# Algebraic Reanalysis of Phonological Processes Described as Output-Oriented

Dakotah Lambert* Jeffrey Heinz†

*Université Jean Monnet, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien, F-42023, Saint-Étienne, France dakotahlambert@acm.org

†Stony Brook University Department of Linguistics and Institute for Advanced Computational Science jeffrey.heinz@stonybrook.edu

## Contribution

OSL is not algebraically well-behaved, but phonological patterns that are OSL fall into some of the simplest classes that are well-behaved: (tier-based) (reverse) definite.

## Algebraic methods for transducers

A deterministic finite-state acceptor is a function $A$ from input strings to states, with the restriction that if $A(x) = A(y)$, then $A(xz) = A(yz)$.

A deterministic finite-state transducer augments an acceptor with another function $f$ mapping a pair consisting of a state and an input symbol to an output string.

$$\mathrm{out} = \prod_{i=0}^{|\mathrm{in}|} f(A(\mathrm{in}_1 \ldots \mathrm{in}_i), \mathrm{in}_{i+1})$$

The $f$ function is a finite lookup table.

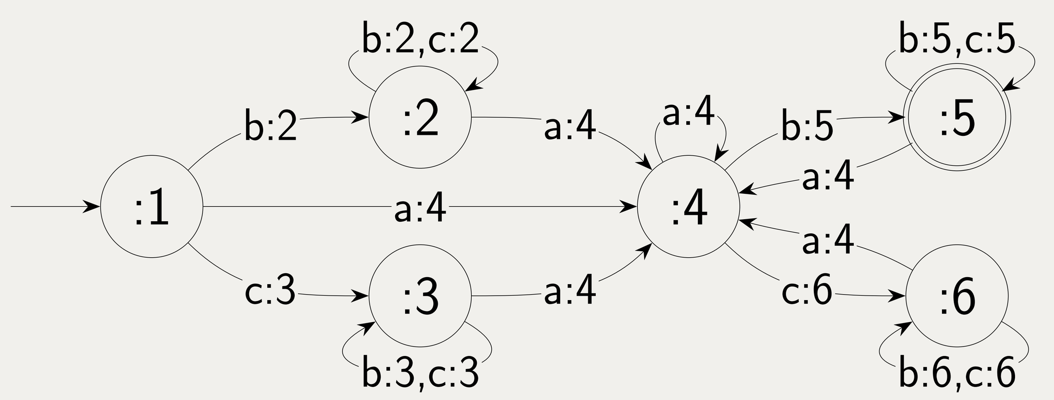
Complexity analysis asks: "What is the structure of $A$?" Given the structure, inference for acceptors is assigning accept/reject status to the states, and inference for transducers is building the lookup table $f$.

## OSL transducers

For output strictly local functions, as well as other output-oriented maps, $A$ can be written in terms of $f$. For OSL, $A$ depends on the $k$ most recent symbols of output.

**Theorem: For every regular language, there is an OSL function with the same structure.**

- Assign to each state $q$ an output letter $x_q$.
- Set the suffixes: $f(q, \ltimes) = x_q$. This prevents any state merges.
- Set inbound edges: $f(A(w), a) = x_q$ when $A(wa) = q$. This ensures that states correspond to the most recent output symbol, that the function is 1-OSL.




($A$: exists "ab" not followed by "a", beyond (T)LTT and PT)

## Input-oriented analyses

Knowing the input structure of a class of finite-state transducers facilitates learning [2], but OSL and other output-oriented classes of maps do not have fixed input structures, and so they require other methods [1]. In algebraic terms, the classes that have fixed input structures are those that have **free objects** when appropriately parameterized. We characterize some well-studied OSL processes according to their input structure to determine which, if any, class is suitable.

### Reverse Definite (K)

Next transformation depends on the $k$ **first** input symbols processed.

Reverse which edge is tracked, not the direction of processing.

### Nilpotent (N)

Both **K** and **D** at once.

Next transformation depends only on whether there have been $k$ input symbols, and if not, the input so far.

### Definite (D)

Next transformation depends on the $k$ **most recent** input symbols.

Better known in linguistics as **input strictly local**.

## Tiers

To lift a class onto a tier, track only the specified types of input symbols, rather than everything. For example, while **K** inspects the first $k$ symbols of any sort, its extension $[\![\mathbf{K}]\!]_T$ might track only the first $k$ **sibilants** encountered. This naturally expresses harmony patterns:

> *If the first sibilant encountered is* [+anterior], *then* [−anterior] *sibilants become* [+anterior], *and vice versa. That is, sibilants agree in anteriority with the leftmost sibilant.*

Symbols can be

- **Salient:** capable of influencing the transformation; on the tier,
- **Neutral:** incapable of influencing the transformation; off the tier, or
- **Undergoers:** surfacing unfaithfully in some environments

## Two analyses of potential trigger/target relationships

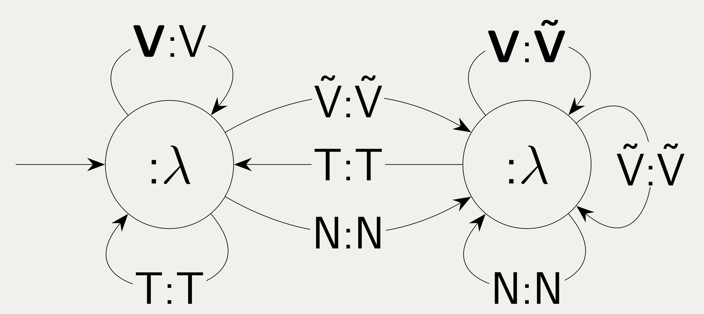
OSL

/pəŋawasan/ ⇒ p ə ŋ a w a s a n ⇒ [pəŋãw̃ãsan]

Tier-Definite

## Undergoers are not inherently salient

For example, in progressive nasal spreading (V → Ṽ/N__), non-nasalized vowels are neutral undergoers.

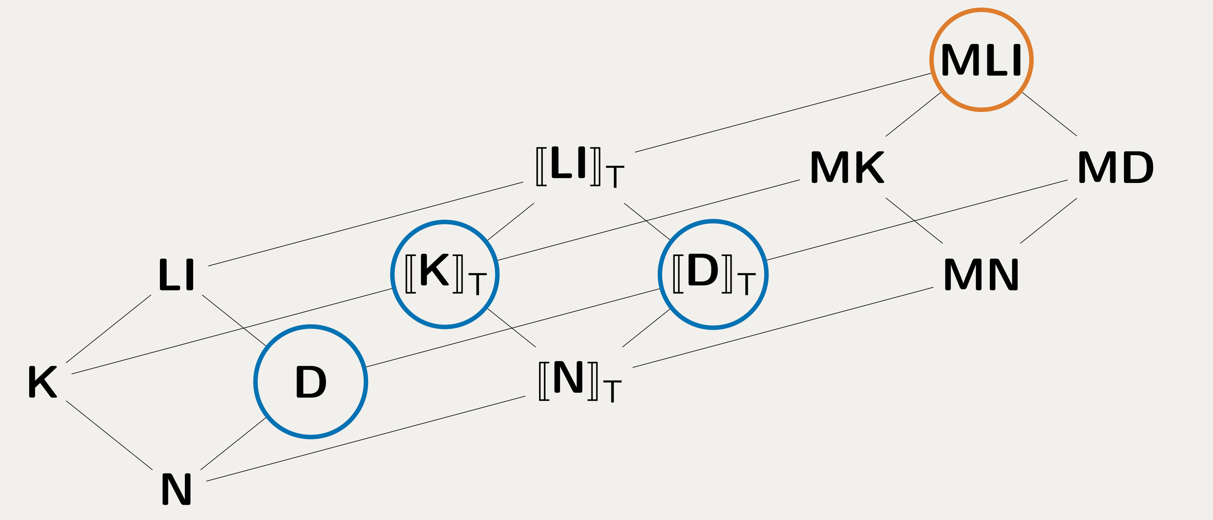



## Classifications

| Pattern | → | ← |
|---|---|---|
| Post-Nasal Voicing | **D** | **D** |
| Progressive Iterative Spreading | $[\![\mathbf{D}]\!]_T$ | – |
| Progressive Symmetric Harmony | $[\![\mathbf{K}]\!]_T$ | – |
| Progressive Asymmetric Harmony | $[\![\mathbf{N}]\!]_T$ | – |
| Pre-Nasal Voicing | **D** | **D** |
| Regressive Iterative Spreading | – | $[\![\mathbf{D}]\!]_T$ |
| Regressive Symmetric Harmony | – | $[\![\mathbf{K}]\!]_T$ |
| Regressive Asymmetric Harmony | – | $[\![\mathbf{N}]\!]_T$ |

## Multitier generalized definite

Given a length parameter $k$, track both the $k$ first and the $k$ most recent input symbols encountered on every tier. The state space is large, but finite. This multitier generalized definite class, **MLI**, is the smallest algebraic class with free objects containing the (tier-based) (reverse) definite classes.




## Why this matters

There are multiple structural interpretations of output-oriented maps, and these interpretations may tell us about linguistic typology or psychological reality.

## References

[1] 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, Toronto, Canada. Association for Computational Linguistics.

[2] Adam Jardine, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the Twelfth International Conference on Grammatical Inference*, volume 34 of *JMLR: Workshop and Conference Proceedings*, pages 94–108.