The Strict Locality of Phonological Processes

Jane Chandlee

University of Delaware

LSA 2014 | Minneapolis

What is possible?

How can we distinguish phonological patterns from logically-possible but non-phonological ones?

- Phonetic naturalness
- Simplicity
- Computational properties

What is possible?

How can we distinguish phonological patterns from logically-possible but non-phonological ones?

- Phonetic naturalness
- Simplicity
- Computational properties

Objectives

- Identify **Strict Locality** as a strong computational property of the UR $\mapsto$ SR mapping in local phonological processes.
- Discuss the empirical coverage of this property based on a database of phonological patterns (P-base v1.95, Mielke 2008).
- Describe the utility of this property in the domain of learning.

Phonological Mappings

underlying form $\mapsto$ *surface form*

	u
Rule A:	u'
Rule B:	u''
Rule C:	s

u	Con A	Con B
☞ s		
s'	!*	
s''	*	!*

Phonological Mappings

$A \Rightarrow B \;/\; C \_\_ D$ *CAD >> FAITH(A⇒B)[1]

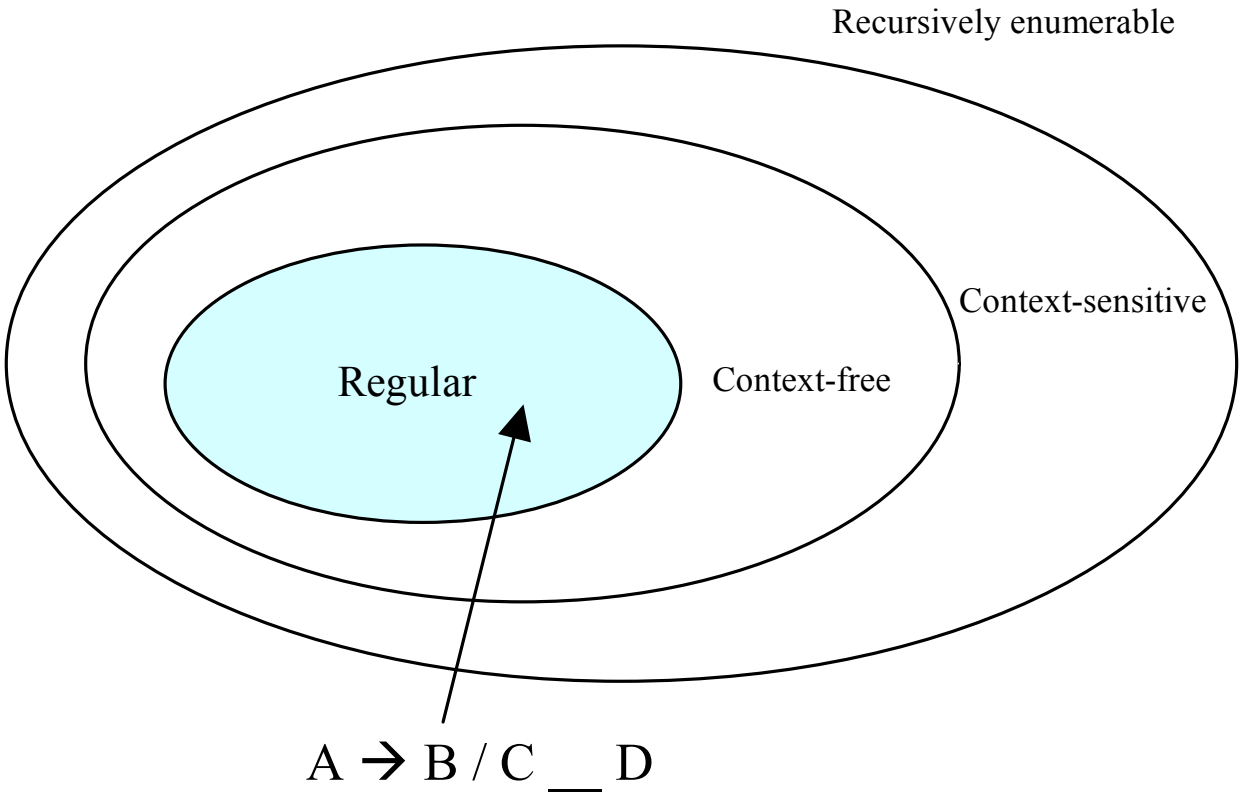
$$f(\text{CAD}) = \text{CBD}$$
$$f(\text{CBD}) = \text{CBD}$$
$$f(\text{CADCAD}) = \text{CBDCBD}$$
$$\ldots$$

[1](Baković 2013)

Phonological Mappings

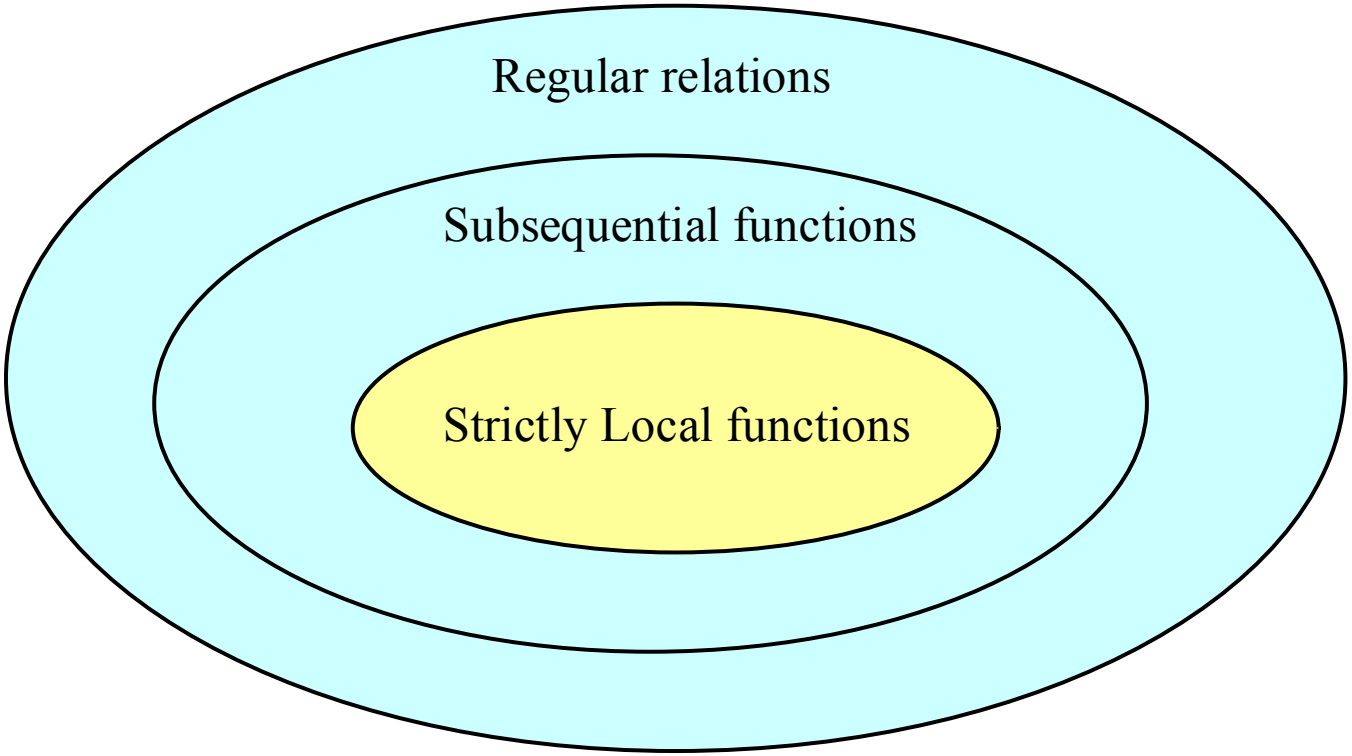
Strict Locality is a property of f.

Formal properties of grammars



(Johnson 1972, Koskenniemi 1983, Kaplan and Kay 1994)

Formal properties of grammars



(Mohri 1997, Chandlee in prep.)

Strictly Local Languages

- The 'Strictly Local' designation comes from the class of formal languages known as Strictly Local languages.
- SL languages can be described with a grammar in the form of a set of substrings up to certain length k (called k-factors).

(1) The language *a# is SL-2 with $\mathcal{G} = \{$#a, #b, aa, ab, bb, ba, b#$\}$

(McNaughton & Papert 1971, Rogers & Pullum 2011, Heinz et al. 2011, Rogers et al. 2012)

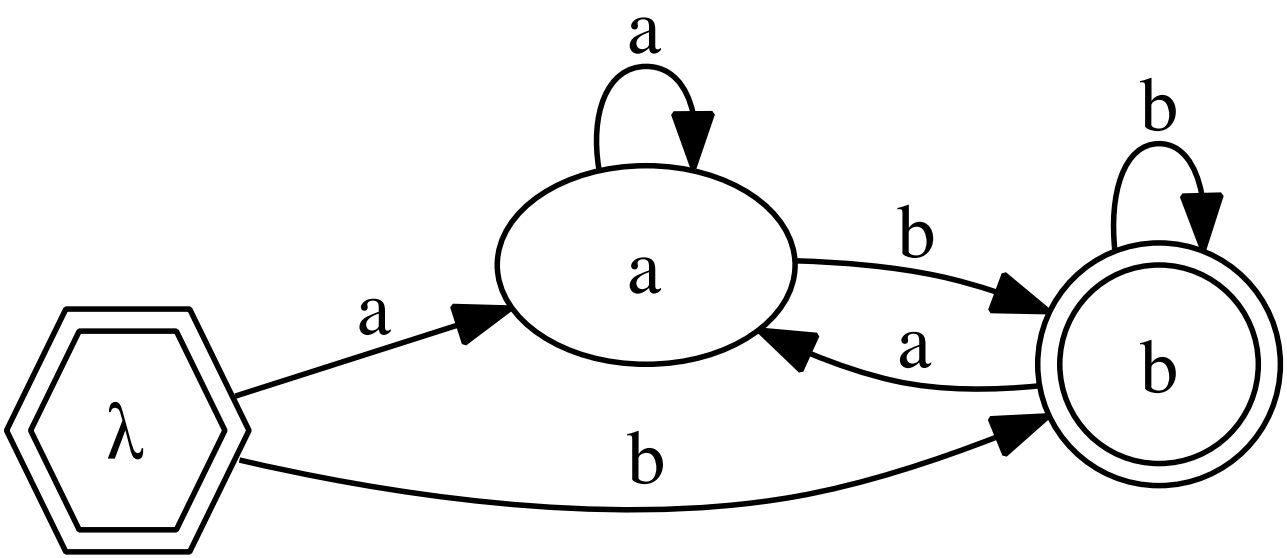
Strictly Local Languages

The language *a# is SL-2 with $\mathcal{G} = \{\#a, \#b, aa, ab, bb, ba, b\#\}$

aba	{#a, ab, ba, a#}	×
aab	{#a, aa, ab, b#}	✓

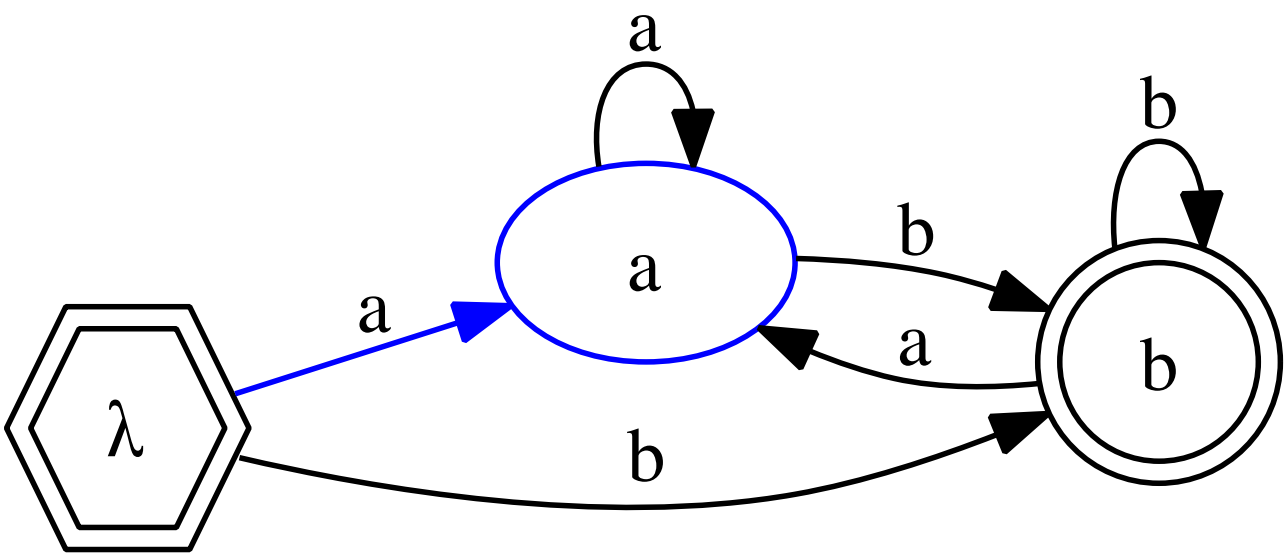
SL Finite State Acceptor

*a#



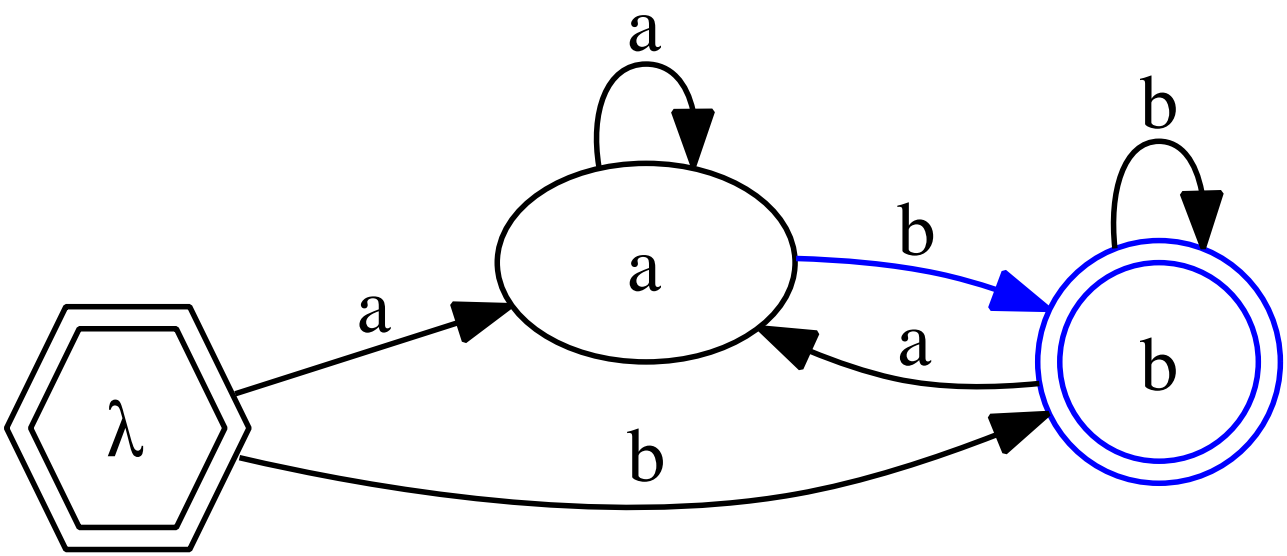
SL Finite State Acceptor

Input: aba



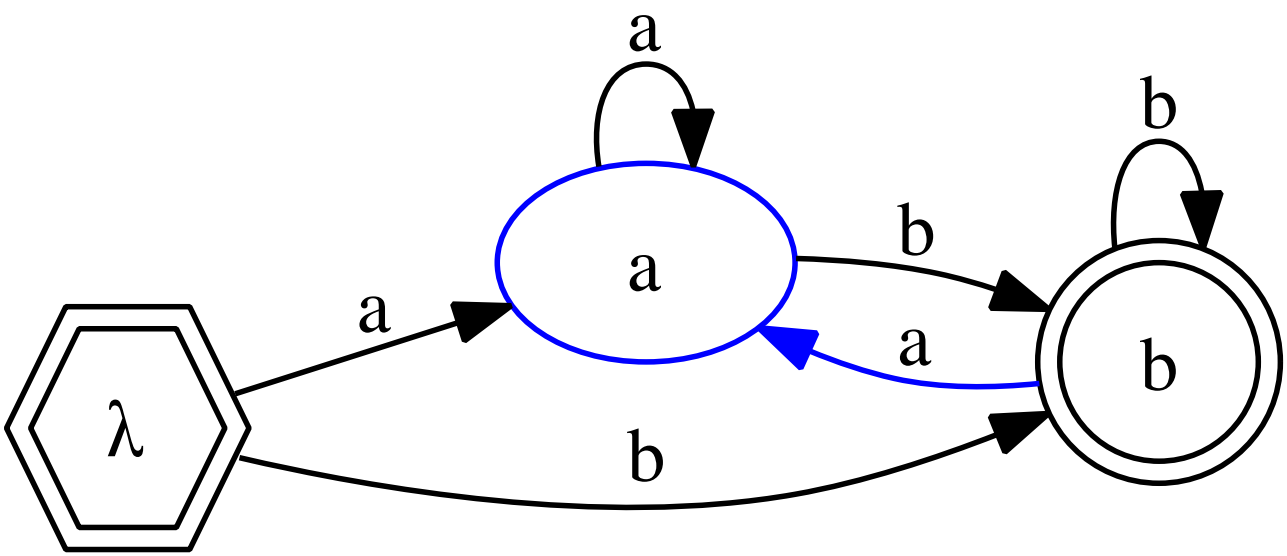
SL Finite State Acceptor

Input: aba



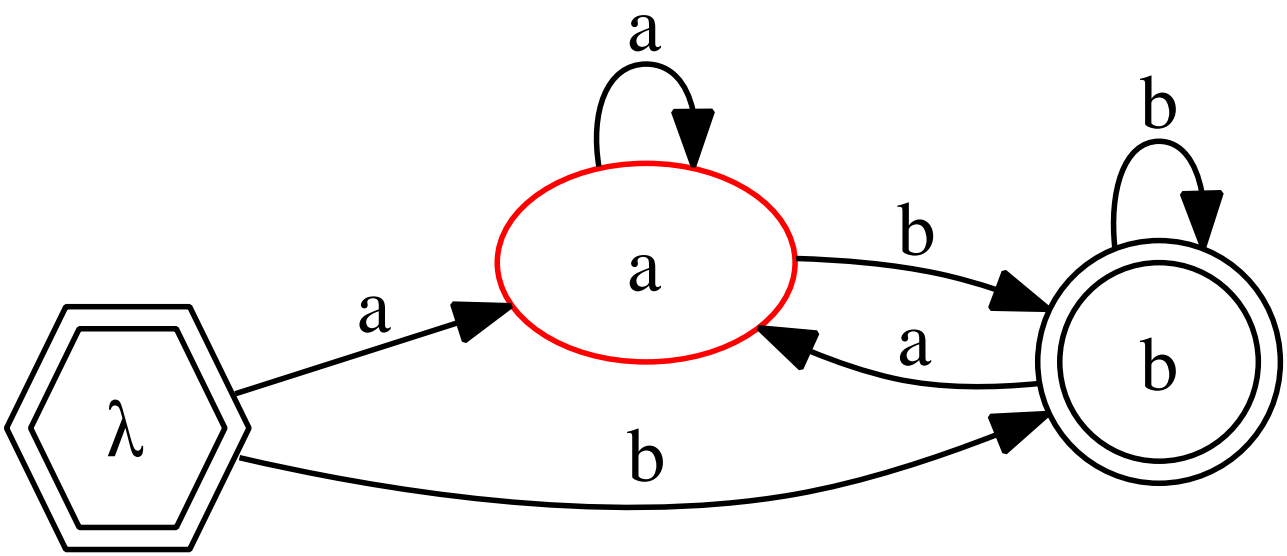
SL Finite State Acceptor

Input: aba



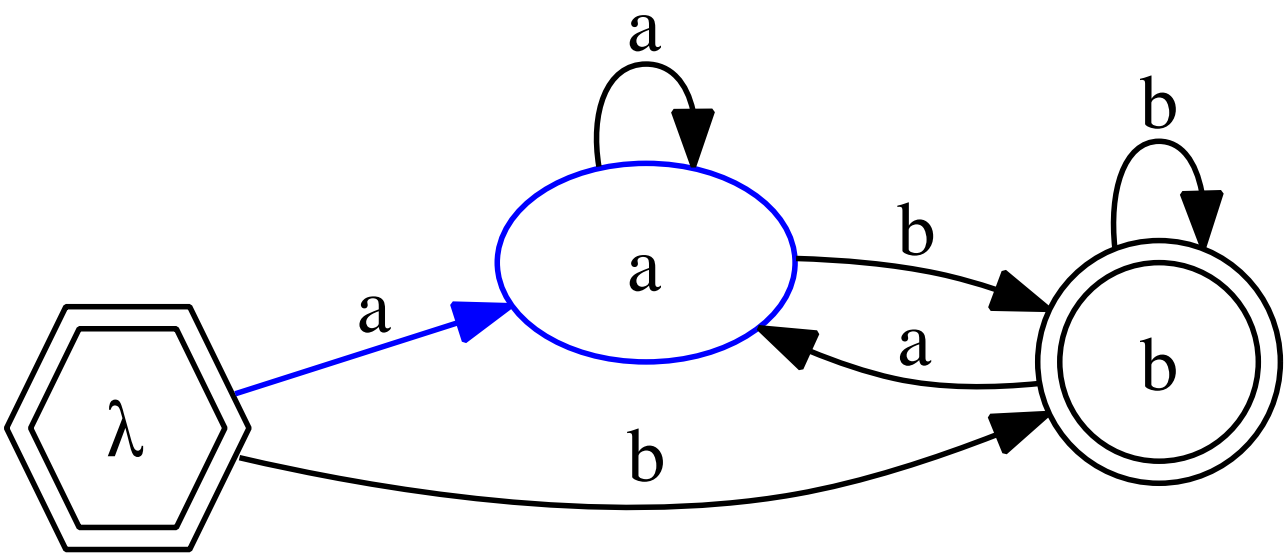
SL Finite State Acceptor

Input: *aba



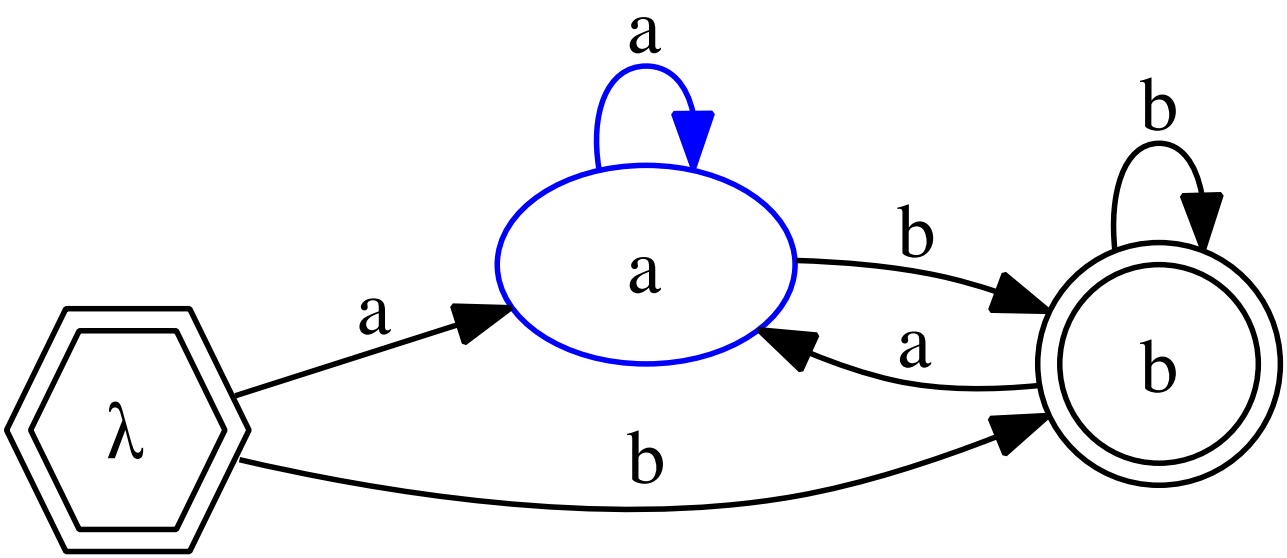
SL Finite State Acceptor

Input: aab



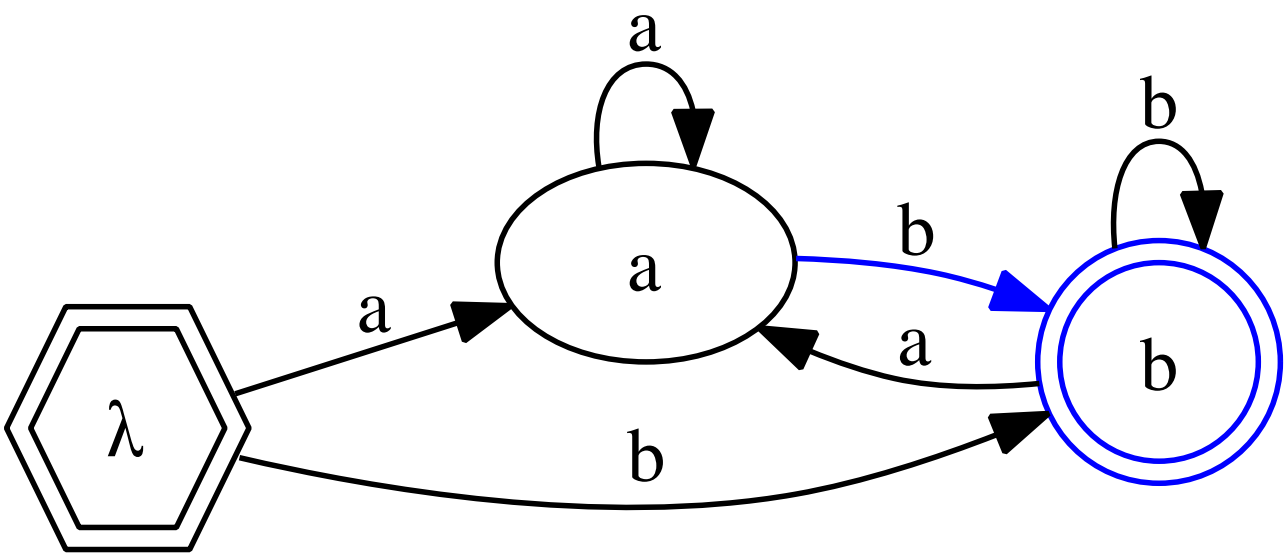
SL Finite State Acceptor

Input: aab



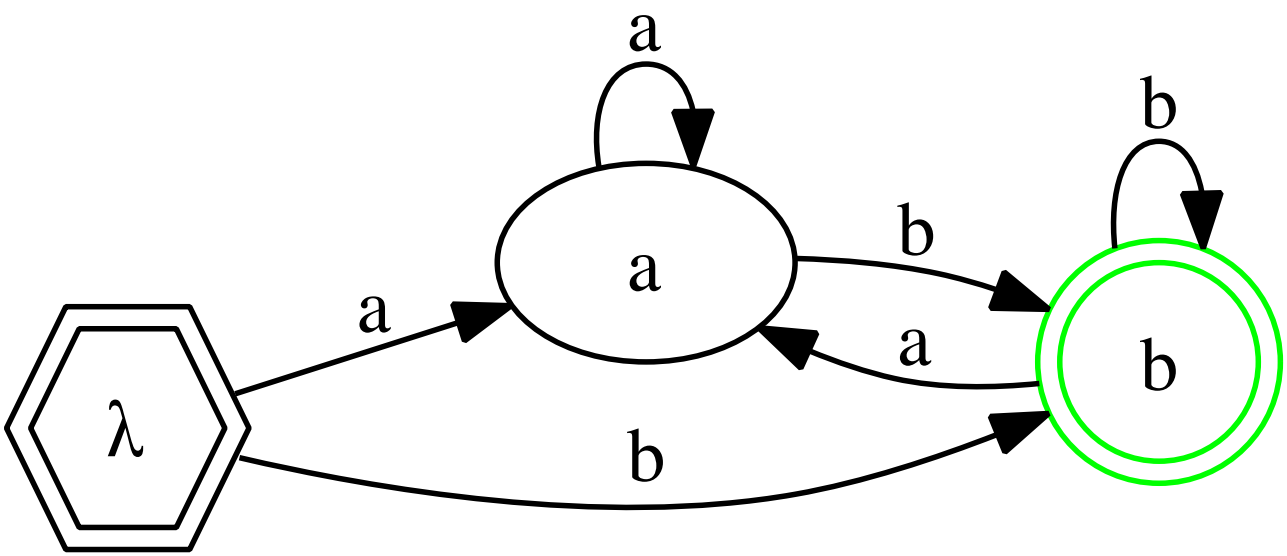
SL Finite State Acceptor

Input: aab



SL Finite State Acceptor

Input: ✓ aab

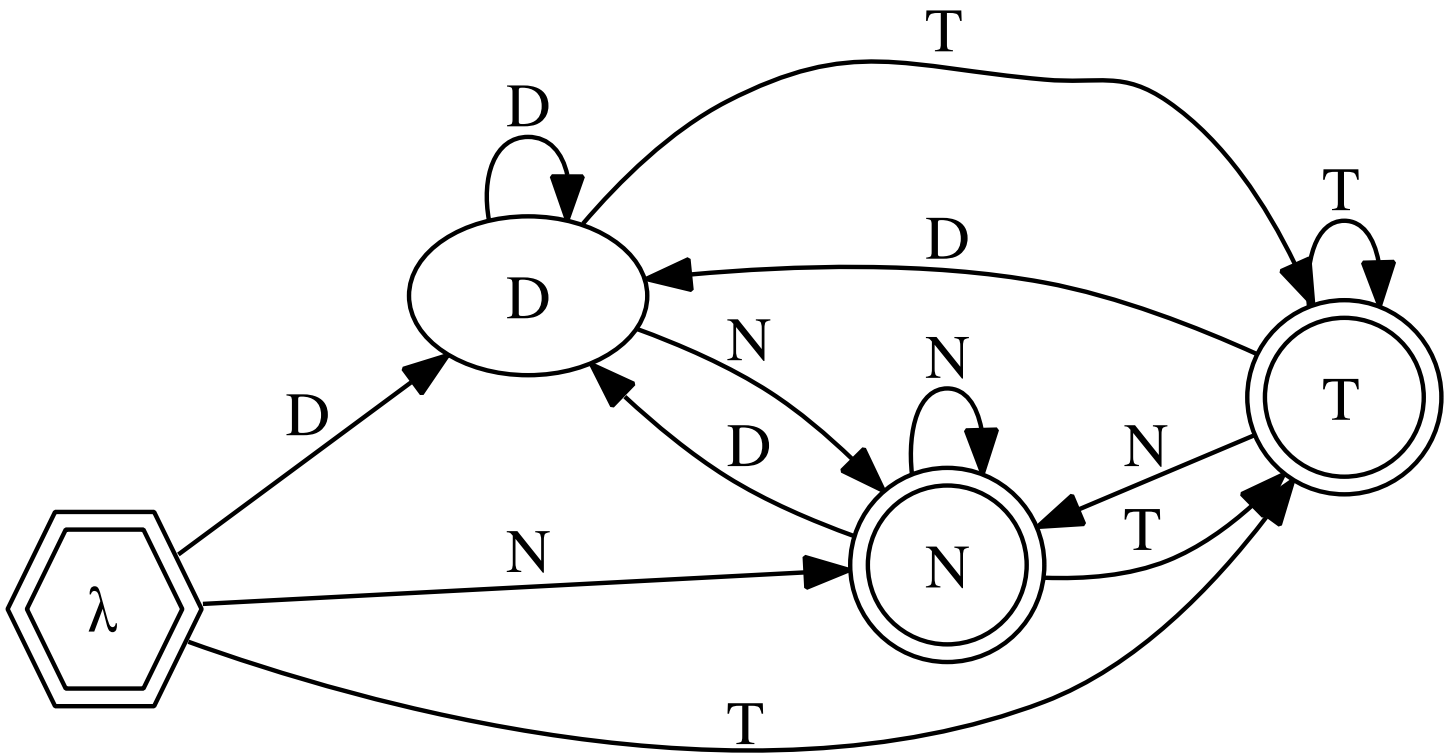


SL Languages and Phonotactics

- SL Languages and FSAs can model natural language phonotactics provided the illegal sequence is a contiguous substring bounded by some length k (i.e., it is local) (Heinz 2010).

SL Languages and Phonotactics

*D#



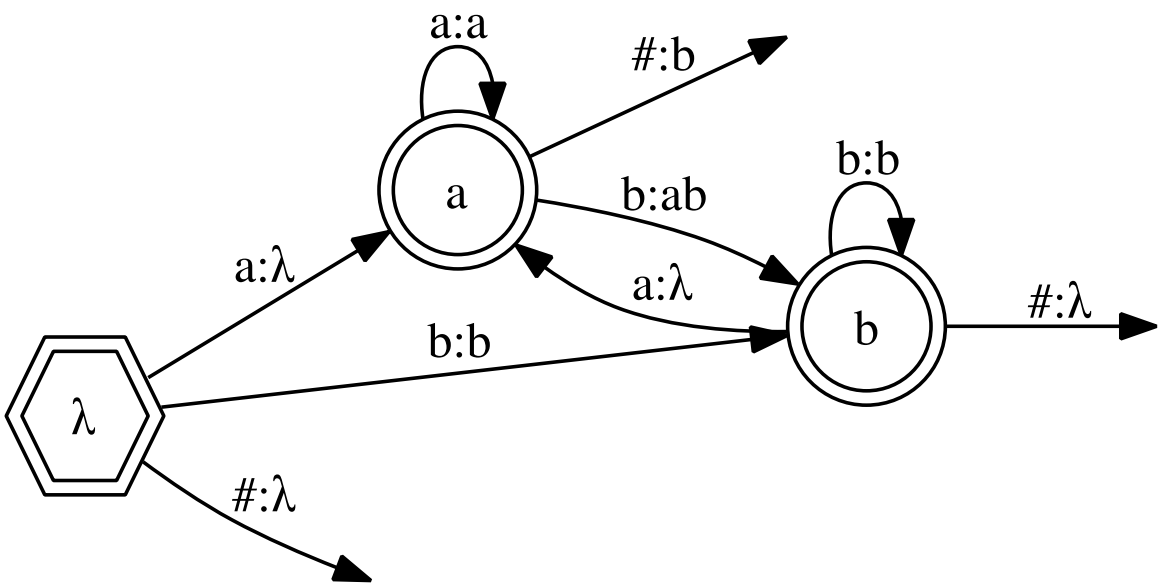
Strictly Local Functions

(2) $a \Rightarrow b \;/\; \_\_ \#$

(3)
a. $f(aba) = abb$
b. $f(aab) = aab$
c. $f(baa) = bab$
d. ...

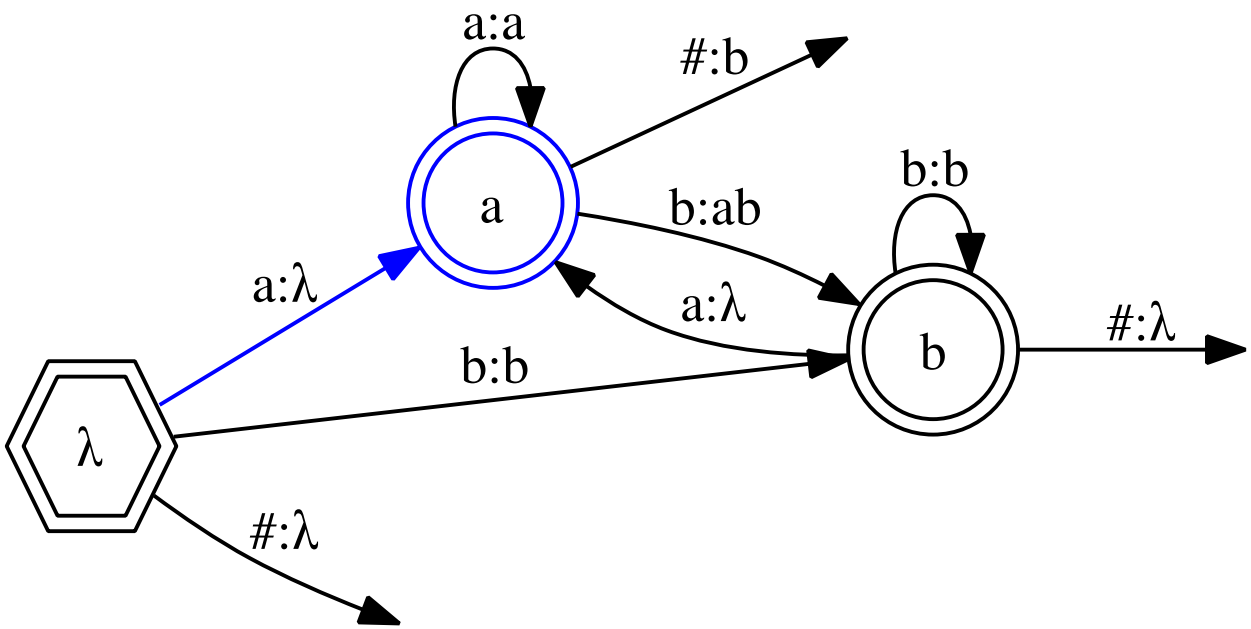
Strictly Local FSTs

(4) $a \Rightarrow b \;/\; \_\_ \#$



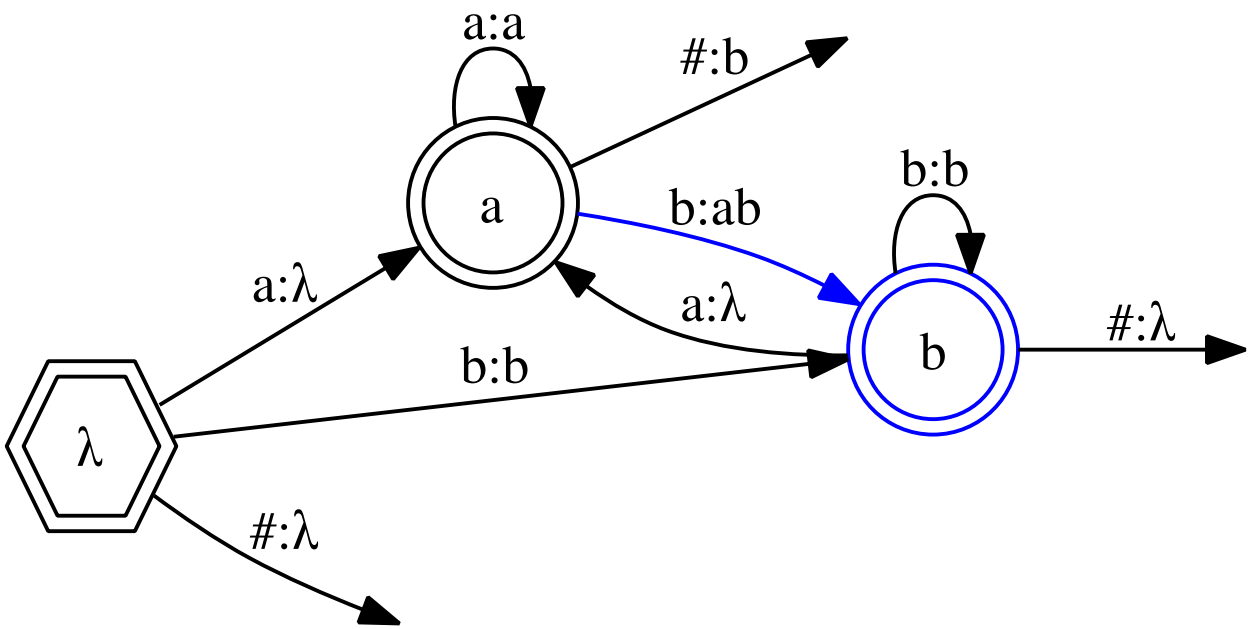
Strictly Local FSTs

Input:	a	b	a	#
Output:	λ			



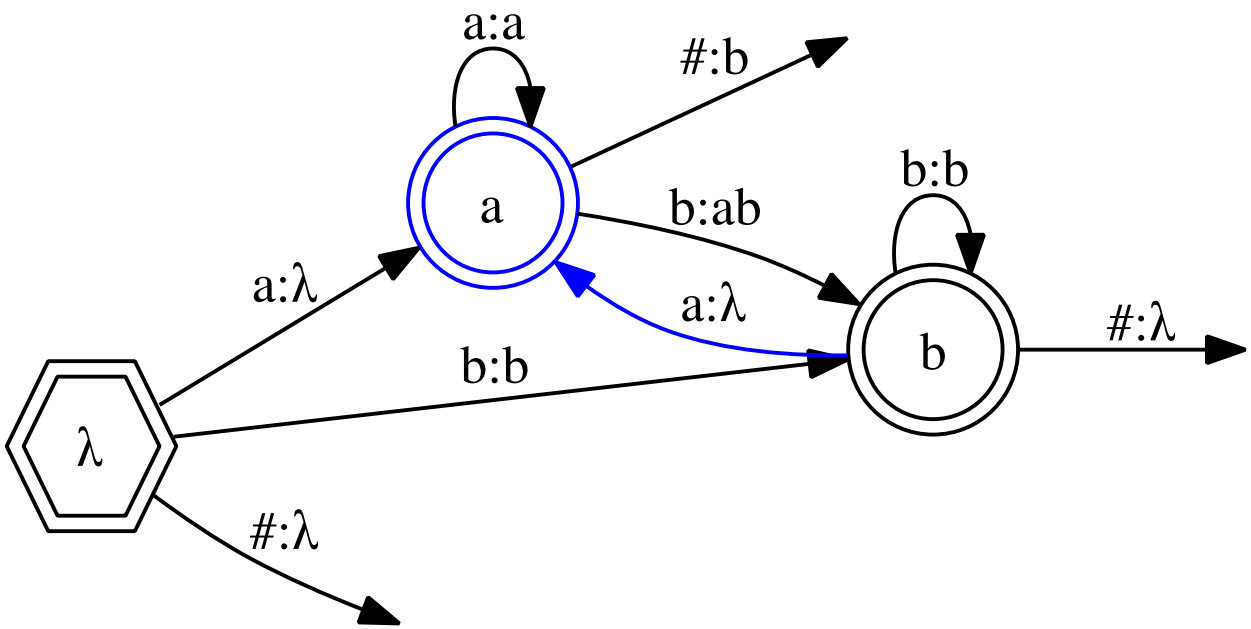
Strictly Local FSTs

Input:	a	b	a	#
Output:	λ	ab		



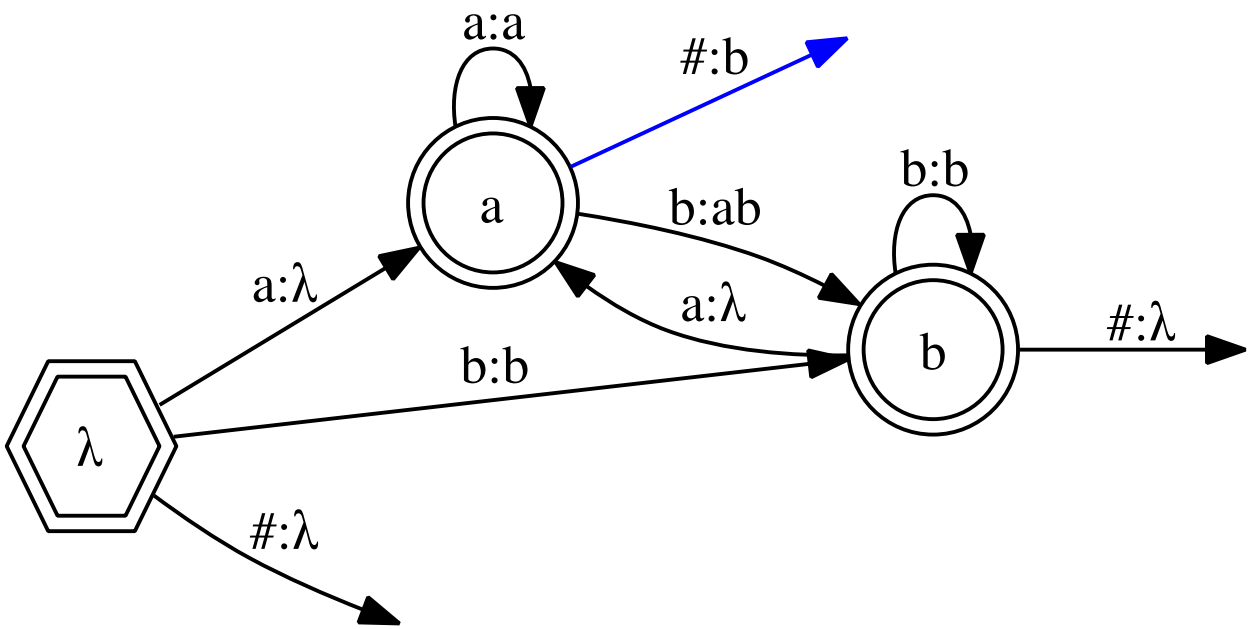
Strictly Local FSTs

Input:	a	b	a	#
Output:	λ	ab	λ	



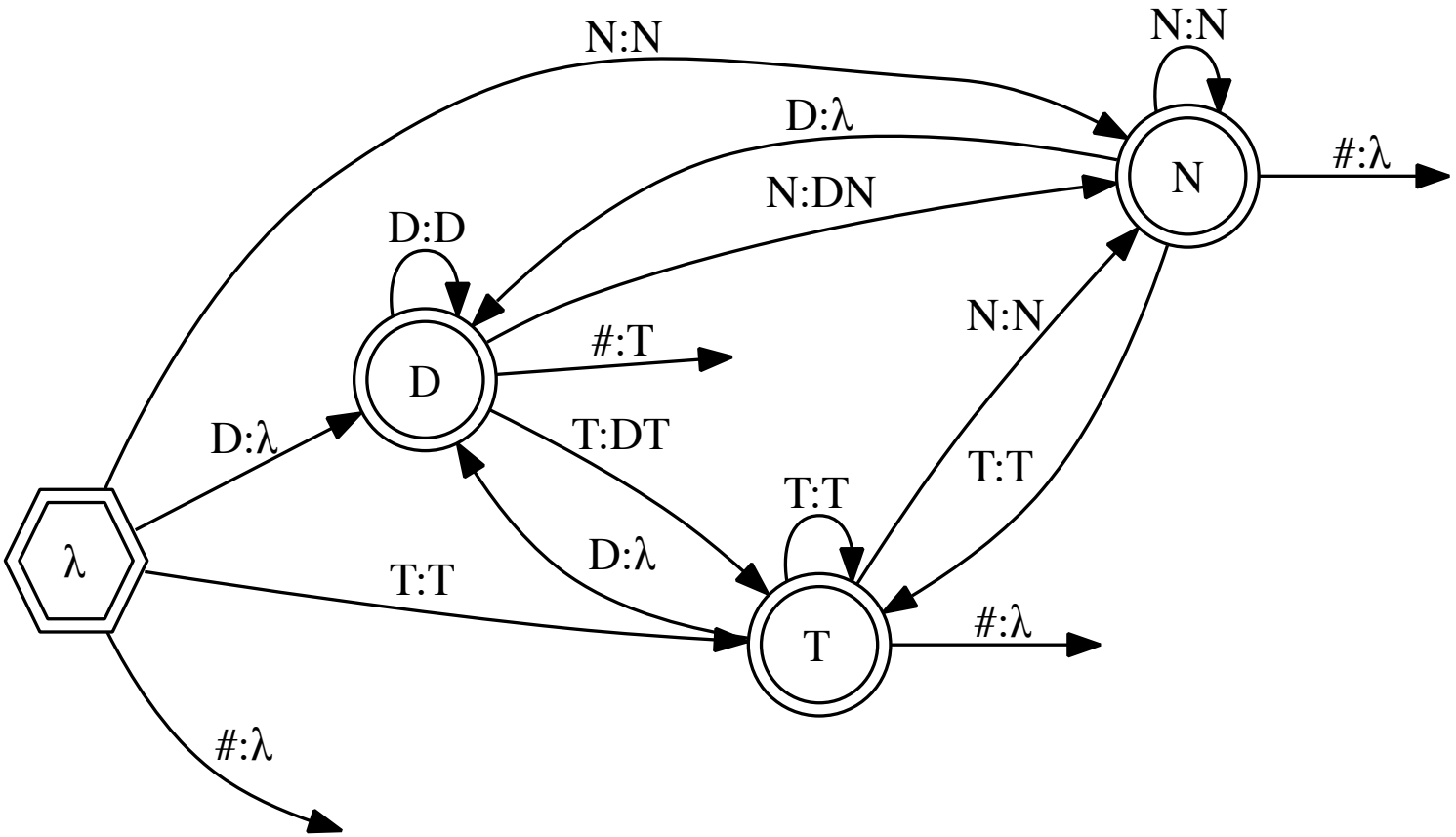
Strictly Local FSTs

Input:	a	b	a	#
Output:	λ	ab	λ	b

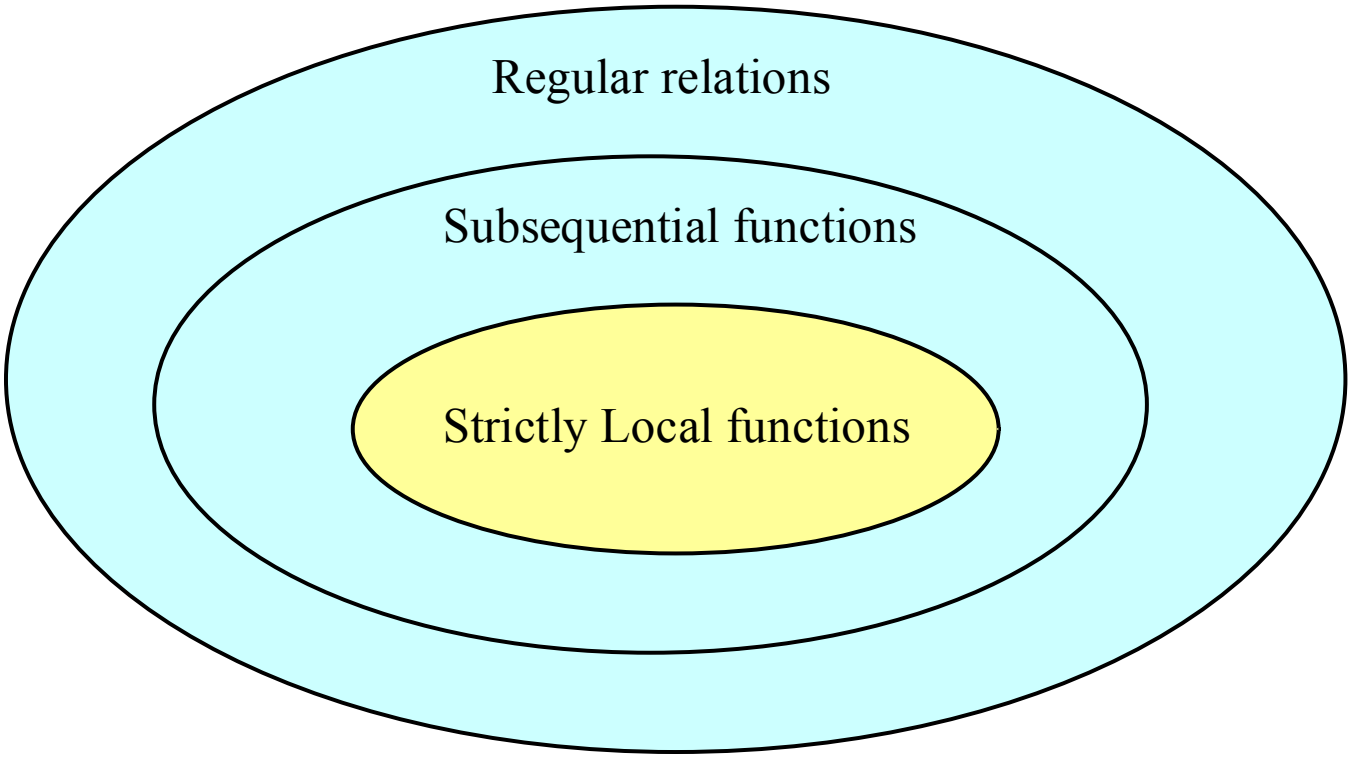


Strictly Local FSTs

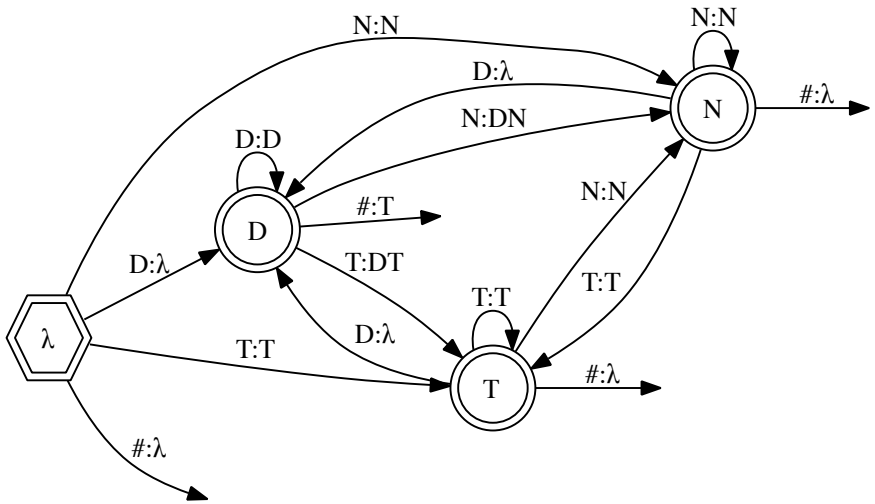
(5) $D \Rightarrow T \; / \; \_\_ \; \#$



Strictly Local FSTs



Strictly Local (Subsequential) FSTs



Subsequential
Deterministic
All states are final
Each state has a transition on #

Strictly Local
States are $k-1$-factors
Destination state of each transition $=$ the $k-1$ most recent input symbols

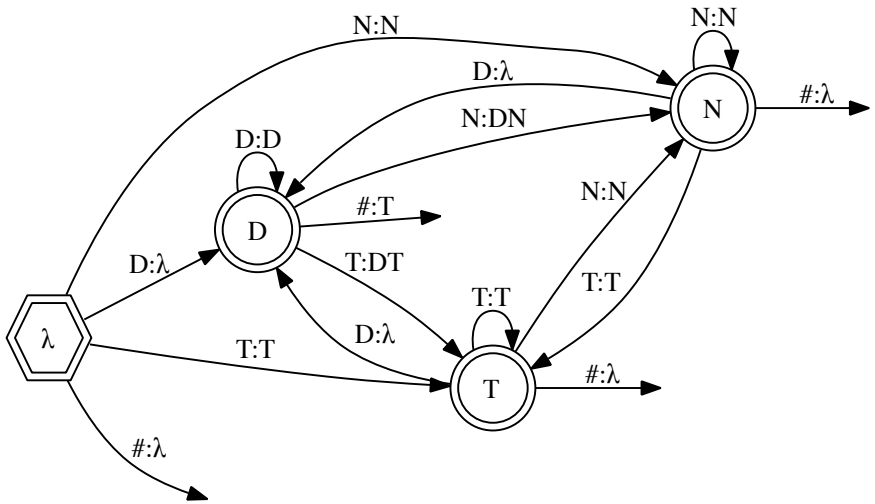
Mode of rule application

(6) $a \Rightarrow b \;/\; a \_\_ a$

Simultaneous	Left-to-right
aaaa $\mapsto$ abba	aaaa $\mapsto$ abaa

(Kaplan & Kay 1994)

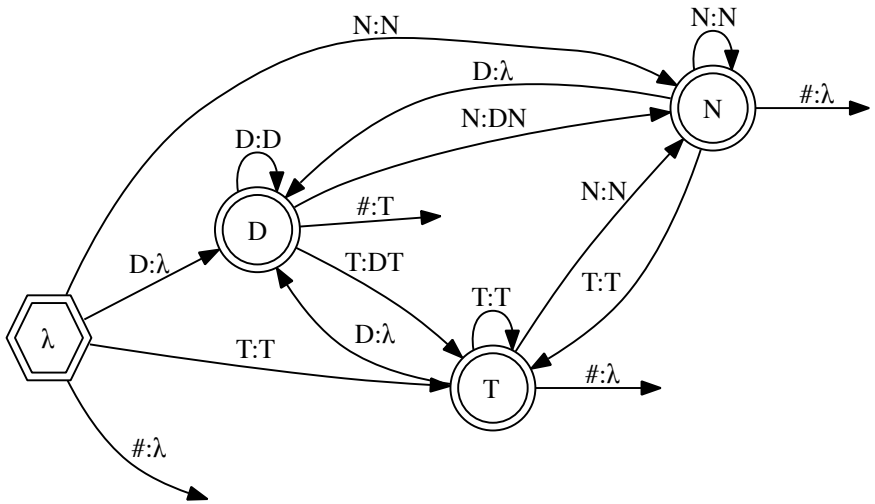
Strictly Local (Subsequential) FSTs



Subsequential
Deterministic
All states are final
Each state has a transition on #

Strictly Local (simultaneous)
States are $k-1$-factors
Destination state of each transition $=$ the $k-1$ most recent input symbols

Strictly Local (Subsequential) FSTs



Subsequential
Deterministic
All states are final
Each state has a transition on #

Strictly Local (left-to-right)
States are $k-1$-factors
Destination state of each transition $=$ the $k-1$ most recent output symbols

How many processes are SL?

- At least 96% of the approx. 5500 processes in P-Base (v1.95, Mielke 2008) are Strictly Local
- SL processes include substitution, deletion, epenthesis, bounded metathesis, local partial reduplication, and general affixation.

How many processes are SL?

(7) $A \Rightarrow B \;/\; C \_\_ D$

SL if CAD is a finite language.

How many processes are SL?

(8) D ⇒ T / __ #

CAD = D# = {b#, d#, g#, z#, ʒ#, dʒ#}

How many processes are SL?

(9) D $\Rightarrow$ T / $\_\_x$ (x has an even number of obstruents)

Not SL.

What *isn't* SL?

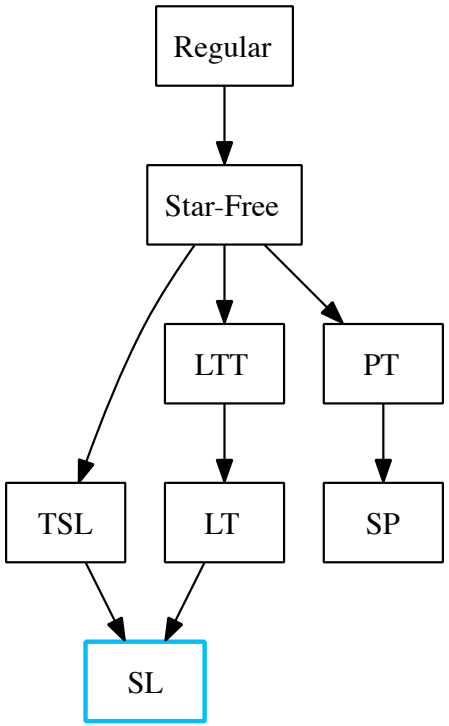
- Vowel harmony with transparent vowels
- Long-distance consonant harmony and dissimilation (Hansson 2001, Rose & Walker 2004, Suzuki 1998, Bennett 2013)
- Some tonal patterns
- Displacement/diachronic metathesis (Blevins & Garrett 1998, Hume 2000, Buckley 2011)
- Nonlocal partial reduplication (Riggle 2003)

What *isn't* SL?

All of these (except one tonal pattern) are still *subsequential*.

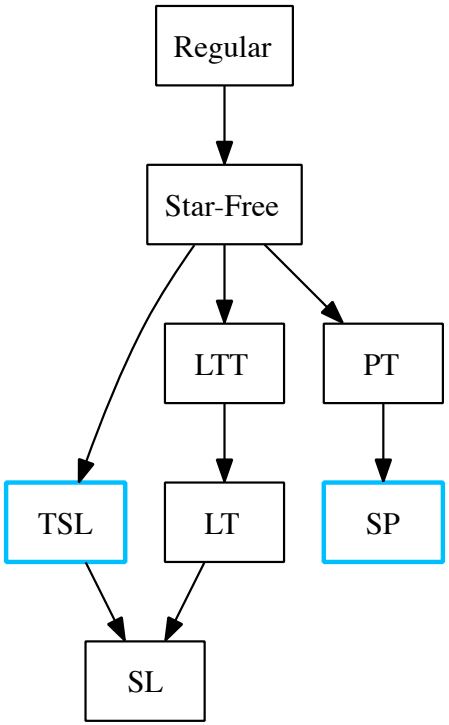
(Chandlee et al. 2012, Chandlee & Heinz 2012, Nevins 2010, Gainor et al. 2012, Heinz & Lai 2013, Luo 2013, Payne 2013, Jardine 2013)

Subregular Hierarchy



(McNaughton & Papert 1971, Rogers & Pullum 2011, Heinz et al. 2011, Rogers et al. 2012)

Subregular Hierarchy

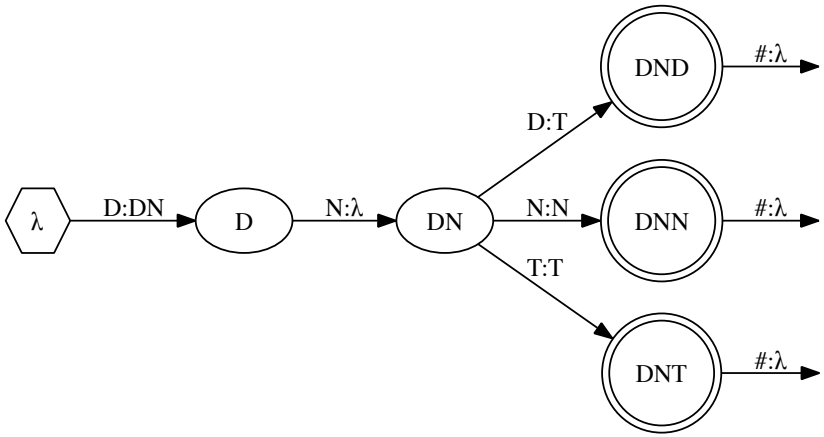


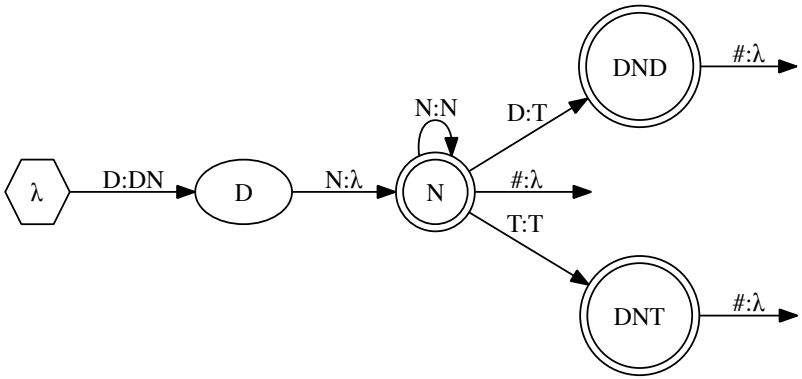
Long-distance phonotactics (Heinz 2010, Heinz, Rawal, Tanner 2011)

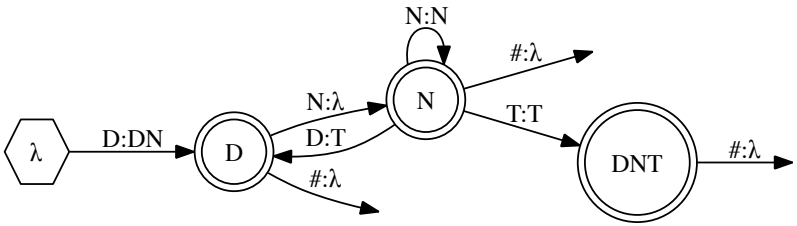
Learning SL

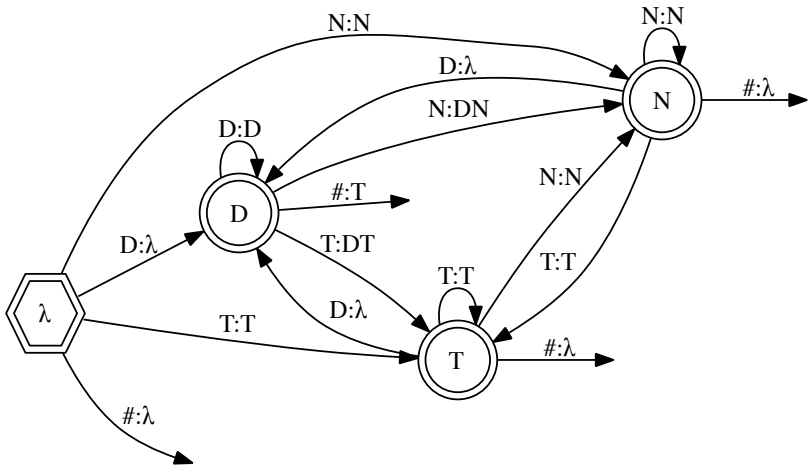
- An algorithm that uses strict locality as an inductive principle can generalize SL functions from finite data (Chandlee & Koirala 2014, Chandlee & Jardine 2014, Chandlee, Heinz, & Eyraud in prep.).
- Data: {(DND, DNT), (DNN, DNN), (DNT, DNT)}

Learning SL


λ
D:DN
D
N:λ
DN
D:T
N:N
T:T
DND
DNN
DNT
#:λ
#:λ
#:λ


λ
D:DN
D
N:λ
N
N:N
D:T
#:λ
T:T
DND
DNT
#:λ
#:λ


λ
D:DN
D
N:λ
D:T
#:λ
N
N:N
#:λ
T:T
DNT
#:λ


λ
D:λ
D
D:D
#:T
N:DN
N:N
N:N
D:λ
N
#:λ
T:DT
T:T
T:T
D:λ
T:T
T
#:λ
#:λ

Conclusions

- Modeling phonological processes with Strictly Local functions identifies locality as a defining property of many phonological UR $\mapsto$ SR mappings.
- This property 1) restricts the power of the grammatical formalism needed to account for local phenomena and 2) aides in learning.

Thank you!

Special thanks to Jeffrey Heinz, as well as James Rogers, Rémi Eyraud, Adam Jardine, the UD Computational Linguistics Group, and the UD Department of Linguistics and Cognitive Science.