Phonology (2025), **xx**:xx 1–38
doi:

PHONOLOGY

ARTICLE

Multitier phonotactics with logic and algebra

Dakotah Lambert[1]

[1]Department of Computer Science, Haverford College, Haverford, PA, USA,
Email: dakotahlambert@acm.org.

Received: 20 March 2024; **Revised:** 8 September 2024; **Accepted:** 13 February 2025

Keywords: computational phonology, stress patterns, Uyghur backness harmony, Karanga Shona tone, propositional logic, tier-based extensions, multitier extensions, subregular hypothesis, algebra

Abstract
The subregular hypothesis posits that all phonological markedness constraints can be described by some principled, learnable subclass of the regular languages. We classify a wide range of attested long-distance phonological markedness constraints, covering stress, harmony and tone, with focus on interactions of constraints over multiple tiers. In doing so, we establish connections between propositional logic over multiple tiers and algebraic properties of formal languages. These techniques allow for mechanical verification or refutation of membership in a class. Modelling the constraints and their induced patterns as formal languages, we demonstrate that the entire range lies within the propositional level, including Uyghur backness harmony and Karanga Shona tone, which have been presented as challenges to aspects of the subregular hypothesis.

Contents

1. Introduction

One core question in phonology pertains to inherent typological restrictions: what kinds of patterns can arise naturally in human language? Here and throughout, the word 'PATTERN' refers generically to a constraint or a formal language. The subregular hypothesis of phonotactics posits that there exists some principled, learnable subclass of the regular languages that contains all attested phonological markedness constraints. Heinz (2018) promoted two significantly stronger hypotheses. First there is the 'strong' subregular hypothesis, that all such constraints can be described with a restricted propositional logic, there exemplified by the strictly local and strictly piecewise classes. Then there is the 'weak' subregular hypothesis, that the constraints can be described using at most some restricted form of first-order logic, there exemplified by the tier-based strictly local class (Heinz *et al.*, 2011).

Mayer & Major (2018) present Uyghur backness harmony as a rebuttal to even the weaker subregular hypothesis, showing that these classes in particular are insufficient. Jardine (2020) also points out aspects of constraints on tone in Karanga Shona that challenge subregular classification. In this article, we introduce a collection of classes at the propositional level, simpler than first-order logic, capable of describing these heretofore challenging patterns. We demonstrate that these classes capture not only these challenging patterns, but also a wide range of constraints in stress and harmony. The simple algebraic structure of these classes greatly facilitated the analysis that led to these results.

At this point, we do not seek one class that contains every possible pattern. Instead, we seek to provide several plausible logical analyses of attested patterns. In doing so, we find a collection of logical building blocks that can be used to describe patterns we encounter in a natural way, and subsequent analysis can locate the patterns relative to others in terms of cognitive complexity. As we shall see in §4, the complexity relations can invert under different logical systems. Understanding where patterns lie when considering different representational assumptions and different logical systems is thus key to understanding the typological and psychological pressures at play. For instance, harmony constraints may require strong logical power to express over adjacent substrings, while a comparatively weak logic may suffice over long-distance subsequences. Our analysis shows a bias toward propositional constraints. These can be learned and perceived with attention only to structures present in the words themselves, without imposing additional logical structure over them.

The initial portion of this work focuses on this logical perspective. However, the analyses in this article heavily relied upon algebraic techniques discussed in §6; these techniques allow us to easily verify or refute class membership, which means that they tell us which kinds of logical formulae are viable for describing a pattern.

The STRICTLY LOCAL patterns, introduced by McNaughton & Papert (1971), have played an important role in phonotactics (Heinz, 2018). Languages in this class are defined by a set of forbidden substrings. A word is rejected if any of its substrings is forbidden; otherwise all of its substrings are allowed and it is accepted. Here and throughout, the terms ‘PREFIX’ and ‘SUFFIX’ refer to substrings anchored to the left or right edge of a string, respectively, and not to the morphological concept by the same name. For instance, ‘c’ is a prefix of ‘concept’, and ‘ept’ is a suffix. An AFFIX is a prefix or suffix in this sense. A strictly k-local pattern can forbid prefixes or suffixes of length up to $k - 1$ and can forbid internal substrings of length up to k.

This work focuses on a simpler class: the DEFINITE patterns introduced by Kleene (1951) and further studied by Perles *et al.* (1963). These are even simpler: a k-definite pattern forbids suffixes of length up to k. We also explore related classes, such as the REVERSE DEFINITE patterns which restrict the set of permissible prefixes.

Heinz *et al.* (2011) apply Goldsmith’s (1976) notion of the phonological tier to strictly local patterns, yielding the TIER-BASED strictly local class. Lambert (2023) demonstrates how to extend this tier-based perspective to other classes. Essentially, some alphabetic symbols, those not on the tier, are invisible to the pattern. When a pattern can be described as a system of tier-based constraints of a given type, but the tiers are not equivalent, then this pattern is a MULTITIER pattern. Multitier perspectives are natural in a logical and algebraic sense, and we show that many attested patterns, including the challenging Uyghur backness harmony and Karanga Shona tone patterns, have simple multitier affix-based descriptions.

The paper proceeds as follows. First §2 formally introduces the definite patterns and related classes, including their tier-based and multitier extensions, in order to provide a new analysis of bounded stress patterns, where primary stress must occur within some fixed distance of a fixed edge of a word. Next, §3 applies the logical techniques introduced in §2 to analyse unbounded stress patterns, where primary stress can appear arbitrarily far from a word edge. After showing that these abstract patterns are captured by logical formulae involving affixes on multiple tiers, we apply these analytical techniques to the StressTyp2 database of Goedemans *et al.* (2015), which demonstrates their wide coverage. Next, §4 characterises some harmony patterns, including Uyghur backness harmony, which we show to be multitier definite. §5 characterises some tone patterns, including that of Karanga Shona, which we show to be multitier generalised definite, using both tier-prefixes and tier-suffixes in its description. Then §6 revisits some of these patterns to introduce the key algebraic techniques that lead to and support these analyses. All analyses have been verified using the Language Toolkit, a publicly available open-source software package designed for answering questions in computational phonology, which implements the algebraic methods described (Lambert, 2024).[1] Finally we conclude with reiteration of our findings that mere propositional logic captures seemingly complicated attested patterns.

2. Analyses of bounded stress

This section presents three simple attested stress patterns: stress-final, where primary stress is fixed to the final syllable, stress-penult, where it is fixed to the penult in words of two or more syllables (else it appears on the only syllable), and stress-initial, where it is fixed to the initial syllable. These patterns are all STRICTLY LOCAL, but this single perspective is limiting. We present long-distance analyses of these patterns in order to introduce more classes of formal languages and the formal logics that define them. In doing so, we introduce multitier extensions of classes, the primary contribution of this work. Under these long-distance perspectives, these bounded stress patterns are not all that different from the unbounded patterns that we explore in the next section.

We can use a simple logical language in the style of Rogers & Lambert (2019) to describe patterns. If we have an alphabet Σ and two distinct end-markers $\rtimes$ and $\ltimes$ that are not in Σ to mark the beginning and end of words, respectively, then a string u over this expanded alphabet is a LITERAL, an atomic term in the logic. On its own, u is a logical formula, and a word w satisfies u, written $w \models u$, if and only if u is a substring of $\rtimes w \ltimes$. In other words, $w \models u$ if and only if there exist strings x and y such that $\rtimes w \ltimes = xuy$. For example, we have that $concept \models ce$, because 'ce' is a substring of '$\rtimes$con**ce**pt$\ltimes$'.

Where the formula u describes a *required* substring, accepting only words that contain u, its negation, written $\neg u$, instead describes a forbidden substring. For instance, $concept \models \neg a$ because 'a' is not a substring of '$\rtimes$concept$\ltimes$'. We can forbid multiple substrings with CONJUNCTION, written $\wedge$ and read as 'and'. For instance, $concept \models \neg a \wedge \neg b$, because 'a' is not a substring of '$\rtimes$concept$\ltimes$' *and* 'b' is not a substring of '$\rtimes$concept$\ltimes$'.

This suffices to describe strictly local languages: each is defined by a conjunction of negated literals. We refer to this as a RESTRICTED PROPOSITIONAL LOGIC. Adding DISJUNCTION and negation gives PROPOSITIONAL LOGIC as a whole. Disjunction, the logical *inclusive or*, is written $\vee$. We say $concept \models a \vee b \vee \rtimes c$ because, while 'a' and 'b' are not substrings of '$\rtimes$concept$\ltimes$', '$\rtimes$c' is a substring. Any disjunct being true suffices to make the entire expression true. More than one is allowed to be true; for instance, $concept \models a \vee \neg b \vee \rtimes c$ as well.

Definition 2.1. The k-DEFINITE languages ($\Sigma^*\mathcal{D}_k$) are those that are definable by propositional formulae over suffixes of length up to k. The DEFINITE languages ($\Sigma^*\mathcal{D}$) are those that are k-definite for some fixed, finite k.

[1]The Language Toolkit is available at https://hackage.haskell.org/package/language-toolkit, and the scripts used in our analyses are available as supplementary material.

It is not a mistake to say that definite patterns may use the full propositional logic rather than the restricted propositional logic. To require a particular suffix is to forbid all incompatible suffixes. It is for this reason that definite patterns remain strictly local despite having access to a more featureful logic.

All of the stress patterns that we discuss in this section are strictly local. However, we present more analyses under different perspectives. We will add more features to this propositional logical as we progress, in order to represent long-distance constraints.

2.1. Stress-final, stress-penult and definite patterns

Consider the following words of Iban, an Austronesian language, as reported by Omar (1969: 78–79).

(1)	a.	$(\sigma\acute{\sigma})$	[bəˈrap]	'to embrace'
	b.	$(\sigma\sigma\sigma\acute{\sigma})$	[səpəməˈrap]	'measure of girth'
	c.	$(\sigma\acute{\sigma})$	[maˈnah]	'beautiful'
	d.	$(\sigma\sigma\sigma\sigma\acute{\sigma})$	[dikəmanahˈka]	'is beautified'

Per Omar (1969: 70), stress is consistently word-final in Iban.

The basic unit is the syllable and there are two types: unstressed syllables (σ) and stressed syllables ($\acute{\sigma}$). The alphabet is $\Sigma = \{\sigma, \acute{\sigma}\}$. In logical terms, this constraint asserts that words end in a stressed syllable: $\acute{\sigma}\ltimes$.

This logical expression is satisfied by the words we have already seen, including $\sigma\acute{\sigma}$ (like [bəˈrap] and [maˈnah]) or $\sigma\sigma\sigma\sigma\acute{\sigma}$ (like [dikəmanahˈka]). It is also satisfied by words like $\acute{\sigma}\acute{\sigma}\acute{\sigma}$, where multiple syllables bear primary stress. Those will be ruled out by a separate constraint, CULMINATIVITY, that will be discussed in §2.3. For now, we ignore culminativity and accept these extraneous words. That is, we now focus on the stress-final constraint in isolation, not the culminative stress-final pattern that more accurately models Iban. The form of our logical expression, a propositional formula involving only suffixes, demonstrates that this stress-final constraint is DEFINITE.

Definition 2.2. The k-SUFFIX of a word is the last k symbols of the word if it is of length at least k, else the entire word. The k-PREFIX is defined analogously for the first symbols.

Proposition 2.3. *A language L is k-definite if and only if whenever two words have the same k-suffix, either both are in L or neither is in L.*

Proposition 2.3 provides a grammar-agnostic characterisation of the definite languages. The stress-final constraint is 1-definite: if the last syllable is stressed, then the word is accepted, else it is rejected. None of the other syllables in the word influence this in any way.

This constraint is equivalently expressed by a negative formula. Instead of requiring the $\acute{\sigma}\ltimes$ suffix, we can forbid the incompatible suffixes: $\neg\rtimes\ltimes \wedge \neg\sigma\ltimes$. The empty word is forbidden because it has no final syllable on which to place stress, and word with unstressed final syllables are forbidden, because this is incompatible with having final stress.

Next, consider the following words of Amara, another Austronesian language, as reported by Thurston (1996: 205).

(2)	a.	$(\sigma\sigma\acute{\sigma}\sigma)$	[ipoˈnei̯ki]	‘it’s full of water’
	b.	$(\sigma\acute{\sigma}\sigma)$	[nɑŋˈgrikŋe]	‘unripe, uncooked’
	c.	$(\acute{\sigma})$	[ˈkin]	‘they drink’

Per Thurston (1996: 205), primary stress in Amara is consistently fixed to the penultimate syllable, except in monosyllables where it must occur on the only syllable. Like with stress-final, we factor out culminativity and focus now on the placement constraint in isolation. We can express this logically as a disjunction: either the word is a stressed monosyllable ($\rtimes\acute{\sigma}\ltimes$) or it has at least two syllables of which the penultimate bears stress ($\acute{\sigma}\Sigma\ltimes$). The latter form is convenient notation to express $\acute{\sigma}\sigma\ltimes \vee \acute{\sigma}\acute{\sigma}\ltimes$, expanding the set Σ into its elements. All together, the formula is $\acute{\sigma}\Sigma\ltimes \vee \rtimes\acute{\sigma}\ltimes$. Application of Proposition 2.3 reveals that this is not 1-definite, as $\acute{\sigma}$ satisfies the constraint but $\sigma\acute{\sigma}$ does not, despite sharing the same 1-suffix: $\acute{\sigma}$. However, it is 2-definite. The word is accepted if and only if the first symbol of its 2-suffix is $\acute{\sigma}$.

Like the stress-final constraint, this stress-penult constraint can alternatively be written in terms of forbidden suffixes. The incompatible suffixes are unstressed monosyllables ($\neg\rtimes\sigma\ltimes$) and unstressed penults ($\neg\sigma\Sigma\ltimes$), as well as the empty word ($\neg\rtimes\ltimes$) so the equivalent restricted propositional formula is $\neg\rtimes\ltimes \wedge \neg\rtimes\sigma\ltimes \wedge \neg\sigma\Sigma\ltimes$.

2.2. Stress-initial and reverse definite patterns

Next we explore a pattern that is not definite. Consider the following words of Finnish, a Uralic language, as reported by Suomi *et al.* (2008: 76).

(3)	a.	$(\acute{\sigma}\sigma)$	[ˈjærki]	‘sense’
	b.	$(\acute{\sigma}\sigma\sigma)$	[ˈjærjet̪øn]	‘senseless’
	c.	$(\acute{\sigma}\sigma\sigma\sigma)$	[ˈjærjest̪elmæ]	‘system’

Per Suomi *et al.* (2008: 75), primary stress in Finnish is consistently fixed to the initial syllable. There is also secondary stress, not notated here.

Let us consider only the placement of primary stress and continue to ignore culminativity. This stress-initial pattern is not definite because it cannot be k-definite for any k. For $k = 1$, consider the accepted word $\acute{\sigma}\sigma$ and the rejected word $\sigma\sigma$. Both share the same length-one suffix σ, so 1-suffixes cannot make the right distinctions. Similarly, for $k = 2$, consider the accepted word $\acute{\sigma}\sigma\sigma$ and the rejected word $\sigma\sigma\sigma$. Both share the same length-two suffix $\sigma\sigma$, so 2-suffixes cannot make the right distinctions. In general, $\acute{\sigma}\sigma^k$ satisfies the constraint while $\sigma\sigma^k$ does not, despite the fact that they share the length-k suffix σ^k.[2] The suffix-oriented definite class cannot handle this prefix-oriented pattern. While these counterexamples were hand-chosen, the algebraic techniques introduced in §6 allow us to derive them automatically.

[2] Here and throughout, parameterised words witnessing nonmembership are chosen for simplicity, not for phonetic viability.

Definition 2.4. The <span>REVERSE</span> k-DEFINITE languages ($\Sigma^*\mathcal{K}_k$)[3] are those that are definable by propositional formulae over prefixes of length up to k. The REVERSE DEFINITE languages ($\Sigma^*\mathcal{K}$) are those that are reverse k-definite for some fixed, finite k.

Proposition 2.5. *A language L is reverse k-definite if and only if whenever two words have the same k-prefix, either both are in L or neither is in L.*

The stress-initial constraint asserts that the initial syllable must be stressed: $\rtimes\acute{\sigma}$. Like definite languages, reverse definite languages are strictly local, so this formula can be written in negative form as well. Rather than requiring an initial primary stress, we forbid the incompatible empty word and initial unstressed syllables: $\neg{\rtimes\ltimes} \land \neg{\rtimes\sigma}$.

Some patterns are both definite and reverse definite. Specifically, any finite set of words or the complement of such a set is both definite and reverse definite. We call these the CO/FINITE languages. They are defined by propositional formulae where all substrings are entire words, anchored on both ends.

Definition 2.6. The CO/FINITE languages ($\Sigma^*\mathcal{N}$) are finite sets of words and the complements of such sets.[4]

2.3. *Culminativity*

In this section, we add culminativity to our bounded stress constraints to demonstrate that strict locality is strictly more powerful than (reverse) definiteness, due to its ability to restrict internal substrings. Culminative bounded stress patterns are strictly local, but we present two alternative analyses as well, first using order-based constraints (piecewise testability) and then again using projection-based constraints (tiers).

2.3.1. Culminative bounded stress with strict locality

Recall that culminativity is the constraint that forbids having multiple syllables with primary stress. In this work, we use the term CULMINATIVITY in the same sense as Hyman (2009: 217): it is not the ONE-STRESS constraint, but only an upper bound on the number of stressed syllables. The lower bound of one is enforced by separate constraint: OBLIGATORINESS. Culminativity is satisfied by the empty word as well as words like σ, $\sigma\acute{\sigma}$, $\sigma\sigma\sigma\sigma\sigma$, and so on, because all have zero or one stressed syllables. But it is not satisfied by $\acute{\sigma}\acute{\sigma}$ or $\acute{\sigma}\sigma\sigma\sigma\acute{\sigma}$, because each of these words has at least two stressed syllables.

Culminativity is not strictly local in isolation. It rejects $\sigma^k\acute{\sigma}\sigma^k\acute{\sigma}\sigma^k$, but no k-factor in this word can be forbidden, as they all appear in the accepted word $\sigma^k\acute{\sigma}\sigma^k$. If some k-factor were forbidden to trigger the rejection, then the latter word would be rejected as well, because it, too, contains that factor. However, when coupled with a bounded stress pattern, the system as a whole is strictly local.

For instance, with the stress-final constraint, the final syllable receives stress. In order to enforce culminativity, stress need only be forbidden in nonfinal positions, before

[3]The name '$\mathcal{K}$' is traditional (*cf.* Almeida, 1995). Its origin is unclear, but given the mathematical tradition of using German names, one likely candidate is the German *kehren*, 'to turn around'.

[4]The name '$\mathcal{N}$' stands for 'nilpotent', an algebraic term for the kind of structure that these languages produce (*cf.* Almeida, 1995).

some other syllable: $\neg\acute{\sigma}\Sigma$. Recall that the stress-final constraint can be written in strictly local form as $\neg\rtimes\ltimes \wedge \neg\sigma\ltimes$, and add this new constraint to describe the culminative stress-final pattern: $\neg\rtimes\ltimes \wedge \neg\sigma\ltimes \wedge \neg\acute{\sigma}\Sigma$.

However, culminativity even alongside the stress-final constraint is not definite, as the system is not k-definite for any k. The crux of the argument is that there can be arbitrarily many unstressed syllables intervening between the stressed final syllable and earlier unwanted stress. We have that $\sigma\sigma^k\acute{\sigma}$ satisfies the constraint but $\acute{\sigma}\sigma^k\acute{\sigma}$ does not, despite the two sharing the same length-k suffix $\sigma^{k-1}\acute{\sigma}$. This demonstrates that strict locality is strictly more powerful than definiteness. From a processing perspective, a mechanism capable of processing a strictly local language must be able to remember whether a forbidden factor was encountered at any point in a word, while definiteness only requires attending to the final k symbols.

Recall that the stress-penult constraint is $\neg\rtimes\ltimes \wedge \neg\rtimes\sigma\ltimes \wedge \neg\sigma\Sigma\ltimes$. This only ensures that the penult is stressed (or the only syllable in the case of monosyllables). To enforce culminativity, we must ensure that stress does not appear in other positions. First, stress should not occur before the penult: $\neg\acute{\sigma}\Sigma\Sigma$. This still allows for stress on the final syllable. We cannot simply forbid that configuration, as it would rule out monosyllables, but we can forbid having two adjacent stressed syllables to achieve the desired effect: $\neg\acute{\sigma}\acute{\sigma}$. If the penult and final syllable were both stressed, then this forbidden configuration would occur. The culminative stress-penult constraint is $\neg\rtimes\ltimes\wedge\neg\rtimes\sigma\ltimes\wedge\neg\sigma\Sigma\ltimes\wedge\neg\acute{\sigma}\Sigma\Sigma\wedge\neg\acute{\sigma}\acute{\sigma}$. Like the culminative stress-final constraint, this is strictly local but not definite.

Finally, stress-initial is the reversal of stress-final and is captured with the following formula: $\neg\rtimes\ltimes\wedge\neg\rtimes\sigma\wedge\neg\Sigma\acute{\sigma}$. Just as the culminative stress-final pattern was not definite, this culminative stress-initial pattern is not reverse definite. We have that $\acute{\sigma}\sigma^k\sigma$ satisfies the constraint while $\acute{\sigma}\sigma^k\acute{\sigma}$ does not, despite sharing the same length-k prefix, $\acute{\sigma}\sigma^{k-1}$. Strict locality is more powerful than reverse definiteness, for exactly the same reason that it is more powerful than definiteness.

The definite and reverse definite classes may have access to a complete system of propositional logical, but they are limited to constraints on suffixes and prefixes, respectively. Having access to more parts of the word gives strict locality more power, even with its weaker restricted propositional logic. That is, a conjunction of negated literals where the literals are arbitrary substrings is strictly more powerful than a full propositional formula where the literals can be only suffixes or only prefixes.

2.3.2. Culminative bounded stress with order

Another logically defined class, incomparable with (reverse) definite and strictly local, is the PIECEWISE TESTABLE languages (Simon, 1975). These are defined by propositional logic not over substrings but over SUBSEQUENCES. Such patterns played a key role in Heinz's (2010) analysis of long-distance phonotactics.

Definition 2.7. A string $u = u_1u_2 \ldots u_n$ is a subsequence of a string w if and only if there exist (possibly empty) strings $v_0, \ldots, v_n$ such that $w = v_0u_1v_1 \ldots u_nv_n$.

In other words, u is a subsequence of w if and only if w contains all of the letters of u in order, but not necessarily adjacently. For example 'net' is a subsequence of '**co**n**c**e**p**t'. We represent subsequences with a new kind of literal: $w \models u_1..u_2..\cdots..u_n$

if $u_1 u_2 \dots u_n$ is a subsequence of w. Boundary symbols are not used in subsequence literals. They are not necessary, as every symbol in a word follows the start marker $\rtimes$ and every symbol precedes the end marker $\ltimes$.

Definition 2.8. A language L is PIECEWISE k-TESTABLE ($L \in \Sigma^*\mathcal{J}_k$)[5] if and only if whenever two words have the same subsequences of length up to k, either both are in L or neither is in L. Then L is PIECEWISE TESTABLE ($L \in \Sigma^*\mathcal{J}$) if and only if it is piecewise k-testable for some fixed, finite k. These are the languages definable by propositional formulae involving only subsequences.

None of the bounded stress constraints discussed so far are piecewise testable, but their culminative versions are. We show the latter by presenting logical descriptions in the appropriate form.

Culminativity forbids the $\acute{\sigma}..\acute{\sigma}$ subsequence (Heinz, 2014). For the culminative stress-final pattern, unstressed syllables are additionally forbidden after the stressed syllable: $\neg\acute{\sigma}..\Sigma$. We must also enforce OBLIGATORINESS, the constraint that some stressed syllable appears: $\acute{\sigma}$. The culminative stress-final pattern is overall $\acute{\sigma} \wedge \neg\acute{\sigma}..\Sigma$. For the culminative stress-initial pattern, we reverse this: $\acute{\sigma} \wedge \neg\Sigma..\acute{\sigma}$.

The culminative stress-penult constraint is similar. Culminativity and obligatoriness apply, and two syllables cannot follow the stressed syllable: $\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma} \wedge \neg\acute{\sigma}..\Sigma..\Sigma$.

2.3.3. Culminative bounded stress with (multiple) tiers

Next we introduce the classes at the core of our contribution, the multitier extensions of classes, which provide another way to describe long-distance constraints. Tier-projection expresses some kinds of long-distance constraints by ignoring irrelevant symbols and enforcing local constraints on the result. For culminativity, inserting or removing unstressed syllables σ can never change a word from accepted to rejected or vice versa. This is a NEUTRAL LETTER in the terminology of Mix Barrington *et al.* (2001), and they are important to the study of tier-based extensions of classes, as neutral letters are the symbols not on the tier.

These neutral letters prevent culminativity from being definite or reverse definite. Clearly, $\acute{\sigma}$ satisfies culminativity while $\acute{\sigma}\acute{\sigma}$ does not. We can freely insert the unstressed σ without changing this status on both the front and the back of these words in order to saturate the length-k prefixes and suffixes: $\sigma^k\acute{\sigma}\sigma^k$ is accepted while $\sigma^k\acute{\sigma}\acute{\sigma}\sigma^k$ is not. They share the same σ^k suffix and prefix and are thus not (reverse) k-definite for any k.

If we could first project to the tier of stressed syllables, then the only valid words on the projection would be the empty string and the stressed monosyllable. In other words, on the stress tier, the formula $\rtimes\ltimes \vee \rtimes\acute{\sigma}\ltimes$ holds. Culminativity is tier-based co/finite on this tier. Heinz *et al.* (2011) offer a formalisation of TIER-PROJECTION, applying a constraint only after projection to some set of salient symbols. Originally applied only to the strictly local languages, Lambert (2023) extends the concept to all classes.

Definition 2.9. For any class C, a language L over Σ is in TIER-BASED C ($\Sigma^*\mathcal{T}C$) if and only if there is some language L' over some subset $\Gamma \subseteq \Sigma$ such that L' is in Γ^*C and all symbols in $\Sigma - \Gamma$ are neutral.

[5]The name '$\mathcal{J}$' is traditional (*cf.* Almeida, 1995). It refers to an algebraic property of the languages in the class.

Additional notation in our propositional logic accounts for tiers: a word $w \models [\varphi]_S$ if and only if the projection of w to S satisfies φ. Culminativity is thus expressed $[\rtimes\ltimes \vee \rtimes\acute{\sigma}\ltimes]_{\{\acute{\sigma}\}}$ and is in $\mathcal{TN}$.

However, tier-based classes per Definition 2.9 use only a single tier of projection. This is limiting. For culminative bounded stress patterns, the stress-placement constraint should operate on the entire word, while culminativity operates only over the tier of stressed syllables.

Therefore we consider a propositional logic where the literals can project to different tiers. As shown below, such a logical system can capture stress-final, stress-penult and stress-initial, each with culminativity.

- Stress-final: $\acute{\sigma}\ltimes \wedge [\rtimes\acute{\sigma}\ltimes]_{\{\acute{\sigma}\}}$
- Stress-penult: $(\acute{\sigma}\sigma\ltimes \vee \rtimes\acute{\sigma}\ltimes) \wedge [\rtimes\acute{\sigma}\ltimes]_{\{\acute{\sigma}\}}$
- Stress-initial: $\rtimes\acute{\sigma} \wedge [\rtimes\acute{\sigma}\ltimes]_{\{\acute{\sigma}\}}$

Note that, as the stress-placement constraint guarantees that stress appears, the stress tier cannot be empty. That is why we used simply $[\rtimes\acute{\sigma}\ltimes]_{\{\acute{\sigma}\}}$ to enforce culminativity. Also note that stress-final and stress-penult with culminativity use only tier-suffixes, so they are multitier definite. Similarly, stress-initial with culminativity uses only tier-prefixes and is thus multitier reverse definite.

Definition 2.10. For any class C, its multitier extension $\mathcal{BTC}$ is the Boolean closure of its tier-based extension $\mathcal{TC}$. The tiers need not be the same between the parts.

Notice that we allow for arbitrary Boolean combinations of projected factors. This differs from earlier treatments of multiple tiers, such as by De Santo & Graf (2019), who explored extensions of the strictly local languages. As strictly local languages are closed under intersection but not under union nor complement, De Santo & Graf (2019) considered only intersection-closure. Because the weaker classes that we consider here are closed under all Boolean operations, it is natural to use the full Boolean closure for their multitier extensions, and as we shall see in §6, this results in useful algebraic properties.

Interestingly, the multitier co/finite languages $\mathcal{BTN}$ form a subclass of the piecewise testable languages $\mathcal{J}$. To show this, consider $u = u_1u_2 \ldots u_n$. The logical formula $\rtimes u\ltimes$ represents the word u, and this can also be achieved using subsequence constraints: $u_1..u_2.. \cdots ..u_n$ asserts that the all of the letters in u occur and are correctly ordered, while $\neg\Sigma..\Sigma.. \cdots ..\Sigma$ forbids words with at least as many symbols as there are Σ. Concretely, $\rtimes abc\ltimes$ is equivalent to $a..b..c \wedge \neg\Sigma..\Sigma..\Sigma..\Sigma$. The three letters occur in order, and valid words do not have four or more letters. To represent $[\rtimes u\ltimes]_\Gamma$, we need only replace the Σ with Γ: instead of saying that valid words do not have $n+1$ letters, we say that valid words do not have $n+1$ letters *on the tier*. Tier-words are expressible formulae over subsequences, so every multitier co/finite language is piecewise testable.

2.4. Generalised definiteness and a small hierarchy

Definite languages are based on suffixes. Reverse definite languages use prefixes. Generalised definite languages extend the propositional logic to allow both. For example, $\rtimes a \vee b \ltimes$ is neither definite nor reverse definite, but it is generalised definite.

Definition 2.11. The GENERALISED k-DEFINITE languages ($\Sigma^*\mathcal{LI}_k$)[6] are those that are definable by propositional formulae over affixes of length up to k. The GENERALISED DEFINITE languages ($\Sigma^*\mathcal{LI}$) are those that are generalised k-definite for some fixed, finite k.

Proposition 2.12. *A language L is generalised k-definite if and only if whenever two words have the same k-prefix and the same k-suffix, then either both are in L or neither is in L.*

Generalised definite patterns extend definiteness in a way that is orthogonal to what strict locality adds. The strictly local class is incomparable with $\mathcal{LI}$, and, unlike strict locality, moving to generalised definiteness does not suffice to describe culminative bounded stress without tiers. Consider stress-final with culminativity. This accepts $\sigma^k\acute{\sigma}$ but rejects $\sigma^k\acute{\sigma}\sigma^k\acute{\sigma}$, despite the two words sharing the same length-k prefix, σ^k, and the same length-k suffix, $\sigma^{k-1}\acute{\sigma}$. Using both prefixes and suffixes will, however, allow us to describe all of the bounded stress patterns under the same logical system. Further, some unbounded stress patterns detailed in the next section require this additional power.

Culminative bounded stress patterns are strictly local (restricted propositional with substrings), multitier definite or multitier reverse definite (propositional with tier-suffixes or tier-prefixes), and piecewise testable (propositional with subsequences). The simplicity of these patterns allows for description under many different propositional logics. The next section explores the extent to which these systems can account for unbounded stress.

3. Unbounded stress

In the previous section, we introduced several classes of formal languages associated with particular forms of logical expressions, including the definite and reverse definite languages and their multitier extensions. Our primary contributions are the logical and algebraic characterisations of multitier classes, as well as the demonstrations that the associated propositional logics describe some complex but attested phonological constraints.

This section classifies unbounded stress patterns with respect to multitier classes. Heinz (2014) shows that simple unbounded stress patterns are piecewise testable as conjunctions of forbidden subsequences and one-stress. We add that they are also at most multitier generalised definite. We further expand upon the work of Rogers & Lambert (2019), who show that most of the stress patterns in the StressTyp2 database (Goedemans *et al.*, 2015) are propositional when using substrings and subsequences

[6]The name '$\mathcal{LI}$' refers to a particular kind of algebraic structure. First, $\mathcal{I}$, the Roman numeral one, refers to there being only one class of strings: everything is accepted or everything is rejected. The $\mathcal{L}$ operator means 'locally' and represents a general operation, much like our $\mathcal{T}$ and $\mathcal{B}$. For more information on $\mathcal{LI}$ and its constituent parts, see Almeida (1995).

simultaneously. The main result is that the multitier generalised definite class, $\mathcal{BTLI}$, provides better coverage of the StressTyp2 database than the piecewise testable class, $\mathcal{J}$.

First we work through the simple typology laid out by Hayes (1995). Unlike the bounded stress patterns, these unbounded stress patterns are quantity-sensitive, so our alphabet must distinguish light syllables (ℓ) from heavy syllables (h). The alphabet is now $\Sigma = \{\ell, h, \acute{\ell}, \acute{h}\}$. We use σ to denote the set $\{\ell, h\}$, and $\acute{\sigma}$ to denote the set $\{\acute{\ell}, \acute{h}\}$. Then we report on the actual patterns described in the StressTyp2 database of Goedemans *et al.* (2015). Scripts used for this analysis are included as supplementary materials.

3.1. *Stress leftmost heavy, else leftmost*

We first investigate monomorphemic words of Amele, a Trans-New Guinea language. Consider the following words, from Roberts (1987: 346,358).

(4)	a.	$(\ell\ell\acute{h})$	[gædɔˈlɔh]	‘edge’
	b.	$(\ell\ell\acute{h})$	[ɪtɪˈtɔm]	‘righteous’
	c.	$(\ell\acute{h}\ell)$	[ɟæˈwælti]	‘wind from north’
	d.	$(\ell\acute{h})$	[duˈæn]	‘cold’
	e.	$(\acute{h}h)$	[ˈtubdoʔ]	‘to join’
	f.	$(\acute{\ell}\ell\ell)$	[ˈʊfio]	‘yam species’

As reported by Roberts (1987: 357–358), primary stress occurs on the leftmost heavy syllable (the leftmost closed syllable) if such a syllable exists, else on the leftmost syllable.

As noted by Heinz (2014), these unbounded stress patterns are piecewise testable. The one-stress constraint applies: $\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma}$. Then the first priority is to stress the leftmost heavy syllable if it exists. This means that no (unstressed) heavy syllable precedes a stressed syllable: $\neg h..\acute{\sigma}$. Further, a stressed light syllable cannot precede an (unstressed) heavy syllable: $\neg\acute{\ell}..h$. This guarantees placement on the leftmost heavy syllable if it exists. The next priority is to ensure that in other cases, the initial syllable is stressed. That is, no syllable precedes a stressed light syllable: $\neg\Sigma..\acute{\ell}$, or simply $\neg\ell..\acute{\ell}$, as the unstressed light syllable ℓ is the only type not covered by other constraints.[7] In sum, monomorphemic words in Amele are described by a piecewise testable formula:

$$\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma} \wedge \neg h..\acute{\sigma} \wedge \neg\acute{\ell}..h \wedge \neg\ell..\acute{\ell}$$

The pattern can also be described with multitier reverse definiteness, which reads more faithfully to the English description. First, the one-stress constraint applies: $[\rtimes\acute{\sigma}\ltimes]_{\acute{\sigma}}$. Next, let us introduce implication: the logical expression $\varphi \rightarrow \psi$ asserts that ψ must be true if φ is true. This is equivalent to $\psi \vee \neg\varphi$, and therefore expressible in our propositional logic. Now, if a heavy syllable exists (if the heavy tier is nonempty), then the leftmost heavy syllable receives stress: $[\neg\rtimes\ltimes]_{\{h,\acute{h}\}} \rightarrow [\rtimes\acute{h}]_{\{h,\acute{h}\}}$. And if there is

[7] We cannot, however, use $\rtimes\acute{\ell}$, as this precludes word-initial heavy syllables. If the positive form is preferred, one might use $(\rtimes\acute{\ell} \vee \neg\acute{\ell})$: a stressed light syllable appears word-initially or not at all.

no heavy syllable (if the heavy tier is empty), then the leftmost (light) syllable receives stress: $[\rtimes\ltimes]_{\{h,\acute{h}\}} \rightarrow \rtimes\acute{\ell}$. All together, the multitier description is as follows.

$$[\rtimes\acute{\sigma}\ltimes]_{\acute{\sigma}} \wedge ([\neg\rtimes\ltimes]_{\{h,\acute{h}\}} \rightarrow [\rtimes\acute{h}]_{\{h,\acute{h}\}}) \wedge ([\rtimes\ltimes]_{\{h,\acute{h}\}} \rightarrow \rtimes\acute{\ell})$$

This expression naturally mirrors the English description of the pattern: there is exactly one stress, and if there is a heavy syllable then the leftmost such syllable is stressed, else (if there is no heavy syllable) then the leftmost syllable overall is stressed (and light). As each factor is a left-anchored substring, demonstrates that the pattern is multitier reverse definite. The expression can be simplified as follows.

$$[\rtimes\acute{\sigma}\ltimes]_{\acute{\sigma}} \wedge ([\rtimes\acute{h}]_{\{h,\acute{h}\}} \vee ([\rtimes\ltimes]_{\{h,\acute{h}\}} \wedge \rtimes\acute{\ell}))$$

This pattern is not strictly local, as $\ell^k\acute{h}\ell^k h\ell^k$ is accepted while $\ell^k h\ell^k\acute{h}\ell^k$ is rejected, despite the two sharing the same set of substrings of length up to k. However, when viewed from a long-distance perspective using order or tiers, it is no more complex than stress-initial with culminativity. It is piecewise testable ($\mathcal{J}$) and multitier reverse definite ($\mathcal{BTK}$).

3.2. *Stress rightmost heavy, else rightmost*

Golin, another Trans-New Guinea language, has the opposite stress pattern of Amele. In Golin, a syllable is heavy if it bears high tone, else it is light. Consider the following words, from Bunn & Bunn (1970: 5).

(5)	a.	$(hh\acute{h})$	[ówáˈré]	‘bat’
	b.	$(\ell\ell\acute{h})$	[oniˈbá]	‘snake’
	c.	$(h\acute{h}\ell)$	[góˈmági]	‘type of sweet potato’
	d.	$(\acute{h}\ell\ell)$	[ˈákola]	‘wild fig tree’
	e.	$(\ell\acute{\ell})$	[keˈba]	‘sweet potato’

In this default-to-same pattern, as described by Bunn & Bunn (1970: 4–5), there is exactly one primary stress in each word, which occurs on the final heavy syllable of each word, if such a syllable exists, else on the final (light) syllable. The formulae that express this pattern are the reversals of the formulae that express the stress pattern of Amele. Using order, it is piecewise testable:

$$\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma} \wedge \neg\acute{\sigma}..h \wedge \neg h..\acute{\ell} \wedge \neg\acute{\ell}..\ell$$

Using tiers, it is multitier definite:

$$[\rtimes\acute{\sigma}\ltimes]_{\acute{\sigma}} \wedge ([\acute{h}\ltimes]_{\{h,\acute{h}\}} \vee ([\rtimes\ltimes]_{\{h,\acute{h}\}} \wedge \acute{\ell}\ltimes))$$

Again, the multitier expression more directly reflects the English description of the pattern: there is exactly one stress, and either the rightmost heavy syllable receives stress or there is no such syllable and stress defaults to the rightmost syllable overall.

These default-to-same unbounded patterns have the same complexity as culminative bounded stress patterns. Both are piecewise testable and either multitier definite or multitier reverse definite, depending on which edge is the relevant boundary. The upcoming default-to-opposite patterns join the two branches, attending to both boundaries simultaneously.

3.3. Stress leftmost heavy, else rightmost

Consider the following words of Kwak'wala, a Wakashan language of Canada, per Grubb (1969: 46).

(6)	a.	$(\ell\acute{\ell})$	[siˈkʲʼæ]	'five'
	b.	$(\ell\acute{h}hh)$	[siˈkʲʼæːgiːju]	'fifteen'
	c.	$(\acute{h}hh\ell)$	[ˈsiːkʲæːgɔːla]	'twenty-five'

Heavy syllables are those containing long vowels, and stress falls on the leftmost heavy syllable if such a syllable exists, else on the rightmost syllable (Grubb, 1969: 46).

Like the default-to-same unbounded patterns of Amele and Golin, this pattern is piecewise testable. As before, one-stress applies: $\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma}$. If there is a heavy syllable, then there is no stressed light syllable; a pair of forbidden subsequences expresses this: $\neg h..\acute{\ell} \wedge \neg\acute{\ell}..h$. Further, an unstressed heavy syllable cannot precede a stressed heavy: $\neg h..\acute{h}$. This can merge with the first conjunct of the previous expression, yielding: $\neg h..\acute{\sigma} \wedge \neg\acute{\ell}..h$. Finally, if there is no heavy syllable, then some light syllable receives stress. To ensure that it is the final such syllable, forbid the occurrence of another (light, unstressed) syllable following a stressed light syllable: $\neg\acute{\ell}..\ell$. This combines with the second conjunct of the previous expression. All together, the following expression describes the stress pattern of Kwak'wala and demonstrates its piecewise testability.

$$\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma} \wedge \neg h..\acute{\sigma} \wedge \neg\acute{\ell}..\sigma$$

A tier-based characterisation more closely follows the English description of the pattern. Again, one-stress applies. Then, either the heavy tier begins with stress – that is, heavy syllables appear and the first one is stressed – or there are no heavy syllables and the rightmost syllable is stressed.

$$[\rtimes\acute{\sigma}\ltimes]_{\acute{\sigma}} \wedge ([\rtimes\acute{h}]_{\{h,\acute{h}\}} \vee ([\rtimes\ltimes]_{\{h,\acute{h}\}} \wedge \acute{\ell}\ltimes))$$

This logical formula uses both tier-prefixes and -suffixes. The pattern is not multitier (reverse) definite, but it is multitier generalised definite.

3.4. Stress rightmost heavy, else leftmost

The other default-to-opposite pattern is attested in Chuvash, a Turkic language. Consider the following words from Krueger (1961: 86–87).

(7)	a.	$(h\acute{h})$	[laˈʃa]	'horse'
	b.	$(\ell h\acute{h})$	[kămaˈka]	'stove'

c.	$(\ell\acute{h}\ell\ell)$	[ĕʃˈlerĕmĕr]	‘we worked’
d.	$(\acute{h}\ell\ell\ell)$	[ˈkalătːămăr]	‘we would say’
e.	$(\acute{\ell}\ell\ell)$	[ˈĕʃlĕpĕr]	‘we shall work’

Here, syllables containing ‘[ă]’ or ‘[ĕ]’ are light, while all others are heavy. Per Krueger (1961: 86), primary stress falls on the rightmost heavy syllable, if such a syllable exists, else on the leftmost syllable. This is the reversal of Kwak’wala stress. The classification is the same: it is piecewise testable and multitier generalised definite. The formulae may be adapted by way of reversal. Using subsequences it is as follows.

$$\acute{\sigma} \wedge \neg\acute{\sigma}..\acute{\sigma} \wedge \neg\acute{\sigma}..h \wedge \neg\sigma..\acute{\ell}$$

Using tier-affixes it is as follows.

$$[\rtimes\acute{\sigma}\ltimes]_{\acute{\sigma}} \wedge ([\acute{h}\ltimes]_{\{h,\acute{h}\}} \vee ([\rtimes\ltimes]_{\{h,\acute{h}\}} \wedge \rtimes\acute{\ell}))$$

All four unbounded stress patterns of this basic four-way typology are easily captured using two kinds of long-distance constraints. Using order, all are piecewise testable. Using tiers, default-to-same patterns are multitier (reverse) definite, while default-to-opposite patterns require multitier generalised definiteness. The higher complexity arises from both tier-prefixes and tier-suffixes being relevant in the latter.

3.5. Other stress patterns

For completeness, the 107 distinct patterns accompanied by automata in the StressTyp2 database (Goedemans *et al.*, 2015) were classified using the Language Toolkit (Lambert, 2024). Files executing this analysis are included as supplementary material. Forty-nine patterns are piecewise testable. Of the remaining fifty-eight, fifty-two are strictly local. Sixty of the total patterns are multitier generalised definite, including all but one of the piecewise testable patterns. The excluded one is index 135, an analysis of Bhojpuri. However, index 134 is another analysis of Bhojpuri, which is piecewise testable, multitier definite, and strictly local. One might conjecture then that index 134 is a better analysis and that unbounded stress patterns universally lie in the intersection $\mathcal{BTLI} \cap \mathcal{J}$ of multitier generalised definiteness and piecewise-testability.

Only three patterns are not strictly local, piecewise testable, nor multitier generalised definite: Yidin and two descriptions of lects of Arabic. The first is a cooccurrence of a default-to-same unbounded pattern with secondary-stress alternation (Goedemans *et al.*, 2015). As alternation is strictly local, this combination is MULTITIER LOCALLY TESTABLE, and still propositional. This class is the Boolean closure of tier-based strictly local, allowing not only tier-affixes but also internal substrings on tiers. The two Arabic patterns describe secondary-stress alternation as well, but state that secondary stress does not surface, lifting the patterns from strictly local to properly regular. If the secondary stress were shown to be present, then these two patterns would be strictly local. See Heinz (2009) for additional discussion on these lects.

4. Harmony patterns

Harmony is another common kind of long-distance pattern. Heinz (2010) describes both symmetric and asymmetric patterns, although more complicated cases exist. One such case is the backness harmony in Uyghur, which is cited for being difficult to analyse with piecewise, local, or tier-local constraints (Mayer & Major, 2018). In the remainder of this section, we analyse each of these with respect to the classes studied here. This analysis creates an interesting inversion, with Uyghur backness harmony lying at a lower complexity level than asymmetric harmony under multitier analysis.

4.1. Symmetric harmony and Navajo

Navajo, an Athabaskan language, exhibits regressive symmetric harmony in sibilants. That is, at any distance before [−anterior] sibilants like ‘ʃ’, [+anterior] sibilants like ‘s’ become [−anterior], and vice versa. The following words from Sapir & Hoijer (1967: 15) demonstrate this.

(8) a. /sì-ʔã́/ → [sìʔã́] ‘a round object lies’
b. /sì-ɣìʃ/ → [ʃìɣìʃ] ‘it is bent, curved’
c. /ʃì-lĩ́ːʔ/ → [ʃìlĩ́ːʔ] ‘my horse’
d. /ʃì-sází/ → [sìsází] ‘my ancestor’

The result is that in surface forms, sibilants will all be [+anterior], represented by ‘s’, or they will all be [−anterior], represented by ‘ʃ’. In short, words do not contain sibilants that disagree in anteriority.

Heinz (2010) describes this in a piecewise testable manner using forbidden subsequences: words contain neither ‘s..ʃ’ nor ‘ʃ..s’. As a logical expression, this is written as follows: ¬s..ʃ ∧ ¬ʃ..s.

Alternatively, we could describe the pattern using tiers. Either the [+anterior] tier is empty, or the [−anterior] tier is empty (or both): $[\rtimes\ltimes]_{\mathrm{s}} \vee [\rtimes\ltimes]_{\mathrm{ʃ}}$. As this involves tier-words over different tiers, the pattern is multitier co/finite.

It is not strictly local, as every substring of length up to k that appears in the rejected word ‘$\mathrm{a}^{k}\mathrm{sa}^{k}\mathrm{ʃa}^{k}$’, also appears in either ‘$\mathrm{a}^{k}\mathrm{sa}^{k}$’ or ‘$\mathrm{a}^{k}\mathrm{ʃa}^{k}$’, which should be accepted. If any of the substrings of the first is forbidden, then one or the other of the latter would also have to be rejected. However, it is tier-based strictly local: $[\neg \mathrm{sʃ} \wedge \neg \mathrm{ʃs}]_{\{\mathrm{s},\mathrm{ʃ}\}}$. This use of internal factors is necessary for single-tier description; while the pattern is both *multitier* definite and *multitier* reverse definite, it is not tier-based (reverse) definite.

4.2. Asymmetric harmony and Tsuut’ina

In the Tsuut’ina language, another Athabaskan language, there is an asymmetric harmony in the sibilants. Like in Navajo, [+anterior] sibilants become [−anterior] at any distance preceding [−anterior] sibilants, but, unlike in Navajo, the reverse does not happen. Consider the following words from Cook (1978: 26).

(9) a. /jí-s-ɣá/ → [jísɣá] ‘I killed them’

b. /nā-s-ɣátʃ/ → [nāʃɣátʃ] ‘I killed them again’

c. /sí-tʃíz+àˀ/ → [ʃítʃídzàˀ] ‘my duck’

Notice the ‘z’ following ‘ʃ’ in example 9c, demonstrating that the assimilation is asymmetric, that [+anterior] sibilants do not trigger harmony.

Per Heinz (2010), like symmetric harmony in Navajo, asymmetric harmony is piecewise testable. One need forbid only one subsequence: ¬s..ʃ. This is tier-based strictly local as well: $\neg[\mathrm{s}\int]_{\{\mathrm{s},\int\}}$.

However, unlike symmetric harmony, this pattern is not multitier generalised definite. The use of internal substrings or subsequences is necessary. To demonstrate this, we can find parameterised words that share the same k-prefix on every tier and the same k-suffix on every tier, where one satisfies the constraint and the other does not. For $k = 1$, we choose the accepted word ‘ʃʃss’ and the rejected word ‘ʃsʃs’. There are three tiers to consider: on tiers containing both ‘s’ and ‘ʃ’, the shared 1-prefix is ‘ʃ’ and the shared 1-suffix is ‘s’. On tiers containing only ‘ʃ’ or only ‘s’, the shared affix is ‘ʃ’ or ‘s’, respectively. The remaining tier, the empty tier, never contains any material.

For $k = 2$, we choose the accepted word ‘ʃʃʃsss’ and the rejected word ‘ʃʃsʃss’. On tiers containing both kinds of sibilants, the shared 2-prefix is ‘ʃʃ’ and the shared 2-suffix is ‘ss’. On tiers containing only one kind, x, the shared affixes are x^2.

In general, ‘$\int^k \int \mathrm{s}\mathrm{s}^k$’ is accepted and ‘$\int^k \mathrm{s}\int \mathrm{s}^k$’ is rejected. On tiers containing both kinds of sibilants, the shared prefix is $\int^k$ and the shared suffix is s^k. On tiers containing only one kind, x, the shared affixes are x^k. The tier-prefix–tier-suffix combinations are not sufficient information to make the required distinctions. The parameterised words witnessing this fact were automatically discovered in software using the algebraic techniques of §6.

While symmetric harmony in Navajo is multitier co/finite, asymmetric harmony in Tsuut’ina is not even multitier generalised definite because its description requires subsequences or internal substrings. With tiers, the pattern requires at least tier-based strict locality.

4.3. Uyghur backness harmony

Mayer & Major (2018) describe a pattern in the Uyghur language, in which morphological suffix forms are determined based on harmonizing vowels, if any, in the stem, else from harmonizing consonants in the stem, if any. They demonstrate that this pattern does not fall into any of the subregular classes commonly cited by computational linguists, nor even in the highly complex structure-sensitive multiple-tier-based strictly local class of De Santo & Graf (2019). Some pertinent examples are as follows, all involving the /-DA/ locative case suffix, taken from Mayer & Major (2018).

(10)	a.	[aʁinædæ]	‘on the friend’
	b.	[mæʃqtæ]	‘on the exercise’
	c.	[qoichida]	‘on the shepherd’
	d.	[rakta]	‘on the shrimp’
	e.	[gezittæ]	‘on the newspaper’

f.	[qirʁizda]	‘on the Kyrgyz’
g.	[itta]	‘on the dog’
h.	[bizdæ]	‘on us’

The following kinds of segments are relevant. Dorsal consonants are either front (‘k’ and ‘g’) or back (‘q’ and ‘ʁ’). Vowels also come in front (‘y’, ‘ø’, and æ) and back (‘u’, ‘o’, and ‘a’) harmonizing varieties, but there are also two transparent vowels: ‘i’ and ‘e’.

Per Mayer & Major (2018), the backness of the morphological suffix is determined as follows:

1. If the stem contains a harmonizing vowel, then the morphological suffix agrees in backness with the rightmost such vowel, as demonstrated in examples 10a–10d.
2. If the stem contains no harmonizing vowels but does contain dorsal consonants, then the morphological suffix agrees in backness with the rightmost such consonant, as demonstrated in examples 10e and 10f.
3. Otherwise, as shown in examples 10g and 10h, this rule does not determine backness. Mayer & Major (2018) state that such forms are arbitrarily specified for backness, with a statistical tendency to be treated as back.

In this analysis, the relevant symbol types are as follows:

$$V_f = \{\text{y}, \text{ø}, \text{æ}\} \qquad V_b = \{\text{u}, \text{o}, \text{ɑ}\} \qquad C_f = \{\text{k}, \text{g}\} \qquad C_b = \{\text{q}, \text{ʁ}\}$$

As the harmony affects only the morphological suffix, we assume, like Mayer & Major (2018), that segments in the morphological suffix are marked as such.[8] Then S_f is $V_f \cup C_f$ marked for being in the morphological suffix, and S_b is analogous.

Mayer & Major (2018) show that this pattern is not (tier-based) strictly local, nor is it piecewise testable, nor does it lie in a number of more complex subregular classes. It is not even in the highly complex STRUCTURE-SENSITIVE MULTIPLE-TIER-BASED STRICTLY LOCAL class of De Santo & Graf (2019), which allows symbols to be conditionally projected based on their local environment, although it still uses only restricted propositional logic. This left STAR-FREE as the only known viable class, which allows for the full first-order logic with order (McNaughton & Papert, 1971).

However, this pattern is multitier definite. The English description can be translated into a series of mutually-exclusive implications.

1. If the stem contains a harmonizing vowel, then the morphological suffix agrees in backness with the rightmost such vowel:

   (a) $[V_f \ltimes]_{V_f \cup V_b} \rightarrow [\rtimes\ltimes]_{S_b}$
   (b) $[V_b \ltimes]_{V_f \cup V_b} \rightarrow [\rtimes\ltimes]_{S_f}$

2. If the stem contains no harmonizing vowels but does contain dorsal consonants, then the morphological suffix agrees in backness with the rightmost such consonant:

   (a) $([\rtimes\ltimes]_{V_f \cup V_b} \wedge [C_f \ltimes]_{C_f \cup C_b}) \rightarrow [\rtimes\ltimes]_{S_b}$
   (b) $([\rtimes\ltimes]_{V_f \cup V_b} \wedge [C_b \ltimes]_{C_f \cup C_b}) \rightarrow [\rtimes\ltimes]_{S_f}$

[8] An alternative analysis not explored here would leave segments unmarked and instead insert an explicit morpheme boundary symbol.

3. Otherwise, this rule does not determine backness.

A multitier definite formula representing this pattern is the conjunction of these constraints in (1) and (2). All involve only tier-suffixes.

Unlike for asymmetric harmony, internal substrings are not needed. When using subsequences, Uyghur backness harmony appears significantly more complex than both symmetric and asymmetric harmony. Under multitier analysis, it is multitier definite, lying strictly between symmetric and asymmetric harmony in terms of complexity.

This inversion of complexity relationships is not uncommon when changing the kind of logic available. Recall from §2.3 that accounting for the culminative stress-final constraint is impossible with full propositional logic over suffixes, but possible with just restricted propositional logic over substrings. It is also possible with propositional logic over tier-suffixes. Choice of representation will affect, and possibly even invert, complexity relationships.

5. Tone patterns

Autosegmental representations as used by Jardine (2017) are likely to provide a better description of tonal patterns than pure string-languages. Nevertheless, in this section we provide propositional analyses for the patterns that Jardine (2020) cited to motivate a class of MELODY LOCAL languages. Some are more amenable to order-based analysis, while others are more simply analysed in terms of tiers. Throughout this section, the alphabet has only two symbols: ℓ and h for low and high tone, respectively. We concern ourselves only with tone strings, not with their associations to segmental content.

5.1. High-tone plateauing in Luganda

First, we examine the high-tone plateauing of Luganda, a Bantu language. Tones can be underlyingly high or unspecified. Following Jardine (2016), we take what Hyman & Katamba (2010) call the 'intermediate' forms to be the output of the phonology. Boundary tones and their effects will not be discussed. There are several other interesting phenomena involved in the full description of Luganda tone, but here we analyse only the high-tone plateauing, as it presents a challenge for multitier analysis.

Consider the following words and phrases, taken from Hyman & Katamba (2010: 71–72; see also Jardine 2016: 252).

(11)	a.	($\ell\ell\ell$)	/ki-tabo/ →	[ki̇̀tàbò]	'book'
	b.	($\ell h\ell$)	/ki-kópo/ →	[ki̇̀kópò]	'cup'
	c.	($hhh\ell$)	/bá-ki-láb-a/ →	[báki̇́lábà]	'they see it'
	d.	($\ell\ell h\ell$)	/ki-siki̇́/ →	[ki̇̀si̇̀ki̇̂]	'log'

Following Jardine (2016), we do not distinguish 'unspecified' from 'low' in output forms.

The shape of the melody arises from two key constraints. First, high tones are not obligatory, but when they do appear, there can be at most one high span; this is the UNBOUNDED TONE PLATEAUING of primary interest to this section. This generalises culminativity from limiting symbol-count to instead limit span-count. Second, the form never ends on a high tone. Notice the falling tone in example 11d that arises from repairing violations of this constraint.

The unbounded tone plateauing constraint itself can be described by a single forbidden subsequence: $\neg h..\ell..h$. The requirement that forms not end on a high tone is definite: $\neg h\ltimes$. In isolation, this latter constraint would not be piecewise testable, as the accepted word $(\ell h)^k\ell$ and the rejected word $(\ell h)^k$ have the same set of length-k subsequences. However, just as the bounded stress patterns of §2 are strictly local despite enforcing a seemingly nonlocal constraint, this finality condition can be enforced by piecewise testable constraints because there is at most a single high span. If there is a high tone, then there must be a later low tone; because no further high tones may follow that low tone, the form necessarily ends low: $h \rightarrow h..\ell$. All together, this aspect of Luganda tone is defined by the following piecewise testable formula.

$$\neg h..\ell..h \wedge (h \rightarrow h..\ell)$$

Neither unbounded tone plateauing nor this pattern as a whole is multitier generalised definite. On every tier, the accepted word $\ell^k hh\ell^k$ and the rejected word $\ell^k h\ell h\ell^k$ have the same tier-k-affixes.

5.2. *Prinmi*

Per Ding (2006: 13), the pitch-accent system of Prinmi is characterised by lexically selecting a position for high tone within a domain (a morpheme or a span of adjacent morphemes) and lexically specifying whether this high tone spreads progressively onto the next syllable. All possible variations are attested in disyllabic through quadrisyllabic domains (Ding, 2006: 14):

(12)	a.	$(h\ell)$	[bɨ́ gè]	‘as for honey’
	b.	(hh)	[bɨ́ gé]	‘as for sun’
	c.	(ℓh)	[tõ̀pú]	‘donkey’
	d.	$(h\ell\ell\ell)$	[bɨ́bʴõ̀bʴõ̀ gè]	‘as for roasted flour with honey’
	e.	$(hh\ell\ell)$	[bɨ́ɬípɜ̀tsɨ̀]	‘sunflower’
	f.	$(\ell h\ell\ell)$	[tʃ’ìn̥ĩ́dʒjẽ̀ ɹə̀]	‘dog-nose groups’
	g.	$(\ell hh\ell)$	[tõ̀púmɜ́ɬè]	‘donkey tail’
	h.	$(\ell\ell h\ell)$	[dʒjõ̀dʒɨ̀mɜ́ɬè]	‘buffalo tail’
	i.	$(\ell\ell hh)$	[ɹə̀tʃɨ̀ʃṍ gé]	‘as for clean liquor’
	j.	$(\ell\ell\ell h)$	[də̀ɹ̥ə̀ɹ̥ĩ̀ sí]	‘concentrated’

As noted by Jardine (2020), an order-based description invokes the same constraint as unbounded tone plateauing in Luganda. A low tone does not appear between two high tones in a domain: $\neg h..\ell..h$. This guarantees that there is at most one high span. It is obligatory: h. To limit its length to a maximum of two syllables, a third high tone in a domain is forbidden: $\neg h..h..h$. All together, this pitch-accent system is piecewise testable as demonstrated by the following propositional formula.

$$h \wedge \neg h..\ell..h \wedge \neg h..h..h$$

Like the system in Luganda, this cannot be captured using multitier generalised definiteness. The same words witness this nonmembership as for high-tone plateauing: $\ell^k hh\ell^k$ is valid but $\ell^k h\ell h\ell^k$ is not, despite the two having the same k-affixes on every tier. On the other hand, the analysis by Ding (2006) assumes maximally quadrisyllabic domains with a great deal of compounding; this restriction would make the system co/finite.

5.3. *Arigibi*

That is not to say that no tonal patterns are multitier generalised definite. Like Prinmi, Arigibi allows only one high-span and incorporates a length-limit on this high-span. However, in Arigibi, the limit is a single mora Jardine (2020). The following words from Donohue (1997: 368) demonstrate the allowed configurations from dimoraic through quadrimoraic forms:

(13)	a.	$(\ell\ell)$	[tùtù]	‘long’
	b.	$(h\ell)$	[nímò]	‘louse’
	c.	(ℓh)	[ùmú]	‘dog’
	d.	$(\ell\ell\ell)$	[vòvòʔò]	‘bird’
	e.	$(h\ell\ell)$	[ŋgíʔɛ̀pù]	‘heart’
	f.	$(\ell h\ell)$	[ìvíò]	‘sun’
	g.	$(\ell\ell h)$	[mùdɛ̀bɛ́]	‘claw’
	h.	$(\ell\ell\ell\ell)$	[ɛ̀làìlà]	‘hot’
	i.	$(h\ell\ell\ell)$	[núʔʌ̀tàmà]	‘bark’
	j.	$(\ell h\ell\ell)$	[ìdómàì]	‘eye’
	k.	$(\ell\ell h\ell)$	[tùnìʔʌ́ʔʌ̀]	‘all’
	l.	$(\ell\ell\ell h)$	[òlàʔòlá]	‘red’

There is at most one mora with high tone in the word, but words with no high tone are allowed. The position of the high tone is lexically specified. The resulting pattern, $\neg h..h$, is exactly analogous to culminativity in isolation, and as such it is piecewise testable and tier-based co/finite, as demonstrated in §2.3.

5.4. *Kagoshima Japanese*

The pitch-accent of Kagoshima Japanese has two lexically specified categories of words with respect to tone placement. There is one and only one high tone per word, and it appears either on the final mora or on the penultimate mora. Consider the following words, from Ding (2006: 28).

(14)	a.	(ℓh)	[sàrɯ́]	'monkey'
	b.	$(h\ell)$	[mízɯ̀]	'water'
	c.	$(\ell\ell\ell h)$	[ìrògàmí]	'coloured paper'
	d.	$(\ell\ell h\ell)$	[kàgàríbì]	'campfire'
	e.	(h)	[é]	'picture'
	f.	$(h\ell)$	[ê]	'handle'

Note that when a monosyllable should receive penultimate high tone, as in example 14f, this surfaces as a falling tone.

The resulting tone strings surface in forms that satisfy analogues of either the stress-penult or stress-final constraints, with culminativity (on high tone) enforced. Because each of these patterns is both piecewise testable and multitier definite, classes closed under the Boolean operations, we know that this pattern must also be both piecewise testable and multitier definite.

Using order, the constraint that exactly one high tone appears is enforced by the cooccurrence of culminativity and obligatoriness: $h \wedge \neg h..h$. The constraint on the placement of this high tone is simply that it is not followed by two (or more) other tones: $\neg h..\ell..\ell$. All together, the piecewise testable formula is as follows.

$$h \wedge \neg h..h \wedge \neg h..\ell..\ell$$

Using tier suffixes, the constraint that exactly one high tone appears is enforced with a tier-based co/finite constraint: $[\rtimes h \ltimes]_{\{h\}}$. Placement is enforced by requiring one of the two permissible suffixes: $h\ell\ltimes \vee h\ltimes$. All together, this is multitier definite, just like the culminative stress-final constraint.

$$[\rtimes h \ltimes]_{\{h\}} \wedge (h\ell\ltimes \vee h\ltimes)$$

5.5. Chuave

Chuave, a Trans-New Guinea language, exhibits OBLIGATORINESS, requiring the appearance of at least one mora with high tone. This is exemplified by the following words of up to three morae from Donohue (1997: 355).

(15)	a.	(h)	[fwí]	'salt'
	b.	(hh)	[gáán]	'child'
	c.	$(h\ell)$	[gáàm]	'skin'
	d.	(ℓh)	[kùbá]	'bamboo species'
	e.	(hhh)	[gíngódí]	'snore'
	f.	$(hh\ell)$	[dénkábù]	'mosquito'
	g.	$(h\ell h)$	[énùgú]	'smoke'
	h.	$(h\ell\ell)$	[kóìòm]	'wing'
	i.	(ℓhh)	[àmámó]	'yam species'
	j.	$(\ell h\ell)$	[kòmárì]	'before'

k. ($\ell\ell h$) [kòi̇̀jóm] ‘navel’

Every word contains at least one high span, but, as witnessed by example 15g, there can be more than one. There is no restriction on where the high tone falls, only that there must be one.

This is piecewise testable (and locally testable) as witnessed by the formula h. It is also tier-based co/finite, as witnessed by the formula $\neg[\rtimes\ltimes]_{\{h\}}$.

5.6. Karanga Shona

Jardine (2020) introduces a class, MELODY LOCAL, which captures all of the patterns of tone assignment discussed to this point.[9] Melody local analysis finds difficulty with Karanga Shona verb stems. Consider the following words from Odden (1984: 258), built from either the ℓ-toned root -bik- ‘cook’ or one of the h-toned roots -p- ‘give’ or -tór- ‘take’. The relevant domain for the present analysis, following Jardine (2020), is the verb stem, the material which follows the hyphen in the examples.

(16) hàndáka-

a.	(ℓh)	[-bìká]	‘I didn’t cook’
b.	($\ell h\ell$)	[-bìkísà]	‘I didn’t make cook’
c.	($\ell hh\ell$)	[-bìkísírà]	‘I didn’t make cook for’
d.	($\ell hh\ell\ell$)	[-bìkísísìrà]	‘I didn’t make cook for a lot’
e.	($\ell hh\ell\ell\ell$)	[-bìkísísìrànà]	‘I didn’t make cook a lot for each other’
f.	($\ell hh\ell\ell\ell\ell$)	[-bìkísírìsìsànà]	‘I didn’t make cook a lot for each other’
g.	(h)	[-pá]	‘I didn’t give’
h.	($h\ell$)	[-tórà]	‘I didn’t take’
i.	($h\ell h$)	[-tórèsá]	‘I didn’t make take’
j.	($hh\ell h$)	[-tóréserá]	‘I didn’t make take for’
k.	($hhh\ell h$)	[-tórésérànà́]	‘I didn’t make take for each other’
l.	($hhh\ell\ell h$)	[-tórésérèsàná]	‘I didn’t make take for each other a lot’
m.	($hhh\ell\ell\ell h$)	[-tórésérèsèsàná]	‘I didn’t make take for each other a lot’

At the tone level, there are seven fully specified words: ℓ, ℓh, $\ell h\ell$, h, $h\ell$, $h\ell h$ and $hh\ell h$. Longer words fall into one of two patterns: $\ell hh\ell\ell^*$ for ℓ-toned roots and $hhh\ell\ell^* h$ for h-toned roots (Jardine, 2020: 1166). Here we show that this surface pattern can be described propositionally. The seven fully specified words form a finite language; as the finite languages are a subclass of every class under consideration in this work, in the following discussion let φ_F be the formula of the appropriate form to specify the set $F = \{\ell, \ell h, \ell h\ell, h, h\ell, h\ell h, hh\ell h\}$ in the chosen logic. We focus on describing the patterns of longer words.

First, we present a piecewise testable expression. For ℓ-toned roots, the long words are of the form $\ell hh\ell\ell^*$. Such words always contain the $\ell..h..h..\ell$ subsequence, and

[9] Jardine (2020) also covers a locally testable pattern from Bemba that is neither piecewise testable nor multitier generalised definite. However, it is locally testable, so it can be defined by propositional logic over substrings.

this might be the entire string. Further, a third high tone may not appear: $\neg h..h..h$. With only these constraints, however, undesired low tones might intervene between the high tones. A word like $\ell h \ell h \ell$ would be accepted. To rule this out without rejecting valid forms, notice that after the two required low tones, no h may appear: $\neg \ell..\ell..h$. This guarantees that the two high tones that appear are adjacent. In sum, the piecewise testable representation for ℓ-toned roots is as follows.

$$L_p = \ell..h..h..\ell \wedge \neg h..h..h \wedge \neg \ell..\ell..h$$

For h-toned roots, the long words are of the form $hhh\ell\ell^* h$. Such words always contain the $h..h..h..\ell..h$ subsequence, and a fifth high tone is forbidden: $\neg h..h..h..h..h$. What remains is to ensure that the low tones are not placed in undesirable positions. If any ℓ is misplaced, there will be an $\ell..h..\ell$ subsequence, so we simply forbid that. In sum, the piecewise testable representation for h-toned roots is as follows:

$$H_p = h..h..h..\ell..h \wedge \neg h..h..h..h..h \wedge \neg \ell..h..\ell$$

Then the entire pattern is $\varphi_F \vee L_p \vee H_p$.

Multitier analysis also takes advantage of the limited number of high tones. For ℓ-toned roots, which again have the form $\ell hh\ell\ell^*$, words begin with $\ell hh\ell$ and there are exactly two high tones.

$$L_m = \rtimes \ell hh\ell \wedge [\rtimes hh \ltimes]_{\{h\}}$$

For h-toned roots, which again have the form $hhh\ell\ell^* h$, words begin with $hhh\ell$, they end with ℓh, and there are exactly four high tones.

$$H_m = \rtimes hhh\ell \wedge \ell h \ltimes \wedge [\rtimes hhhh \ltimes]_{\{h\}}$$

In sum, the pattern is multitier generalised definite because both tier-prefixes and tier-suffixes are relevant, and the witnessing formula is $\varphi_F \vee L_m \vee H_m$.

Tiers can be eschewed in favour of internal substrings. The pattern is locally testable, captured by a propositional formula over substrings. Without getting lost in the details, the pattern is described by $\varphi_F \vee L_s \vee H_s$ where L_s and H_s are defined as follows.

$$L_s = \rtimes \ell hh\ell \wedge \neg h\ell h \wedge \neg \ell\ell h$$
$$H_s = \rtimes hhh\ell \wedge \ell h \ltimes \wedge \neg \ell h\ell \wedge \neg \ell hh$$

Like culminative bounded stress, this pattern is propositional in many ways. The surface patterns of Karanga Shona verb stems that prove difficult for melody local description are nonetheless simple to describe using substrings, subsequences, or tier-affixes.

6. Introduction to algebraic phonotactics

In this section, we briefly introduce algebraic tools and techniques that facilitated analyses performed in this work. We first describe how to construct an algebraic model of a language, both conceptually and mechanically. This algebraic model is known as the SYNTACTIC SEMIGROUP. From there, we revisit the classes discussed so far in this work; other than the strictly local class, all of them are definable algebraically by a system of equations. Systems of equations are provided for the definite, reverse definite and generalised definite classes as well as their multitier extensions. The algebraic model automatically provides a way to decide whether a given pattern belongs to a class. A language is in the class if and only if its algebraic model satisfies all of the equations, regardless of how variables are instantiated. If any instantiation fails, the associated words witness the nonmembership. In this way, given *any* logical description of a language, we can determine whether it is definable under a chosen kind of logic, and if not, why not. In other words, this section lays out the techniques that led to the analyses in the previous sections.

6.1. The syntactic semigroup

In phonology, we often use MINIMAL PAIRS to determine whether two phones represent distinct phonemes. For instance, Finnish has contrastive length on both vowels and consonants as demonstrated by the words [t̪uli] 'fire', [t̪uːli] 'wind' and [t̪ulːi] 'customs', differing only in length. The t̪__li environment distinguishes [u] from [uː], and the t̪u__i environment distinguishes [l] from [lː]. In algebra, we do the same thing, except that rather than looking for differences in meaning, we seek differences in acceptability. Given arbitrary strings x and y, we seek an environment $u\_\_v$ where uxv is accepted and uyv is rejected, or vice versa. If such an environment can be found, then x and y are distinct (Rabin & Scott, 1959). Otherwise, they are MYHILL-EQUIVALENT with respect to the language.

Definition 6.1. In a language L, two words x and y are MYHILL-EQUIVALENT with respect to L, written $x \equiv y$, if and only if for all u and v it holds that $uxv \in L$ whenever $uyv \in L$ and vice versa.

Recall the stress-penult constraint, without culminativity, from §2.1. This is satisfied by stressed monosyllables or by longer words whose penult is stressed, regardless of which other syllables bear stress. For analysis of this constraint, we used the two-letter alphabet $\Sigma = \{\sigma, \acute{\sigma}\}$. Clearly, $\acute{\sigma}$ and σ are distinct, because in the trivial context, __, the former is accepted while the latter is rejected. We have [ˈkin] 'they drink' but not [kin]. Then σ and $\sigma\sigma$ are distinguished by the $\acute{\sigma}\_\_$ context, as $\acute{\sigma}\sigma$ satisfies the constraint while $\acute{\sigma}\sigma\sigma$ does not. But no context can distinguish $\acute{\sigma}\sigma\sigma$ from $\sigma\sigma$; the only accepting contexts $u\_\_v$ for either string involve v itself being two or more syllables long with a stressed penult, and the other string is necessarily accepted in the same context. We have $\acute{\sigma}\sigma\sigma \equiv \sigma\sigma$.

The equivalence class of a string is the set of all strings to which it is equivalent, including that string itself. For example, with respect to the stress-penult constraint, the equivalence class of $\sigma\sigma$, written $[\sigma\sigma]$, is the set of all strings which end with $\sigma\sigma\ltimes$: $[\sigma\sigma] = \Sigma^*\sigma\sigma = \{\sigma\sigma, \sigma\sigma\sigma, \acute{\sigma}\sigma\sigma, \ldots\}$. Similarly, the equivalence class of $\acute{\sigma}\sigma$ is $[\acute{\sigma}\sigma] = \Sigma^*\acute{\sigma}\sigma = \{\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \acute{\sigma}\acute{\sigma}\sigma, \ldots\}$. The SYNTACTIC SEMIGROUP is the collection of equivalence classes. The most important property of Myhill-equivalence is that concatenation is well-defined not only at the string level but also at the equivalence-class level (Rabin & Scott, 1959). That is, for any choice of class representatives, their concatenation will always be in the same class as it would be if other representatives were chosen: if $a \equiv b$ and $c \equiv d$ then $ac \equiv bd$. For example, we have $\acute{\sigma}\sigma \equiv \sigma\acute{\sigma}\sigma$ and $\sigma\sigma\sigma \equiv \acute{\sigma}\sigma\sigma$, so it must be the case that $\acute{\sigma}\sigma\ \sigma\sigma\sigma \equiv \sigma\acute{\sigma}\sigma\ \acute{\sigma}\sigma\sigma$, and this is true; each ends with $\sigma\sigma\ltimes$ so each is in $[\sigma\sigma]$.

Proposition 6.2. *If $a \equiv b$ and $c \equiv d$ then $ac \equiv bd$.*

Proof. Suppose that $a \equiv b$ and $c \equiv d$. Because $a \equiv b$, it is the case that for any context $u\_\_v$, we have that uav is accepted if and only if ubv is accepted. Specifically, this holds for $v = cx$ for any x. That is, $u(ac)x$ is accepted if and only if $u(bc)x$ is accepted. In other words, $ac \equiv bc$.

Similarly, because $c \equiv d$, it is the case that for any context $u\_\_v$, we have that ucv is accepted if and only if udv is accepted. Specifically, this holds for $u = tb$ for any t. That is, $t(bc)v$ is accepted if and only if $t(bd)v$ is accepted. In other words, $bc \equiv bd$.

Putting these together, we have $ac \equiv bc \equiv bd$, proving the statement. □

This proves that we can concatenate equivalence classes as a whole in a way that makes sense. Every regular language partitions Σ^* into finitely many equivalence classes (Rabin & Scott, 1959). Algebraic analysis of subregular systems takes full advantage of this fact. Rather than face the daunting task of generalising properties over infinitely many strings, one need only investigate finitely many equivalence classes.

That was the conceptual overview of the syntactic semigroup. To construct it mechanically, begin with a DETERMINISTIC FINITE-STATE AUTOMATON representing the target language. There are many software packages that assist with this step, including the Language Toolkit of Lambert (2024) and *foma* from Hulden (2009). We use the Language Toolkit both because it is designed to work with the propositional logics discussed in this work and because it implements algebraic decision procedures. The automaton representation of the stress-penult constraint, without culminativity, is shown in Figure 1. The five states are represented by circles. Here they are numbered, for ease of reference. The arrow from nowhere points to the initial state, q_0, from which all computation begins. For each state, there is exactly one outgoing arrow for each symbol in the alphabet. These arrows represent the δ function: given a symbol x and a state q, $\delta_x(q)$ is the state reached by following the arrow labeled by symbol x from state q. This letter-based δ function recursively extends to a word-based $\hat{\delta}$ as follows. As a base case, $\hat{\delta}_\lambda(q) = q$, where λ represents the empty string. Longer strings are processed left-to-right: $\hat{\delta}_{xv}(q) = \hat{\delta}_v(\delta_x(q))$. Accepting states are indicated by extra thick borders; a word w is accepted if and only if $\hat{\delta}_w(q_0)$ is an accepting state.

Next we use the automaton to construct the syntactic semigroup. The functions $\hat{\delta}_w$ for nonempty words w exactly pick out the elements of the syntactic semigroup (McNaughton & Papert, 1971). That is, $u \equiv v$ if and only if $\hat{\delta}_u$ and $\hat{\delta}_v$ are the same function. To find them all, begin with the single-symbol words, and append symbols one at a time until no new elements are generated. For the stress-penult constraint as depicted in Figure 1, notice that $\hat{\delta}_\sigma$ maps states 1, 3 and 5 to state 3, and maps states 2 and 4 to state 5. For brevity we write this as $\hat{\delta}_\sigma = \langle 3, 5, 3, 5, 3\rangle$. The first item in the list is the destination from state 1, the second from state 2, and so on. We also

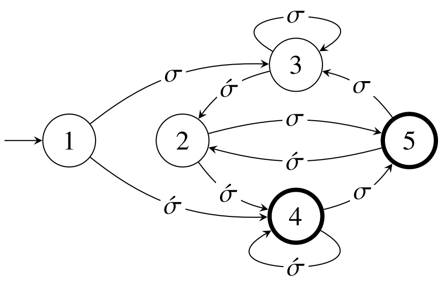

Figure 1. *The stress-penult constraint.*

Table 1. *Elements of the syntactic semigroup of stress-penult.*

w	$\hat{\delta}_w$	w	$\hat{\delta}_w$
σ	$\langle 3,5,3,5,3\rangle$	$\sigma\sigma\sigma$	$\langle 3,3,3,3,3\rangle = \hat{\delta}_{\sigma\sigma}$
$\acute{\sigma}$	$\langle 4,4,2,4,2\rangle$	$\sigma\sigma\acute{\sigma}$	$\langle 2,2,2,2,2\rangle = \hat{\delta}_{\sigma\acute{\sigma}}$
$\sigma\sigma$	$\langle 3,3,3,3,3\rangle$	$\sigma\acute{\sigma}\sigma$	$\langle 5,5,5,5,5\rangle = \hat{\delta}_{\acute{\sigma}\sigma}$
$\sigma\acute{\sigma}$	$\langle 2,2,2,2,2\rangle$	$\sigma\acute{\sigma}\acute{\sigma}$	$\langle 4,4,4,4,4\rangle = \hat{\delta}_{\acute{\sigma}\acute{\sigma}}$
$\acute{\sigma}\sigma$	$\langle 5,5,5,5,5\rangle$	$\acute{\sigma}\sigma\sigma$	$\langle 3,3,3,3,3\rangle = \hat{\delta}_{\sigma\sigma}$
$\acute{\sigma}\acute{\sigma}$	$\langle 4,4,4,4,4\rangle$	$\acute{\sigma}\sigma\acute{\sigma}$	$\langle 2,2,2,2,2\rangle = \hat{\delta}_{\sigma\acute{\sigma}}$
		$\acute{\sigma}\acute{\sigma}\sigma$	$\langle 5,5,5,5,5\rangle = \hat{\delta}_{\acute{\sigma}\sigma}$
		$\acute{\sigma}\acute{\sigma}\acute{\sigma}$	$\langle 4,4,4,4,4\rangle = \hat{\delta}_{\acute{\sigma}\acute{\sigma}}$

Table 2. *Concatenation table for stress-penult, brackets omitted.*

	σ	$\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
σ	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\sigma\sigma$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}\sigma$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$

have $\hat{\delta}_{\acute{\sigma}} = \langle 4,4,2,4,2\rangle$. Table 1 lists all possible words of length up to three and their associated functions. The three-letter words introduced no new functions. This means that the syntactic semigroup for the stress-penult constraint has only the six elements $[\sigma]$, $[\acute{\sigma}]$, $[\sigma\sigma]$, $[\sigma\acute{\sigma}]$, $[\acute{\sigma}\sigma]$ and $[\acute{\sigma}\acute{\sigma}]$.

The concatenation of $[x]$ and $[y]$ is $[xy]$; in terms of these functions, this is found via the composition $\hat{\delta}_y \circ \hat{\delta}_x$. Table 2 depicts the concatenation table where the element at row $[x]$, column $[y]$, is $[xy]$. At this point, we have constructed a semigroup mechanically from a deterministic finite-state automaton and presented its concatenation table. This algebraic object represents the language's structure, and the presentation as a concatenation table will be useful in the next section to demonstrate or refute membership in various classes.

6.2. Classification through equations

We can translate logical characterisations of certain classes of formal languages into algebraic characterisations. This section explains how this works with definite languages, beginning with k-definite before abstracting away from k to establish definiteness as a whole. In the k-definite languages, membership is determined by the final k symbols. For any nonempty strings $x_1 \ldots x_k$ and for any strings u and s, the strings $ux_1 \ldots x_k$ and $usx_1 \ldots x_k$ share the same final k symbols, so they must be treated identically.

Further, adding any suffix v preserves the property. That is, $usx_1 \ldots x_k v$ is accepted if and only if $ux_1 \ldots x_k v$ is as well. Recall the definition of Myhill-equivalence. What we have just argued is that in any k-definite language, for any string s and any nonempty strings $x_1 \ldots x_k$, we have $sx_1 \ldots x_k \equiv x_1 \ldots x_k$. In terms of the syntactic semigroup (which does not include the empty string), that means $[s][x_1] \ldots [x_k] = [sx_1 \ldots x_k] = [x_1 \ldots x_k]$ for all $[s]$ and $[x_i]$. When the x_i are individual letters, this shows that each word is equivalent to its k-suffix.

This property can determine whether a given regular language is k-definite, by exhaustively checking all instantiations of semigroup elements. If any instantiation fails to satisfy the equation, the language is not k-definite. Recall the syntactic semigroup of the stress-penult constraint, shown in Table 2, and consider $k = 1$. Notice that $[\sigma][\sigma] = [\sigma\sigma] \neq [\sigma]$. This establishes that the stress-penult constraint is not 1-definite, as for $s = x_1 = \sigma$, the property that holds for all 1-definite languages does not hold. But with $k = 2$ we can capture this stress-penult constraint. For every possible instantiation of s, x_1 and x_2 as nonempty strings, we find that $[s][x_1][x_2] = [x_1x_2]$. Every word is Myhill-equivalent to its own 2-suffix.

We write $\mathcal{D}_k = [\![sx_1 \ldots x_k = x_1 \ldots x_k]\!]$ to mean that $\mathcal{D}_k$ is the class of languages whose syntactic semigroups satisfy the equation $[s][x_1] \ldots [x_k] = [x_1] \ldots [x_k]$ for all possible variable instantiations as nonempty strings. At this point, we have only shown that this condition is necessary, not that it is sufficient. We still need to show the latter.

Proposition 6.3. *A language is k-definite if and only if it is in* $[\![sx_1 \ldots x_k = x_1 \ldots x_k]\!]$.

Proof. As argued above, this condition is necessary: all k-definite languages are in the class described by the equation. We now show that it is sufficient, that all languages satisfying the equation are k-definite. Suppose that L satisfies the equation for all instantiations of its variables. We want to show that if strings a and b have the same k-suffix, then either both are in L or neither is in L.

If a has length less than k, then the k-suffix of a is a itself. If this is also the k-suffix of b, then $a = b$ and they are certainly treated the same.

Otherwise, $a = a'x_1 \ldots x_k$ and $b = b'x_1 \ldots x_k$ for some strings a' and b' and for some symbols x_i. In particular, when a' is nonempty, s is instantiated as a' and the x_i are instantiated accordingly, we have that $a'x_1 \ldots x_k \equiv x_1 \ldots x_k$, which further means that for all strings u and v, we have that $ua'x_1 \ldots x_k v$ is in L if and only if $ux_1 \ldots x_k v$ is in L. (This holds even if a' is the empty string.) In particular, this holds when u and v are the empty string. That is, $a = a'x_1 \ldots x_k$ is accepted if and only if $x_1 \ldots x_k$ is as well. Similarly, $b = b'x_1 \ldots x_k$ is accepted if and only if $x_1 \ldots x_k$ is as well.

Taken together, we have $a \in L$ if and only if $b \in L$ whenever the two share the same k-suffix. By Definition 2.1 then, L is k-definite. □

To determine whether a language is k-definite, construct its syntactic semigroup and check its concatenation table. If the equation always holds, then it is k-definite. Otherwise, there is some instantiation of variables where it does not hold. For example, in showing that the stress-penult constraint is not 1-definite, we found that $[\sigma][\sigma] \neq [\sigma]$, that $\sigma\sigma \not\equiv \sigma$. This is because some context distinguishes the two strings. One finds the distinguishing context by finding a state for which $\hat{\delta}_{\sigma\sigma}$ and $\hat{\delta}_{\sigma}$ differ in their output. Looking at Table 1, we see that $\hat{\delta}_{\sigma}(4) = 5$ while $\hat{\delta}_{\sigma\sigma}(4) = 3$. The left side of the context is then any string leading to state 4; we choose $\acute{\sigma}$ because it is short. The right side of the context is any string which leads one of the output states to acceptance and the other to rejection; in this case, our right side can be the empty string, because state 5 is accepting while state 3 is rejecting. The $\acute{\sigma}\_\_$ context distinguishes σ and $\sigma\sigma$ from one another: $\acute{\sigma}\sigma$ is accepted while $\acute{\sigma}\sigma\sigma$ is not. This shows algebraically that stress-penult is 2-definite but not 1-definite, and mechanically derives a pair of words witnessing this nonmembership.

6.2.1. Definiteness in the limit

For any chosen factor-size k, k-definiteness is queried using the equational characterisation $\mathcal{D}_k = [\![sx_1 \ldots x_k = x_1 \ldots x_k]\!]$. There are two apparent weaknesses of this approach. First, as k grows, the number of variables to check grows alongside it, and therefore so too does the number of checks to make. It can take a long time to decide whether a given pattern is in $\mathcal{D}_{1\,000\,000}$! Second, we often wish to know whether a pattern is definite *at all*, not just whether it is k-definite for a specific k. In that case, even if we do determine that the pattern is not in $\mathcal{D}_{1\,000\,000}$, how do we know that it is not in $\mathcal{D}_{1\,000\,001}$? This section, following Straubing (1985), presents a two-variable characterisation of $\mathcal{D}$ in general, abstracting away the parameter.

The key to this abstraction comes from IDEMPOTENT elements. These are the equivalence classes $[x]$ where $[xx] = [x]$. For example, in the stress-penult constraint as depicted in Table 2, the idempotents correspond to strings of length two: $[\sigma\sigma]$, $[\sigma\acute{\sigma}]$, $[\acute{\sigma}\sigma]$ and $[\acute{\sigma}\acute{\sigma}]$.

Definition 6.4. An element e of a semigroup is IDEMPOTENT if and only if $ee = e$.

Recall that the k-definite languages $\mathcal{D}_k$ are those that satisfy $[\![sx_1 \ldots x_k = x_1 \ldots x_k]\!]$. Specifically, if s is instantiated as $x_1 \ldots x_k$, then $[x_1 \ldots x_k] = [x_1 \ldots x_k][x_1 \ldots x_k]$. Because the x_i were arbitrary, it follows that every string of length at least k is idempotent. Moreover, any suffix-length can be saturated by any idempotent: if $[e] = [e][e]$, then a single $[e]$ is duplicable as desired: $[e] = [e][e] = [e][e][e] = \cdots = [e] \ldots [e]$. In any k-definite language then, if we have a string $[s]$ and an idempotent $[e]$, then $[s][e] = [s][e] \ldots [e] = [e] \ldots [e] = [e]$. Because definite languages $\mathcal{D}$ are those that are k-definite for some finite k, this characterises definiteness. This essentially restates a theorem of Straubing (1985).

Unfortunately, the equational system of characterisation does not allow for restrictions like 's ranges over all elements, but e ranges over only idempotents'. All variables must range over all elements. In arithmetic, there are real numbers, such as $\sqrt{2}$, that cannot be represented as a fraction of whole numbers, but we are still allowed to talk about $\sqrt{2}$ in a formula. In the same way, there are semigroup elements, such as our idempotents,

Table 3. *Concatenation table for stress-initial, brackets omitted.*

	σ	$\acute{\sigma}$
σ	σ	σ
$\acute{\sigma}$	$\acute{\sigma}$	$\acute{\sigma}$

that cannot be represented as a finite concatenation of variables, but we still want to be able to talk about them. Much like we can add the square root operation to numeric equations, we can add a new operation to semigroup equations. For every element $[x]$ of a finite semigroup, the sequence $[x]$, $[xx]$, $[xxx]$, and so on, is guaranteed to eventually reach a unique idempotent (Almeida, 1995: 72). We denote this idempotent x^{ω}. In a finite semigroup, this operation maps every element to an idempotent, and in particular every idempotent maps to itself, so all of them are covered.

Definition 6.5. The UNIQUE IDEMPOTENT POWER x^{ω} of x is the unique element of the form x^k such that $x^{\omega}x^{\omega} = x^{\omega}$.

For example, for the stress-penult constraint we might choose $x = [\sigma]$. Then x is not idempotent, because $x = [\sigma]$, $xx = [\sigma\sigma]$, and $\sigma\sigma \not\equiv \sigma$. But xx is idempotent, because $\sigma\sigma\sigma\sigma \equiv \sigma\sigma$. So $[\sigma]^{\omega} = [\sigma\sigma]$. Similarly, $[\acute{\sigma}]^{\omega} = [\acute{\sigma}\acute{\sigma}]$. The other four elements are idempotent and map to themselves. With this new operation, we restate the algebraic characterisation of definiteness.

Proposition 6.6. *A language is definite if and only if it is in* $\mathcal{D} = [\![sx^{\omega} = x^{\omega}]\!]$.

This equation has a consequence that is easy to verify with a concatenation table. In the concatenation table of a definite language, idempotents fill their respective columns. For the stress-penult constraint shown in Table 2, idempotents are the elements represented by length-two strings such as $[\sigma\sigma]$ and $[\sigma\acute{\sigma}]$. Each fills its own column; no other elements are present.

6.2.2. Reverse definiteness and generalised definiteness

The stress-initial constraint is not definite, but it is reverse 1-definite. Its syntactic semigroup contains two elements: $[\sigma]$ (the class of strings that begin with σ) and $[\acute{\sigma}]$ (the class of strings that begin with $\acute{\sigma}$). These elements concatenate as shown in Table 3.

Both elements are idempotent: $[\sigma]^{\omega} = [\sigma]$ and $[\acute{\sigma}]^{\omega} = [\acute{\sigma}]$. This is not definite, as it does not universally satisfy $[\![sx^{\omega} = x^{\omega}]\!]$. Say we have $s = [\acute{\sigma}]$ and $x = [\sigma]$. Plugging those assignments into the formula yields $sx^{\omega} = [\acute{\sigma}][\sigma]^{\omega} = [\acute{\sigma}][\sigma] = [\acute{\sigma}]$. But $x^{\omega} = [\sigma]^{\omega} = [\sigma]$, and $\acute{\sigma} \not\equiv \sigma$. These are not the same element, so the equation is not satisfied.

In the previous section, when we found an instantiation of variables that falsified an equation, the result was a pair of words that witnessed nonmembership. Recall that when we showed that the stress-penult constraint was not 1-definite, we found that $\sigma \not\equiv \sigma\sigma$. Then we found that these elements are distinguished in the $\acute{\sigma}\_\_$ context. This gave us the accepted word $\acute{\sigma}\sigma$ and the rejected word $\acute{\sigma}\sigma\sigma$ which share the same 1-suffix, witnessing nonmembership in the 1-definite class. In the same way, in showing that the stress-initial constraint is not definite at all, we found two elements, $[\acute{\sigma}][\sigma]^{\omega} = [\acute{\sigma}]$ and $[\sigma]^{\omega} = [\sigma]$, which are distinguished by the trivial context. An idempotent like

$[\sigma]^\omega$ can be duplicated as many times as desired without changing the value. We can just as well use $[\acute{\sigma}]([\sigma]^\omega)^k$ and $([\sigma]^\omega)^k$ for any positive whole number k, and this is what gives us our parameterised words. We have that $\acute{\sigma}\sigma^k$ is accepted but σ^k itself is rejected, despite sharing the same length-k suffix σ^k.

Definiteness is based on suffixes. Reverse definiteness is based on prefixes. The algebraic characterisations for the reverse definite languages are exactly reversed from those for definite languages.

Proposition 6.7 (Almeida, 1995: 90)**.** *A language is reverse k-definite if and only if it is in $\mathcal{K}_k = [\![x_1 \ldots x_k s = x_1 \ldots x_k]\!]$. In the limit, it is reverse definite if and only if it is in $\mathcal{K} = [\![x^\omega s = x^\omega]\!]$.*

For a reverse definite language, idempotents fill their respective rows in the concatenation table. In a language that is not reverse definite, there will be at least one idempotent whose row has at least one position that is not that same idempotent. For example in the stress-penult constraint, depicted in Table 2 on page 27 we have that $[\sigma\sigma]$ is idempotent but $[\sigma\sigma][\acute{\sigma}] = [\sigma\acute{\sigma}] \neq [\sigma\sigma]$, violating the reverse definiteness condition. The strings $\sigma\acute{\sigma}$ and $\sigma\sigma$ are distinguished in the $\_\sigma$ context, so our mechanically derived parameterised words witnessing nonmembership are the accepted word $(\sigma\sigma)^k\sigma\acute{\sigma}\sigma$ and the rejected word $(\sigma\sigma)^k\sigma\sigma\sigma$, which share the length-k prefix σ^k. In simple examples like this, choosing witnesses by hand is simple enough, but for more complex classes it is convenient to have this mechanical derivation available.

The generalised definite languages can attend to both prefixes and suffixes at the same time. It can be shown that these languages are $\mathcal{LI} = [\![x^\omega s z^\omega = x^\omega z^\omega]\!]$. When there is some sufficiently long prefix x^ω and some sufficiently long suffix z^ω, intervening material s is irrelevant.

This equation can be simplified by considering what happens when $z = x$; this gives $x^\omega s x^\omega = x^\omega x^\omega = x^\omega$. That is, any idempotent can be duplicated with arbitrary material stuffed between the pair. And the reverse holds; when there are two copies of an idempotent, even when spread apart by intervening content, the entire span can collapse to a single copy of that idempotent. We show that this two-variable equation implies that the original three-variable equation holds. Assume that $x^\omega s x^\omega = x^\omega$ holds for all instantiations of variables, and consider the element $x^\omega s z^\omega$ for a new idempotent z^ω which may not be equal to x^ω. Duplicate the z^ω idempotent and insert x^ω between: $x^\omega s z^\omega = x^\omega s z^\omega x^\omega z^\omega$. Then collapse the x^ω pair: $x^\omega s z^\omega x^\omega z^\omega = x^\omega z^\omega$. This shows algebraically that $[\![x^\omega s z^\omega = x^\omega z^\omega]\!]$ and $[\![x^\omega s x^\omega = x^\omega]\!]$ define the same class.

Proposition 6.8 (Straubing, 1985: 60)**.** *A language is generalised k-definite if and only if it is in $\mathcal{LI}_k = [\![x_1 \ldots x_k s x_1 \ldots x_k = x_1 \ldots x_k]\!]$. In the limit, it is generalised definite if and only if it is in $\mathcal{LI} = [\![x^\omega s x^\omega = x^\omega]\!]$.*

Finally, the co/finite languages are those that are both definite and reverse definite simultaneously.

Proposition 6.9. *A language is co/finite if and only if it is in $\mathcal{N} = [\![s x^\omega = x^\omega = x^\omega s]\!]$.*

This concludes algebraic characterisation for the basic affix-based classes. The equational characterisations provide a decision procedure for the classes as well as

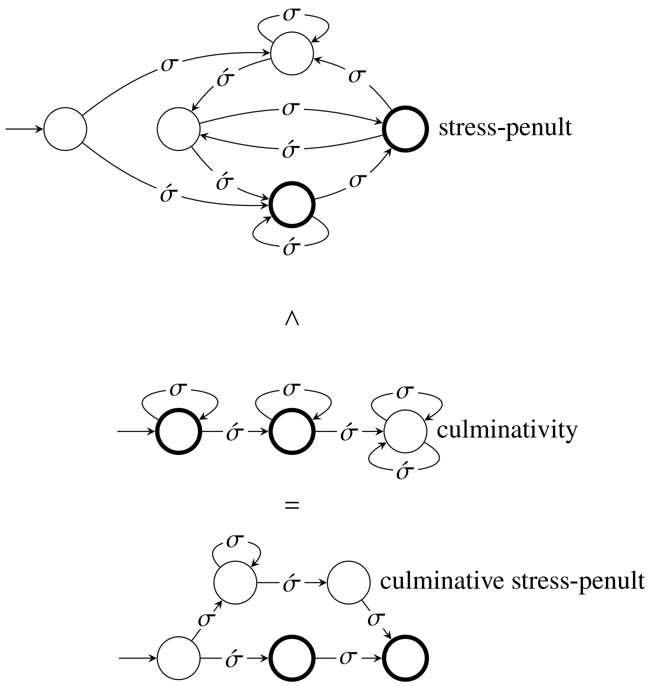

Figure 2. *Stress-penult combines with the culminativity constraint. Missing edges in the result go to a rejecting sink state, not depicted.*

Table 4. *Concatenation table for culminativity, brackets omitted.*

	σ	$\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$
σ	σ	$\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}$	$\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$

counterexample generators in cases of nonmembership. The classes explored in this section suffice to describe bounded stress patterns without culminativity. In the next section, we analyse culminativity to explore how tiers interact with our analyses.

6.3. Tiers

This section develops systems of equations for multitier classes. Recall from §2.3 that culminativity is the constraint that allows stress to appear at most once in a word, and that this constraint is tier-based co/finite. Its automaton is shown at centre in Figure 2, from which we derive a syntactic semigroup with three elements: $[\sigma]$, $[\acute{\sigma}]$ and $[\acute{\sigma}\acute{\sigma}]$. These concatenate as shown in Table 4.

This is not generalised definite, and therefore also not (reverse) definite, because it does not satisfy the equation $[\![x^{\omega}sx^{\omega} = x^{\omega}]\!]$. Specifically, let $x = x^{\omega} = [\sigma]$ and $s = [\acute{\sigma}]$. We have $x^{\omega}sx^{\omega} = [\acute{\sigma}] \neq [\sigma] = x^{\omega}$. Notice that σ never changes state

Table 5. *Concatenation table for culminative stress-penult, brackets omitted and idempotents shaded.*

	σ	$\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
σ	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$
$\sigma\sigma$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\sigma\sigma$	$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$
$\sigma\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}\sigma$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$
$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$	$\acute{\sigma}\acute{\sigma}$

in culminativity as shown in Figure 2. This is what it means to be neutral, and this neutrality is what precludes a generalised definite (or, indeed, co/finite) description. Lambert (2023) characterises tier-based extensions based on this: remove the neutral element, retain all of the others, and if there are nonneutral x and y such that xy is neutral, then bring the neutral element back. If the pattern over this potentially reduced alphabet is in class C, then the original pattern was in $\mathcal{T}C$. For culminativity, the nonneutral elements are $[\acute{\sigma}]$ and $[\acute{\sigma}\acute{\sigma}]$, and no combination of them results in the neutral $[\sigma]$. After removing this neutral element, there is just one idempotent, $[\acute{\sigma}\acute{\sigma}]$, which fills both its column and its row; culminativity is in $\mathcal{TN}$ and tier-based co/finite.

The culminative stress-penult pattern is the intersection of the stress-penult constraint with culminativity. An automaton representing this is shown at bottom in Figure 2, from which we derive the semigroup represented in Table 5. This is not generalised definite, because $[\sigma\sigma]$ is idempotent, but $[\sigma\sigma][\acute{\sigma}][\sigma\sigma] = [\acute{\sigma}\acute{\sigma}] \neq [\sigma\sigma]$. Further, no element is neutral, so it is not tier-based generalised definite. But we know from §2.3.3 that it is multitier definite.

In characterising definite languages, we noticed that idempotent elements expand to saturate a suffix of any length: no matter what k we wish to explore, any idempotent can expand to have at least k symbols, making other content irrelevant. With only a minor modification, we handle tier-suffixes rather than overall suffixes: the idempotent can still expand to be as long as needed to fill the suffix, but in doing so it only hides material on tiers that it occupies. For example in this pattern, no number of copies of $[\sigma\sigma]$ can ever influence the $\{\acute{\sigma}\}$-tier in any way. We want to say that if an idempotent *contains* a symbol or string x, then any preceding instance of x is irrelevant. And we can do exactly that.

Proposition 6.10. *A language is multitier definite if and only if it is in* $\mathcal{BTD} = [\![xv(sxt)^{\omega} = v(sxt)^{\omega}]\!]$.

Almeida (1995) arrives at the same class from another perspective. Rather than thinking about phonological tiers and projections, Almeida considers MONOIDS, semigroups with neutral elements. The insight here is recognising that projecting to multiple tiers and generalising to monoids are the same thing. With some algebraic manipulation, we arrive at a simpler characterisation of multitier (reverse) definiteness. Multitier generalised definiteness brings more complexity, as there are two equations that must be satisfied rather than just one. Where we use $\mathcal{BT}$ to denote the Boolean closure of tier-based extensions, Almeida uses $\mathcal{M}$ to denote generalising to monoids.

Proposition 6.11 (Almeida, 1995: 212). *A language is multitier definite if and only if it is in* $\mathcal{BTD} = [\![xyx^{\omega} = yx^{\omega}]\!]$.

Proposition 6.12 (Almeida, 1995: 212). *A language is multitier reverse definite if and only if it is in* $\mathcal{BTK} = [\![x^{\omega}yx = x^{\omega}y]\!]$.

Proposition 6.13 (Almeida, 1995: 212). *A language is multitier generalised definite if and only if it is in* $\mathcal{BTLI} = [\![x^{\omega}sxtx^{\omega} = x^{\omega}stx^{\omega}; x^{\omega}sxztz^{\omega} = x^{\omega}szxtz^{\omega}]\!]$. *The semicolon is an 'and' operator separating the two equations that must be satisfied.*

The culminative stress-penult constraint of Table 5 satisfies the equation for multitier definite no matter how the variables are instantiated. This proves membership in the class. However, this is not so for multitier reverse definite. Let $x = x^{\omega} = [\sigma\sigma]$ and $y = [\acute{\sigma}]$. Then $x^{\omega}yx = [\sigma\sigma][\acute{\sigma}][\sigma\sigma] = [\acute{\sigma}\acute{\sigma}]$ while $x^{\omega}y = [\sigma\sigma][\acute{\sigma}] = [\sigma\acute{\sigma}]$. These differ, so the pattern is not multitier reverse definite. They are distinguished by the $\_\sigma$ context, so the mechanically derived parameterised words witnessing nonmembership are the accepted words $(\sigma\sigma)^k\acute{\sigma}\sigma$ (from $x^{\omega}y$ in this context) and the rejected words $(\sigma\sigma)^k\acute{\sigma}\sigma\sigma$ (from $x^{\omega}yx$ in this context). Each has the length-k prefix σ^k on the $\{\sigma\}$-tier and on the $\{\sigma, \acute{\sigma}\}$-tier, while on the $\{\acute{\sigma}\}$-tier the k-prefix is $\acute{\sigma}$ and on the empty tier it is the empty string. They have the same k-prefixes on every tier, so tier-prefixes cannot make necessary distinctions.

6.4. Summary

The process shown here was used to analyse every pattern in the present work. First, a description is given in any logical form, perhaps even mixing substrings, subsequences and tier-substrings all in the same formula. For Uyghur backness harmony in particular, the pattern was described by Mayer & Major (2018) as a regular expression. In any case, the logical form is translated directly into a minimal deterministic finite-state automaton. Using the Language Toolkit of Lambert (2024), we determine class membership by providing the equations and querying whether they are satisfied. If not, then some instantiation of the variables provides parameterised words witnessing nonmembership. Otherwise, the language is in the class and there is some logical expression in the appropriate form. At the moment, creating this logical expression is not automated, but we can at least be confident that one can be created, that in trying to create one we are not wasting time trying to do the impossible.

As a summary, Table 6 presents equational algebraic characterisations for affix-based classes and their multitier extensions. Per Eilenberg (1976), every set of algebraic equations defines a class closed under the Boolean operations. Which ones are linguistically relevant is an open question.

7. Conclusions

This article explored attested phonotactic patterns from a diverse set of languages across stress, harmony and tone, including two patterns that have proven challenging in prior computational linguistics literature: Uyghur backness harmony, as studied by Mayer & Major (2018), and Karanga Shona tone, as studied by Jardine (2020).

Table 6. *Summary of algebraic characterisations.*

	$\mathcal{C}$	Base ($\mathcal{C}$)	Multitier ($\mathcal{BTC}$)
Co/finite	$\mathcal{N}$	$[\![x^\omega y = x^\omega = yx^\omega]\!]$	$[\![x^\omega y = yx^\omega; x^\omega x = x^\omega]\!]$
Definite	$\mathcal{D}$	$[\![yx^\omega = x^\omega]\!]$	$[\![xyx^\omega = yx^\omega]\!]$
Reverse def.	$\mathcal{K}$	$[\![x^\omega y = x^\omega]\!]$	$[\![x^\omega yx = x^\omega y]\!]$
Generalised def.	$\mathcal{LI}$	$[\![x^\omega yx^\omega = x^\omega]\!]$	$[\![x^\omega uxvx^\omega = x^\omega uvx^\omega;$ $x^\omega uxzvz^\omega = x^\omega uzxvz^\omega]\!]$

Rather than taking strict locality as a fundamental base class, we began with a simpler subclass, definiteness, and built up a hierarchy from variations on that form. For each pattern, we presented one or more propositional formulae representing that pattern to demonstrate membership in particular subregular classes. Notably, Uyghur backness harmony is multitier definite, no more complex than a simple bounded stress pattern, and Karanga Shona tone is multitier generalised definite, no more complex than the default-to-opposite unbounded stress patterns. Through our analysis, we have provided evidence for the subregular hypothesis, that there is some principled, learnable subclass of the regular languages that suffices to capture phonotactics.

Some patterns are most simply described using tiers, others are most simply described using subsequences, and others are best described in other ways. Those invested in using a single class for everything might consider the multitier extension of the class of languages of dot-depth at most one. DOT-DEPTH ONE generalises subsequences such that the ordered elements are not individual symbols but entire substrings (Straubing, 1985: 79–80), *e.g.*, $\neg ab..cd$ forbids words which contain an 'ab' substring at any position that precedes a 'cd' substring at any distance. In other words, dot-depth one is what Rogers & Lambert (2019) call 'piecewise locally testable'. As this logic naturally encodes both substrings and subsequences, its multitier extension contains every class discussed in the present work.

However, no pattern we have encountered requires the full power of multitier dot-depth one. Further, every Boolean-closed class we have discussed has a corresponding learning algorithm in the style of Heinz *et al.* (2012). Create a grammar by collecting the sets of factors of the appropriate size and type, and accept words whose factor set is attested. As the grammars produced accept all input words, with sufficient data the correct algorithm infers the target grammar and others infer a superset. By intersecting the results, the target language is reached with potentially far less overhead than reaching up to a common superclass. Moreover, while every class we have discussed has sufficient power to describe unattested patterns, focusing on simpler classes minimises the extent of this issue. We focus instead on providing a set of building-blocks from which more complex classes can be derived as needed.

That said, the algebraic techniques in this work allow us to initially describe a pattern in any logical form, then mechanically determine whether that pattern lies in any chosen class, and if not, to mechanically derive a set of parameterised words witnessing nonmembership. When a new pattern is to be analysed, the analyst need not decide upon a desired logical form up-front, because these tools inform the researcher where to look. When the pattern is known to be in a given class, only then should a description in the associated form be sought.

Areas of future work are numerous. In order to fully understand the typology of phonotactic patterns and the psychological pressures that influence it, we should continue to develop a library of attested patterns, classified in as many ways as possible. Lifting the analyses to more general graph-based structures may allow us to better account for tone (Jardine, 2017) and other patterns that arise in morphophonology, such as stress-attracting morphological suffixes like the '-ic' in English 'atomic'. We still seek equational characterisations for two interesting classes with known logical characterisations: multitier locally testable and multitier dot-depth one. Finally, one can analyse finite-state functions using the same algebraic techniques as we have used for analysis of finite-state languages (Lambert, 2022). Understanding the typology of phonological processes is a key area of ongoing research.

The logical and algebraic techniques described in this work will not account for all aspects of typology. When two constraints have the same form, such as forbidding voiced or voiceless obstruents immediately following a nasal (∗ND and ∗NT, respectively), these techniques will not distinguish them. The broader attestation of the latter then is likely not due to its form, but due to other factors, including the physical mechanisms involved in production and perception. Using a different alphabetic representation, however, may result in the two having different forms. Featural and graph-based alphabets may be of relevance here.

Supplementary material. The algorithms discussed in this work are implemented in the Language Toolkit software (Lambert, 2024). Files executing the analyses are available as supplementary material.

Acknowledgements. This work has been greatly improved thanks to the constructive comments from three anonymous reviews and the associate editor. I additionally thank Jeffrey Heinz and Rémi Eyraud for the thoughtful discussions they brought regarding preliminary versions of this work.

Competing interests. The author declares no competing interests.

References

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

Bunn, Gordon & Ruth Bunn (1970). Golin phonology. In Gordon Bunn, Ruth Bunn, Alan Pence, Elaine Geary, Doris Bjorkman, Harry Weimer, Natalia Weimer, O. R. Claassen, & K. A. McElhanon (eds.) *Papers in New Guinea Linguistics, No. 11*, Canberra, Australia: The Australian National University, chapter 1, 1–7.

Cook, Eung-Do (1978). Palatalizations and related rules in Sarcee. In Eung-Do Cook & Jonathan Kaye (eds.) *Linguistic Studies of Native Canada*, Vancouver, Canada: University of British Columbia Press, 19–35.

De Santo, Aniello & Thomas Graf (2019). Structure sensitive tier projection: Applications and formal properties. In *Formal Grammar 2019*. Springer Verlag, volume 11668 of *Lecture Notes in Computer Science*, 35–50.

Ding, Picus Sizhi (2006). A typological study of tonal systems of Japanese and Prinmi: Towards a definition of pitch-accent languages. *Journal of Universal Language* **7:2**. 1–35.

Donohue, Mark (1997). Tone systems in New Guinea. *Linguistic Typology* **1:3**. 347–386.

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

Goedemans, R. W. N., Jeffrey Heinz & Harry van der Hulst (2015). StressTyp2. http://st2.ullet.net/.

Goldsmith, John A. (1976). *Autosegmental Phonology*. PhD dissertation, Swarthmore College.

Grubb, David McClintock (1969). *A Kwakiutl Phonology*. MA dissertation, University of Victoria.

Hayes, Bruce (1995). *Metrical Stress Theory: Principles and Case Studies*. Chicago, IL: The University of Chicago Press.

Heinz, Jeffrey (2009). On the role of locality in learning stress patterns. *Phonology* **26:2**. 303–351.

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

Heinz, Jeffrey (2014). Culminativity times harmony equals unbounded stress. In Harry van der Hulst (ed.) *Word Stress: Theoretical and Typological Issues*, Cambridge, UK: Cambridge University Press.

Heinz, Jeffrey (2018). The computational nature of phonological generalizations. In Larry Hyman & Frank Plank (eds.) *Phonological Typology*, Mouton de Gruyter, volume 23 of *Phonetics and Phonology*, 126–195.

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

Heinz, Jeffrey, Chetan Rawal & 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*. Portland, Oregon: Association for Computational Linguistics, volume 2, 58–64.

Hulden, Mans (2009). Foma: A finite-state compiler and library. In *Proceedings of the Demonstrations Session at EACL 2009*. Athens, Greece: Association for Computational Linguistics, 29–32.

Hyman, Larry M. (2009). How (not) to do phonological typology: The case of pitch-accent. *Language Sciences* **31:2–3**. 213–238.

Hyman, Larry M. & Francis X. Katamba (2010). Tone, syntax, and prosodic domains in Luganda. *ZAS Papers in Linguistics* **53**. 69–98.

Jardine, Adam (2016). Computationally, tone is different. *Phonology* **33:2**. 247–283.

Jardine, Adam (2017). The local nature of tone-association patterns. *Phonology* **34:2**. 385–405.

Jardine, Adam (2020). Melody learning and long-distance phonotactics in tone. *NLLT* **38**. 1145–1195.

Kleene, S. C. (1951). Representation of events in nerve nets and finite automata. Technical Report RM-704, U.S. Air Force Project RAND.

Krueger, John Richard (1961). *Chuvash Manual: Introduction, Grammar, Reader, and Vocabulary*, volume 7 of *Uralic and Altaic Series*. Bloomington: Indiana University.

Lambert, Dakotah (2022). *Unifying Classification Schemes for Languages and Processes with Attention to Locality and Relativizations Thereof.* PhD dissertation, Stony Brook University.

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

Lambert, Dakotah (2024). System description: A theorem-prover for subregular systems: The Language Toolkit and its interpreter, plebby. In *Functional and Logic Programming, 17th International Symposium, FLOPS 2024*. Kumamoto, Japan.

Mayer, Connor & Travis Major (2018). A challenge for tier-based strict locality from Uyghur backness harmony. In *Formal Grammar 2018*. Springer, volume 10950 of *Lecture Notes in Computer Science*, 62–83.

McNaughton, Robert & Seymour Papert (1971). *Counter-Free Automata*. Cambridge, MA: MIT Press.

Mix Barrington, David A., Neil Immerman, Clemens Lautemann, Nicole Schweikardt & Denis Thérien (2001). The Crane Beach conjection. In *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science*. Boston, MA, 187–196.

Odden, David (1984). Stem tone assignment in Shona. In George N. Clements & John Goldsmith (eds.) *Autosegmental Studies in Bantu Tone*, Dordrecht, The Netherlands: Foris Publications.

Omar, Asmah Haji (1969). *The Iban Language of Sarawak: A Grammatical Description*. PhD dissertation, University of London.

Perles, M., M. O. Rabin & E. Shamir (1963). The theory of definite automata. *IEEE Transactions on Electronic Computers* **12:3**. 233–243.

Rabin, M. O. & D. Scott (1959). Finite automata and their decision problems. *IBM Journal of Research and Development* **3:2**. 114–125.

Roberts, John R. (1987). *Amele*. New York, New York: Croom Helm.

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

Sapir, Edward & Harry Hoijer (1967). *The Phonology and Morphology of the Navaho Language*, volume 50 of *Linguistics*. Berkeley, California: University of California Press.

Simon, Imre (1975). Piecewise testable events. In Helmut Brakhage (ed.) *Automata Theory and Formal Languages*, Berlin: Springer-Verlag, volume 33 of *Lecture Notes in Computer Science*, 214–222.

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

Suomi, Kari, Juhani Toivanen & Riikka Ylitalo (2008). *Finnish Sound Structure: Phonetics, phonology, phonotactics and prosody*. Studia Humaniora Ouluensia. Oulu University Press.

Thurston, William R. (1996). Amara: An Austronesian language of Northwestern New Britain. In Malcolm D. Ross (ed.) *Studies in languages of New Britain and New Ireland*, The Australian National University, volume C-135 of *Pacific Linguistics*, 197–248.