)*, pages 7096–7116. Association for Computational Linguistics.

Stephen L. Bloom. 1976. Varieties of ordered algebras. *Journal of Computer and System Sciences*, 13(2):200–212.

Janusz Antoni Brzozowski. 1964. Derivatives of regular expressions. *Journal of the ACM*, 11(4):481–494.

Janusz Antoni Brzozowski and Faith Ellen Fich. 1984. On generalized locally testable languages. *Discrete Mathematics*, 50:153–169.

Phillip Burness and Kevin McMullin. 2020. Modelling non-local maps as strictly piecewise functions. In *Proceedings of the Society for Computation in Linguistics*, volume 3, pages 493–495, New Orleans, Louisiana.

Pascal Caron. 2000. Families of locally testable languages. *Theoretical Computer Science*, 242(1–2):361–376.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Eung-Do Cook. 1978. The synchronic and diachronic status of Sarcee γ^y. *International Journal of American Linguistics*, 44(3):192–196.

Aldo De Luca and Antonio Restivo. 1980. A characterization of strictly locally testable languages and its application to subsemigroups of a free semigroup. *Information and Control*, 44(3):300–319.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In Raffaella Bernardi, Greg Kobele, and Sylvain Pogodalla, editors, *Formal Grammar 2019*, volume 11668 of *Lecture Notes in Computer Science*, pages 35–50. Springer Verlag.

Matt Edlefsen, Dylan Leeman, Nathan Myers, Nathaniel Smith, Molly Visscher, and David Wellcome. 2008. Deciding strictly local (SL) languages. In *Proceedings of the 2008 Midstates Conference for Undergraduate Research in Computer Science and Mathematics*, pages 66–73.

Samuel Eilenberg. 1976. *Automata, Languages, and Machines*, volume B. Academic Press, New York, New York.

Jie Fu, Jeffrey Heinz, and Herbert G. Tanner. 2011. An algebraic characterization of strictly piecewise languages. In Mitsunori Ogihara and Jun Tarui, editors, *Theory and Applications of Models of Computation*, volume 6648 of *Lecture Notes in Computer Science*, pages 252–263. Springer Berlin / Heidelberg.

Pedro García and José Ruiz. 2006. Learning in varieties of the form $V * LI$ from positive data. *Theoretical Computer Science*, 362(1–3):100–114.

R. W. N. Goedemans, Jeffrey Heinz, and Harry van der Hulst. 2015. StressTyp2.

Thomas Graf. 2017. The power of locality domains in phonology. *Phonology*, 34(2):385–405.

Leonard H. Haines. 1969. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6(1):94–98.

Jeffrey Heinz. 2007. *Inductive Learning of Phonotactic Patterns*. Ph.D. thesis, University of California, Los Angeles.

Jeffrey Heinz. 2010a. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Jeffrey Heinz. 2010b. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906, Uppsala, Sweden. Association for Computational Linguistics.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, volume 23 of *Phonetics and Phonology*, chapter 5, pages 126–195. Mouton de Gruyter.

Jeffrey Heinz, Anna Kasprzik, and Timo Kötzing. 2012. Learning in the limit with lattice-structured hypothesis spaces. *Theoretical Computer Science*, 457:111–127.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*, volume 2, pages 58–64, Portland, Oregon. Association for Computational Linguistics.

Jeffrey Heinz and James Rogers. 2013. Learning subregular classes of languages with factored deterministic automata. In *Proceedings of the 13th Meeting on the Mathematics of Language*, pages 64–71, Sofia, Bulgaria. Association for Computational Linguistics.

Adam Jardine. 2020. Melody learning and long-distance phonotactics in tone. *Natural Language & Linguistic Theory*, 38:1145–1195.

Jacob K. Johnson and Aniello De Santo. 2024. Online learning of ITSL grammars. In *Proceedings of the Society for Computation in Linguistics*, volume 7, pages 257–267, Irvine, California.

Ondřej Klíma and Libor Polák. 2019. Syntactic structures of regular languages. *Theoretical Computer Science*, 800:125–141.

Dakotah Lambert. 2021. Grammar interpretations and learning TSL online. In *Proceedings of the Fifteenth International Conference on Grammatical Inference*, volume 153 of *Proceedings of Machine Learning Research*, pages 81–91.

Dakotah Lambert. 2023. Relativized adjacency. *Journal of Logic, Language and Information*, 32(4):707–731.

Dakotah Lambert. 2024. System description: A theorem-prover for subregular systems: The Language Toolkit and its interpreter, plebby. In *Functional and Logic Programming: 17th Annual Symposium, FLOPS 2024*, volume 14659 of *Lecture Notes in Computer Science*, pages 311–328, Kumamoto, Japan. Springer, Singapore.

Dakotah Lambert. in press. Multitier phonotactics with logic and algebra. *Phonology*.

Dakotah Lambert and Jeffrey Heinz. 2023. An algebraic characterization of total input strictly local functions. In *Proceedings of the Society for Computation in Linguistics*, volume 6, pages 25–34, Amherst, Massachusetts.

Dakotah Lambert and Jeffrey Heinz. 2024. Algebraic reanalysis of phonological processes described as output-oriented. In *Proceedings of the Society for Computation in Linguistics*, volume 7, pages 129–138, Irvine, California.

Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9(1):151–194.

Fang-Kuei Li. 1930. A study of Sarcee verb-stems. *International Journal of American Linguistics*, 6(1):3–27.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In Annie Foret, Greg Kobele, and Sylvain Pogodalla, editors, *Formal Grammar 2018*, volume 10950 of *Lecture Notes in Computer Science*, pages 62–83. Springer.

Robert McNaughton. 1974. Algebraic decision procedures for local testability. *Mathematical Systems Theory*, 8(1):60–76.

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

Jean-Éric Pin. 1995. A variety theorem without complementation. *Russian Mathematics (Izvestija vuzov. Matematika)*, 39:80–90.

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.

Jean-Éric Pin. 2017. The dot-depth hierarchy, 45 years later. In Stavros Konstantinidis, Nelma Moreira, Rogério Reis, and Jeffrey Shallit, editors, *The Role of Theory in Computer Science*, pages 177–201. World Scientific.

Jean-Éric Pin and Pascal Weil. 1996. A Reiterman theorem for pseudovarieties of finite first-order structures. *Algebra Universalis*, 35:577–595.

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.

Michael Oser Rabin and Dana Scott. 1959. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125.

Jan Reiterman. 1982. The Birkhoff theorem for finite algebras. *Algebra Universalis*, 14:1–10.

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.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar: 17th and 18th International Conferences FG 2012 Opole, Poland, August 2012, Revised Selected Papers and FG 2013 Düsseldorf, Germany, August 2013, Proceedings*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer-Verlag.

James Rogers and Dakotah Lambert. 2019a. Extracting Subregular constraints from Regular stringsets. *Journal of Language Modelling*, 7(2):143–176.

James Rogers and Dakotah Lambert. 2019b. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada. Association for Computational Linguistics.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20(3):329–342.

Howard Straubing. 1985. Finite semigroup varieties of the form $V * D$. *Journal of Pure and Applied Algebra*, 36:53–94.

Howard Straubing and Pascal Weil. 2021. Varieties. In Jean-Éric Pin, editor, *Handbook of Automata Theory*, volume 1, chapter 16, pages 569–614. EMS Press, Berlin, Germany.

Bret Tilson. 1987. Categories as algebra: An essential ingredient in the theory of monoids. *Journal of Pure and Applied Algebra*, 48(1–2):83–198.

Charles Torres and Richard Futrell. 2023. L_0-regularization induces subregular biases in LSTMs. In *Proceedings of the Society for Computation in Linguistics*, volume 6, pages 394–396, Amherst, Massachusetts.

Sam van der Poel, Dakotah Lambert, Kalina Kostyszyn, Tiantian Gao, Rahul Verma, Derek Andersen, Joanne Chau, Emily Peterson, Cody St. Clair, Paul Fodor, Chihiro Shibata, and Jeffrey Heinz. 2024. ML-RegTest: A benchmark for the machine learning of regular languages. *Journal of Machine Learning Research*, 25(283):1–45.