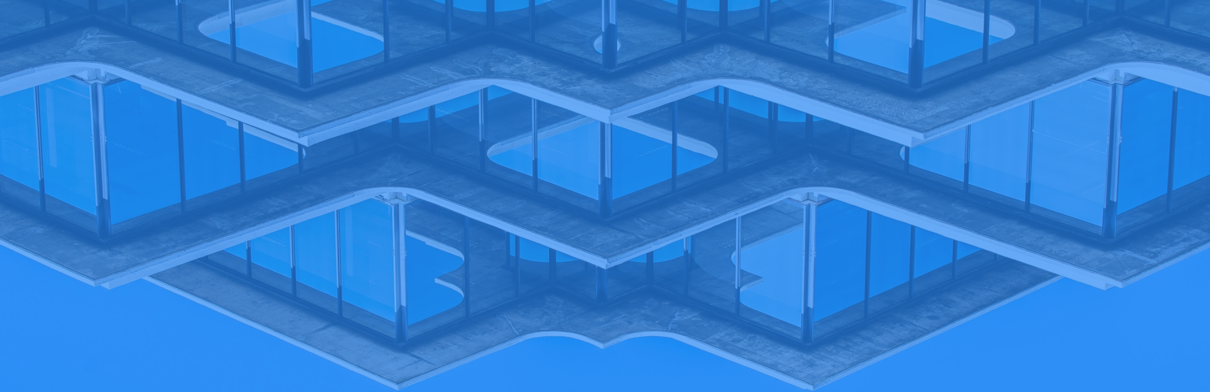


Computational Modeling of Phonological Grammar

Eric Baković | CSS 201S | Aug. 16, 2022 | ebakovic@ucsd.edu

Say what now?

- **Phonological grammar** refers to the mental capacity that is assumed to be developed (as a part of language acquisition) and deployed (as a part of language use) to describe the **sound patterns** of a spoken language.
    - NB. The same applies, *mutatis mutandis*, to **sign patterns** of a signed language.
- **Phonological representations** consist of **speech sounds**, assumed here to be strings of **phonetic symbols**, cross-classified by **phonological features**.
- Grammars are **computational functions**, from representations to **grammaticality judgments** (here assumed categorical/binary, $\{1, 0\}$), or **functional mappings** between different representations.

Today's plan

- **First hour-ish:**
  speech sounds, phontactics, and finite state acceptors
  - + exercises, followed by a short break
- **Second hour-ish:**
  phonological features, alternations, and finite state transducers
  - + exercises, followed by a short break
- **Third hour-ish:**
  computational expressivity and complexity
  - + discussion, *fin*

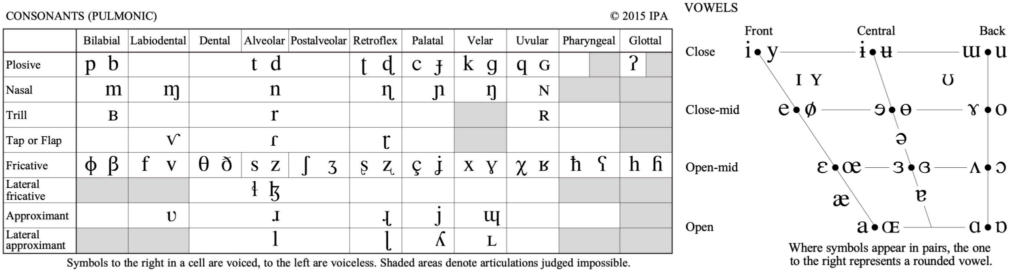
Speech sounds and symbols

Kinda like letters, but kinda not. For example:

letters		**sounds**	*ph*	*o*	*t*	*c*
phonetics	=	[fəˈnɛɾɪks]	[f]	[ə]	[ɾ]	[k]
phonetician	=	[ˌfoʊnəˈtʰɪʃən]	[f]	[oʊ]	[tʰ]	[ʃ]
phoneticist	=	[fəˈnɛɾɪˌsist]	[f]	[ə]	[ɾ]	[s]

There are rough correspondences between orthographic letters and phonetic symbols representing speech sounds, but they are not always consistent and the reasons for the inconsistencies are not always obvious.

The International Phonetic Alphabet



CONSONANTS (PULMONIC) © 2015 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ	▨	ʔ ▨
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ	▨	▨
Trill	ʙ			r				▨	ʀ		▨
Tap or Flap		ⱱ		ɾ		ɽ		▨			▨
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative	▨	▨		ɬ ɮ						▨	▨
Approximant		ʋ		ɹ		ɻ	j	ɰ			▨
Lateral approximant	▨	▨		l		ɭ	ʎ	ʟ		▨	▨

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

VOWELS

	Front	Central	Back
Close	i • y	ɨ • ʉ	ɯ • u
	ɪ ʏ		ʊ
Close-mid	e • ø	ɘ • ɵ	ɤ • o
		ə	
Open-mid	ɛ • œ	ɜ • ɞ	ʌ • ɔ
	æ	ɐ	
Open	a • ɶ		ɑ • ɒ

Where symbols appear in pairs, the one to the right represents a rounded vowel.

Concatenated string representations

Elements of meaning are associated with mentally-stored strings of speech sound symbols, and these strings are concatenated to form words.

The phonological grammar takes these mentally-stored (= ‘underlying’) concatenated strings as input and renders to-be-pronounced (= ‘surface’) strings as output — think of these as the instructions for your vocal tract.

orthographic	**underlying / input**	**surface / output**
phonetic-s	/foʊnɛtɪk+z/	[fəˈnɛɾɪks]
phonetic-ian	/foʊnɛtɪk+jən/	[ˌfoʊnəˈtʰɪʃən]
phonetic-ist	/foʊnɛtɪk+ɪst/	[fəˈnɛɾɪˌsɪst]

Not all strings are created equal

Some strings are grammatical (= allowed by the grammar), others are not.

- Attested strings are self-evidently grammatical:

  e.g. *flake* = [fleɪk]

- Some unattested strings are self-evidently ungrammatical:

  e.g. *fnake* = *[fneɪk]

- Some unattested strings are *possible*, and thus grammatical:

  e.g. *frake* = [fɹeɪk]

Phonotactic conditions

Ungrammaticality can typically be localized to particular ungrammatical (because unattested) *substrings*.

For example, the issue with *fnake* *[fneɪk] is that [fn] is not a possible word-initial consonant cluster in English.

On the other hand, word-initial [fɹ] is amply attested (*frame, fry, frown, …*), which is what makes *frake* [fɹeɪk] possible (albeit unattested, as yet).

Phonotactic conditions

Two general methods for expressing these *phonotactic conditions*.

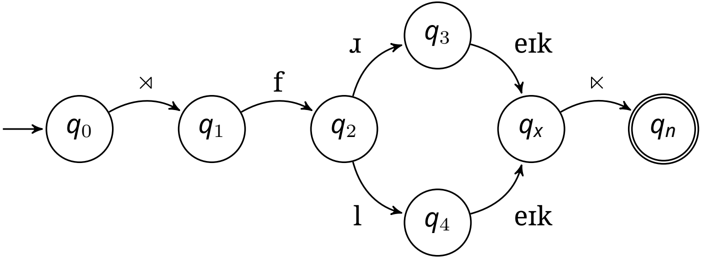
- **Exclusion:** All strings that include an ungrammatical substring are ungrammatical (= 0); all others are grammatical (= 1).

  e.g. *⋊[fn] (⋊ = word-initial boundary)

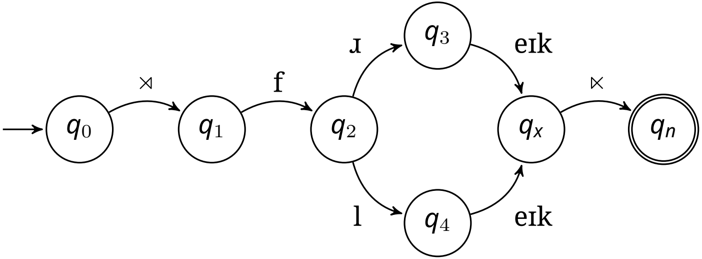
- **Inclusion:** All strings that include only grammatical substrings are grammatical (= 1); all others are ungrammatical (= 0).

Finite-state acceptors (FSAs)

- A finite set of *states*, including a distinguished start state (here designated 'q_0') and at least one final (= accepting) state (here designated 'q_n').
- A finite alphabet Σ of symbols (here, a designated subset of the IPA).
- A state-transition function (read a symbol, move to a next state).



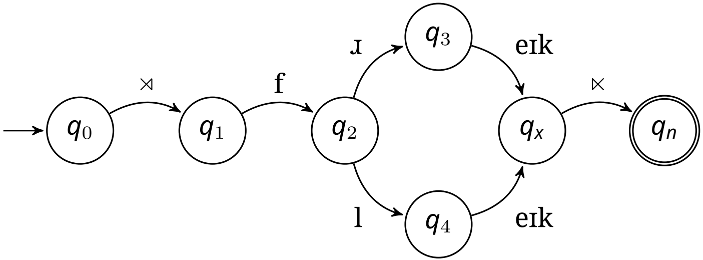
Finite-state acceptors (FSAs)



Every grammatical word should have a path through the FSA to an accepting state.

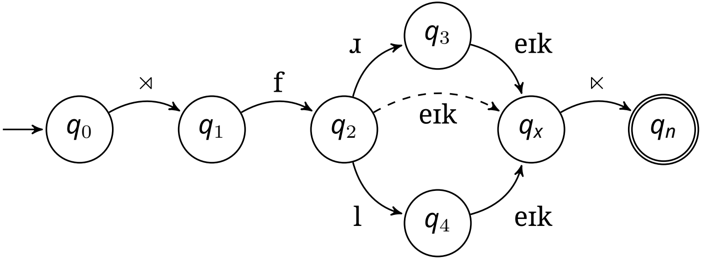
No ungrammatical word should have a path through the FSA to an accepting state.

Exercises



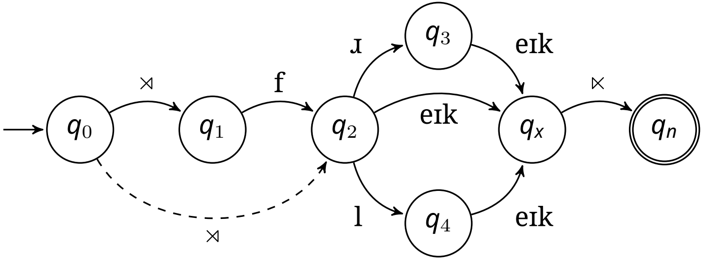
- How can this FSA be amended to allow the word *fake* = [feɪk]?

Exercises



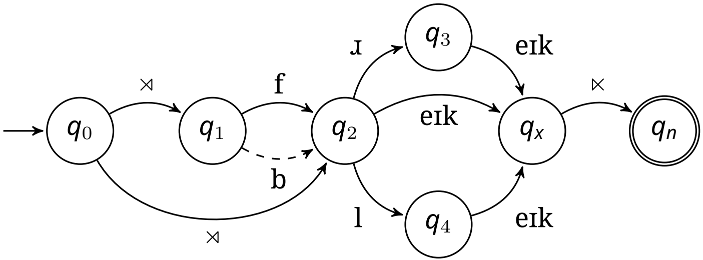
- ~~How can this FSA be amended to allow the word *fake* = [feɪk]?~~
- . . . to allow the words *rake* = [ɹeɪk], *lake* = [leɪk], and *ache* = [eɪk]?

Exercises



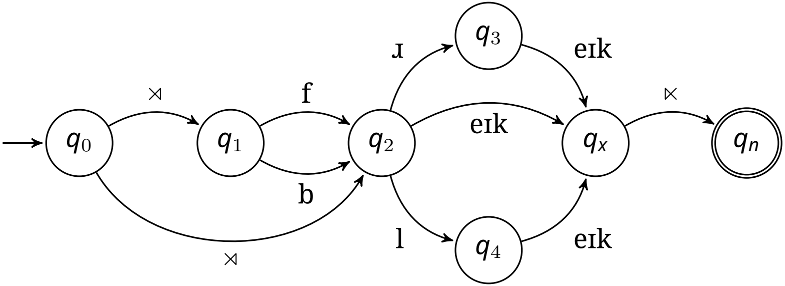
- ~~How can this FSA be amended to allow the word *fake* = [feɪk]?~~
- ~~. . . to allow the words *rake* = [ɹeɪk], *lake* = [leɪk], and *ache* = [eɪk]?~~
- . . . to allow the words *brake* = [bɹeɪk], *Blake* = [bleɪk], and *bake* = [beɪk]?

Exercises



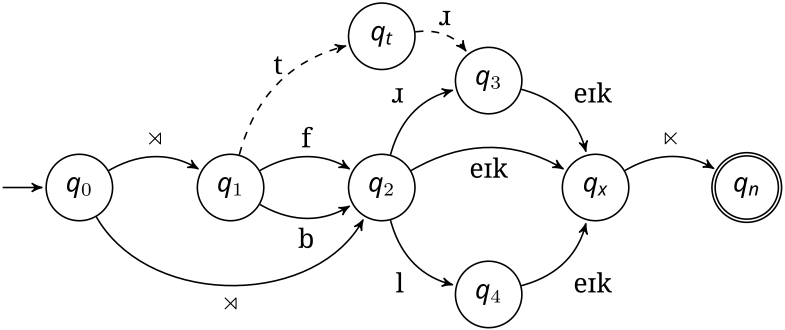
- ~~How can this FSA be amended to allow the word *fake* = [feɪk]?~~
- ~~. . . to allow the words *rake* = [ɹeɪk], *lake* = [leɪk], and *ache* = [eɪk]?~~
- ~~. . . to allow the words *brake* = [bɹeɪk], *Blake* = [bleɪk], and *bake* = [beɪk]?~~

Exercises



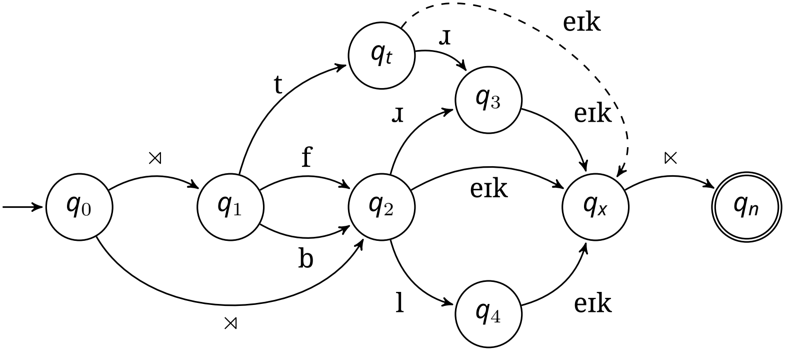
- How can this FSA be amended to account for the fact that word-initial [t] can be followed by [ɹ] (e.g. *trach* = [tɹeɪk]) but not by [l] (*tlake* = *[tleɪk])?

Exercises



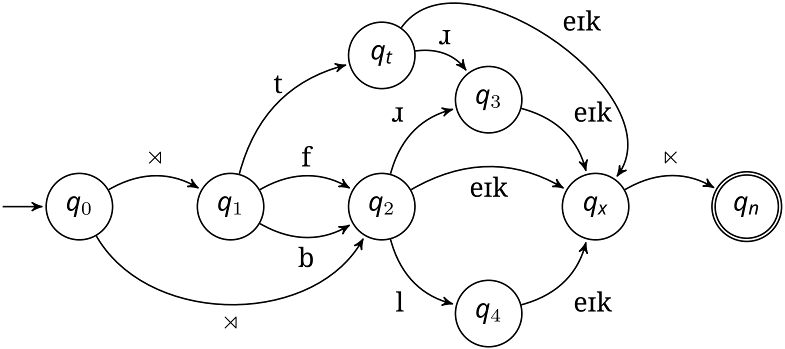
- ~~How can this FSA be amended to account for the fact that word-initial [t] can be followed by [ɹ] (e.g. *trach* = [tɹeɪk]) but not by [l] (*tlake* = *[tleɪk])?~~
- Does this amended FSA account for the grammaticality of *take* = [teɪk]? If not, how can it be further amended to do so?

Exercises



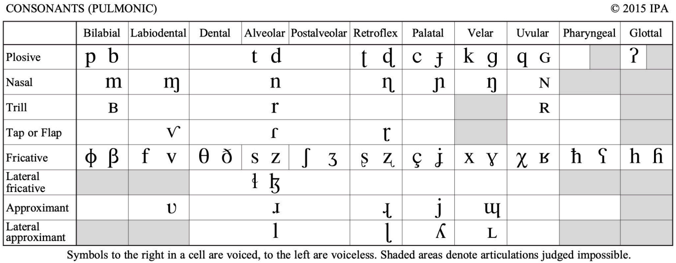
- ~~How can this FSA be amended to account for the fact that word-initial [t] can be followed by [ɹ] (e.g. *trach* = [tɹeɪk]) but not by [l] (*tlake* = *[tleɪk])?~~
- ~~Does this amended FSA account for the grammaticality of *take* = [teɪk]? If not, how can it be further amended to do so?~~

Some further exercises



- *slake* = [sleɪk], *sake* = [seɪk], *snake* = [sneɪk], but *[sɹeɪk]
- *stake* = [steɪk], *stray* = [stɹeɪ], but *[stleɪ]

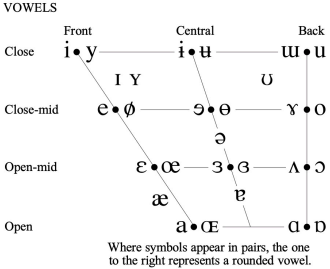
Cross-classification of speech sound symbols



CONSONANTS (PULMONIC) © 2015 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ	▨	ʔ ▨
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ	▨	▨
Trill	ʙ			r				▨	ʀ		▨
Tap or Flap		ⱱ		ɾ		ɽ		▨			▨
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative	▨	▨		ɬ ɮ						▨	▨
Approximant		ʋ		ɹ		ɻ	j	ɰ			▨
Lateral approximant	▨	▨		l		ɭ	ʎ	ʟ		▨	▨

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.



Where symbols appear in pairs, the one to the right represents a rounded vowel.

Focus on English consonants

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Palatal	Velar	Glottal
Plosive	p b			t d			k g	
Nasal	m			n			ŋ	
Tap				ɾ				
Fricative		f v	θ ð	s z	ʃ ʒ			h
Affricate					ʧ ʤ			
Approximant	(w)			ɹ		j	(w)	
Lateral				l				

Symbols to the left in a cell are voiceless, all other symbols are voiced.

. . . with phonological features

	+labial	+coronal +anterior +distributed	+coronal +anterior −distributed	+coronal −anterior +distributed	+dorsal −back	+dorsal +back	+spread glottis
−continuant −sonorant	p b		t d	ʧ ʤ		k g	
+continuant −sonorant	f v	θ ð	s z	ʃ ʒ			h
−continuant +sonorant	m		n			ŋ	
+continuant +sonorant	(w)		ɹ		j	(w)	
+liquid −lateral			ɾ				
+liquid +lateral			l				

Symbols to the left in a cell are [−voice], all other symbols are [+voice].

Natural class behavior

Any single consonant (except [ŋ, ɾ]) can begin a word.[1]

peep = [piːp]	*beep* = [biːp]	*team* = [tiːm]	*deem* = [diːm]	
cheap = [ʧiːp]	*jeep* = [ʤiːp]	*cool* = [kuːl]	*ghoul* = [guːl]	
feel = [fiːl]	*veal* = [viːl]	*thin* = [θɪn]	*this* = [ðɪs]	
scene = [siːn]	*zine* = [ziːn]	*shoe* = [ʃuː]	*jus* = [ʒuː]	
meet = [miːt]	*neat* = [niːt]	*wield* = [wiːld]	*yield* = [jiːld]	*heat* = [hiːt]

[1] Initial [ʒ] is limited to words borrowed from French.

Natural class behavior

Only a subset can cluster with [ɹ].

prow = [pɹaʊ]	*brow* = [bɹaʊ]	*train* = [tɹeɪn]	*drain* = [dɹeɪn]	
? [ʧɹ. . .]	? [ʤɹ. . .]	*cruel* = [kɹuːl]	*gruel* = [gɹuːl]	
free = [fɹi]	* [vɹ. . .]	*three* = [θɹiː]	* [ðɹ. . .]	
* [sɹ. . .]	* [zɹ. . .]	*shred* = [ʃɹɛd]	* [ʒɹ. . .]	
* [mɹ. . .]	* [nɹ. . .]	* [wɹ. . .]	* [jɹ. . .]	* [hɹ. . .]

- **Pattern:** [−sonorant]; if [+continuant], then [−voice] (except [s]).

Natural class behavior

Only a subset can cluster with [l].

plume	= [pluːm]	*bloom*	= [bluːm]		* [tl. . .]		* [dl. . .]	
	* [ʧl. . .]		* [ʤl. . .]	*clue*	= [kluː]	*glue*	= [ɡluː]	
flu	= [fluː]		* [vl. . .]		* [θl. . .]		* [ðl. . .]	
slow	= [sloʊ]		* [zl. . .]	*schlep*	= [ʃlɛp]		* [ʒl. . .]	
	* [ml. . .]		* [nl. . .]		* [wl. . .]		* [jl. . .]	* [hl. . .]

- **Pattern:** [−sonorant]; if [+continuant], then [−voice, −coronal] (except [s]).

Surface speech sounds vary by context

English plural nouns (and third person singular verbs)

schwa-+voice	**−voice**	**+voice**
kisses = [kɪsəz]	*dips* = [dɪps]	*dibs* = [dɪbz]
fizzes = [fɪzəz]	*bits* = [bɪts]	*bids* = [bɪdz]
wishes = [wɪʃəz]	*wicks* = [wɪks]	*wigs* = [wɪgz]
zhuzhes = [ʒʊʒəz]	*cliffs* = [klɪfs]	*gives* = [gɪvz]
matches = [mæʧəz]	*myths* = [mɪθs]	*bathes* = [beɪðz]
badges = [bæʤəz]		*times* = [taɪmz]
		tines = [taɪnz]
		tires = [taɪɹz]
		tiles = [taɪlz]
		ties = [taɪz]

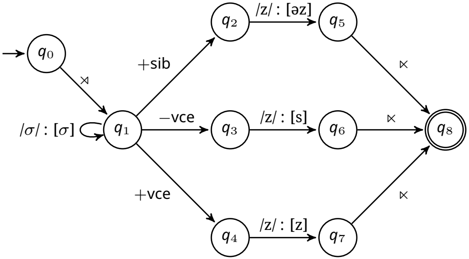
Surface speech sounds vary by context

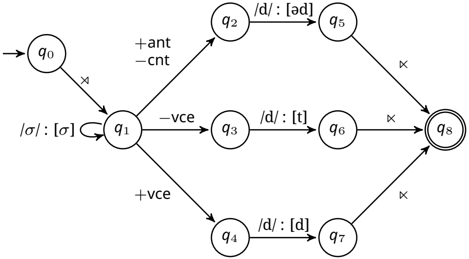
English past tense verbs (and past participles)

schwa-+voice	−voice	+voice	
fitted = [fɪtəd]	*dipped* = [dɪpt]	*fibbed* = [fɪbd]	*signed* = [saɪnd]
glided = [glaɪdəd]	*picked* = [pɪkt]	*rigged* = [ɹɪgd]	*tired* = [taɪɹd]
	riffed = [ɹɪft]	*lived* = [lɪvd]	*tiled* = [taɪld]
	mythed = [mɪθt]	*bathed* = [beɪðd]	*tied* = [taɪd]
	kissed = [kɪst]	*fizzed* = [fɪzd]	
	wished = [wɪʃt]	*zhuzhed* = [ʒʊʒd]	
	itched = [ɪʧt]	*edged* = [ɛʤd]	

Finite-state transducers (FSTs)

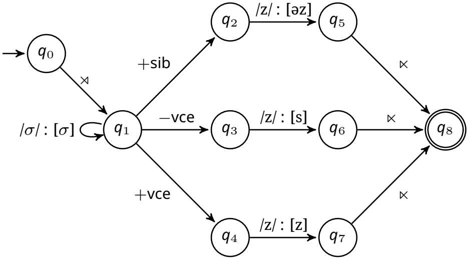
- A finite set of *states*, including a distinguished start state (here designated ‘q_0’) and at least one final state (here designated ‘q_8’).
- A finite alphabet Σ of symbols (here, a designated subset of the IPA).[2]
- A transition function (read a symbol, write a symbol, move to a next state).

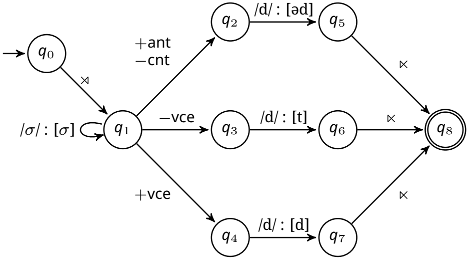




[2] Technically there are separate input and output alphabets.

Challenges





- The two FSTs are obviously related. Could they be merged? How?
- The choice between the [−voice] ($q_3 \rightarrow q_6$) and [+voice] ($q_4 \rightarrow q_7$) forms of both suffixes is clearly a case of *assimilation*. Is that clear from the FSTs?
- The choice between the schwa-+voice ($q_2 \rightarrow q_5$) and the other two forms of both suffixes is clearly a case of *similarity avoidance*. Is that clear from the FSTs?
- The [−voice] and [+voice] sets are complementary, but the [+sib] and [+ant, −cont] sets intersect with both. How can we account for e.g. /wɪʃ+z/ ⟶ *[wɪʃs] rather than [wɪʃəz] = *wishes*?

Exercises

English adjective negation

before **vowels**	*inaudible*	= [ɪnˈɔːɾəbəl]
before **bilabials**	*impossible*	= [ɪmˈpɑːsəbəl]
before **labiodentals**	*infeasible*	= [ɪɱˈfiːzəbəl]
before **alveolars**	*indelible*	= [ɪnˈdɛləbəl]
before **velars**	*inconceivable*	= [ɪŋkənˈsiːvəbəl]

- Construct (the essential elements of) an FST describing this pattern.
- Does one type of FST account for the assimilation better than another?
- How is the ‘default’ nature of the [ɪn] form of the prefix accounted for?

Exercises

Spanish nasal place assimilation

before **vowels**	*con agua*	= [koˈnaɣwa]	‘with water’
before **bilabials**	*con pollo*	= [komˈpoʎo]	‘with chicken’
before **labiodentals**	*con frío*	= [koɱˈfɾio]	‘feeling cold’
before **dentals**	*con tiempo*	= [kon̪ˈt̪jempo]	‘with time (leftover)’
before **alveolars**	*con salsa*	= [konˈsalsa]	‘with salsa’
before **postalveolars**	*con chile*	= [kon̻ˈʧile]	‘with chile’
before **palatals**	*con yema*	= [koɲˈjema]	‘with egg yolk’
before **velars**	*con calma*	= [koŋˈkalma]	‘calmly’

Exercises

Spanish nasal place assimilation

before **vowels**	*con agua*	= [koˈnaɣwa]	‘with water’
before **bilabials**	*con pollo*	= [komˈpoʎo]	‘with chicken’
before **labiodentals**	*con frío*	= [koɱˈfɾio]	‘feeling cold’
before **dentals**	*con tiempo*	= [kon̪ˈt̪jempo]	‘with time (leftover)’
before **alveolars**	*con salsa*	= [konˈsalsa]	‘with salsa’
before **postalveolars**	*con chile*	= [kon̻ˈʧile]	‘with chile’
before **palatals**	*con yema*	= [koɲˈjema]	‘with egg yolk’
before **velars**	*con calma*	= [koŋˈkalma]	‘calmly’

- Does the FST for this pattern differ from the one for the English pattern?

Computational expressivity and complexity

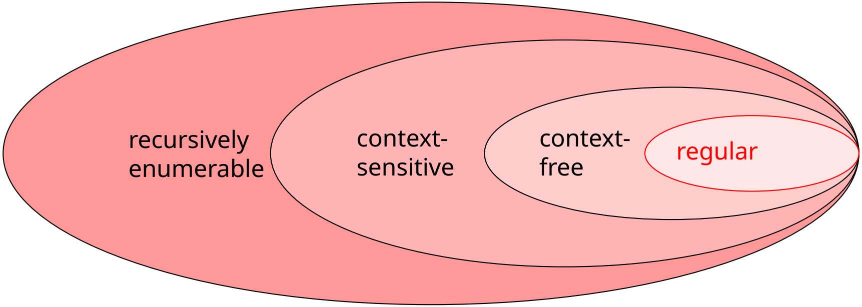
- Typical *expressivity* questions asked of a theory of phonological grammar:
  - Is it powerful enough to correctly describe patterns presumed possible?
    - If not, how can its expressive power be appropriately extended?
  - Is it so powerful that it can incorrectly describe patterns presumed impossible?
    - If so, how can the theory's expressive power be appropriately constrained?
- Recent **Formal Language Theory (FLT)** approach to these questions:
  - Characterize possible and impossible patterns in terms of well-defined, theory-independent mappings between computational classes of string sets.
  - Formally delimit the boundaries between possible and impossible patterns, and between different classes of possible patterns.

How (computationally) complex is phonology?

FLT provides tools to describe the minimum level of **complexity** necessary to **express** phonological patterns (= input-output maps) in precise terms.

- Ordered rule-based phonological grammars are **regular** functions

(Johnson 1972, Kaplan & Kay 1994)



The regular and the subregular

Any regular function can be described by the composition of a left-subsequential function and a right-subsequential function.

(Elgot & Mezei 1965)

one function takes the output of the other function as input

Most phonological patterns require (far) less expressivity than this.

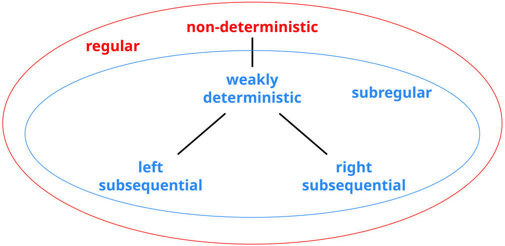
e.g., only bounded string access on either or both sides of any index in the string

e.g., only compositions of subsequential functions with unbounded access to the same side of any index in the string

e.g., only compositions of contradirectional subsequential functions with other restrictions imposed on them

A subregular function is sufficiently complex to express such patterns.

(Heinz 2011, 2018; Heinz & Lai 2013, Jardine 2016, *a.m.o.*)



The regular and the subregular

Any regular function can be described by the composition of a left-subsequential function and a right-subsequential function.

(Elgot & Mezei 1965)

a function with *unbounded string access* to the *left* of any index in the string (and bounded access to the right)

Most phonological patterns require (far) less expressivity than this.

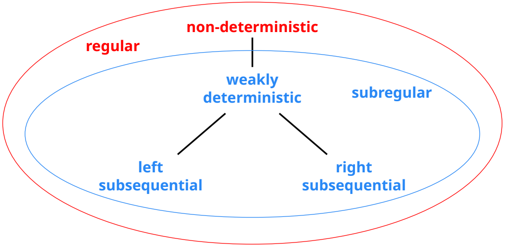
e.g., only bounded string access on either or both sides of any index in the string

e.g., only compositions of subsequential functions with unbounded access to the same side of any index in the string

e.g., only compositions of contradirectional subsequential functions with other restrictions imposed on them

A subregular function is sufficiently complex to express such patterns.

(Heinz 2011, 2018; Heinz & Lai 2013, Jardine 2016, *a.m.o.*)



The regular and the subregular

Any regular function can be described by the composition of a left-subsequential function and a right-subsequential function.

(Elgot & Mezei 1965)

a function with *unbounded string access* to the *right* of any index in the string (and bounded access to the left)

Most phonological patterns require (far) less expressivity than this.

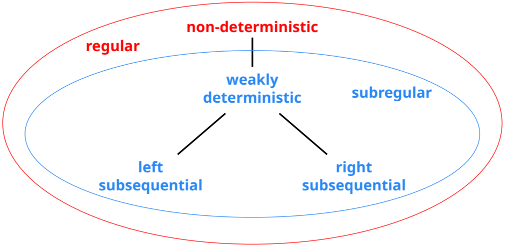
e.g., only bounded string access on either or both sides of any index in the string

e.g., only compositions of subsequential functions with unbounded access to the same side of any index in the string

e.g., only compositions of contradirectional subsequential functions with other restrictions imposed on them

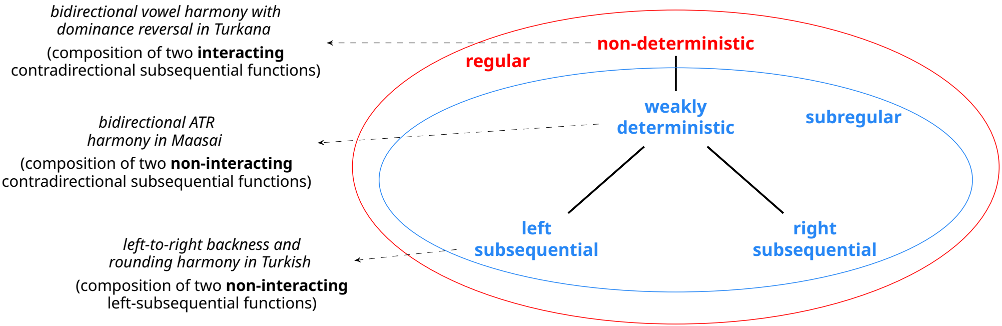
A subregular function is sufficiently complex to express such patterns.

(Heinz 2011, 2018; Heinz & Lai 2013, Jardine 2016, *a.m.o.*)



Complexity and interaction

Non-deterministic (= fully regular) functions require **interaction** between the composed contradirectional subsequential functions that define them; subregular functions do not.



bidirectional vowel harmony with dominance reversal in Turkana

(composition of two **interacting** contradirectional subsequential functions)

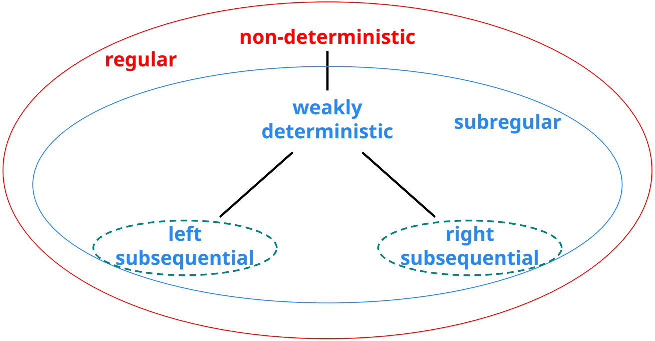
bidirectional ATR harmony in Maasai

(composition of two **non-interacting** contradirectional subsequential functions)

left-to-right backness and rounding harmony in Turkish

(composition of two **non-interacting** left-subsequential functions)

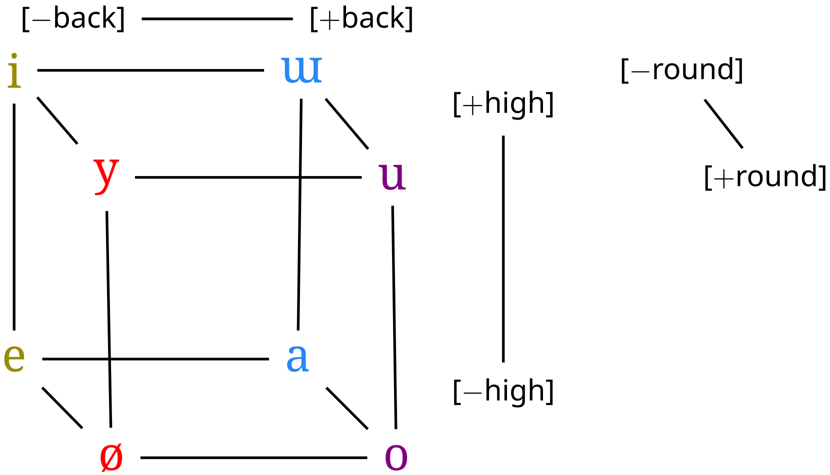
Subsequential functions



The identity of the output element for every input element is (in principle unboundedly) determined by elements from *one side* of the string.

This class contains compositions of functions that share a directionality, compositions of only left-subsequential functions or of only right-subsequential functions.

Turkish vowels



Turkish left-to-right harmony

$\begin{bmatrix}-\text{back}\\-\text{round}\end{bmatrix}$ **root**

a.	ip+in+i	‘rope+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+LOC’

$\begin{bmatrix}+\text{back}\\-\text{round}\end{bmatrix}$ **root**

f.	kɯz+ɯn+ɯ	‘girl+2S.POSS+ACC’
g.	sap+ɯn+ɯ	‘stalk+2S.POSS+ACC’
h.	kɯz+lar+ɯ	‘girl+PL+ACC’
i.	sap+lar+ɯ	‘stalk+PL+ACC’
j.	kɯz+ɯn+da	‘girl+2S.POSS+LOC’

$\begin{bmatrix}-\text{back}\\+\text{round}\end{bmatrix}$ **root**

k.	jyz+yn+y	‘face+2S.POSS+ACC’
l.	køj+yn+y	‘village+2S.POSS+ACC’
m.	jyz+ler+i	‘face+PL+ACC’
n.	køj+ler+i	‘village+PL+ACC’
o.	jyz+yn+de	‘face+2S.POSS+LOC’

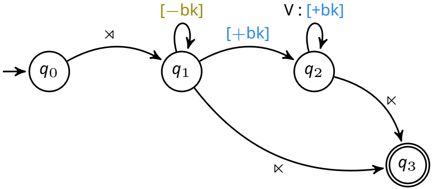
$\begin{bmatrix}+\text{back}\\+\text{round}\end{bmatrix}$ **root**

p.	pul+un+u	‘stamp+2S.POSS+ACC’
q.	son+un+u	‘end+2S.POSS+ACC’
r.	pul+lar+ɯ	‘stamp+PL+ACC’
s.	son+lar+ɯ	‘end+PL+ACC’
t.	pul+un+da	‘stamp+2S.POSS+LOC’

Focus on backness harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+LOC’

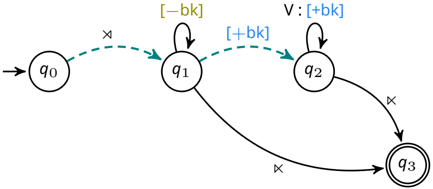
f.	kɯz+ɯn+ɯ	‘girl+2S.POSS+ACC’
g.	sap+ɯn+ɯ	‘stalk+2S.POSS+ACC’
h.	kɯz+lar+ɯ	‘girl+PL+ACC’
i.	sap+lar+ɯ	‘stalk+PL+ACC’
j.	kɯz+ɯn+da	‘girl+2S.POSS+LOC’



input ⋊ k ɯ z i n d e ⋉

Focus on backness harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+LOC’
f.	kɯz+ɯn+ɯ	‘girl+2S.POSS+ACC’
g.	sap+ɯn+ɯ	‘stalk+2S.POSS+ACC’
h.	kɯz+lar+ɯ	‘girl+PL+ACC’
i.	sap+lar+ɯ	‘stalk+PL+ACC’
j.	kɯz+ɯn+da	‘girl+2S.POSS+LOC’

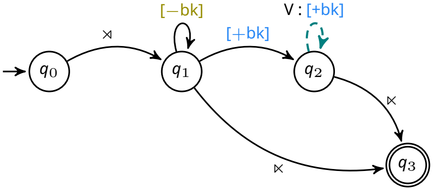


input ⋊ k ɯ z i n d e ⋉

⋊ k ɯ z

Focus on backness harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’	f.	kɯz+ɯn+ɯ	‘girl+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’	g.	sap+ɯn+ɯ	‘stalk+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’	h.	kɯz+lar+ɯ	‘girl+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’	i.	sap+lar+ɯ	‘stalk+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+LOC’	j.	kɯz+ɯn+da	‘girl+2S.POSS+LOC’



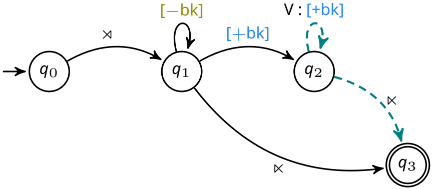
input ⋊ k ɯ z i n d e ⋉

⋊ k ɯ z

⋊ k ɯ z ɯ n

Focus on backness harmony

a.	ip+in+i	'rope+2s.poss+acc'	f.	kɯz+ɯn+ɯ	'girl+2s.poss+acc'
b.	el+in+i	'hand+2s.poss+acc'	g.	sap+ɯn+ɯ	'stalk+2s.poss+acc'
c.	ip+ler+i	'rope+pl+acc'	h.	kɯz+lar+ɯ	'girl+pl+acc'
d.	el+ler+i	'hand+pl+acc'	i.	sap+lar+ɯ	'stalk+pl+acc'
e.	ip+in+de	'rope+2s.poss+loc'	j.	kɯz+ɯn+da	'girl+2s.poss+loc'

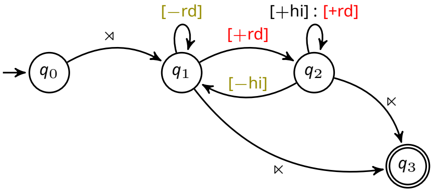


input	⋊	k	ɯ	z	i	n	d	e	⋉
	⋊	k	ɯ	z					
	⋊	k	ɯ	z	ɯ	n			
output	⋊	k	ɯ	z	ɯ	n	d	a	⋉

Focus on rounding harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+ACC’

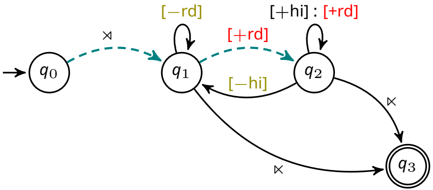
k.	jyz+yn+y	‘face+2S.POSS+ACC’
l.	køj+yn+y	‘village+2S.POSS+ACC’
m.	jyz+ler+i	‘face+PL+ACC’
n.	køj+ler+i	‘village+PL+ACC’
o.	jyz+yn+de	‘face+2S.POSS+LOC’



initial input ⋊ j y z i n d e ⋉

Focus on rounding harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’	k.	jyz+yn+y	‘face+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’	l.	køj+yn+y	‘village+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’	m.	jyz+ler+i	‘face+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’	n.	køj+ler+i	‘village+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+ACC’	o.	jyz+yn+de	‘face+2S.POSS+LOC’



initial input

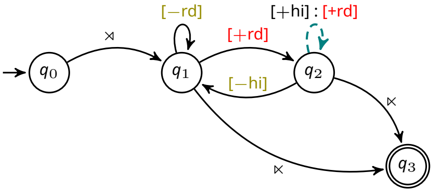
⋊ j y z i n d e ⋉

⋊ j y z

Focus on rounding harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+ACC’

k.	jyz+yn+y	‘face+2S.POSS+ACC’
l.	køj+yn+y	‘village+2S.POSS+ACC’
m.	jyz+ler+i	‘face+PL+ACC’
n.	køj+ler+i	‘village+PL+ACC’
o.	jyz+yn+de	‘face+2S.POSS+LOC’



initial input ⋊ j y z i n d e ⋉

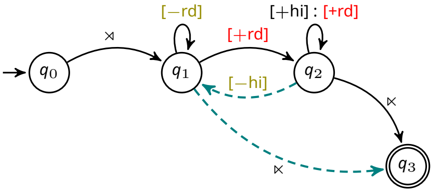
⋊ j y z

⋊ j y z y n

Focus on rounding harmony

a.	ip+in+i	‘rope+2S.POSS+ACC’
b.	el+in+i	‘hand+2S.POSS+ACC’
c.	ip+ler+i	‘rope+PL+ACC’
d.	el+ler+i	‘hand+PL+ACC’
e.	ip+in+de	‘rope+2S.POSS+ACC’

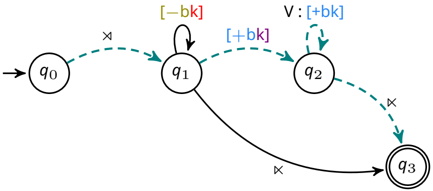
k.	jyz+yn+y	‘face+2S.POSS+ACC’
l.	køj+yn+y	‘village+2S.POSS+ACC’
m.	jyz+ler+i	‘face+PL+ACC’
n.	køj+ler+i	‘village+PL+ACC’
o.	jyz+yn+de	‘face+2S.POSS+LOC’



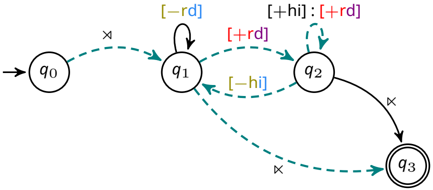
initial input	⋊	j	y	z	i	n	d	e	⋉
	⋊	j	y	z					
	⋊	j	y	z	y	n			
final output	⋊	j	y	z	y	n	d	e	⋉

Backness first, rounding second

$R(B(x)), R \circ B(x)$



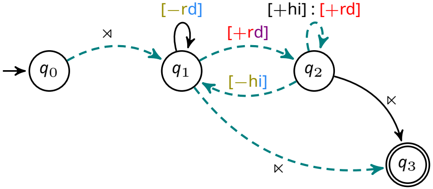
initial input	⋊	p	u	l	i	n	d	e	⋉
	⋊	p	u	l					
	⋊	p	u	l	ɯ	n			
intermediate	⋊	p	u	l	ɯ	n	d	a	⋉



intermediate	⋊	p	u	l	ɯ	n	d	a	⋉
	⋊	p	u	l					
	⋊	p	u	l	u	n			
final output	⋊	p	u	l	u	n	d	a	⋉

Rounding first, backness second

$B(R(x))$, $B \circ R(x)$

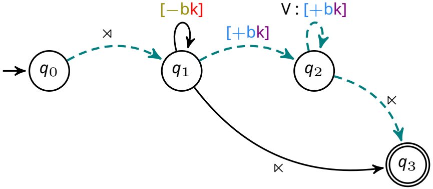


initial input ⋊ p u l i n d e ⋉

⋊ p u l

⋊ p u l y n

intermediate ⋊ p u l y n d e ⋉



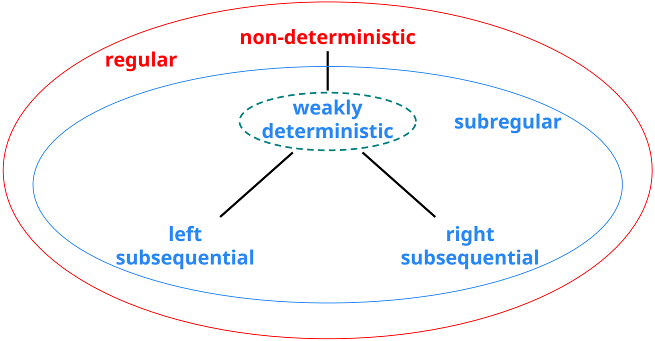
intermediate ⋊ p u l y n d e ⋉

⋊ p u l

⋊ p u l u n

final output ⋊ p u l u n d a ⋉

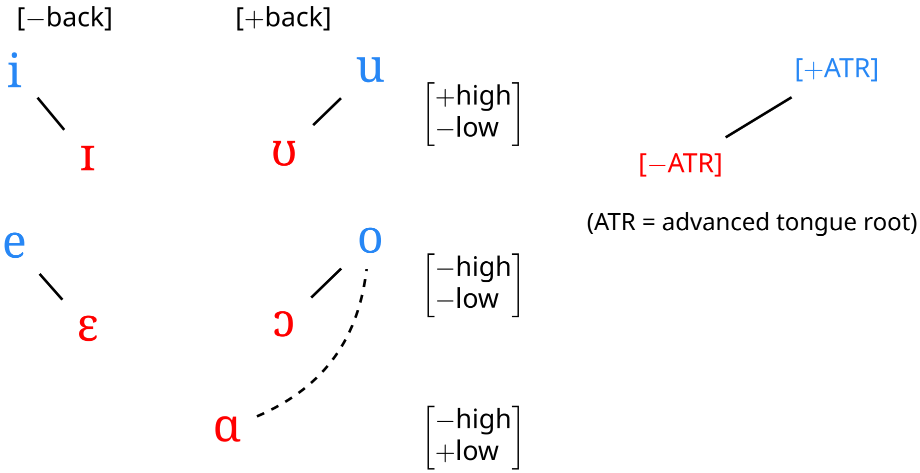
Weakly deterministic functions



The identity of the output element for every input element is (in principle unboundedly) determined by elements from *one side or the other* of the string.

Composition of two contradirectional subsequential functions *that don't interact*.

Maasai vowels



[−back] [+back]

i u

ɪ ʊ

$\begin{bmatrix}+\text{high}\\-\text{low}\end{bmatrix}$

e o

ɛ ɔ

$\begin{bmatrix}-\text{high}\\-\text{low}\end{bmatrix}$

ɑ

$\begin{bmatrix}-\text{high}\\+\text{low}\end{bmatrix}$

[+ATR]

[−ATR]

(ATR = advanced tongue root)

Maasai bidirectional harmony

a.	/mɪ+kɪ+rɑɲ/	⟶	[mɪ+kɪ+rɑɲ]	‘NEG+1PL+sing’	root-controlled
b.	/mɪ+kɪ+itoki/	⟶	[mi+ki+intoki]	‘NEG+1PL+do.again’	
c.	/kɪ+ɪdɪm+ʊ/	⟶	[kɪ+ɪdɪm+ʊ]	‘1PL+be.able+PRES’	
d.	/kɪ+ɲorr+ʊ/	⟶	[ki+ɲorr+u]	‘1PL+love+PRES’	
e.	/ɪsʊj+ɪʃɔ/	⟶	[ɪsʊj+ɪʃɔ]	‘wash+INTR’	suffix-controlled
f.	/ɪsʊj+ɪʃɔ+re/	⟶	[isuj+iʃo+re]	‘wash+INTR+APPL’	
g.	/ɛ+tɛ+bɛl+ɑ/	⟶	[ɛ+tɛ+bɛl+ɑ]	‘3SG+PF+break+PF’	
h.	/ɛ+tɛ+bɛl+ie/	⟶	[e+te+bel+ie]	‘3SG+PF+break+INST’	

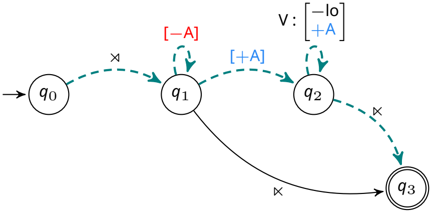
Maasai bidirectional harmony

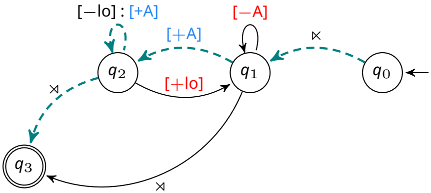
i.	/kɪ+dot+ʊn+ie/	⟶	[ki+dot+uɲ+ie]	‘1PL+pull+MT+APPL’	/ɑ/ blocks [+ATR] spreading from the right
j.	/kɪ+tɑ+dot+ʊn+ie/	⟶	[kɪ+tɑ+dot+uɲ+ie]	‘1PL+PST+pull+MT+APPL’	
k.	/ɪ+duŋ+ɪʃɔ+re/	⟶	[i+duŋ+iʃo+re]	‘2SG+cut+INTR+APPL’	
l.	/ɪ+ɑs+ɪʃɔ+re/	⟶	[ɪ+ɑs+iʃo+re]	‘2SG+do+INTR+APPL’	
m.	/ɪn+lɪpɔŋ+ɑ/	⟶	[ɪn+lɪpɔŋ+ɑ]	‘F.PL+female+PROD’	/ɑ/ raises to [o] when [+ATR] spreads from the left
n.	/ɔl+mɛn+ɑ/	⟶	[ɔl+mɛn+ɑ]	‘M.SG+despise+PROD’	
o.	/ɪn+mudoŋ+ɑ/	⟶	[in+mudoŋ+o]	‘F.PL+relative+PROD’	
p.	/ɛŋ+komon+ɑ/	⟶	[eŋ+komon+o]	‘F.SG+pray+PROD’	
q.	/ɑ+duŋ+ɑkɪn+ie/	⟶	[ɑ+duŋ+okiɲ+ie]	‘3SG+cut+DAT+APPL’	/ɑ/ raises to [o] when flanked by [+ATR] vowels
r.	/ɛ+isud+ɑ+rɪ+ie/	⟶	[e+isud+o+ri+jie]	‘3SG+hide+MA+N+APPL’	

Left-to-right first, right-to-left second

$R(L(x));\ R \circ L(x)$

/ɛ+isud+ɑ+rɪ+ie/ ⟶ [e+isud+o+ri+jie] ‘3SG+hide+MA+N+APPL’





→

in	⋊	ɛ	i	s	u	d	ɑ	r	ɪ	i	e	⋉
	⋊	ɛ										
	⋊	ɛ	i	s	u	d						
	⋊	ɛ	i	s	u	d	o					
	⋊	ɛ	i	s	u	d	o	r	i			
int	⋊	ɛ	i	s	u	d	o	r	i	i	e	⋉

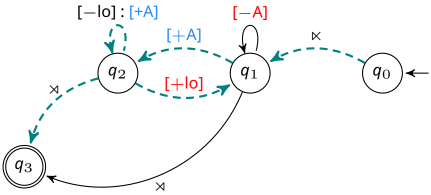
←

int	⋊	ɛ	i	s	u	d	o	r	i	i	e	⋉
										i	e	⋉
								r	i	i	e	⋉
							o	r	i	i	e	⋉
			i	s	u	d	o	r	i	i	e	⋉
out	⋊	e	i	s	u	d	o	r	i	i	e	⋉

Right-to-left first, left-to-right second

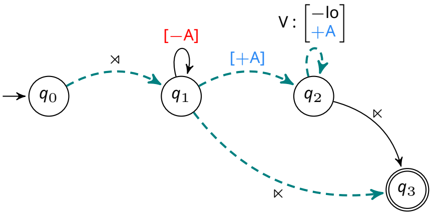
$L(R(x))$; $L \circ R(x)$

/ɛ+isud+ɑ+rɪ+ie/ ⟶ [e+isud+o+ri+jie] ‘3SG+hide+MA+N+APPL’



←

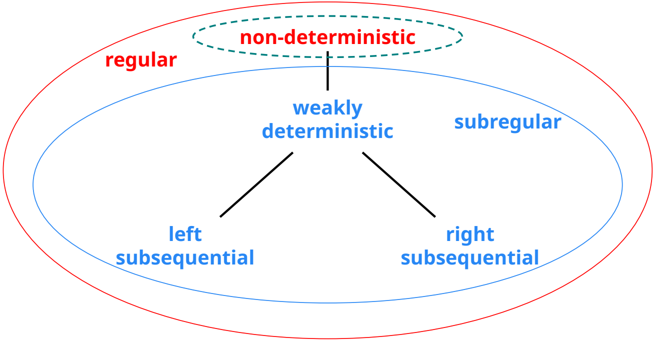
in	⋊	ɛ	i	s	u	d	ɑ	r	ɪ	i	e	⋉
										i	e	⋉
								r	i	i	e	⋉
							ɑ	r	i	i	e	⋉
			i	s	u	d	ɑ	r	i	i	e	⋉
int	⋊	e	i	s	u	d	ɑ	r	i	i	e	⋉



→

int	⋊	e	i	s	u	d	ɑ	r	i	i	e	⋉
	⋊	e										
	⋊	e	i	s	u	d						
	⋊	e	i	s	u	d	o					
	⋊	e	i	s	u	d	o	r	i			
out	⋊	e	i	s	u	d	o	r	i	i	e	⋉

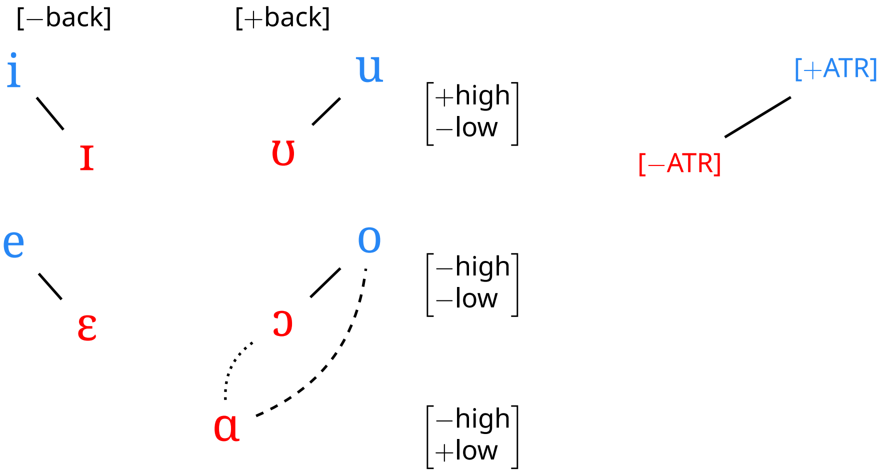
Non-deterministic functions



The identity of the output element for every input element is (in principle unboundedly) determined by elements from *both sides* of the string.

Composition of two contradirectional subsequential functions *that crucially interact*.

Turkana vowels



[−back] [+back]

i u

ɪ ʊ

$\begin{bmatrix}+\text{high}\\-\text{low}\end{bmatrix}$

[+ATR]

[−ATR]

e o

ɛ ɔ

$\begin{bmatrix}-\text{high}\\-\text{low}\end{bmatrix}$

ɑ

$\begin{bmatrix}-\text{high}\\+\text{low}\end{bmatrix}$

Turkana bidirectional harmony

a.	/ɛ-tɪm-ɑt/	[ɛ-tɪm-ɑt]	‘FEM.SG-shoe-SG’	root-controlled, raising of /ɑ/ to [o] from left, blocking by /ɑ/ from right
b.	/ɛ-pur-ɑt/	[e-pur-ot]	‘MASC.SG-beer-SG’	
c.	/ɑ-dɔk-ʊn/	[ɑ-dɔk-ʊn]	‘INF-climb-VEN’	
d.	/ɑ-lim-ʊn/	[ɑ-lim-un]	‘3-tell-VEN-ASP-VOI’	
e.	/ɑ-kɪ-dɔk/	[ɑ-kɪ-dɔk]	‘INF-K-climb’	suffix-controlled, raising of /ɑ/ to [o] from left, blocking by /ɑ/ from right, raising of /ɑ/ to [o] when flanked
f.	/ɛ-dɔk-ʊn-e/	[e-dok-un-e]	‘MASC.SG-climb-VEN-GER’	
g.	/ɑ-kɪ-rɑm/	[ɑ-kɪ-rɑm]	‘INF-K-beat’	
h.	/ɛ-rɑm-e/	[ɛ-rɑm-e]	‘MASC.SG-beat-GER’	
i.	/ɑ-tur-ɑɑn-u/	[ɑ-tur-oon-u]	‘GER-agile-HAB-NOM’	

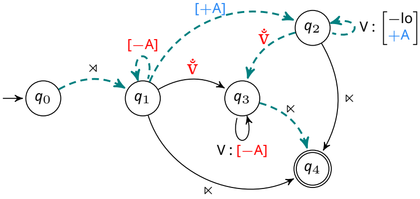
Turkana bidirectional harmony

j.	/ido-ʊn-ɪt/	[ido-un-it]	‘give.birth-VEN-ASP’	exceptional [–ATR] spreading
k.	/ɑ-k-ido-ʊn-**ɛ̆̊**t/	[ɑ-k-ɪdɔ-ʊn-**ɛ̆̊**t]	GER-K-give.birth-VEN-INST.LOC’	
l.	/ɑ-k-ɪnɔk-**ɑ**-kɪn-ɪ/	[ɑ-k-ɪnɔk-**ɑ**-kɪn-ɪ]	‘GER-K-light.fire-EPI-DAT-V’	exceptional [–ATR] spreading and raising of /ɑ/ to [o]
m.	/ɛ-ibus-**ɑ**-kɪn-ɪ/	[e-ibus-**o**-kin-i]	‘3-drop-EPI-DAT-V’	
n.	/ɛ-ibus-**ɑ**-kɪn-**ɑ̆̊**]	[ɛ-ɪbʊs-**ɔ**-kɪn-**ɑ̆̊**]	‘GER-drop-EPI-DAT-VOI’	

Left-subsequential function first

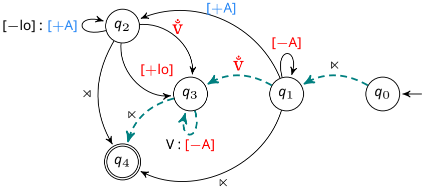
$R(L(x)); R \circ L(x)$

/ɛ-ibus-ɑ-kɪn-ɑ̊] ⟶ [ɛ-ɪbʊs-ɔ-kɪn-ɑ̊] ‘GER-drop-EPI-DAT-VOI’



⟶

in	⋊	ɛ	i	b	u	s	ɑ	k	ɪ	n	ɑ̊	⋉
	⋊	ɛ										
	⋊	ɛ	i	b	u	s						
	⋊	ɛ	i	b	u	s	o					
	⋊	ɛ	i	b	u	s	o	k	i	n		
int	⋊	ɛ	i	b	u	s	o	k	i	n	ɑ̊	⋉



⟵

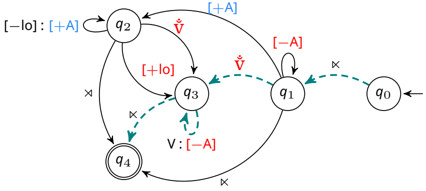
int	⋊	ɛ	i	b	u	s	o	k	i	n	ɑ̊	⋉
											ɑ̊	⋉
								k	ɪ	n	ɑ̊	⋉
							ɔ	k	ɪ	n	ɑ̊	⋉
			ɪ	b	ʊ	s	ɔ	k	ɪ	n	ɑ̊	⋉
out	⋊	ɛ	ɪ	b	ʊ	s	ɔ	k	ɪ	n	ɑ̊	⋉

✓

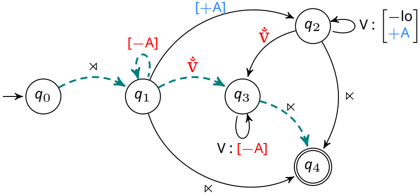
Right-subsequential first

$L(R(x)); L \circ R(x)$

/ε-ibus-ɑ-kɪn-ɑ̌̊] ⟶ [ε-ɪbʊs-ɔ-kɪn-ɑ̌̊] ‘GER-drop-EPI-DAT-VOI’



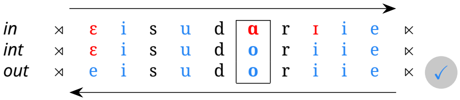
in	⋊	ε	i	b	u	s	ɑ	k	ɪ	n	ɑ̌̊	⋉
											ɑ̌̊	⋉
								k	ɪ	n	ɑ̌̊	⋉
							ɑ	k	ɪ	n	ɑ̌̊	⋉
			ɪ	b	ʊ	s	ɑ	k	ɪ	n	ɑ̌̊	⋉
int	⋊	ε	ɪ	b	ʊ	s	ɑ	k	ɪ	n	ɑ̌̊	⋉



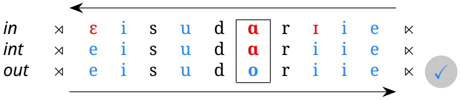
int	⋊	ε	ɪ	b	ʊ	s	ɑ	k	ɪ	n	ɑ̌̊	⋉
	⋊	ε										
	⋊	ε	ɪ	b	ʊ	s						
	⋊	ε	ɪ	b	ʊ	s	ɑ					
	⋊	ε	ɪ	b	ʊ	s	ɑ	k	ɪ	n		
out	⋊	ε	ɪ	b	ʊ	s	ɑ	k	ɪ	n	ɑ̌̊	⋉
							*					

Summary

No interaction (Maasai): weakly deterministic



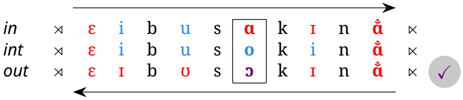
in	⋊	ɛ	i	s	u	d	ɑ	r	ɪ	i	e	⋉
int	⋊	ɛ	i	s	u	d	o	r	i	i	e	⋉
out	⋊	e	i	s	u	d	o	r	i	i	e	⋉ ✓



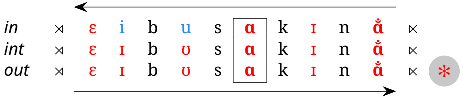
in	⋊	ɛ	i	s	u	d	ɑ	r	ɪ	i	e	⋉
int	⋊	e	i	s	u	d	ɑ	r	i	i	e	⋉
out	⋊	e	i	s	u	d	o	r	i	i	e	⋉ ✓

$$/\text{ɑ}/ \longrightarrow \begin{cases} [\text{o}] \;/\; [+\text{ATR}] \;\ldots\; \_\_ \\ [\text{ɑ}] \;/\; \textit{otherwise} \end{cases}$$

Interaction (Turkana): non-deterministic



in	⋊	ɛ	i	b	u	s	ɑ	k	ɪ	n	ɑ̐	⋉
int	⋊	ɛ	i	b	u	s	o	k	i	n	ɑ̐	⋉
out	⋊	ɛ	ɪ	b	ʊ	s	ɔ	k	ɪ	n	ɑ̐	⋉ ✓



in	⋊	ɛ	i	b	u	s	ɑ	k	ɪ	n	ɑ̐	⋉
int	⋊	ɛ	ɪ	b	ʊ	s	ɑ	k	ɪ	n	ɑ̐	⋉
out	⋊	ɛ	ɪ	b	ʊ	s	ɑ	k	ɪ	n	ɑ̐	⋉ *

$$/\text{ɑ}/ \longrightarrow \begin{cases} [\text{ɔ}] \;/\; [+\text{ATR}] \;\ldots\; \_\_ \;\ldots\; \text{v̐} \\ [\text{o}] \;/\; [+\text{ATR}] \;\ldots\; \_\_ \;\; \textit{otherwise} \\ [\text{ɑ}] \;/\; \textit{otherwise} \end{cases}$$

Summary

1. **Subsequential (deterministic) functions**

   The identity of the output element for every input element is (in principle unboundedly) determined by elements from *one side* of the string.

2. **Weakly deterministic functions**

   The identity of the output element for every input element is (in principle unboundedly) determined by elements from *one side or the other* of the string.

   - Can be described with a pair of composed contradirectional subsequential functions, **with no interaction between the two functions**.

3. **Non-deterministic functions**

   The identity of the output element for every input element is (in principle unboundedly) determined by elements from *both sides* of the string.

   - Can be described with a pair of composed contradirectional subsequential functions, **with interaction between the two functions**.

fin

Questions, etc. — `ebakovic@ucsd.edu`