Proceedings of the Society for Computation in Linguistics

Volume 3 Article 60

2020

Modelling Non-local Maps as Strictly Piecewise Functions

Phillip A. Burness
University of Ottawa, Canada, pburn036@uottawa.ca

Kevin McMullin
kevin.mcmullin@uottawa.ca

Follow this and additional works at: https://scholarworks.umass.edu/scil

Part of the Computational Linguistics Commons, and the Phonetics and Phonology Commons

Recommended Citation

Burness, Phillip A. and McMullin, Kevin (2020) "Modelling Non-local Maps as Strictly Piecewise Functions," *Proceedings of the Society for Computation in Linguistics*: Vol. 3 , Article 60.
DOI: https://doi.org/10.7275/xtrm-ny22
Available at: https://scholarworks.umass.edu/scil/vol3/iss1/60

This Extended Abstract is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Proceedings of the Society for Computation in Linguistics by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

Modelling Non-Local Maps as Strictly Piecewise Functions

Phillip Burness
University of Ottawa
pburn036@uottawa.ca

Kevin McMullin
University of Ottawa
kevin.mcmullin@uottawa.ca

A growing body of research aims to describe phonological patterns with *subregular* classes of formal languages or functions. Of particular note in this subregular hierarchy are the Strictly Local languages (SL; McNaughton and Papert, 1971; Rogers and Pullum, 2011; Rogers et al., 2013), and the Strictly Piecewise languages (SP; Heinz, 2010; Rogers et al., 2010, 2013), both of which are described in more detail below. The SL languages were extended to functions by Chandlee (2014) and Chandlee et al. (2014, 2015); the work presented here investigates whether the SP languages can be extended to functions in a similar way. While the SP functions may be able to capture non-local phonological processes that are not SL, it is somewhat difficult to achieve a straightforward definition of an SP function. As part of this work in progress, we first define a more powerful type of function and investigate whether the intended SP properties can be obtained by imposing specific restrictions.

The SL_k languages are those that ban certain contiguous sequences of length k, and have been put forth as a characterization of locally bounded phonotactic restrictions. A key property of SL_k languages is what is known as Suffix Substitution Closure (Rogers and Pullum, 2011; Rogers et al., 2013): any two well-formed strings in an SL_k language that share a suffix of length $k-1$ can both be legally continued by the same set of strings.

Chandlee (2014) and Chandlee et al. (2014, 2015) expanded on this property to define the Strictly Local functions, in which the the output associated with an input segment is determined by the immediately preceding $k-1$ elements on either the input side (ISL_k) or output side (OSL_k). These functions can model many local phonological processes such as substitution, deletion, and epenthesis. A major limitation of the SL languages and functions, though, is that they cannot model long-distance patterns such as sibilant harmony in Aari (e.g., /ʃed-er-s-it/ → [ʃederʃit], 'I was seen'; Hayward, 1990).

One proposed means of capturing long-distance patterns is to eliminate the requirement of contiguity. The SP_k languages operate in this manner, banning certain sequences of length k whether contiguous or not. For example, sibilant harmony in Aari can be described as a ban on output strings that contain the subsequence [ʃ...s]. As many nonlocal phonotactic dependencies can be characterized as SP languages (Heinz, 2010), this raises the question of whether the language class can be extended to functions as well. Preliminary work suggests that it may be possible to do so, though our line of inquiry faces an interesting challenge. Namely, the SP_k languages do not exhibit a property directly analogous to Suffix Substitution Closure, which makes it difficult to extend them to functions with the same approach that has been used for the ISL_k or OSL_k functions.

Interestingly, the class of Piecewise Testable languages (PT; Simon, 1975; Rogers et al., 2010, 2013), which are the boolean closure of the SP languages, *do* have such a suffix-related closure property. Rather than simply banning specific subsequences, a PT_k language is one that excludes strings with an impermissible *set* of subsequences of up to length k. Rogers et al.'s (2013) Theorem 7 states that given a PT_k language L and any two strings with a matching set of subsequences of up to length k, either both strings are in L or neither string is in L. A corollary of this Theorem is that any two well-formed strings in a PT_k language that have matching sets of subsequences of up to length k can both be legally continued by the same set of strings. The SP_k languages are effectively a restricted type of PT_k language, and we propose to define the SP_k functions as PT_k functions that satisfy certain restrictions.

Proceedings of the Society for Computation in Linguistics (SCiL) 2020, pages 493-495.
New Orleans, Louisiana, January 2-5, 2020

Intuitively, an Output Piecewise k-Testable (OPT_k) function would keep track of the subsequences of up to length k produced so far, which would dictate the output for any subsequent input segment. For example, consider Figure 1 which shows how a hypothetical OPT_1 function would model the Aari sibilant harmony from above. Each circle represents the strings of length 1 (i.e., the individual segments) produced thus far, and an arrow labelled $x : y$ acts as instruction to output y and move to the indicated state upon reading x.

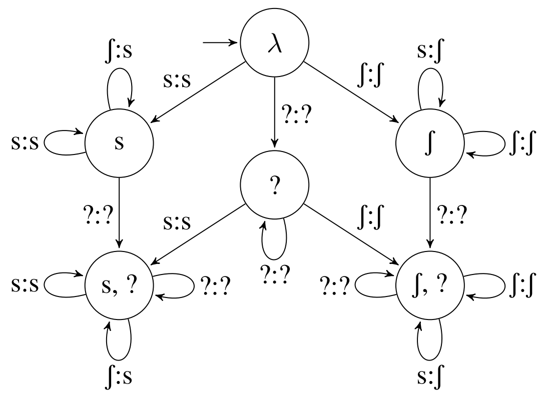

Figure 1: Aari sibilant harmony as an OPT_1 function, where ? denotes any non-sibilant

The key difference between an SP_k and a PT_k language is that a given k-subsequence has a consistent effect on well-formedness in an SP_k language; this is not necessarily true in a PT_k language. Since a PT_k language is defined with reference to *sets* of subsequences, it is possible for a PT_k language to exclude (i.e., treat as ill-formed) all strings containing a given k-length subsequence u, except those strings that also contain a different k-length subsequence v. Such conditional well-formedness of a k-length subsequence is impossible to describe using an SP_k language.

We therefore propose that the SP_k functions could be defined by restricting the PT_k functions such that the presence of a given subsequence has a consistent effect on the output for some input segment. A preliminary definition of the Output Strictly k-Piecewise (OSP_k) functions along these lines is provided below. $\texttt{Sub}_{\leq k-1}(x)$ denotes the set of subsequences of up to length $k-1$ in a string x, and $\texttt{cont}(\sigma, w)$ denotes the *contribution* of σ relative to w, or the output produced upon reading σ after having read w.

Definition 1. *A function $f : \Sigma^* \to \Delta^*$ is OSP_k iff:*

1. *If $\texttt{cont}(\sigma, w)$ is undefined, then $\texttt{cont}(\sigma, w')$ is undefined for all w' such that $\texttt{Sub}_{\leq k-1}(f(w')) \supseteq \texttt{Sub}_{\leq k-1}(f(w))$*
2. *If $\texttt{Sub}_{\leq k-1}(f(w_1)) \neq \texttt{Sub}_{\leq k-1}(f(w_2))$ and $\texttt{cont}(\sigma, w_1) \neq \texttt{cont}(\sigma, w_2)$, then either:*
   - *$\texttt{cont}(\sigma, w_3) = \texttt{cont}(\sigma, w_1)$ for all w_3 such that $\texttt{Sub}_{\leq k-1}(f(w_3)) \supseteq (\texttt{Sub}_{\leq k-1}(f(w_1)) \cup \texttt{Sub}_{\leq k}(f(w_2)))$*
   - *$\texttt{cont}(\sigma, w_3) = \texttt{cont}(\sigma, w_2)$ for all w_3 such that $\texttt{Sub}_{\leq k-1}(f(w_3)) \supseteq (\texttt{Sub}_{\leq k-1}(f(w_1)) \cup \texttt{Sub}_{\leq k}(f(w_2)))$*

The first point ensures that if some instance of an output subsequence causes the contribution of a following input element to be undefined in one case, then all instances of that output subsequence will cause the contribution of that following input element to be undefined. The second point ensures that when two subsequences have a different *defined* effect on the contribution of some input element, one of these is *dominant* and will apply to all mappings containing both subsequences. If this is indeed an appropriate definition of the OSP_k functions, an automata-theoretic characterization as in Figure 1 seems likely achievable.

Future work could then compare the OSP functions and the Output Tier-based Strictly Local functions (OTSL; Chandlee et al., 2017; Chandlee and McMullin, 2018; Burness and McMullin, 2019), which have also been put forth as a means of capturing non-local phonological maps. Like their name suggests, these are functional extensions of the Tier-based Strictly Local languages (TSL; Heinz et al., 2011; McMullin and Hansson, 2016). Rather than eschewing contiguity, the TSL languages and functions capture long-distance patterns by augmenting the SL languages and functions with a *tier*—a subset of the alphabet that allows us to ignore irrelevant elements that stand between interacting elements.

A decisive outcome from this comparison would have interesting consequences for phonological theory. If the OSP functions offer a better fit to the typology, it would suggest that local and non-local phonological maps are fundamentally different: the former operating according to strict precedence and the latter operating according to general precedence. If, on the other hand, the OTSL functions offer a better fit to the typology, it would suggest that all phonological maps operate according to strict precedence, relative to certain elements.

References

Phillip Burness and Kevin McMullin. 2019. Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90. Association for Computational Linguistics.

Jane Chandlee. 2014. *Strictly Local phonological processes*. Ph.D. thesis, University of Delaware.

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

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language*.

Jane Chandlee, Jeffrey Heinz, Adam Jardine, and Kevin McMullin. 2017. Modeling long-distance alternations with Tier-based Strictly Local functions. Paper presented at the 91st Annual Meeting of the Linguistic Society of America, Austin, TX.

Jane Chandlee and Kevin McMullin. 2018. Output-based computation and unbounded phonology. Presented at the *2018 Annual Meeting on Phonology*, San Diego, CA.

Richard J. Hayward. 1990. Notes on the Aari language. In Richard J. Hayward, editor, *Omotic language studies*, pages 425–493. School of Oriental and African Studies, University of London, London.

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

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

Kevin McMullin and Gunnar Ólafur Hansson. 2016. Long-distance phonotactics as Tier-based Strictly 2-Local languages. In *Proceedings of the 2014 Annual Meeting on Phonology*, Washington, DC. Linguistic Society of America.

Robert McNaughton and Seymour A. Papert. 1971. *Counter-free automata*. MIT Press, Cambridge, MA.

James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, *The mathematics of language*, pages 255–265. Springer, Berlin.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

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

Imre Simon. 1975. Piecewise testable events. In *Automata Theory and Formal Languages: 2nd Grammatical Inference conference*, pages 214–222. Springer-Verlag.