Some Notes on Piecewise Local Sets (Supplement to: A logical and computational methodology for exploring systems of phonotactic constraints) January 5, 2019

Dakotah Lambert
Earlham College
Richmond, Indiana, USA
djlambe11@earlham.edu

James Rogers
Earlham College
Richmond, Indiana, USA
jrogers@cs.earlham.edu

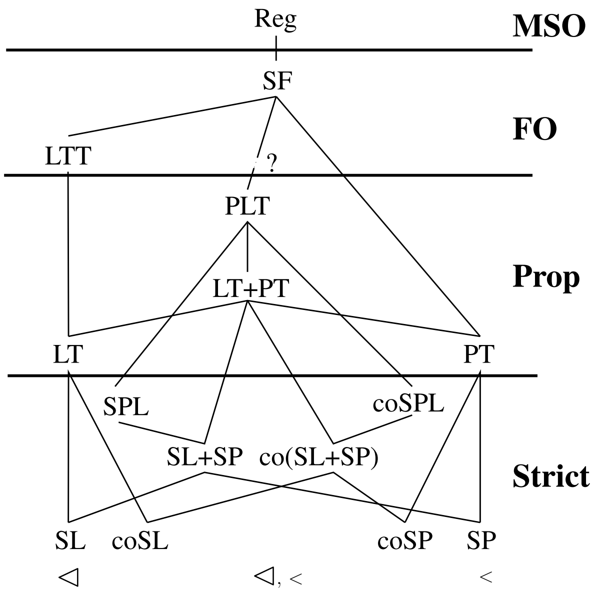

Figure 1: The Piecewise Local Hierarchy

A The Piecewise Local Hierarchy

From a model-theoretic perspective, a string is just a finite linearly ordered domain with a unary predicate for each alphabet symbol that picks out the subset of the domain at which that symbol occurs. The signature also includes binary predicates for successor (adjacency, denoted here by $\triangleleft$ or $+1$), for less-than (precedence, denoted $<$ or $..$) or both.

The Piecewise Local Hierarchy is a hierarchy of classes of stringsets distinguished by definability with respect to these signatures using logical machinery of varying strength. On the Local side of the hierarchy (LTT, LT, SL, coSL) the signature is reduced to just successor. On the Piecewise side it is reduced to just less-than. Formulae in the classes in the center of the hierarchy can employ both.

At the First Order level definitions are quantificational formulae in which variables range over positions in the domain. At this level, $\triangleleft$ is definable from $<$ but not v.v. The sets of strings that are First Order definable over a signature including only $\triangleleft$ are all and only the Locally Threshold Testable sets, while those that are First Order definable over a signature that includes $<$ are all and only the Star-Free sets.

At the Monadic Second Order level, definitions are quantificational formulae with two sorts of variables: those that range over positions and those that range over sets of positions. At this level it makes no difference which predicate defines the order, $\triangleleft$ and $<$ are each definable from the other. Sets of strings MSO definable over strings are all and only the Regular stringsets.

At the weaker levels of the hierarchy quantification is not available. Definitions are Propositional formulae in which the atomic propositions are *factors*, fragments of the string that are connected in the graph-theoretic sense, i.e., every position is reachable from every other position by a sequence of the ordering relations taken in either direction.

In the Local classes, since the domain is ordered by successor, these factors are substrings: sequences of symbols that occur consecutively in the string. In these classes the string boundaries are significant. We mark them with adjoined points denoted $\rtimes$ and $\ltimes$ with the first position in the string as the successor of $\rtimes$ and $\ltimes$ as the successor of the last position. The size of the factor is the length of the substring including the endmarkers, if any.

The set of local factors of length k or less that occur in a string is denoted $\mathrm{F}^{\triangleleft}_{k}$.

The Piecewise classes are ordered by less-than. These factors are subsequences: sequences of symbols that occur in order in the string, but not necessarily consecutively. Since there is no builtin sense of adjacency, the string boundaries are not significant. Again, the size of the factor is the length of the subsequence.

The set of piecewise factors of length k or less that occur in a string is denoted $\mathrm{F}^{<}_{k}$.

Each of these atomic propositions is satisfied in a string iff the factor occurs somewhere in the string.

At the Strict level, the stringsets are definable in terms of their *forbidden factors* the factors which may not occur in the string. Hence these sets are definable with formulae that are conjunctions of negative literals (negated factors).

The co-Strict classes are stringsets that are complements of Strictly definable stringsets. Hence these sets are definable with formulae that are disjunctions of positive literals. Note that, since these are disjunctions, no single factor is required to occur in every string, but every string must include at least one of the factors.

The Testable classes are Boolean combinations of the Strictly definable stringsets, hence these sets are definable by arbitrary propositional formulae over factors of the appropriate type. It should be noted that the Testable classes are not just the union of their respective Strict and co-Strict classes. At the Strict level negation applies only to individual factors; at the Testable level it applies to arbitrary formulae.

Characterizing the Stress Patterns in StressTyp2

As we have seen in the main paper, the stress pattern of Yidin (as defined in StressTyp2) is definable by a combination of Strict Local, co-Strict Local and Strict Piecewise constraints. Constraints in these classes (and the co-Strict Piecewise constraints) are extremely simple cognitively, requiring only the recognition of individual factors in the string in isolation. (In contrast to the Testable constraints, which depend on the entire set of factors occurring in the string.) The workbench described in the paper includes a tool that can, given a finite-state automaton recognizing a pattern, extract sets of strict and co-strict local and piecewise factors that approximate that pattern. If the pattern is definable at this level the approximation is exact, otherwise it is minimal in a particular sense.

Using this, we have shown that nearly all of these stress patterns are definable by co-occurrence of strict constraints. (In the figure the co-occurence classes are indicated using ‘$+'$. Only a few of these classes are included there.) The exceptions include two lects of Arabic in which unstressed syllables with secondary stress are required to alternate in certain contexts, but in which the secondary stress does not surface. Consequently, certain spans of unstressed syllables are required to be of odd length. This is a strictly regular pattern.

The remaining exceptions, certain lects of Bhojpuri, Buriat, Cheremis, Hindi, Mongolean and Sindhi involve a constraint of the form $\acute{\mathrm{H}}\ltimes \rightarrow \neg X$ (syllables of a certain type do not occur in words with a final primary stressed heavy syllable). While this can be expressed with an LT_2 formula, it cannot be expressed with conjunctions of strict and co-strict constraints using adjacency or precedence alone. If we consider factors of models with both adjacency and precedence, though, it can be expressed as $\neg(X \mathrel{..} \acute{\mathrm{H}}\ltimes)$ (where ‘$..'$ denotes precedence). This is a Strict Piecewise Local constraint, one in which both precedence and adjacency occur in the same factor.

Propositional formulae over factors of this mixed type form the middle of the hierarchy, the Piecewise Local stringsets. These formulae have two parameters: $\mathrm{SPL}_{j,k}$ and $\mathrm{PLT}_{j,k}$, where the factors include no more than j pieces, each of which is no more than k symbols wide: $\neg(X \mathrel{..} \acute{\mathrm{H}}\ltimes)$ is a $\mathrm{SPL}_{2,2}$ constraint. ($\mathrm{SL}_k = \mathrm{SPL}_{1,k}$ and $\mathrm{SP}_k = \mathrm{SPL}_{k,1}$.) As with the other types of strict constraints, recognizing violations of SPL constraints still involves only being sensitive to the occurrence of specific factors in isolation, thus they are extremely simple from a cognitive perspective. With the exception of the Arabic outliers, all of the lects that are represented by automata in the StressTyp2 database are definable with the co-occurrence of SPL and coSL formulae.

Abstract Characterization of SPL and PLT

Let $\mathrm{F}^{<,\triangleleft}_{j,k}(w)$ be the set of factors of w with no more than j pieces, none of which is more than k symbols wide.

$\mathrm{SPL}_{j,k}$ is defined by conjunctions of negative literals in $\mathrm{F}^{<,\triangleleft}_{j,k}(\{\rtimes\}\Sigma^*\{\ltimes\})$.

$\mathrm{PLT}_{j,k}$ is defined by Boolean combinations of $\mathrm{SPL}_{j,k}$ stringsets.

Theorem 1 (Piecewise Local Testability) *Every class $\mathbb{L}$ of stringsets that is defined by Boolean formulae over Piecewise Local factors is characterized by the following way:*

$$\begin{aligned}&L \in \mathbb{L} \Leftrightarrow \\ &\quad (F^{<,\lhd}_{j,k}(w) = F^{<,\lhd}_{j,k}(v) \;\Rightarrow\; (w \in L \Leftrightarrow v \in L)\,).\end{aligned}$$

This simply follows from the definition of satisfaction for piecewise local factors.

Theorem 2 (Downward Closure) *Every class of stringsets $\mathbb{L}$ that is defined by conjunctions of negative literal factors is closed in the following way:*

$$\begin{aligned}&L \in \mathbb{L} \Rightarrow \\ &\quad (F^{<,\lhd}_{j,k}(v) \subseteq F^{<,\lhd}_{j,k}(w) \;\Rightarrow\; (w \in L \Rightarrow v \in L)\,).\end{aligned}$$

This follows from the fact that strict formulae are negative constraints. If a set of factors does not include a forbidden factor, then no subset of that set does either.

While downward closure characterizes SPL, it has a weakness in that it characterizes sets of strings in terms of a property in the space of sets of factors, ordered by subset. But that space of sets of factors is richer than the corresponding space of sets of strings: not every set of factors is realized by a well-formed string.

This is not an issue for the SP stringsets, since every subsequence of a string is also a string. But it is an issue for SL since an arbitrary subset of the set of local factors in a string may not include sufficient factors to build a string starting with $\rtimes$ and ending with $\ltimes$. (Indeed, it could fail to include any initial or final factors.) Consequently, SL is better characterized by Suffix Substitution Closure, which guarantees that the integrity of the strings is maintained. SPL requires a similar closure condition based directly on the set of strings.

The following is a necessary condition for a stringset to be SPL. While the structure of a proof of sufficiency is reasonably clear, we have not yet filled it out. Consequently the hypothesis may need to be strengthened slightly before we can establish it as a full characterization.

Theorem 3 (Generalized SSC) *If $L \in SPLj,k$ then for all $x \in \Sigma^k$, $u_1, u_2 \in \{\rtimes\}\Sigma^*$, $v_1, v_2 \in \Sigma^*\{\ltimes\}$:*

$$\begin{array}{ll} (& u_1 \cdot x \cdot v_1 \in L \\ \wedge & u_2 \cdot x \cdot v_2 \in L \\ \wedge & (F^{<,\lhd}_{j,k}(u_1) \subseteq F^{<,\lhd}_{j,k}(u_2) \\ & \quad \vee F^{<,\lhd}_{j,k}(v_2) \subseteq F^{<,\lhd}_{j,k}(v_1)) \qquad) \\ \Rightarrow & u_1 \cdot x \cdot v_2 \in L. \end{array}$$

Proof 1 *Let $f \in F^{<,\lhd}_{j,k}(u_1 \cdot x \cdot v_2)$. Then, by cases:*

- *If $f \in F^{<,\lhd}_{j,k}(u_1 \cdot x)$ then $f \in F^{<,\lhd}_{j,k}(u_1 \cdot x \cdot v_1)$ and f is not a forbidden factor of L.*
- *The $f \in F^{<,\lhd}_{j,k}(x \cdot v_2)$ case is similar.*
- *Otherwise $f = f_1 \,.\,.\, f_2$ where $f_1 \in F^{<,\lhd}_{j,k}(u_1 \cdot x)$ and $f_2 \in F^{<,\lhd}_{j,k}(x \cdot v_2)$. Then neither f_1 nor f_2 is a forbidden factor of L and,* a fortiori, *neither is $f_1 \,.\,.\, f_2$.*

Since none of the (j,k)-factors of $u_1 \cdot x \cdot v_2$ is forbidden by L, $u_1 \cdot x \cdot v_2 \in L$.

Example

Since to show that a stringset is in the class SPL one needs only to demonstrate an SPL formula that defines it, the primary value of the abstract characterization is in establishing that a given stringset is not SPL. For this a necessary condition is all that is needed. The following example shows that a pattern based on our impression of Latin liquid dissimulation is not SPL. The counterexample could be sharpened, but we have not bothered to. We believe that a similar counterexample will suffice to show that it is not PLT either.

Latin liquid dissimulation (LLD): every pair of ‘l’s is separated by at least one ‘r’ and every pair of ‘r’s is separated by at least one ‘l’:

$$\begin{array}{ll} (\forall x, y)[& (x < y \wedge l(x) \wedge l(y)) \\ & \quad \rightarrow (\exists z)[x < z \wedge z < y \wedge r(z)]\;] \\ \quad\wedge & \\ (\forall x, y)[& (x < y \wedge r(x) \wedge r(y)) \\ & \quad \rightarrow (\exists z)[x < z \wedge z < y \wedge l(z)]\;] \end{array}$$

Let

$$w_1 = \rtimes(s^{jk}ls^{jk}r)^{jk} \cdot s^{jk} \cdot ls^{jk}r(s^{jk}ls^{jk}r)\ltimes$$

and

$$w_2 = \rtimes(s^{jk}ls^{jk}r)^{jk}s^{jk}l \cdot s^{jk} \cdot r(s^{jk}ls^{jk}r)\ltimes.$$

Both w_1, $w_2 \in L$, but

$$\rtimes(s^{jk}ls^{jk}r)^{jk} \cdot s^{jk} \cdot r(s^{jk}ls^{jk}r)\ltimes \notin L.$$

Therefore, LLD is not SPLj,k for any j and k.