Quantifier-free least fixed point functions for phonology

Jane Chandlee and Adam Jardine
Haverford Rutgers

16th Mathematics of Language
18 July 2019
University of Toronto

Introduction

- What kind of functions are phonological UR-SR maps?
- Automata-theoretic characterizations have focused on **subsequentiality** (Heinz and Lai, 2013; Payne, 2017; Chandlee and Heinz, 2018)
- **Logical** characterizations of *sets* provide representation-independent complexity hypotheses
- No previous logical characterizations of *functions* approach subsequentiality

- The subregular class of **input strictly local (ISL)** functions can be captured with **quantifier-free** (QF) first order (FO) logic

- We generalize this with **least fixed-point** extension of QF functions (QFLFP)

- QFLFP offers recursive, *output-based* definitions of functions

- As a proper subclass of the subsequential functions, QFLFP is a better fit to the typology of phonological functions

Motivation

- Connections between logical transductions and **finite state string transducers (FSTs)**: (Filiot and Reynier, 2016)

MSO = two-way FSTs

(Engelfriet and Hoogeboom, 2001)

order-preserving MSO = one-way (non-det.) FSTs

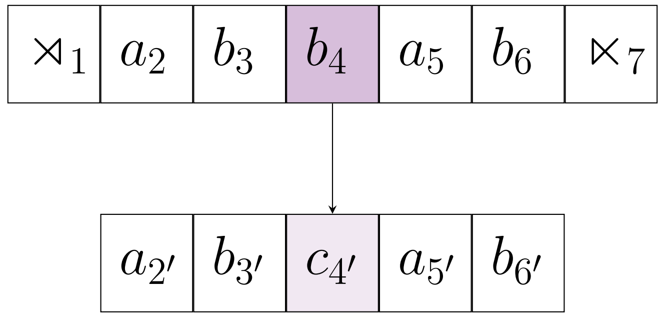
(Filiot, 2015)

- No previous characterization for **deterministic** FSTs

Logical definitions of functions

$\rtimes_1$	a_2	b_3	b_4	a_5	b_6	$\ltimes_7$

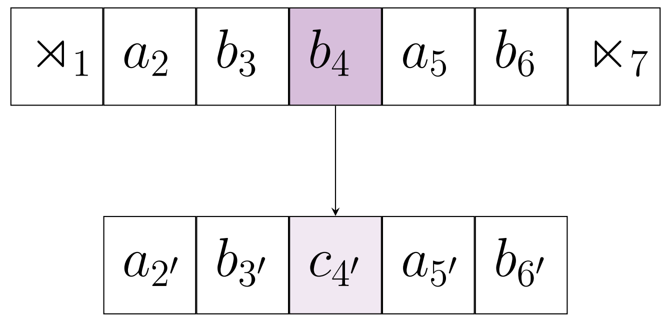
- **Model** of a string over Σ:
  - $D = \{1, 2, ..., n\}$ $\qquad D = \{1, 2, 3, 4, 5, 6, 7\}$
  - $P_\sigma \subseteq D$ for each $\sigma \in \Sigma, \rtimes, \ltimes$ $\qquad P_b = \{3, 4, 6\}$
  - A *predecessor* function p $\qquad p(2) = 1$, etc.



- A **logical transduction** defines an output structure in the logic of the input structure (Courcelle, 1994; Courcelle et al., 2012)

$$
\begin{aligned}
a'(x) &\overset{\text{def}}{=} a(x) \\
b'(x) &\overset{\text{def}}{=} b(x) \wedge \neg(b(p(x))) \\
c'(x) &\overset{\text{def}}{=} b(x) \wedge (b(p(x)))
\end{aligned}
$$

- $b \rightarrow c$ / $b\_\_$



- A **logical transduction** defines an output structure in the logic of the input structure (Courcelle, 1994; Courcelle et al., 2012)

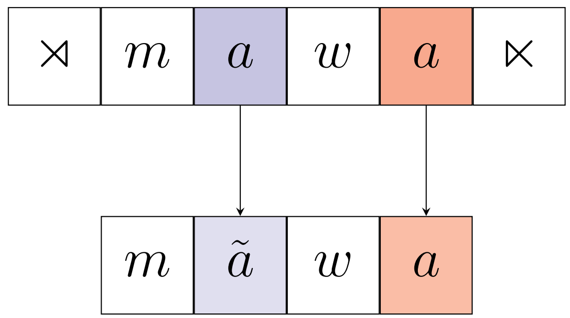
$$
\begin{aligned}
a'(x) &\stackrel{\text{def}}{=} a(x) \\
b'(x) &\stackrel{\text{def}}{=} b(x) \wedge \neg(b(p(x))) \\
c'(x) &\stackrel{\text{def}}{=} b(x) \wedge (b(p(x)))
\end{aligned}
$$

- QF transductions capture ISL functions (Chandlee, 2014).

- Long-distance patterns are not ISL/QF

- Iterative nasal spreading in Malay (Onn, 1980)

/mawa/ $\rightarrow$ [mãw̃ã]



$$\tilde{a}'(x) \overset{\text{def}}{=} a(x) \wedge \texttt{nasal}(p(x))$$

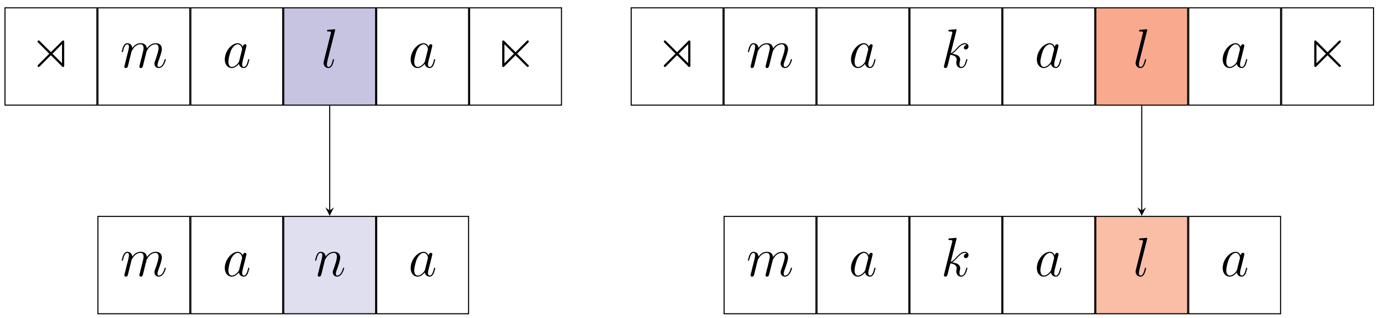
$$a'(x) \overset{\text{def}}{=} a(x) \wedge \neg\texttt{nasal}(p(x))$$

- $\texttt{nasal}(x) \overset{\text{def}}{=} m(x) \vee \tilde{a}'(x) \vee \tilde{w}'(x)$

- Long-distance patterns are not ISL/QF

- Nasal harmony in Kikongo (Ao, 1991)

/mala/ $\rightarrow$ [mana] /makala/ $\rightarrow$ [makana]



$$n'(x) \stackrel{\text{def}}{=} n(x) \lor \big(l(x) \land \mathtt{nasal}(p(p(x)))\big)$$

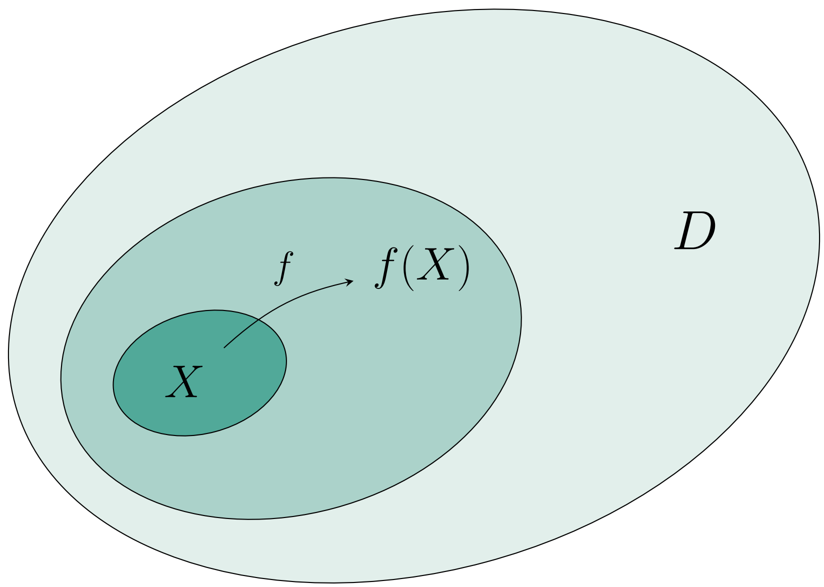
$$l'(x) \stackrel{\text{def}}{=} l(x) \land \neg\mathtt{nasal}(p(p(x)))$$

- $\mathtt{nasal}(x) \stackrel{\text{def}}{=} m(x) \lor n(x)$

- **Least-fixed point** logic allows:
  - reference to output structures;
  - definition of precedence from predecessor (p)
- Restriction to QF keeps the logic weak

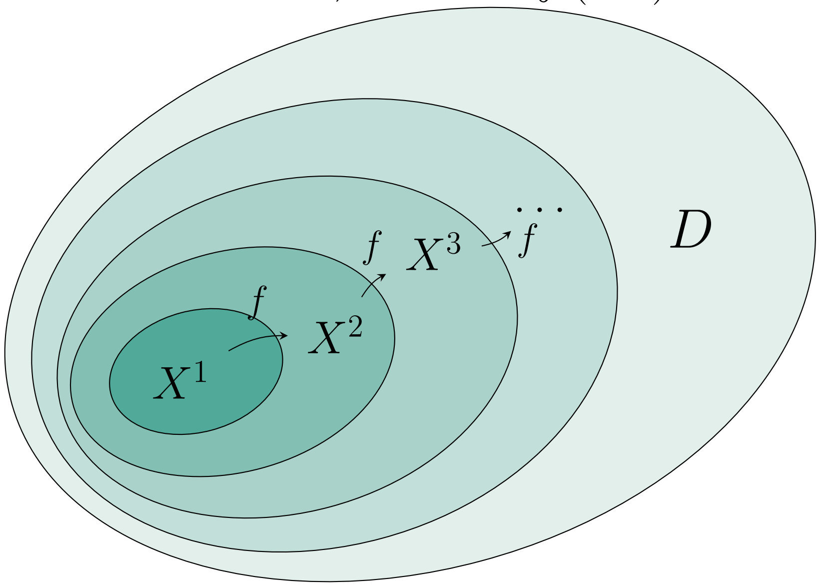
Least fixed point logic

- An **operator** on D is a function $f : \mathcal{P}(D) \to \mathcal{P}(D)$



- The **least fixed point** of f is $\texttt{lfp}(f) = \bigcup_i X^i$, where

$$X^0 = \emptyset, X^{i+1} = f(X^i)$$



- If f is **monotone** then it has a least fixed point

$$X \subseteq Y \Rightarrow f(X) \subseteq f(Y)$$

- $\varphi(A, y)$ with a special predicate $A(y)$ induces an operator

$$f_\varphi(X) = \left\{ d \in D \;\middle|\; \varphi(A, y) \text{ is satisfied with } A \mapsto X, d \mapsto y \right\}$$

- if A is under the scope of an even number of negations, then f_φ is monotone

- f_φ is applied recursively until it converges on the least fixed point (`lfp`)

Example

$\rtimes_1$	a_2	b_3	a_4	a_5	a_6	c_7	a_8	$\ltimes_9$

$$\varphi(A, y) = a(y) \wedge \big(b(p(y)) \vee A(p(y))\big)$$

$$\begin{aligned}
f_\varphi(\emptyset) &= \{4\} && X^1 \\
f_\varphi(\{4\}) &= \{4, 5\} && X^2 \\
f_\varphi(\{4, 5\}) &= \{4, 5, 6\} && X^3 \\
f_\varphi(\{4, 5, 6\}) &= \{4, 5, 6\} && X^4 = X^5 = \ldots
\end{aligned}$$

$$\texttt{lfp}(f_\varphi) = \{4, 5, 6\}$$

- QFLFP is QF extended with predicates of the form

$$\left[_{\texttt{lfp}}\varphi(A, y)\right](x)$$

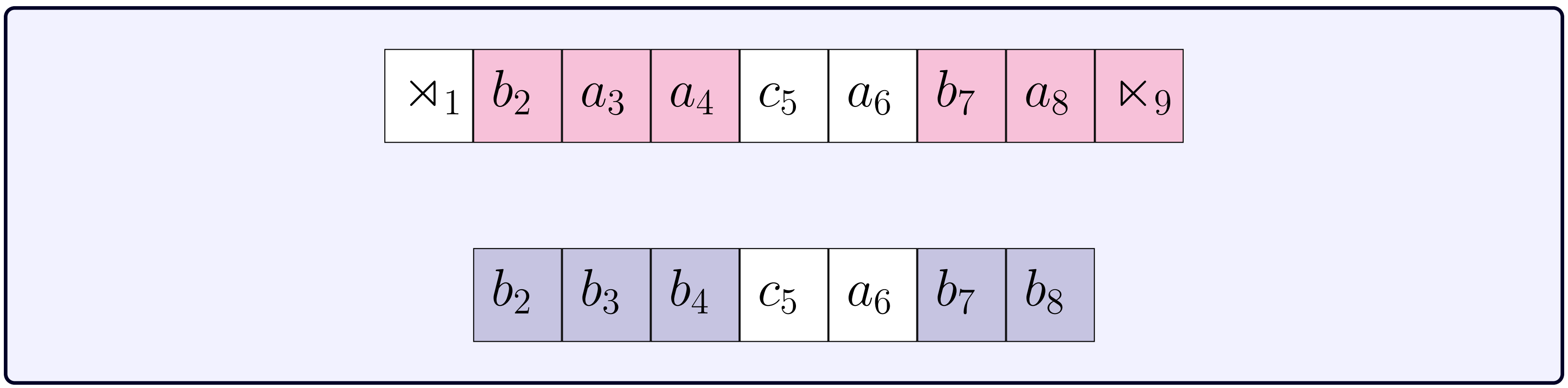
$$\left[_{\texttt{lfp}} a(y) \wedge \big(b(p(y)) \vee A(p(y))\big)\right](x)$$

$\rtimes_1$	a_2	b_3	a_4	a_5	a_6	c_7	a_8	$\ltimes_9$

Iterative spreading (with blocking)

baaa $\mapsto$ bbbb
baaca $\mapsto$ bbbca
baacaba $\mapsto$ bbbcabb

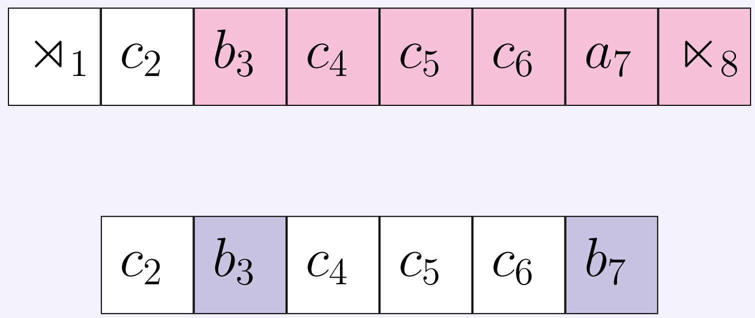
$$b'(x) \overset{\mathsf{def}}{=} [_{\mathtt{lfp}}(b(y) \vee (A(p(y)) \wedge \neg c(y)))](x)$$



Long-distance agreement

cbccca $\mapsto$ cbcccb

$$b'(x) \overset{\text{def}}{=} [_{\texttt{lfp}}(b(y) \vee A(p(y)))](x) \wedge \neg c(x)$$



Spreading with blocking:

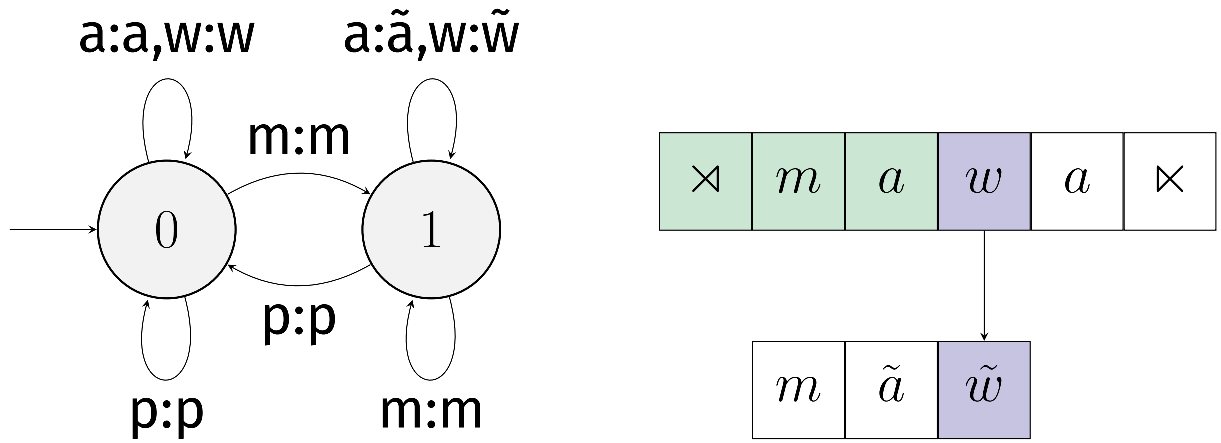
$$b'(x) \stackrel{\text{def}}{=} [_{\texttt{lfp}}(b(y) \vee (A(p(y)) \wedge \boldsymbol{\neg c(y)}))](x)$$

LD agreement:

$$b'(x) \stackrel{\text{def}}{=} [_{\texttt{lfp}}(b(y) \vee A(p(y)))](x) \wedge \boldsymbol{\neg c(x)}$$

Theorem: QFLFP is subsequential

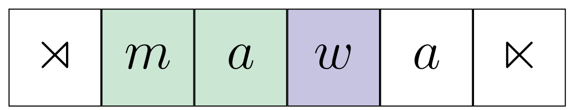
- **Subsequential functions** have some **deterministic** finite-state transducer (Schützenberger, 1977; Mohri, 1997)



- We *immediately* know the output at each position
- This output is based on some finite-state (=MSO) control

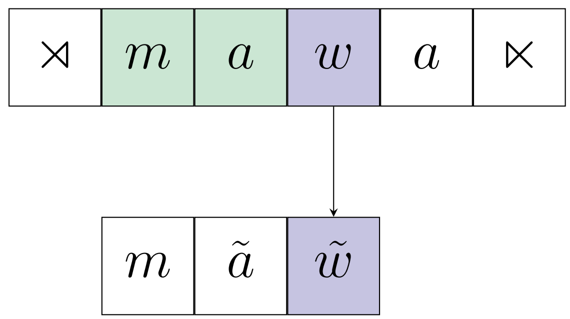
- **Lemma 1:** For any $\varphi(x) \in \mathrm{QFLFP}$, whether a position satisfies $\varphi(x)$ depends entirely on the *preceding* information in the input

$$\left[_{\texttt{lfp}}\big(w(y) \vee a(y)\big) \wedge \big(m(y) \vee A(p(y))\big)\right](x)$$



- For QFLFP, reading left-to-right, we *immediately* know the output at each position

$$\tilde{w}(x) \overset{\text{def}}{=} \left[_{\texttt{lfp}} \big(w(y) \vee a(y)\big) \wedge \big(m(y) \vee A(p(y))\big)\right](x)$$



- Any LFP predicate can be translated into MSO

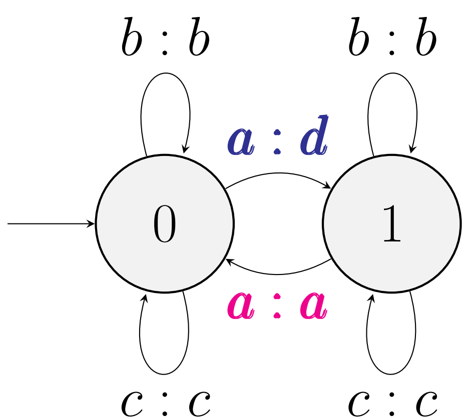
$$[\texttt{lfp}\ \varphi(A, y)](x)$$
$$\Updownarrow$$
$$(\exists X, \forall y)\ \Big[\Big(\varphi(X/A, y) \rightarrow X(y)\Big) \wedge X(x)\Big]$$

- QFLFP functions are deterministic left-to-right, and have MSO (=finite state) control

- Thus, they are subsequential

Conjecture: Subsequential is not QFLFP

- Keeping track of even and odd-numbered elements of a particular type over arbitrary distances is subsequential



$$bbbbbb\boldsymbol{a} \mapsto bbbbbb\boldsymbol{d}$$
$$abbbbb\boldsymbol{a} \mapsto dbbbbb\boldsymbol{a}$$

- There is likely no QFLFP definition for this function

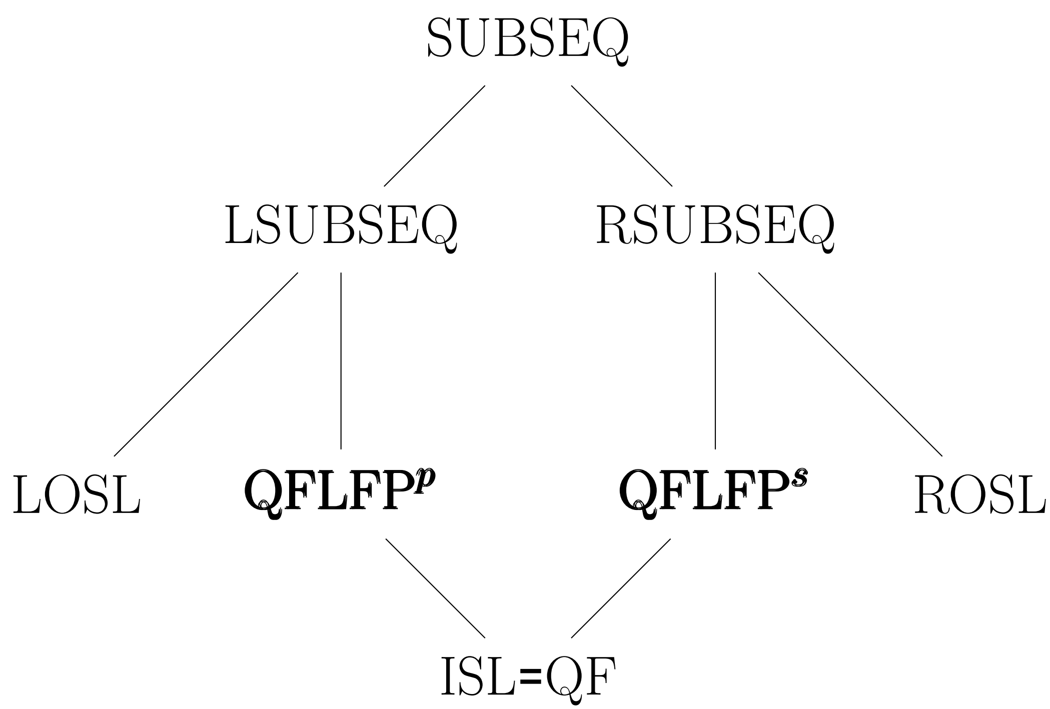
- This is a good *phonological* prediction of QFLFP; functions like “odd-numbered sibilants harmonize” are not attested.

- *But*, QFLFP *can* capture ‘local’ even/odd counting (for, e.g., iterative stress)

$$\left[_{\texttt{lfp}} \rtimes (p(y)) \vee A(p(p(y)))\right](x)$$

$\rtimes_1$	a_2	a_3	a_4	a_5	a_6	a_7	a_8	$\ltimes_9$

The general picture



OSL = output strictly local functions (Chandlee, 2014; Chandlee et al., 2015)

Conclusions & Discussion

- $\mathrm{QFLFP} \subseteq \mathrm{SUBSEQ}$ is a restrictive theory for phonology based on recursive definitions of local structures
- Because $\mathrm{QFLFP} \subseteq \mathrm{SUBSEQ}$, it is learnable (Oncina et al., 1993)
- Remaining theoretical questions:
  - Not likely closed under compositon
  - What is an abstract definition of QFLFP?
  - What is expressivity of $\mathrm{QFLFP}^{p,s}$?

- Logic can be applied to non-string structures:
  - Features
  - Autosegmental representations
  - Metrical structure
  - Others?
- What do we get with two-place predicates and QFLFP (Koser et al., AMP)?

Conclusion

- QFLFP combines the restrictiveness of QF with the ability to recursively reference the output structure.
- Allows us to model non-ISL phenomena such as LD agreement and iterative spreading.
- This class of functions appears to cross-cut several subregular classes that have been applied to the modeling of phonological processes.
- As a subset of subsequential, it is also learnable.

Acknowledgements

Thanks to the following for helpful thoughts and discussion: Jeffrey Heinz, Bill Idsardi, Steven Lindell, Jim Rogers, Jon Rawski, Siddharth Bhaskar, and the audiences at NECPhon 2018 and the computational phonology workshops at Rutgers and Stony Brook