Three subregular classes of formal languages for phonology

Jeffrey Heinz
heinz@udel.edu

University of Delaware

University of Pennsylvania
Linguistics Speaker Series
January 20, 2011

Collaborators and support

James Rogers (Earlham College)

Angeliki Athanasopoulou, Jane Chandlee, Jie Fu, Brian Gainor, Cesar Koirala, Regine Lai, Darrell Larsen, Tim O'Neill, Chetan Rawal (University of Delaware)

This research is supported by NSF#CPS-1035577.

Three subregular classes of formal languages for phonology

This talk:

1. provides a *tighter* computational characterization for possible phonological patterns what currently exists (Johnson 1972 and Kaplan and Kay 1994)
2. discusses the implications for *learnability* and for *measuring pattern complexity*

Caveat

This talk synthesizes the current state of a research program which is still unfolding. Some details are still being worked out…

Outline of talk

Outline

Theories of Phonology

$$F_1 \times F_2 \times \cdots \times F_n = P$$

Theories of Phonology - The Factors

$$F_1 \times F_2 \times \cdots \times F_n = P$$

The factors are the *individual* generalizations.

SPE These are *rules*.

OT, HG, HS These are markedness and faithfulness *constraints*.

(Chomsky and Halle 1968, Prince and Smolenksy 1993/2004, Legendre et al. 1990, Pater et al. 2007, McCarthy 2000, 2006 et seq.)

Theories of Phonology - The Interaction

$$F_1 \times F_2 \times \cdots \times F_n = P$$

SPE The output of one rule becomes the input to the next.

OT Optimization over ranked constraints.

HG Optimization over weighted constraints.

HS Repeated incremental changes w/OT optimization until convergence.

Theories of Phonology - The Whole Phonology

$$F_1 \times F_2 \times \cdots \times F_n = P$$

- The whole phonology is an *input/output mapping* given by the product of the factors.
- SPE, OT, HG, and HS grammars map underlying forms to surface forms.
- What kind of mapping is this?

Questions for theories of phonology

1. What is the nature of whole phonologies?
2. What is the nature of the individual generalizations?
   - I.e. what is the theory of possible rules?
   - Or what is the theory of CON?
3. How can these things be learned?

Example constraint: No word-final consonant clusters

$$*CC\#$$

As a formal language, this constraint is represented as an infinite set: all logically possible words which don't end in two consonants:

$$\{ \text{a, b, ki, kaki, bık, blık, } \ldots \}$$

Example constraint: No word-final consonant clusters

$$*\mathrm{CC}\#$$

Equivalently, we can represent this with a function:

$$\begin{array}{ccc} & f & \\ \mathrm{a} & \rightarrow & 1 \\ \mathrm{b} & \rightarrow & 1 \\ \mathrm{aa} & \rightarrow & 1 \\ \mathrm{bb} & \rightarrow & 0 \\ & \ldots & \\ \mathrm{bl\imath k} & \rightarrow & 1 \\ \mathrm{bl\imath kt} & \rightarrow & 0 \\ & \ldots & \end{array}$$

Example constraint: No word-final consonant clusters

$$*\mathrm{CC}\#$$

or a stochastic function:

$$\begin{array}{ccc} & f & \\ \mathrm{a} & \rightarrow & 0.01 \\ \mathrm{b} & \rightarrow & 0.01 \\ \mathrm{aa} & \rightarrow & 0.01 \\ \mathrm{bb} & \rightarrow & 0 \\ & \ldots & \\ \mathrm{bl\imath k} & \rightarrow & 0.001 \\ \mathrm{bl\imath kt} & \rightarrow & 0 \\ & \ldots & \end{array}$$

Example process: optional word-final consonantal deletion in English

$$[\text{+coronal,-continuant}] \longrightarrow \emptyset \ / \ \text{C} \ \underline{\qquad} \ \#$$

There are many factors which condition its application including whether the following word begins with a consonant, etcetera (Guy 1980, Guy and Boberg 1997, Coetzee 2004). These are overlooked here.

Relational and functional characterizations

obligatory [+coronal,-continuant] $\longrightarrow \emptyset$ / C ___ #

	R			f	
wɛst	$\rightarrow$	wɛs	(wɛst,wɛs)	$\rightarrow$	1
			(wɛst,wɛst)	$\rightarrow$	0
			(wɛst,wɛsk)	$\rightarrow$	0
			(wɛst,wɛ)	$\rightarrow$	0
	...		...		
pɪwɛst	$\rightarrow$	pɪwɛs	(pɪwɛst,pɪwɛs)	$\rightarrow$	1
			(pɪwɛst,pɪwɛst)	$\rightarrow$	0
			(pɪwɛst,pɪwɛsk)	$\rightarrow$	0
			(pɪwɛst,pɪwɛ)	$\rightarrow$	0
	...		...		

Figure: Fragments of the relational and functional characterization of the rule. The function f maps a pair (x, y) to 1 if and only if $x \rightarrow y$ belongs to R.

Relational and functional characterizations

optional [+coronal,-continuant] $\longrightarrow \emptyset$ / C ___ #

	R			f	
wɛst	$\rightarrow$	wɛs	(wɛst,wɛs)	$\rightarrow$	1
wɛst	$\rightarrow$	wɛst	(wɛst,wɛst)	$\rightarrow$	1
			(wɛst,wɛsk)	$\rightarrow$	0
			(wɛst,wɛ)	$\rightarrow$	0
	...		...		
pɪwɛst	$\rightarrow$	pɪwɛs	(pɪwɛst,pɪwɛs)	$\rightarrow$	1
pɪwɛst	$\rightarrow$	pɪwɛst	(pɪwɛst,pɪwɛst)	$\rightarrow$	1
			(pɪwɛst,pɪwɛsk)	$\rightarrow$	0
			(pɪwɛst,pɪwɛ)	$\rightarrow$	0
	...		...		

Figure: Fragments of the relational and functional characterization of the rule. The function f maps a pair (x, y) to 1 if and only if $x \rightarrow y$ belongs to R.

Relational and functional characterizations

optional, stochastic [+coronal,-continuant] $\longrightarrow \emptyset$ / C ___ #

Figure: Fragments of the functional characterization of the rule. Requiring

$$\sum_{x \in \Sigma^*} f(x) = 1$$

ensures a well-formed probability distribution.

	f	
(wɛst,wɛs)	$\to$	0.4
(wɛst,wɛst)	$\to$	0.6
(wɛst,wɛsk)	$\to$	0
(wɛst,wɛ)	$\to$	0
...		
(pɪwɛst,pɪwɛs)	$\to$	0.4
(pɪwɛst,pɪwɛst)	$\to$	0.6
(pɪwɛst,pɪwɛsk)	$\to$	0
(pɪwɛst,pɪwɛ)	$\to$	0
...		

The computational perspective

The functional characterizations
are the central objects of interest.

What kinds of sets, relations, and functions are individual phonological generalizations (and these whole phonologies)?

Distinguishing sets from relations

Phonotactic constraints are characterized by subsets of, or probability distributions over, all logically possible words (i.e. formal languages or formal stochastic languages).

Sets are total functions with domain Σ^*.

Phonological processes are characterized by formal relations, or formal stochastic relations.

Relations are total functions with domain $\Sigma_1^* \times \Sigma_2^*$.

Distinguishing sets from relations

Phonotactic constraints are characterized by subsets of, or probability distributions over, all logically possible words (i.e. formal languages or formal stochastic languages).

Sets are total functions with domain Σ^*.

Phonological processes are characterized by formal relations, or formal stochastic relations.

Relations are total functions with domain $\Sigma_1^* \times \Sigma_2^*$.

Henceforth, I use the word pattern to refer ambiguously to such functions, be their co-domain real or boolean.

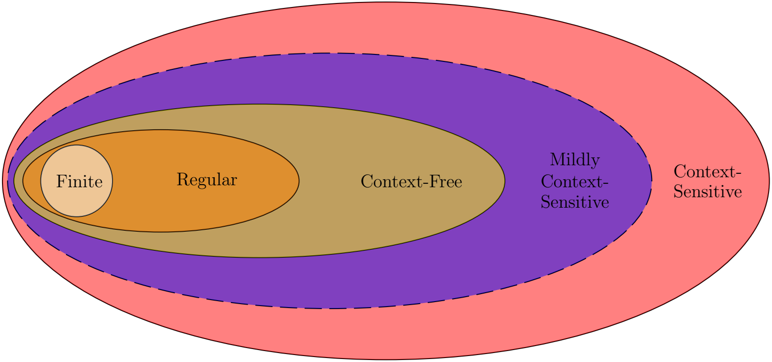
Patterns under consideration

1. Local processes:
   - 1.1 substitution (assimilation, dissimilation)
   - 1.2 epenthesis
   - 1.3 deletion
   - 1.4 metathesis
   - 1.5 bounded stress assignment
2. Long-distance processes:
   - 2.1 consonsantal harmony
   - 2.2 vowel harmony with no neutral vowels
   - 2.3 vowel harmony with transparent vowels
   - 2.4 vowel harmony with opaque vowels
   - 2.5 long-distance dissimilation
   - 2.6 unbounded stress assignment
3. Sets of surface forms derived from such processes.

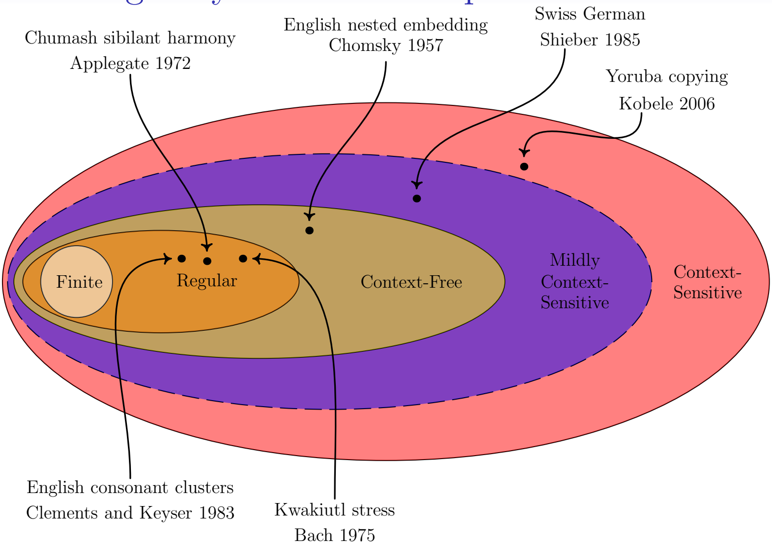
Patterns NOT under consideration

1. Reduplication, arguably morphological (Inkelas and Zoll 2005)
2. long-distance metathesis, synchronic status is questionable (Buckley in press)

Logically Possible Computable Patterns

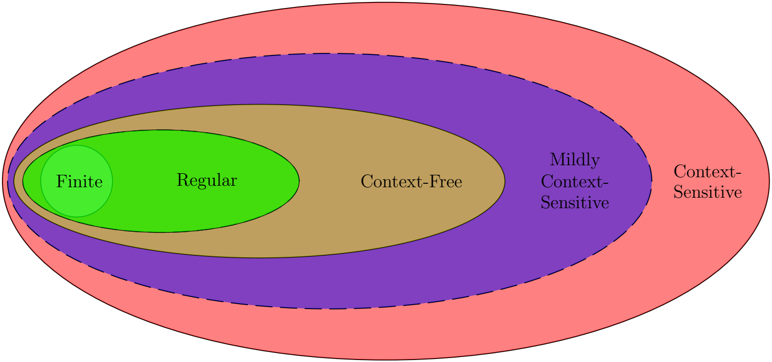


Logically Possible Computable Patterns



Where is phonology?

Logically Possible Computable Patterns



Where is phonology?

Johnson 1972, Kaplan and Kay 1994

Phonology is regular (Kaplan and Kay 1994)

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Optional, left-to-right, right-to-left, and simultaneous application of rules A $\longrightarrow$ B / C ____ D (where A,B,C,D are regular expressions) *describe regular relations*, provided the rule cannot reapply to the locus of its structural change.
2. Rule ordering is functional composition (finite-state transducer composition).
3. Regular relations are closed under composition.
4. SPE grammars (finitely many ordered rewrite rules of the above type) can describe virtually all phonological patterns.
5. Therefore, phonology is regular (both F_i and P).

Phonology is regular (Kaplan and Kay 1994)

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Optional, left-to-right, right-to-left, and simultaneous application of rules A $\longrightarrow$ B / C ____ D (where A,B,C,D are regular expressions) *describe regular relations*, provided the rule cannot reapply to the locus of its structural change.
2. Rule ordering is functional composition (finite-state transducer composition).
3. Regular relations are closed under composition.
4. SPE grammars (finitely many ordered rewrite rules of the above type) can describe virtually all phonological patterns.
5. Therefore, phonology is regular (both F_i and P).

Phonology is regular (Kaplan and Kay 1994)

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Optional, left-to-right, right-to-left, and simultaneous application of rules A $\longrightarrow$ B / C ____ D (where A,B,C,D are regular expressions) *describe regular relations*, provided the rule cannot reapply to the locus of its structural change.
2. Rule ordering is functional composition (finite-state transducer composition).
3. Regular relations are closed under composition.
4. SPE grammars (finitely many ordered rewrite rules of the above type) can describe virtually all phonological patterns.
5. Therefore, phonology is regular (both F_i and P).

Phonology is regular (Kaplan and Kay 1994)

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Optional, left-to-right, right-to-left, and simultaneous application of rules A $\longrightarrow$ B / C ____ D (where A,B,C,D are regular expressions) *describe regular relations*, provided the rule cannot reapply to the locus of its structural change.
2. Rule ordering is functional composition (finite-state transducer composition).
3. Regular relations are closed under composition.
4. SPE grammars (finitely many ordered rewrite rules of the above type) can describe virtually all phonological patterns.
5. Therefore, phonology is regular (both F_i and P).

Phonology is regular (Kaplan and Kay 1994)

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Optional, left-to-right, right-to-left, and simultaneous application of rules A $\longrightarrow$ B / C ____ D (where A,B,C,D are regular expressions) *describe regular relations*, provided the rule cannot reapply to the locus of its structural change.
2. Rule ordering is functional composition (finite-state transducer composition).
3. Regular relations are closed under composition.
4. SPE grammars (finitely many ordered rewrite rules of the above type) can describe virtually all phonological patterns.
5. Therefore, phonology is regular (both F_i and P).

Outline

Hypothesis: Phonology is Subregular.

1. The individual factors and the whole phonologies cannot be *any* regular pattern. Instead they belong to well-defined subregular regions.

Logically possible and regular phonotactic patterns

Attested	**Unattested and Weird**
Words do not have NT strings.	Words do not have 3 NT strings (but 2 is OK).
Words must have a vowel (or a syllable).	Words must have an even number of vowels (or consonants, or sibilants, ...).
If a word has sounds with [F] then they must agree with respect to [F]	If the first and last sounds in a word have [F] then they must agree with respect to [F].
Words have exactly one primary stress.	These six arbitrary words $\{w_1, w_2, w_3, w_4, w_5, w_6\}$ are well-formed.

(Pater 1996, Dixon and Aikhenvald 2002, Baković 2000, Rose and Walker 2004, Liberman and Prince 1977)

Examples of regular, but weird processes

1. $\text{t} \longrightarrow \widehat{\text{ts}}$ / C ___ D where C is all strings containing an odd number of vowels and D is all strings containing an even number of sibilants.

   /atiʃos/ → [at͡siʃos] but /apatiʃos/ → [apatiʃos]

2. l ⟶ r / #l[+seg]* ___ #

   /lalitol/ → [lalitor] but /palitol/ → [palitol]

3. ‘Sour Grapes’ vowel harmony (Padgett 1995, Wilson 2003). The example below is based on Finnish front/back harmony except that /i,e/ now block vowel harmony as opposed to being transparent to it.

   /kurœpælœ/ → [kuropalo] but /kurœpæli/ → [kurœpæli]

Examples of regular, but weird processes

1. $t \longrightarrow \widehat{ts}$ / C ____ D where C is all strings containing an odd number of vowels and D is all strings containing an even number of sibilants.

   /atiʃos/ → [a͡tsiʃos] but /apatiʃos/ → [apatiʃos]

2. l ⟶ r / #l[+seg]* ____ #

   /lalitol/ → [lalitor] but /palitol/ → [palitol]

3. 'Sour Grapes' vowel harmony (Padgett 1995, Wilson 2003). The example below is based on Finnish front/back harmony except that /i,e/ now block vowel harmony as opposed to being transparent to it.

   /kurœpælœ/ → [kuropalo] but /kurœpæli/ → [kurœpæli]

Examples of regular, but weird processes

1. $\text{t} \longrightarrow \widehat{\text{ts}}$ / C ____ D where C is all strings containing an odd number of vowels and D is all strings containing an even number of sibilants.

   /atiʃos/ → [at͡siʃos] but /apatiʃos/ → [apatiʃos]

2. $\text{l} \longrightarrow \text{r}$ / #l[+seg]* ____ #

   /lalitol/ → [lalitor] but /palitol/ → [palitol]

3. ‘Sour Grapes’ vowel harmony (Padgett 1995, Wilson 2003). The example below is based on Finnish front/back harmony except that /i,e/ now block vowel harmony as opposed to being transparent to it.

   /kurœpælœ/ → [kuropalo] but /kurœpæli/ → [kurœpæli]

Hypothesis: Phonology is Subregular.

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. The individual factors and the whole phonologies cannot be *any* regular pattern. Instead they belong to well-defined subregular regions.
2. We ought to characterize necessary and sufficient properties of these regions.
3. We ought to aim to prove that these regions are feasibly learnable (under various definitions).
4. We ought to investigate the empirical consequences from a psycholinguistic perspective.

What is at stake if phonology is subregular?

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. We obtain more precise characterizations of possible phonological patterns.

- We can decide whether some logically possible pattern is a possible phonological one.
- We can *cross-classify* to help understand *why* this is so. For example, we can formulate more precise theories which ground phonology in (articulatory or perceptual) phonetics.

What is at stake if phonology is subregular?

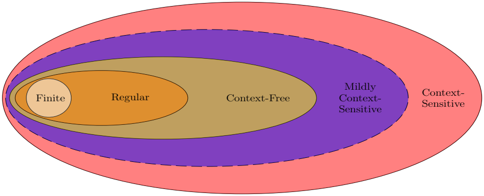
$$F_1 \times F_2 \times \cdots \times F_n = P$$

2. The computational complexity issues may resolve.

- The complexity problems noticed by Barton et al., Eisner and Idsardi stem from the the known fact that the intersection/composition of arbitrarily-many arbitrary regular sets/relations is NP-Hard.
- But if actual phonological patterns belong to more "well-behaved" subregular regions, these issues may disappear.

(Barton et. al 1997, Eisner 1997, Idsardi 2006, Heinz et al. 2009)

What is at stake if phonology is subregular?

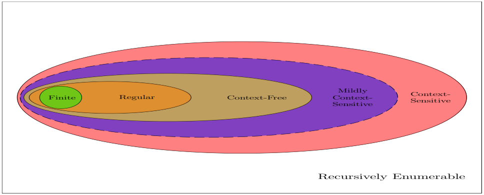


3. The learning problems may become easier to solve.

- No superfinite class of languages is identifiable in the limit from positive data (or with probability $p > 2/3$)
- The finite languages are not PAC-learnable.
- While the class of r.e. languages and stochastic languages is identifiable from positive data from computable classes of texts,
  - these learners are not feasible, and
  - the learning criteria is much weaker than these others
- But many non-superfinite classes of languages are feasibly learnable and include patterns found in natural language (proofs are often constructive)

(Gold 1967, Horning 1969, Angluin 1980, 1982, 1988, Osherson et al. 1984, Wiehagen et. al 1984, Pitt 1985, Valiant 1984, Blum et. al 1989, Garcia et al. 1990, Muggleton 1990, Jain et. al 1999, Kearns and Vazirani 1994, Yokomori 2003, Clark and Thollard 2004, Oates et al. 2006, Niyogi 2006, Chater and Vitanyí 2007, Clark and Eryaud 2007, Heinz 2008, 2010, Yoshinaka 2008, Case et al. 2009, de la Higuera 2010)

What is at stake if phonology is subregular?

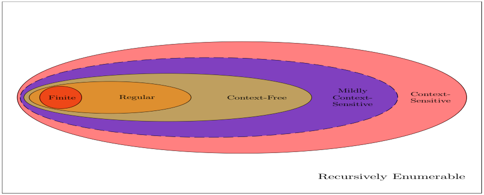


3. The learning problems may become easier to solve.

- No superfinite class of languages is identifiable in the limit from positive data (or with probability $p > 2/3$)
- The finite languages are not PAC-learnable.
- While the class of r.e. languages and stochastic languages is identifiable from positive data from computable classes of texts,
  - these learners are not feasible, and
  - the learning criteria is much weaker than these others
- But many non-superfinite classes of languages are feasibly learnable and include patterns found in natural language (proofs are often constructive)

(Gold 1967, Horning 1969, Angluin 1980, 1982, 1988, Osherson et al. 1984, Wiehagen et. al 1984, Pitt 1985, Valiant 1984, Blum et. al 1989, Garcia et al. 1990, Muggleton 1990, Jain et. al 1999, Kearns and Vazirani 1994, Yokomori 2003, Clark and Thollard 2004, Oates et al. 2006, Niyogi 2006, Chater and Vitanyí 2007, Clark and Eryaud 2007, Heinz 2008, 2010, Yoshinaka 2008, Case et al. 2009, de la Higuera 2010)

What is at stake if phonology is subregular?

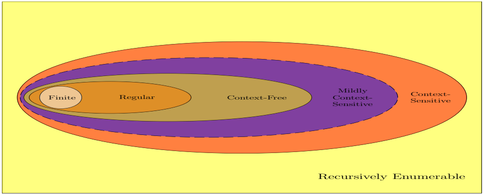


3. The learning problems may become easier to solve.

- No superfinite class of languages is identifiable in the limit from positive data (or with probability $p > 2/3$)
- The finite languages are not PAC-learnable.
- While the class of r.e. languages and stochastic languages is identifiable from positive data from computable classes of texts,
  - these learners are not feasible, and
  - the learning criteria is much weaker than these others
- But many non-superfinite classes of languages are feasibly learnable and include patterns found in natural language (proofs are often constructive)

(Gold 1967, Horning 1969, Angluin 1980, 1982, 1988, Osherson et al. 1984, Wiehagen et. al 1984, Pitt 1985, Valiant 1984, Blum et. al 1989, Garcia et al. 1990, Muggleton 1990, Jain et. al 1999, Kearns and Vazirani 1994, Yokomori 2003, Clark and Thollard 2004, Oates et al. 2006, Niyogi 2006, Chater and Vitanyí 2007, Clark and Eryaud 2007, Heinz 2008, 2010, Yoshinaka 2008, Case et al. 2009, de la Higuera 2010)

What is at stake if phonology is subregular?

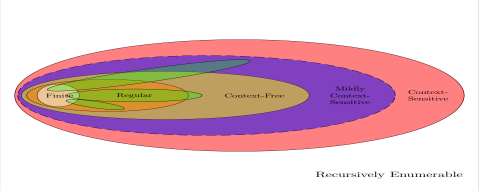


3. The learning problems may become easier to solve.

- No superfinite class of languages is identifiable in the limit from positive data (or with probability $p > 2/3$)
- The finite languages are not PAC-learnable.
- While the class of r.e. languages and stochastic languages is identifiable from positive data from computable classes of texts,
  - these learners are not feasible, and
  - the learning criteria is much weaker than these others
- But many non-superfinite classes of languages are feasibly learnable and include patterns found in natural language (proofs are often constructive)

(Gold 1967, Horning 1969, Angluin 1980, 1982, 1988, Osherson et al. 1984, Wiehagen et. al 1984, Pitt 1985, Valiant 1984, Blum et. al 1989, Garcia et al. 1990, Muggleton 1990, Jain et. al 1999, Kearns and Vazirani 1994, Yokomori 2003, Clark and Thollard 2004, Oates et al. 2006, Niyogi 2006, Chater and Vitanyí 2007, Clark and Eryaud 2007, Heinz 2008, 2010, Yoshinaka 2008, Case et al. 2009, de la Higuera 2010)

What is at stake if phonology is subregular?

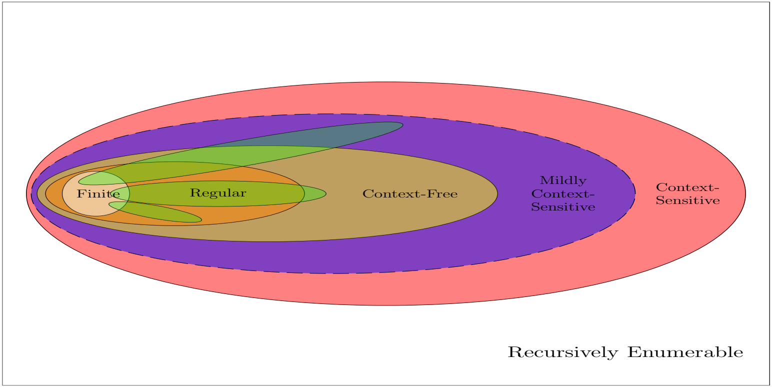


3. The learning problems may become easier to solve.

- No superfinite class of languages is identifiable in the limit from positive data (or with probability $p > 2/3$)
- The finite languages are not PAC-learnable.
- While the class of r.e. languages and stochastic languages is identifiable from positive data from computable classes of texts,
  - these learners are not feasible, and
  - the learning criteria is much weaker than these others
- But many non-superfinite classes of languages are feasibly learnable and include patterns found in natural language (proofs are often constructive)

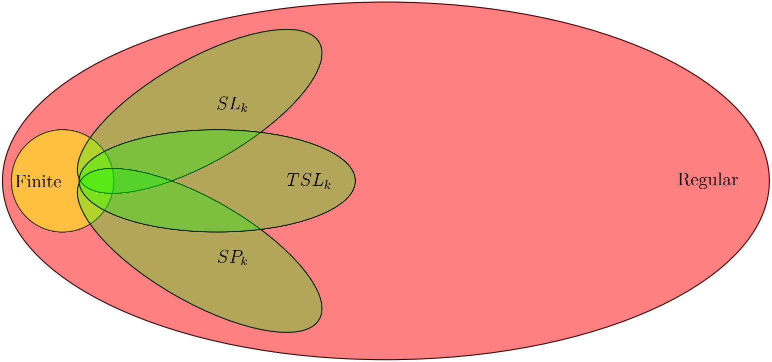
(Gold 1967, Horning 1969, Angluin 1980, 1982, 1988, Osherson et al. 1984, Wiehagen et. al 1984, Pitt 1985, Valiant 1984, Blum et. al 1989, Garcia et al. 1990, Muggleton 1990, Jain et. al 1999, Kearns and Vazirani 1994, Yokomori 2003, Clark and Thollard 2004, Oates et al. 2006, Niyogi 2006, Chater and Vitanyí 2007, Clark and Eryaud 2007, Heinz 2008, 2010, Yoshinaka 2008, Case et al. 2009, de la Higuera 2010)

What is at stake if phonology is subregular?



4. The learning solutions can help explain the limits of phonological variation (Heinz 2009, 2010).

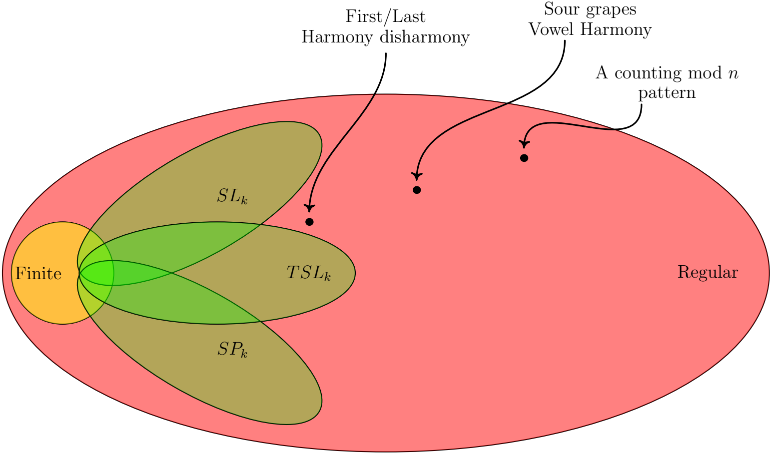
Three classes for phonology



SL_k means Strictly k-Local patterns.
SP_k means Strictly k-Piecewise patterns.
TSL_k means Tier-based SL_k patterns.

These classes are incomparable

Three classes for phonology



SL_k means Strictly k-Local patterns.
SP_k means Strictly k-Piecewise patterns.
TSL_k means Tier-based SL_k patterns.

These classes are incomparable and demonstrably exclude many unattested, weird regular patterns.

Specific claims

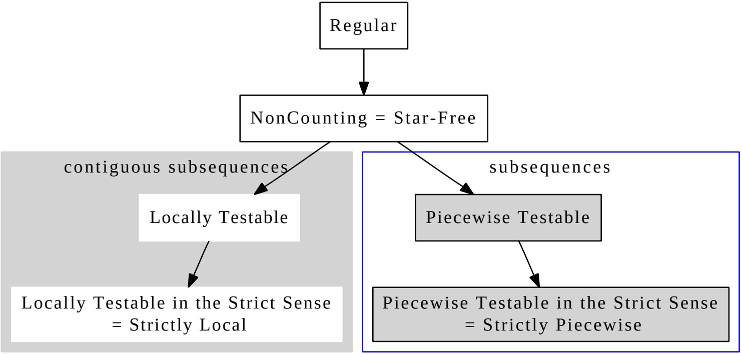
1. Local processes belong to SL_k
   1.1 substitution (assimilation, dissimilation)
   1.2 epenthesis
   1.3 deletion
   1.4 metathesis
   1.5 bounded stress assignment
2. Long-distance processes belong to SL_k or TSL_k
   2.1 consonsantal harmony (SP_k, TSL_k)
   2.2 vowel harmony with no neutral vowels (SP_k, TSL_k)
   2.3 vowel harmony with transparent vowels (SP_k)
   2.4 vowel harmony with opaque vowels (TSL_k)
   2.5 long-distance dissimilation (TSL_k)
   2.6 unbounded stress assignment (SP_k, TSL_k?)
3. Sets of surface forms derived from such processes.

Explaining the details

1. Subregular sets (Subregular Hierarchies) for phonotactic patterns
    - Strictly Local
    - Strictly Piecewise
    - Tier-based Strictly Local
2. Subregular relations for phonological alternations
    - Subsequential relations
    - Subclasses of the subsequential relations based on the above three classes
3. Of course there are senses in which even these three subregular classes are “too big”, but they provide a substantially tighter bound that what previously existed.

Outline

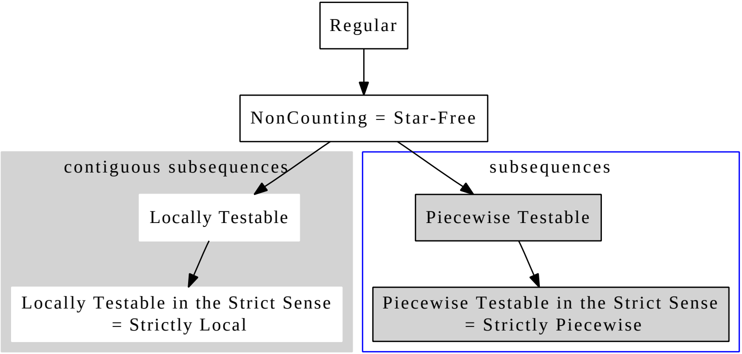
Dual hierarchies of subregular sets (simplified)



- Each class has independent, equivalent characterizations from formal language theory, group theory, logic, and automata theory.

  (McNaughton and Papert 1971, Simon 1975, Rogers and Pullum 2007, Rogers et. al 2010)

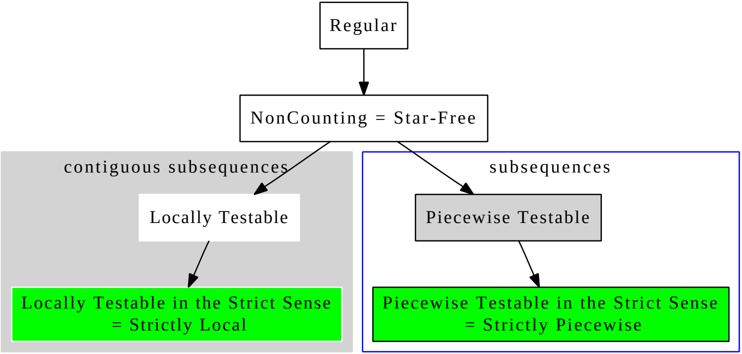
Dual hierarchies of subregular sets (simplified)



- Decision procedures and closure properties under intersection, concatenation, etc. are known.

  (McNaughton and Papert 1971, Simon 1975, Rogers and Pullum 2007, Rogers et. al 2010)

Dual hierarchies of subregular sets (simplified)

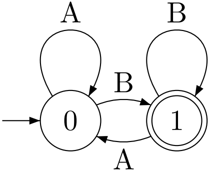


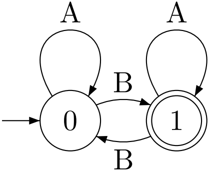
- We introduce the Tier-based Strictly Local, which is properly Noncounting.

  (McNaughton and Papert 1971, Simon 1975, Rogers and Pullum 2007, Rogers et. al 2010)

Measures of complexity

Sequences of As and Bs which end in B	Sequences of As and Bs with an odd number of Bs
$(A+B)^*B$	$(A^*BA^*BA^*)^*A^*BA^*$
Minimal deterministic finite-state automata	Minimal deterministic finite-state automata





Conclusion: The size of the DFA as given by the Nerode equivalence relation doesn't seem appropriate.

Strictly k-Local: Adjacency—Substrings

$$\rtimes \quad \mathrm{C} \quad \mathrm{V} \quad \mathrm{C} \quad \mathrm{V} \quad \ltimes$$

Definition

u is a factor of w iff $w = xuy$ for some $x, y \in \Sigma^*$.

u is a k-factor of w iff u is a factor and $|u| = k$.

The container of u is

$$C(u) = \{w \in \{\rtimes\}\Sigma^*\{\ltimes\} : u \text{ is a factor of } w\}$$

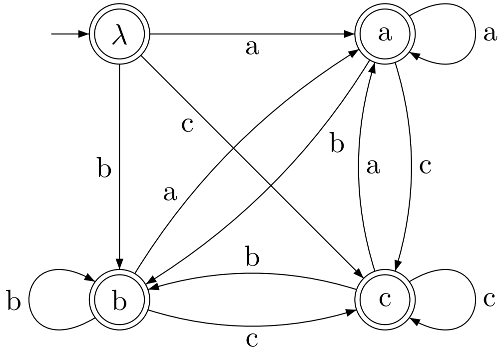
Note $\overline{C(u)}$ is the set of all words *not* containing the factor u.

$L \in SL_k$ iff there exists a finite $S \subseteq \{\rtimes\}\Sigma^{<k} \cup \Sigma^{\leq k} \cup \Sigma^{<k}\{\ltimes\}$ such that

$$L = \bigcap_{u \in S} \overline{C(u)}$$

FSA illustration of SL_2

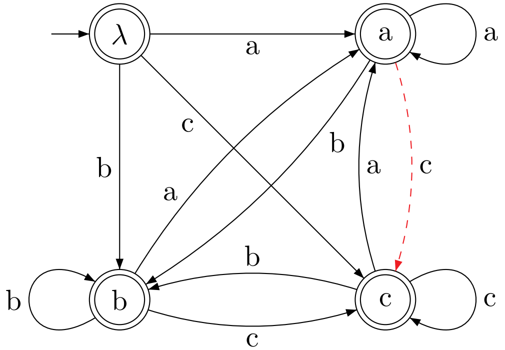
Let $\Sigma = \{a, b, c\}$. Forbidding factors is equivalent to considering subgraphs of the following.
Examples.



FSA illustration of SL_2

Let $\Sigma = \{a, b, c\}$. Forbidding factors is equivalent to considering subgraphs of the following.

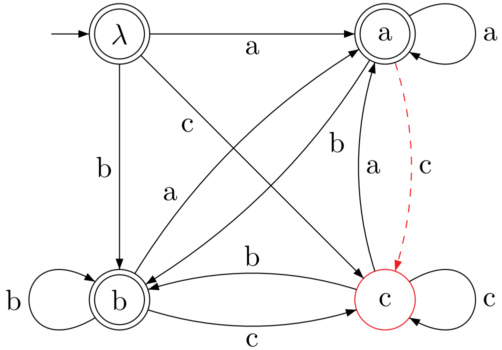
Examples. Suppose ac is forbidden.



FSA illustration of SL_2

Let $\Sigma = \{a, b, c\}$. Forbidding factors is equivalent to considering subgraphs of the following.

Examples. Suppose ac and $c\ltimes$ is forbidden.



FSA characterization of SL_k languages

$L \in SL_k$ iff it is a subgraph of the following automata

- The states $Q = \Sigma^{<k}$
- The initial states are $I = \{\lambda\}$
- The final states are $F = Q$
- The transition function $\delta(a_1 \cdots a_n, b) =$

  $a_2 \cdots a_n b$ iff $|a_1 \cdots a_n b| \geq k$ and $|a_1 \cdots a_n b|$ otherwise

Examples: Strictly K-Local Markedness Constraints

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. *a is SL_1.
2. *NT is SL_2.
3. *$\acute{\sigma}\#$ is SL_2.
4. *CCC is SL_3.

More Examples: Stress Patterns

$$F_1 \times F_2 \times \cdots \times F_n = P$$

Edlefsen et. al (2008) classify the 109 patterns in the Stress Pattern Database (Heinz 2007,2009).

9 are SL_2	Abun West, Afrikans, Maranungku, Cambodian, ...
44 are SL_3	Alawa, Arabic (Bani-Hassan), ...
24 are SL_4	Arabic (Cairene), ...
3 are SL_5	Asheninca, Bhojpuri, Hindi (Fairbanks)
1 is SL_6	Icua Tupi
28 are not SL	Amele, Bhojpuri (Shukla Tiwari),
	Arabic Classical, Hindi (Keldar), Yidin, ...

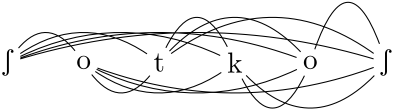
72% are SL_k for $k \leq 6$. 49% are SL_3.

What is not SL_k

For any k:

1. Unbounded stress patterns (because the primary stress may occur arbitrarily far from a word edge)
2. Long-distance harmony and disharmony patterns (because arbitrarily long material may occur between segments)

Strictly Piecewise (Rogers et al. 2010)



Definition

u is a subsequence of w iff $u = a_0 a_1 \cdots a_n$ and $w \in \Sigma^* a_0 \Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$.

u is a k-subsequence of w iff u is a subsequence of w and $|u| = k$.

The shuffle ideal of u is $SI(u) = \{w : u \text{ is a subsequence of } w\}$.

Note $\overline{SI(u)}$ is the set of all words *not* containing the subsequence u.

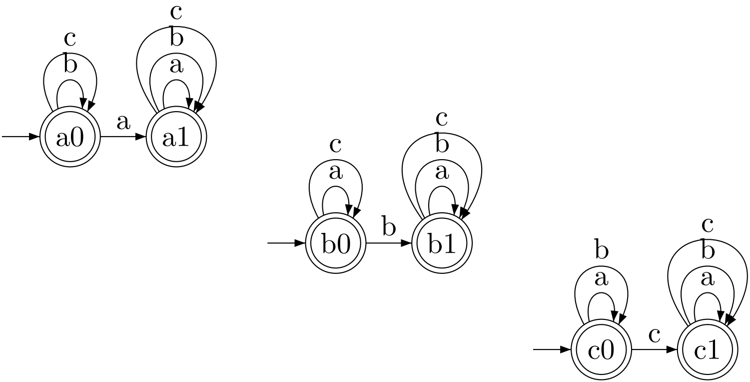
$L \in SP_k$ iff there exists a finite set $S \subset \Sigma^{\leq k}$ such that

$$L = \bigcap_{w \in S} \overline{SI(w)}$$

FSA representation of SP_2.

Let $\Sigma = \{a, b, c\}$. Forbidding subsequences is equivalent to removing transitions from the rightmost states in the acceptors below and then taking the intersection of these acceptors.

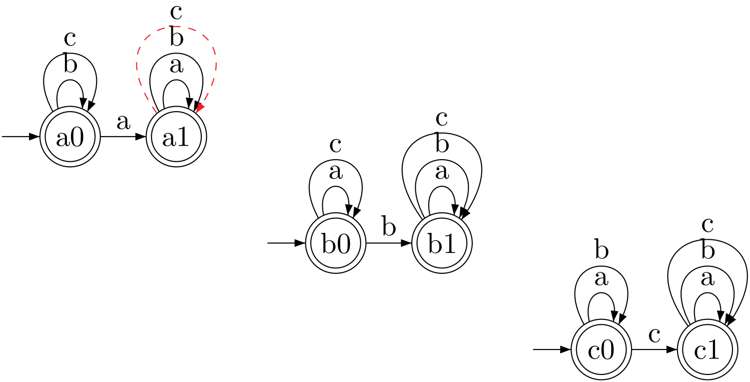
Examples.



FSA representation of SP_2.

Let $\Sigma = \{a, b, c\}$. Forbidding subsequences is equivalent to removing transitions from the rightmost states in the acceptors below and then taking the intersection of these acceptors.

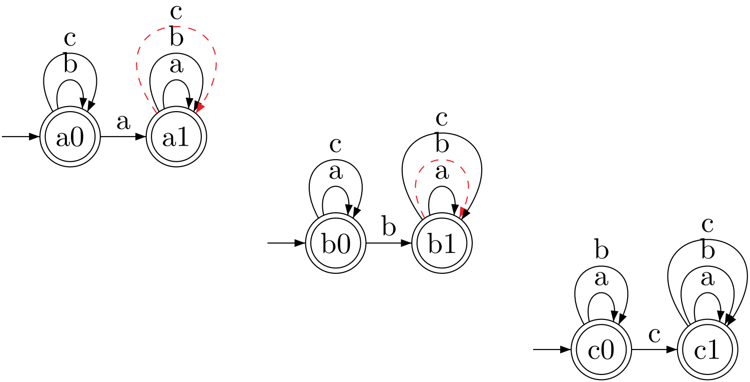
Examples. Suppose ac is forbidden.



FSA representation of SP_2.

Let $\Sigma = \{a, b, c\}$. Forbidding subsequences is equivalent to removing transitions from the rightmost states in the acceptors below and then taking the intersection of these acceptors.

Examples. Suppose *ac* and *bb* is forbidden.



Examples: What is SP_k (Heinz 2010)

SP_2 includes phonotactic patterns derived from:

1. Asymmetric consonantal harmony
    - Sibilant Harmony in Sarcee (Cook 1978a,b, 1984)
    - E.g. forbidding the subsequence *sʃ
2. Symmetric consonantal harmony
    - Sibilant Harmony in Navajo (Sapir and Hojier 1967, Fountain 1998)
    - E.g forbidding the subsequences ʃs and sʃ
3. Vowel harmony patterns with transparent vowels
    - Finnish, Korean sound-symbolic harmony, ...
4. Unbounded stress patterns, once *culminativity* is factored out (Heinz, in prep). (wrinkle: Culminativity is properly Piecewise Testable.)

Examples: What is SP_k (Heinz 2010)

SP_2 includes phonotactic patterns derived from:

1. Asymmetric consonantal harmony
    - Sibilant Harmony in Sarcee (Cook 1978a,b, 1984)
    - E.g. forbidding the subsequence *sʃ
2. Symmetric consonantal harmony
    - Sibilant Harmony in Navajo (Sapir and Hojier 1967, Fountain 1998)
    - E.g forbidding the subsequences ʃs and sʃ
3. Vowel harmony patterns with transparent vowels
    - Finnish, Korean sound-symbolic harmony, …
4. Unbounded stress patterns, once *culminativity* is factored out (Heinz, in prep). (wrinkle: Culminativity is properly Piecewise Testable.)

Examples: What is SP_k (Heinz 2010)

SP_2 includes phonotactic patterns derived from:

1. Asymmetric consonantal harmony
   - Sibilant Harmony in Sarcee (Cook 1978a,b, 1984)
   - E.g. forbidding the subsequence *sʃ
2. Symmetric consonantal harmony
   - Sibilant Harmony in Navajo (Sapir and Hojier 1967, Fountain 1998)
   - E.g forbidding the subsequences ʃs and sʃ
3. Vowel harmony patterns with transparent vowels
   - Finnish, Korean sound-symbolic harmony, ...
4. Unbounded stress patterns, once *culminativity* is factored out (Heinz, in prep). (wrinkle: Culminativity is properly Piecewise Testable.)

Examples: What is SP_k (Heinz 2010)

SP_2 includes phonotactic patterns derived from:

1. Asymmetric consonantal harmony
   - Sibilant Harmony in Sarcee (Cook 1978a,b, 1984)
   - E.g. forbidding the subsequence *sʃ
2. Symmetric consonantal harmony
   - Sibilant Harmony in Navajo (Sapir and Hojier 1967, Fountain 1998)
   - E.g forbidding the subsequences ʃs and sʃ
3. Vowel harmony patterns with transparent vowels
   - Finnish, Korean sound-symbolic harmony, ...
4. Unbounded stress patterns, once *culminativity* is factored out (Heinz, in prep). (wrinkle: Culminativity is properly Piecewise Testable.)

What is not SP_k

1. Vowel harmony with blocking, i.e. with opaque vowels
2. Long-distance dissimilation

   Example. Latin Liquid Dissimilation ([l,r] on own tier):

a.	/nav-alis/	nav-alis	'naval'
b.	/episcop-alis/	episcop-alis	'episcopal'
c.	/infiti-alis/	infiti-alis	'negative'
d.	/sol-alis/	sol-aris	'solar'
e.	/lun-alis/	lun-aris	'lunar'
f.	/milit-alis/	milit-aris	'military'

   However, the rule does not apply if an [r] intervenes.

g.	/flor-alis/	flor-alis	'floral'	*flor-aris
h.	/sepulkr-alis/	sepulkr-alis	'funereal'	*sepulkr-aris
i.	/litor-alis/	litor-alis	'of the shore'	*litor-aris

   Note there are some exceptions; e.g. 'fili-alis' and 'glute-alis'.

Tier-based SL_k Patterns

Let T be a subset of Σ. T is the tier.

Definition

First, define the Erasing function for all $w = a_1 \cdots a_n$:

$$E_T(w) = v_1 \cdots v_n \text{ where } v_i = a_i \text{ iff } a_i \in T \text{ and } v_i = \lambda \text{ otherwise}$$

The tier-based container of $u \in \{\rtimes\}T^*\{\ltimes\}$ is

$$TC(u) = \{w \in \{\rtimes\}\Sigma^*\{\ltimes\} : u \text{ is a factor of } E_T(w)\}$$

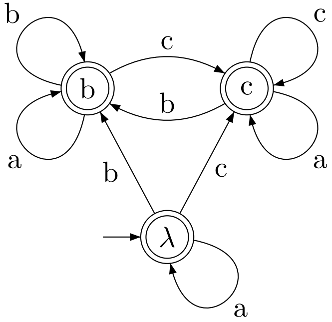
Note $\overline{TC(u)}$ is the set of all words *not* containing the factor u on tier T.

$L \in TSL_k$ iff there exists a finite $S \subseteq \{\rtimes\}\Sigma^{<k} \cup \Sigma^{\leq k} \cup \Sigma^{<k}\{\ltimes\}$ such that

$$L = \bigcap_{u \in S} \overline{TC(u)}$$

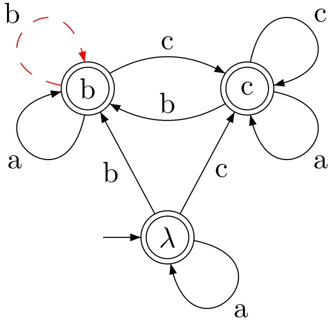
FSA representation of TSL_2

Let $\Sigma = \{a, b, c\}$ and $T = \{b, c\}$. Forbidding factors on the tier T is equivalent to considering subgraphs of the following. Examples.



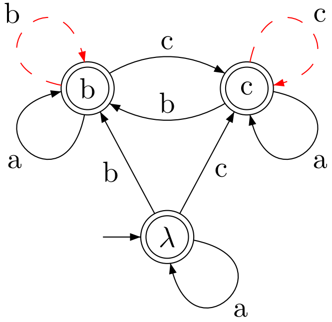
FSA representation of TSL_2

Let $\Sigma = \{a, b, c\}$ and $T = \{b, c\}$. Forbidding factors on the tier T is equivalent to considering subgraphs of the following.
Examples. Suppose bb is forbidden.



FSA representation of TSL_2

Let $\Sigma = \{a, b, c\}$ and $T = \{b, c\}$. Forbidding factors on the tier T is equivalent to considering subgraphs of the following.
Examples. Suppose *bb* and *cc* is forbidden.



What is TSL_k

Phonotactic patterns derivable from

1. Vowel harmony with opaque vowels.
2. Long-distance dissimilation patterns

What is not TSL_k

Phonotactic patterns derivable from

1. Vowel harmony with transparent vowels
   (unless the transparent vowels are off the tier).

Learning Results for these Subregular Sets

Theorem

Given k, SL_k and SP_k are identifiable in the limit from positive data by an incremental learner which is efficient in the size of the sample and which has many other desirable properties. They are also PAC-learnable.

(Garcia et al. 1990, Heinz 2010b, Kasprzik and Kötzing 2010)

Learning Results for these Subregular Sets

Theorem

Given k, and a sample S drawn according to a stochastic SL_k (SP_k) language, there is an efficient procedure which finds the parameter values which maximizes the likelihood of S w.r.t the SL_k (SP_k) family of distributions.

(Jurafsky and Martin 2008, Heinz and Rogers 2010)

Learning Results for these Subregular Sets

Theorem

Given k and the tier T, TSL_k is identifiable in the limit from positive data by an incremental learner which is efficient in the size of the sample and which has many other desirable properties. They are also PAC-learnable.

(Heinz 2010b, Kasprzik and Kötzing 2010)

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Three subregular classes circumscribe virtually all attested phonotactic patterns (F_i), and exclude many weird, regular ones.
2. These classes are learnable under established definitions, and provide better measures of pattern complexity than DFA size.
3. Noncounting is closed under intersection so if the interaction ($\times$) is intersection ($\cap$) then at worst P is Noncounting

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Three subregular classes circumscribe virtually all attested phonotactic patterns (F_i), and exclude many weird, regular ones.
2. These classes are learnable under established definitions, and provide better measures of pattern complexity than DFA size.
3. Noncounting is closed under intersection so if the interaction ($\times$) is intersection ($\cap$) then at worst P is Noncounting

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. Three subregular classes circumscribe virtually all attested phonotactic patterns (F_i), and exclude many weird, regular ones.
2. These classes are learnable under established definitions, and provide better measures of pattern complexity than DFA size.
3. Noncounting is closed under intersection so if the interaction ($\times$) is intersection ($\cap$) then at worst P is Noncounting

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

4. Some loose ends:
   - Graf 2009 (PLC) observes a counting stress pattern
   - Culminativity "exactly one" is properly Piecewise Testable.
5. Linguistically motivated open question: Optimization over regular constraints yield nonregular patterns (Frank and Satta 1998, Kartunnen 1998). What about optimization over constraints drawn from these classes?
6. What about regular relations describing phonological processes?

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

4. Some loose ends:
   - Graf 2009 (PLC) observes a counting stress pattern
   - Culminativity "exactly one" is properly Piecewise Testable.
5. Linguistically motivated open question: Optimization over regular constraints yield nonregular patterns (Frank and Satta 1998, Kartunnen 1998). What about optimization over constraints drawn from these classes?
6. What about regular relations describing phonological processes?

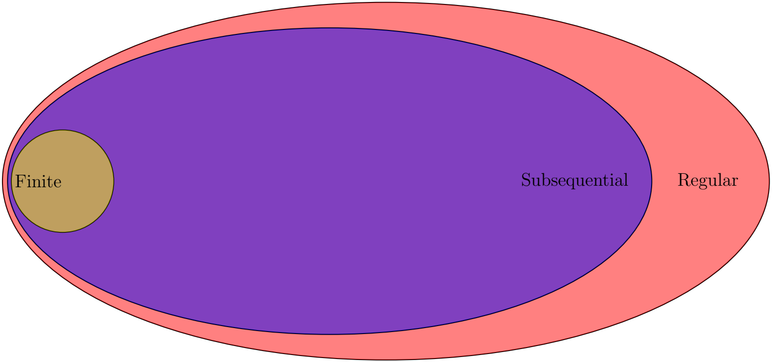
Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

4. Some loose ends:
   - Graf 2009 (PLC) observes a counting stress pattern
   - Culminativity "exactly one" is properly Piecewise Testable.
5. Linguistically motivated open question: Optimization over regular constraints yield nonregular patterns (Frank and Satta 1998, Kartunnen 1998). What about optimization over constraints drawn from these classes?
6. What about regular relations describing phonological processes?

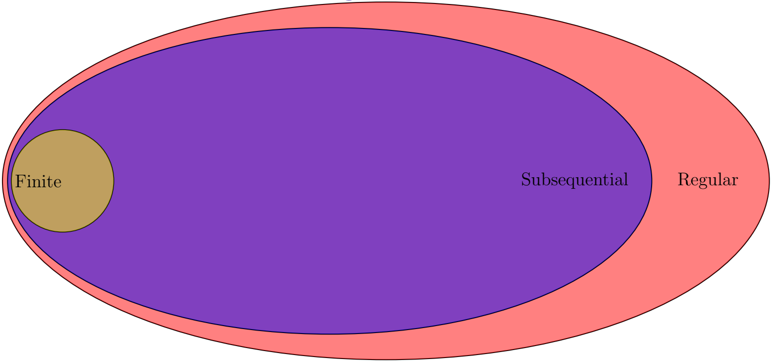
Outline

Subregular relations



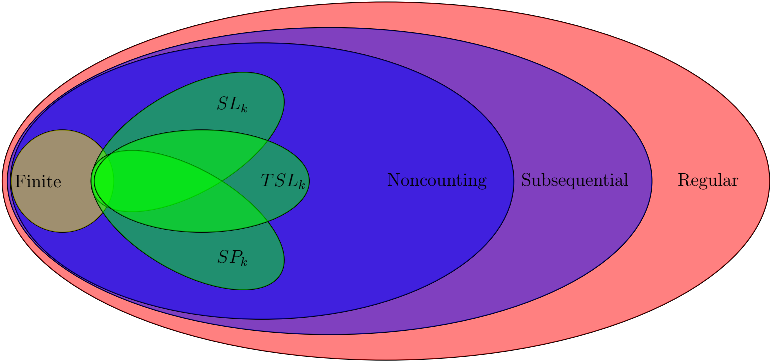
Subsequential functions are those describable by finite-state transducers that are deterministic on the input.

Subregular relations



Subsequential functions are identifiable in the limit from positive data (where they are defined) (Oncina et al. 1993), though the sample necessary for learning phonological patterns does not appear to be present in natural language corpora (Gildea and Jurafsky 1996).

Subregular relations

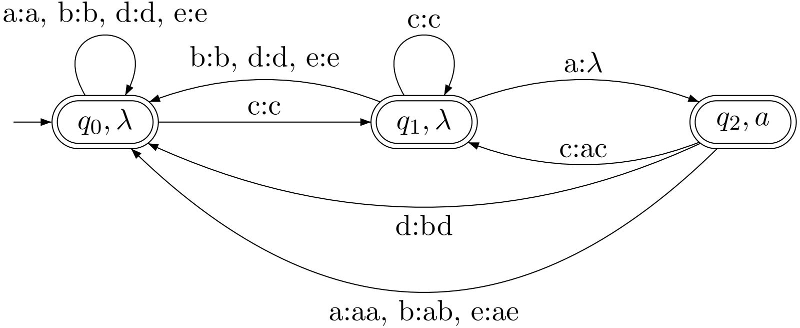


We are defining subsequential counterparts to SL_k, SP_k, TSL_k.

Subsequential Transducers

One definition permits each state to be associated with a string. The action of "ending" in a state appends this string to the output.

Example: $\Sigma = \{a, b, c, d, e\}$ and the rule a $\longrightarrow$ b / c ___ d

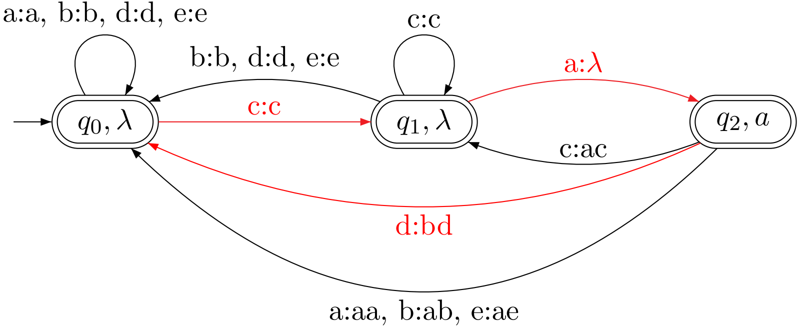


Subsequential Transducers

One definition permits each state to be associated with a string. The action of "ending" in a state appends this string to the output.

Example: $\Sigma = \{a, b, c, d, e\}$ and the rule a $\longrightarrow$ b / c ___ d

/cad/→[cbd]

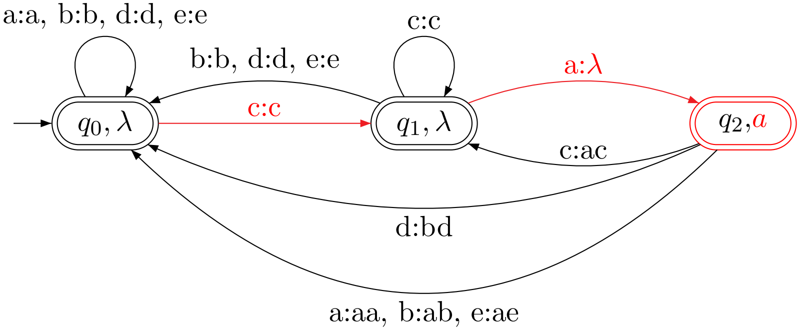


Subsequential Transducers

One definition permits each state to be associated with a string. The action of "ending" in a state appends this string to the output.

Example: $\Sigma = \{a, b, c, d, e\}$ and the rule a $\longrightarrow$ b / c ____ d

/ca/→[ca]



What is and what is not subsequential

Subsequential processes

1. epenthesis, deletion, substitution, local metathesis
2. consonantal harmony
3. long-distance dissimilation
4. Vowel harmony patterns (verified for all in Nevins 2010 (46 languages))

Non-subsequential processes

1. "sour grapes" vowel harmony process
2. unbounded long-distance metathesis (not even regular)

Strictly k-Local Subsequential Relations

A relation $R \subseteq \Sigma_1^* \times \Sigma_2^*$ is *input-based* Strictly k-Local iff

(above the line is same as SL_k sets)

- The states $Q = \Sigma^{<k}$
- The initial states are $I = \{\lambda\}$
- The final states are $F = Q$
- The transition function $\delta(a_1 \cdots a_n, b) =$

  $a_2 \cdots a_n b$ iff $|a_1 \cdots a_n b| \geq k$ and $|a_1 \cdots a_n b|$ otherwise

---

- The output of each transition belongs to Σ_2^*.
- Final appending strings belong to Σ_2^*.

Examples of Strictly k-Local Subsequential Relations

Simultaneous application:

1. Substitution: rules of the form $\mathrm{a} \longrightarrow \mathrm{b}$ / u ___ v where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uav|$.
2. Deletion: Rules of the form $\mathrm{a} \longrightarrow \emptyset$ / u ___ v where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uav|$.
3. Epenthesis: Rules of the form $\emptyset \longrightarrow \mathrm{b}$ / u ___ v where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uv|$.
4. Metathesis: Rules of the form $\mathrm{ab} \longrightarrow \mathrm{ba}$ / u ___ v where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uabv|$.

Examples of Strictly k-Local Subsequential Relations

Simultaneous application:

1. Substitution: rules of the form $\mathrm{a} \longrightarrow \mathrm{b}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uav|$.
2. Deletion: Rules of the form $\mathrm{a} \longrightarrow \emptyset\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uav|$.
3. Epenthesis: Rules of the form $\emptyset \longrightarrow \mathrm{b}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uv|$.
4. Metathesis: Rules of the form $\mathrm{ab} \longrightarrow \mathrm{ba}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uabv|$.

Examples of Strictly k-Local Subsequential Relations

Simultaneous application:

1. Substitution: rules of the form $\mathrm{a} \longrightarrow \mathrm{b}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uav|$.
2. Deletion: Rules of the form $\mathrm{a} \longrightarrow \emptyset\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uav|$.
3. Epenthesis: Rules of the form $\emptyset \longrightarrow \mathrm{b}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uv|$.
4. Metathesis: Rules of the form $\mathrm{ab} \longrightarrow \mathrm{ba}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
   - This alternation is Strictly k-Local with $k = |uabv|$.

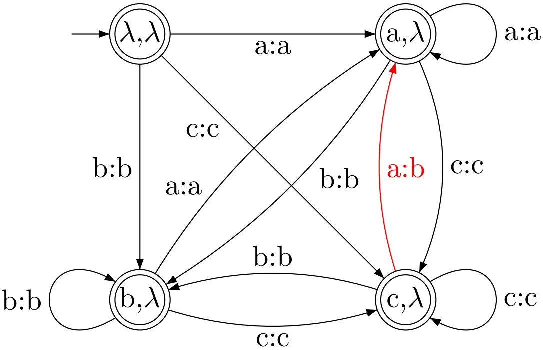
Examples of Strictly k-Local Subsequential Relations

Simultaneous application:

1. Substitution: rules of the form $\mathrm{a} \longrightarrow \mathrm{b}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
    - This alternation is Strictly k-Local with $k = |uav|$.
2. Deletion: Rules of the form $\mathrm{a} \longrightarrow \emptyset\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
    - This alternation is Strictly k-Local with $k = |uav|$.
3. Epenthesis: Rules of the form $\emptyset \longrightarrow \mathrm{b}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
    - This alternation is Strictly k-Local with $k = |uv|$.
4. Metathesis: Rules of the form $\mathrm{ab} \longrightarrow \mathrm{ba}\ /\ \mathrm{u}\ \_\_\_\ \mathrm{v}$ where u and v are strings.
    - This alternation is Strictly k-Local with $k = |uabv|$.

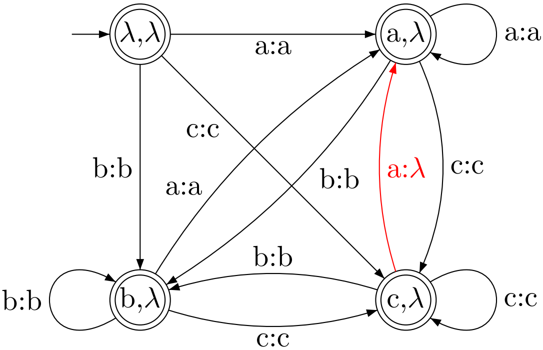
Example of SL_2 subsequential transducers

Substitution: a $\longrightarrow$ b / c ____



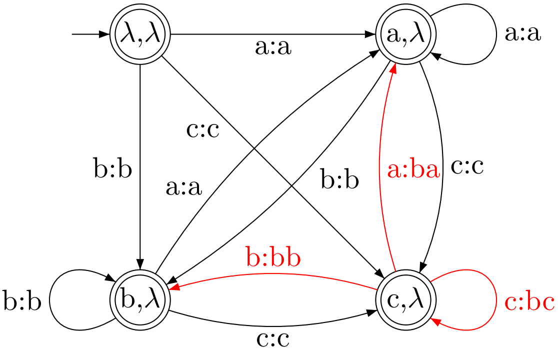
Example of SL_2 subsequential transducers

Deletion: $a \longrightarrow \emptyset$ / c ___



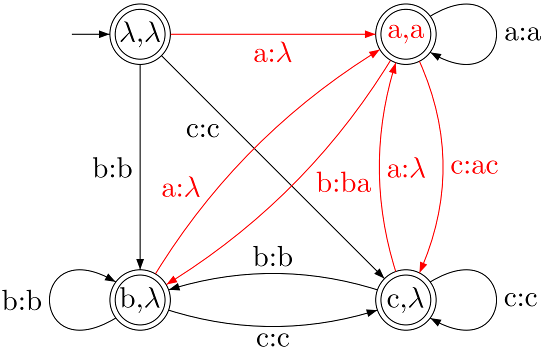
Example of SL_2 subsequential transducers

Epenthesis: $\emptyset \longrightarrow$ b / c ___



Example of SL_2 subsequential transducers

Metathesis: ab $\longrightarrow$ ba / ___



What is not a Strictly k-Local Subsequential Relation for any k

1. Rules which apply left-to-right or right-to-left (self-feeding rules)
2. Long distance processes
   2.1 Consonantal harmony
   2.2 Vowel harmony
   2.3 Long-distance dissimilation
   2.4 *Most* subsequential relations including counting ones, first/last agreement, etc.!

Current research

1. Defining output-based SL_k for left-to-right patterns.
2. Defining reverse, output-based SL_k for right-to-left patterns.
3. Establishing that these input-based SL_k relations are not closed under intersection but are under composition.
4. Designing learners for input-based SL_k that require reasonable sample sizes for success (cf. Gildea and Jurafsky 1996).
5. Decision procedures: Given a (subsequential) transducer, is it SL_k and for what k?

SP_k and TSL_k subsequential relations

These are defined analogously:

1. The states of the SP_k subsequential transducer keeps track of the $(k-1)$-subsequences observed
2. The states of the TSL_k subsequential transducer keeps track of the last $(k-1)$ symbols on tier T observed.

SP_k and TSL_k subsequential relations

Can describe:

- Consonant harmony (SP_2, TSL_2)
- Long-distance dissimilation (TSL_2)
- Vowel harmony without neutral vowels (SP_2, TSL_2)
- Vowel harmony with opaque vowels (TSL_2)
- Vowel harmony with transparent vowels (SP_2)

SP_k and TSL_k subsequential relations

Can describe:

- Consonant harmony (SP_2, TSL_2)
- Long-distance dissimilation (TSL_2)
- Vowel harmony without neutral vowels (SP_2, TSL_2)
- Vowel harmony with opaque vowels (TSL_2)
- Vowel harmony with transparent vowels (SP_2)

SP_k and TSL_k subsequential relations

Can describe:

- Consonant harmony (SP_2, TSL_2)
- Long-distance dissimilation (TSL_2)
- Vowel harmony without neutral vowels (SP_2, TSL_2)
- Vowel harmony with opaque vowels (TSL_2)
- Vowel harmony with transparent vowels (SP_2)

SP_k and TSL_k subsequential relations

Can describe:

- Consonant harmony (SP_2, TSL_2)
- Long-distance dissimilation (TSL_2)
- Vowel harmony without neutral vowels (SP_2, TSL_2)
- Vowel harmony with opaque vowels (TSL_2)
- Vowel harmony with transparent vowels (SP_2)

SP_k and TSL_k subsequential relations

Can describe:

- Consonant harmony (SP_2, TSL_2)
- Long-distance dissimilation (TSL_2)
- Vowel harmony without neutral vowels (SP_2, TSL_2)
- Vowel harmony with opaque vowels (TSL_2)
- Vowel harmony with transparent vowels (SP_2)

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. We can profitably extend the classes in the subregular hierarchy to subclasses of subsequential functions to describe phonological processes
2. Some details remain (e.g. left-to-right, right-to-left application)
3. Again, we can think of these subclasses as constraining the factors (F_i).
4. If, as anticipated, Noncounting subsequential relations are closed under composition, then again P is constrained to the Noncounting class.

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. We can profitably extend the classes in the subregular hierarchy to subclasses of subsequential functions to describe phonological processes
2. Some details remain (e.g. left-to-right, right-to-left application)
3. Again, we can think of these subclasses as constraining the factors (F_i).
4. If, as anticipated, Noncounting subsequential relations are closed under composition, then again P is constrained to the Noncounting class.

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. We can profitably extend the classes in the subregular hierarchy to subclasses of subsequential functions to describe phonological processes
2. Some details remain (e.g. left-to-right, right-to-left application)
3. Again, we can think of these subclasses as constraining the factors (F_i).
4. If, as anticipated, Noncounting subsequential relations are closed under composition, then again P is constrained to the Noncounting class.

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

1. We can profitably extend the classes in the subregular hierarchy to subclasses of subsequential functions to describe phonological processes
2. Some details remain (e.g. left-to-right, right-to-left application)
3. Again, we can think of these subclasses as constraining the factors (F_i).
4. If, as anticipated, Noncounting subsequential relations are closed under composition, then again P is constrained to the Noncounting class.

Interim Summary

$$F_1 \times F_2 \times \cdots \times F_n = P$$

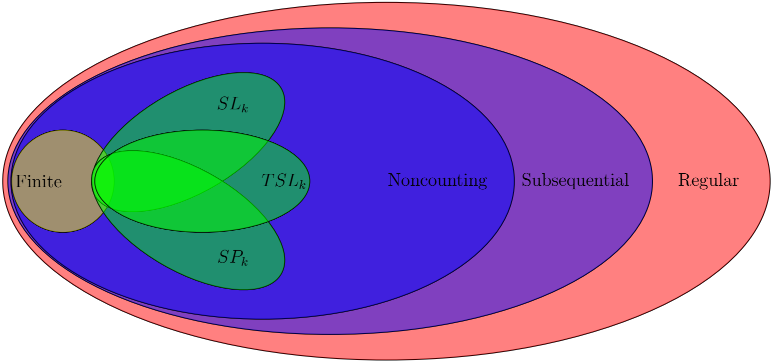
5. These new bounds, when studied thoroughly, promise to lead to new techniques for learning phonological patterns and to new independent, computationally sound measures of pattern complexity.

Future Work

1. Establish decision procedures for these classes
2. Continue to determine properties of these subregular classes
3. In particular: establish the feasible learnability of classes of subregular relations and demonstrate on natural language corpora!
4. Classify phonological patterns in P-base (Mielke 2007) according to these classes
5. Examine the psychological reality of these limits
    - by staying alert to patterns observed cross-linguistically and
    - with artificial language-learning experiments.

Last Words

1. The computational perspective provides a universal theory for defining the dual notions of pattern expressivity and restrictedness.
2. Theoretical linguistics and theoretical computer science mutually inform each other.
3. Phonological processes and constraints are not arbitrary and in fact appear to belong to well-defined *subregular* classes.
4. If correct, this significantly improves on the computational bound established by Kaplan and Kay 1994 and I predict it carries significant implications for learning phonological patterns.



Thanks!