How Much Homophony Is Normal?*

Abby Kaplan
University of California, Santa Cruz

CUNY Conference on the Word in Phonology
January 14, 2010

1 Introduction: Contrast Maintenance in Phonology

- Phonological patterns sensitive to the need to maintain contrast
  - Dispersion Theory (Flemming 2002): constraints that require contrast among phonemic categories oppose constraints against expending articulatory effort
  - Phonological patterns sometimes serve to maximize contrasts among potential lexical items (Padgett 2003; Padgett and Tabain 2005)
  - Some patterns seem especially prone to avoiding neutralization (e.g., lenition, Gurevich (2004))
- This sensitivity usually modeled at level of *potential* lexical items, not *actual* ones
  - Don't want to predict unattested phonological patterns:
    * Dispersed inventories only for lexical items in minimal pairs
    * Rules that apply only where they don't create homophones

      "These questions arise when we take the domain of explanation to be the set of actual lexical items in a language. But this is in fact not the practice in generative phonology. Instead, theories model the set of *possible* words of a language...." (Padgett 2003, 78-79)
  - Could there be other types of homophony avoidance?

*Many thanks to Aaron Kaplan, Yongeun Lee, Jaye Padgett, Dan Silverman, Paul Willis, and members of Phlunch at UC Santa Cruz. All shortcomings are my own. This research was funded by an NSF Graduate Research Fellowship.

- Silverman (to appear) argues that phonological patterns avoid creating homophones among *actual* words
  - Korean has many neutralizing phonological alternations
  - But these alternations result in only a relative handful of homophones
  - Homophony avoidance in lexical statistics rather than formalization of phonological pattern

- Remaining question: how can we be sure number of homophones in Korean is really unexpectedly low?

  ⇒ Perhaps *any* set of neutralizations would create very few homophones

- To establish a 'baseline' level of expected homophony, count homophones produced by alternative phonological rules and compare to actual rule; at least two ways to do this:

  1. Hand-select a few alternative rules for comparison
     - Silverman's method; alternatives produce more homophony than actual rules
     - Pro: can select phonologically plausible rules that are similar to actual rules
     - Con: results highly dependent on which alternatives we happen to pick
  2. Compute homophony for large number of alternative rules ('brute-force' method)
     - Method used for this talk
     - Pro: cover a lot more ground than hand-selection approach
     - Con: hard to filter out implausible rules

2 Method

2.1 Korean

- Why Korean?
  - Large number of neutralizing phonological alternations

(1) a. (i) /nɑʧ-i/ → [nɑʧi] 'day.NOM'
(ii) /nɑʧ-k'wɑ/ → [nɑtk'wɑ] 'day and'
b. (i) /nɑʧʰ-i/ → [nɑʧʰi] 'face.NOM'
(ii) /nɑʧʰ-k'wɑ/ → [nɑtk'wɑ] 'face and'

  - Writing system roughly morphophonemic: orthography can be used to approximate underlying forms

Table 1: Surface realizations of underlying coda-onset sequences in Korean

<table>
<tr><th></th><th>p^h</th><th>p</th><th>p’</th><th>t^h</th><th>t</th><th>t’</th><th>s</th><th>s’</th><th>ʧ^h</th><th>ʧ</th><th>ʧ’</th><th>k^h</th><th>k</th><th>k’</th><th>m</th><th>n</th><th>l</th><th>h</th><th>∅</th></tr>
<tr><td>p</td><td rowspan="12">p^h</td><td rowspan="12" colspan="2">p’</td><td rowspan="5">pt^h</td><td rowspan="5" colspan="2">pt’</td><td rowspan="5" colspan="2">ps’</td><td rowspan="5">pʧ^h</td><td rowspan="5" colspan="2">pʧ’</td><td rowspan="5">pk^h</td><td rowspan="5" colspan="2">pk’</td><td rowspan="12">pm</td><td rowspan="5">pn</td><td rowspan="5">pl</td><td>p^h</td><td>p</td></tr>
<tr><td>p^h</td><td>p^{h}h</td><td>p^h</td></tr>
<tr><td>ps</td><td>ps’h</td><td>ps’</td></tr>
<tr><td>lp</td><td>lp^h</td><td>lp</td></tr>
<tr><td>lp^h</td><td>lp^{h}h</td><td>lp^h</td></tr>
<tr><td>h</td><td rowspan="7">t^h</td><td rowspan="7" colspan="2">t’</td><td rowspan="7" colspan="2">s’</td><td rowspan="7">ʧ^h</td><td rowspan="7" colspan="2">ʧ’</td><td rowspan="12">k^h</td><td rowspan="12" colspan="2">k’</td><td rowspan="7">tn</td><td rowspan="7">tl</td><td>hh</td><td>∅</td></tr>
<tr><td>t</td><td>t^h</td><td>t</td></tr>
<tr><td>t^h</td><td>t^{h}h</td><td>t^h</td></tr>
<tr><td>s</td><td>sh</td><td>s</td></tr>
<tr><td>s’</td><td>s’h</td><td>s’</td></tr>
<tr><td>ʧ</td><td>ʧ^h</td><td>ʧ</td></tr>
<tr><td>ʧ^h</td><td>ʧ^{h}h</td><td>ʧ^h</td></tr>
<tr><td>k</td><td rowspan="5">kp^h</td><td rowspan="5" colspan="2">kp’</td><td rowspan="5">kt^h</td><td rowspan="5" colspan="2">kt’</td><td rowspan="5" colspan="2">ks’</td><td rowspan="5">kʧ^h</td><td rowspan="5" colspan="2">kʧ’</td><td rowspan="5">km</td><td rowspan="5">kn</td><td rowspan="5">kl</td><td rowspan="2">k^h</td><td>k</td></tr>
<tr><td>k’</td><td>k’</td></tr>
<tr><td>k^h</td><td>k^{h}h</td><td>k^h</td></tr>
<tr><td>ks</td><td>ksh</td><td>ks’</td></tr>
<tr><td>lk</td><td>lk^h</td><td>lk</td></tr>
<tr><td>m</td><td>mp^h</td><td>mp</td><td>mp’</td><td>mt^h</td><td>mt</td><td>mt’</td><td>ms</td><td>ms’</td><td>mʧ^h</td><td>mʧ</td><td>mʧ’</td><td>mk^h</td><td>mk</td><td>mk’</td><td>mm</td><td>mn</td><td>ml</td><td colspan="2">m</td></tr>
<tr><td>n</td><td rowspan="3">np^h</td><td>np</td><td rowspan="3">np’</td><td rowspan="3">nt^h</td><td>nt</td><td rowspan="3">nt’</td><td>ns</td><td rowspan="3">ns’</td><td rowspan="3">nʧ^h</td><td>nʧ</td><td rowspan="3">nʧ’</td><td rowspan="3">nk^h</td><td>nk</td><td rowspan="3">nk’</td><td rowspan="3">nm</td><td rowspan="3">nn</td><td rowspan="3">nl</td><td colspan="2">n</td></tr>
<tr><td>nh</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td>nhh</td><td></td></tr>
<tr><td>nʧ</td><td>nʧ^h</td><td>nʧ</td></tr>
<tr><td>ŋ</td><td>ŋp^h</td><td>ŋp</td><td>ŋp’</td><td>ŋt^h</td><td>ŋt</td><td>ŋt’</td><td>ŋs</td><td>ŋs’</td><td>ŋʧ^h</td><td>ŋʧ</td><td>ŋʧ’</td><td>ŋk^h</td><td>ŋk</td><td>ŋk’</td><td>ŋm</td><td>ŋn</td><td>ŋl</td><td colspan="2">ŋ</td></tr>
<tr><td>lm</td><td rowspan="5">lp^h</td><td rowspan="2">lp</td><td rowspan="5">lp’</td><td rowspan="5">lt^h</td><td rowspan="2">lt</td><td rowspan="5">lt’</td><td rowspan="2">ls</td><td rowspan="5">ls’</td><td rowspan="5">lʧ^h</td><td rowspan="2">lʧ</td><td rowspan="5">lʧ’</td><td rowspan="5">lk^h</td><td rowspan="2">lk</td><td rowspan="5">lk’</td><td rowspan="5">lm</td><td rowspan="5">ln</td><td rowspan="5">ll</td><td colspan="2">lm</td></tr>
<tr><td>l</td><td colspan="2">l</td></tr>
<tr><td>lh</td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td>lhh</td><td></td></tr>
<tr><td>lt^h</td><td>lt^{h}h</td><td>lt^h</td></tr>
<tr><td>ls</td><td>lsh</td><td>ls</td></tr>
<tr><td>∅</td><td>p^h</td><td>p</td><td>p’</td><td>t^h</td><td>t</td><td>t’</td><td>s</td><td>s’</td><td>ʧ^h</td><td>ʧ</td><td>ʧ’</td><td>k^h</td><td>k</td><td>k’</td><td>m</td><td>n</td><td>l</td><td colspan="2">∅</td></tr>
</table>

- Table 1 shows surface realizations of underlying coda-onset sequences
  - Reflect nine ordered rules from descriptions in Sohn (1994): resyllabification, [h]-aspiration, coda neutralization, sibilation, tensification, consonant cluster simplification, decoronization, [h]-weakening, reduction
  - Some other rules suppressed:
    - Non-neutralizing alternations (e.g., intervocalic voicing)
    - Morphologically conditioned alternations (e.g., lateralization)
- Data on Korean lexicon
  - Data from Korean National Database (Lee 2006) (uninflected stems in Korean orthography; no morpheme boundaries)
  - Phonological rules applied via Java scripts (collaboration w/ Paul Willis)
  - Rules applied only stem-internally: application to codas may be bled by vowel-initial suffixes, of which there are many (Albright and Kang 2009)

2.2 Procedure

- Same procedure for each rule; illustrated here with data for overall neutralization pattern

2.2.1 Step 1: Count Homophones Created by Rule

- Measures of homophony

  Homophones Number of words in lexicon w/ at least one homophone

  Weighted Homophones Sum of frequencies of words w/ at least one homophone

  Homophone Pairs Number of pairs of homophones in lexicon

  Homophone Sets Number of (maximal) sets of words in lexicon that all neutralize to the same thing

- Table 2 shows levels of homophony underlyingly and after all rules have applied

  ⇒ 'New': number of homophones added by rules

- For results of simulations, only 'homophones' and 'weighted homophones' shown

  ⇒ 'Homophone pairs' and 'homophone sets' very similar to 'homophones'

Table 2: Homophony in Korean lexicon

Measure	**Underlying**	**Surface**	**New**
Homophones	6201	6646	445
Weighted Homophones	291681	319835	28154
Homophone Pairs	4308	4692	384
Homophone Sets	2794	2975	181

2.2.2 Step 2: Simulate Alternative Rules

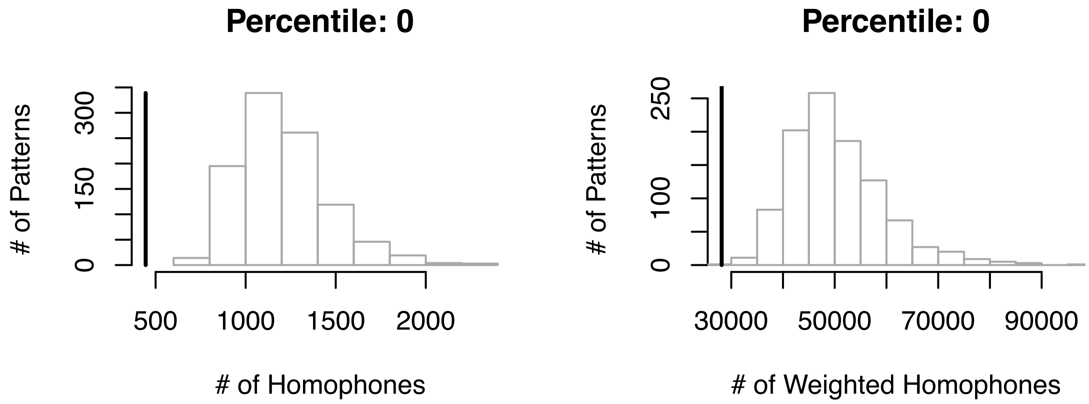
- Three simulations; each creates 1000 random patterns w/ same number of neutralizations as actual rule
- ‘a’-series simulation: neutralize random sets of coda-onset sequences
  - Table 3 illustrates this for overall neutralization pattern
  - Randomly distribute onset-coda sequences in table; superimpose grid from actual pattern
  - All sequences in same box neutralize with each other
  - (Some cells neutralize w/ nothing – left blank; sequences here don’t matter)
  - Figure 1 shows results for overall pattern
    - Histograms show level of homophony across simulated patterns
    - Vertical lines show actual number of homophones
    - Title gives percentile rank of actual number of homophones among simulated patterns: smaller percentile → actual level of homophony surprisingly low

  ⇒ For both measures, every simulated pattern resulted in *more* homophony than actual pattern

Table 3: Example of neutralization patterns in an ‘a’-series simulation

ʧʧ’	hʧ	np’	lkh	ls	pp	tp’	ʧl	lhk^h	t^{h}ʧ’	s’s’	t^{h}s’	nʧp^h	lp^{h}t^h	lst^h	k’p	t^{h}t’	lp^{h}t	mʧ^h
ʧ^{h}l	ls’	nhh	psn	nʧs	ksp	kt^h	lkp^h	ʧ^{h}m	sl	t^{h}t’	k’k^h	mn	ʧk^h	lmp’	ksp’	ʧ^{h}k’		mm
pk’	kss’	nk^h	lmʧ	k’s’	ll	ksk^h	psʧ’	p^h∅	st^h	lhp^h	ʧt’	k’ʧ’	ʧn	tʧ^h	∅ʧ^h	pk		pk^h
nʧl	ŋʧ^h	∅t	kp^h	nʧp’	hk’	mh	nl	lkn	s’m	k^{h}p^h	ŋh	lkm	ŋt	hp’	p^{h}p^h	ʧp	kʧ^h	nt’
ks	pss’	hs’	ls∅	psk^h	lpʧ^h	k^{h}s’	pl	hs	lk∅	s’t^h	lt^{h}k’	ŋʧ’	lkʧ	ms’	ŋk^h	lʧ^h		
lp∅	lp^{h}s	ŋp’	lmt^h	pp^h	s’ʧ	mk’	s’p’	lhs’	kst’	lt^{h}l	ʧ^{h}ʧ^h	lt^h∅	lkʧ^h	k’t	ps∅	lpp		∅∅
k’p’	lk^h	lsn	p^{h}m	∅s’	lhʧ^h	lkk^h	s’k	lʧ	mt	s’l	lpn	m∅	lss’	kst	s’t’	h∅	ss	nm
kʧ’	k^{h}p’	lsʧ^h	pʧ’	k’∅	lsk^h	pst	nhp	lpp’	mt’	ʧ∅	lmk	tn	sp	sm	lp^{h}ʧ^h	k^{h}ʧ		lp^{h}l
ht	s’∅	lt’	k’p’	lmt’	∅s	ʧ^{h}ʧ	psk	p^{h}ʧ’	ksʧ	p^{h}p	kk	kp	p^{h}k	k^{h}ʧ^h	ʧ^{h}t’	ss’		t∅
psk’	lsʧ	lt^{h}k	lht’	ks∅	nht^h	∅ʧ	lmʧ^h	sp^h	ŋn	lm	nhs	tp	p^{h}t^h	∅k’	nn	hh		t^{h}s
∅ʧ’	nʧk’	lpt’	psp	lh	lmh	lt^{h}p’	nʧ’	np	k’m	mk	mp	hm	psh	lms	∅n	kh	nʧh	lpl
kk’	k^h∅	ʧ^{h}k	lpʧ	k’k	ksk’	lp^{h}k^h	nhʧ	lsl	p^{h}n	nʧs’	ht^h	t^h∅	lhs	hʧ^h	k^{h}t’	lkt^h		nʧʧ’
lp^{h}ʧ’	k’t^h	p^{h}h	sk	nhʧ^h	nhk’	ʧs	pt’	nhp^h	ʧ^h∅	lt^{h}ʧ’	t^{h}l	tk	lps	ŋt’	th	nk	∅k^h	lsk
t^{h}ʧ^h	nk’	lt^h	kst^h	lt^{h}ʧ	s’t	ʧʧ^h	lks	tʧ’	nʧk^h	lhl	tʧ	ʧʧ	∅l	nʧp	ʧk	nh	k’k’	lpt
lsp	kʧ	k’s	ʧ^{h}s	lsk’	ʧk’	s’p^h	sʧ	psm	t^{h}h	kt	sp’	nʧk	lp^{h}n	ps	np^h	t^{h}k^h		nʧt
st’	nt^h	t^{h}p	ʧt	ŋl	lp^{h}k’	ksʧ’	ʧp’	lt^{h}k^h	ʧ^{h}ʧ’	lmp	lsʧ	s’k^h	∅k	ʧ^{h}p’	lpm	ksm		nh∅
ksn	s’n	∅t^h	lhm	k^{h}s	ŋs’	lss	k’ʧ^h	lhʧ’	nhl	lp^{h}k	∅p’	nhk	hp	ksh	nhn	lkʧ’	ph	lsm
																	psp^h	lm∅
lt^{h}h		t^{h}k’	∅h		lsp’		lpt^h	∅m		lmt	np		ʧ^{h}p	s’ʧ’	nhk^h	sk’	n∅	nʧm
ŋp^h	ms	lhp’	s’ʧ^h	p^{h}ʧ^h	hk	lp	nʧʧ^h	lt^{h}m	nt	hn	ml	t^{h}m	∅p^h	s’p	lt^{h}s’	pʧ	p^{h}t’	ʧm
mʧ’	pn	lt^{h}t’	lp’	hʧ’	lh∅	∅p	lkp’	nʧ∅	hk^h	k^{h}k	sk^h	ps’	tt’	lst’	psl	st	k’∅	tm
																	lps’	k^{h}t^h
nht	lpʧ’	t^{h}p^h	lp^{h}s’	p^{h}ʧ	lmp^h	k^{h}t	lpp^h	k’l	s’k’	tk^h	tk’	ŋt^h	mp^h	psp’	k^{h}k^h	lsp^h	lt	ts
mt^h	k^{h}l	nʧt’	lms’	ŋk	p^{h}s’	ns’	∅t’	lhn	k^{h}ʧ’	ʧ^{h}k^h	lp^{h}ʧ	lp^h	sn	ʧ^{h}s’	nhm	k’p^h	lkl	lkt’
ʧ^{h}h	nhʧ’	s’s	k’h	nhs’	mʧ	k^{h}k’	lt^{h}t	t^{h}ʧ	pst^h	p^{h}k^h	p^{h}p’	lp^{h}m	lml	pp’	nʧ	hp^h	kk^h	ksk
lmʧ’	tl	ʧs’	pt	p∅	k’ʧ	pst’	k^{h}p	nʧʧ	p^{h}k’	lp^{h}p’	nʧ^h	ŋʧ	s∅	ŋk’	lhh	ʧ^{h}p^h		ts’
kl	t^{h}ʧ^h	lk’	pss	ʧ^{h}t	ns	nʧt^h	ht	tp^h	lp^h∅	psʧ^h	s’h	k^{h}h	km	ksʧ^h	ln	t^{h}n		kn
lt^{h}s	t^{h}p’	lmm	lhk	ʧ^{h}n	lph	pm	kss	lp^{h}p	sʧ’	lkt	lt^{h}ʧ^h	ksl	kn	lht^h	psʧ	lt^{h}t^h	lp^{h}h	lt^{h}n

Figure 1: Amount of homophony in simulation a



- ‘b’-series simulation: randomly shuffle onsets, codas separately
  - Table 4 illustrates for overall pattern
  - Neutralized sequences more similar than in ‘a’-series: more phonologically plausible
  - Figure 2 shows results
    - ⇒ Almost every simulated pattern results in more homophony than actual pattern

Figure 2: Amount of homophony in simulation b

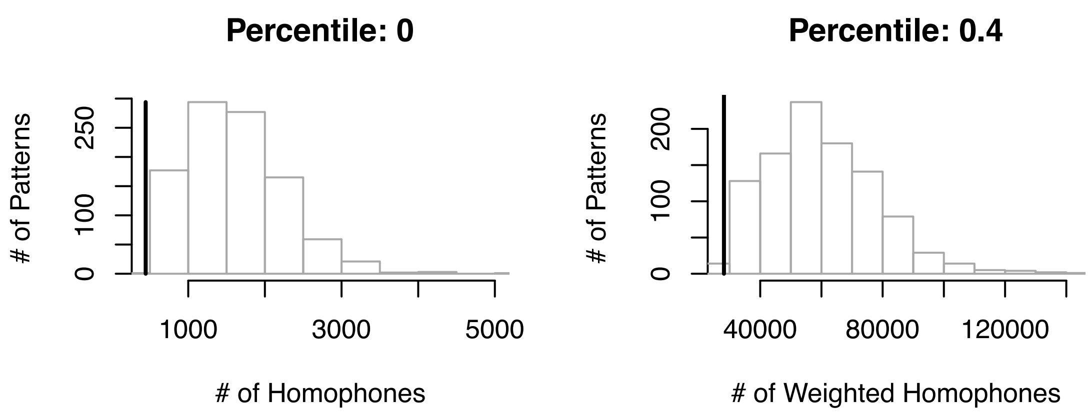


Table 4: Example of neutralization patterns in a ‘b’-series simulation

	p	**s’**	**l**	**∅**	**ʧ^h**	**p^h**	**n**	**k**	**ʧ’**	**t’**	**k^h**	**s**	**k’**	**h**	**t**	**p’**	**ʧ**	**t^h**	**m**
p	pp	ps’	pl	p∅	pʧ^h	pp^h	pn	pk	pʧ’	pt’	pk^h	ps	pk’	ph	pt	pp’	pʧ	pt^h	pm
n	np	ns’	nl	n∅	nʧ^h	np^h	nn	nk	nʧ’	nt’	nk^h	ns	nk’	nh	nt	np’	nʧ	nt^h	nm
lk	lkp	lks’	lkl	lk∅	lkʧ^h	lkp^h	lkn	lkk	lkʧ’	lkt’	lkk^h	lks	lkk’	lkh	lkt	lkp’	lkʧ	lkt^h	lkm
p^h	p^{h}p	p^{h}s’	p^{h}l	p^h∅	p^{h}ʧ^h	p^{h}p^h	p^{h}n	p^{h}k	p^{h}ʧ’	p^{h}t’	p^{h}k^h	p^{h}s	p^{h}k’	p^{h}h	p^{h}t	p^{h}p’	p^{h}ʧ	p^{h}t^h	p^{h}m
lm	lmp	lms’	lml	lm∅	lmʧ^h	lmp^h	lmn	lmk	lmʧ’	lmt’	lmk^h	lms	lmk’	lmh	lmt	lmp’	lmʧ	lmt^h	lmm
m	mp	ms’	ml	m∅	mʧ^h	mp^h	mn	mk	mʧ’	mt’	mk^h	ms	mk’	mh	mt	mp’	mʧ	mt^h	mm
ŋ	ŋp	ŋs’	ŋl	ŋ∅	ŋʧ^h	ŋp^h	ŋn	ŋk	ŋʧ’	ŋt’	ŋk^h	ŋs	ŋk’	ŋh	ŋt	ŋp’	ŋʧ	ŋt^h	ŋm
k	kp	ks’	kl	k∅	kʧ^h	kp^h	kn	kk	kʧ’	kt’	kk^h	ks	kk’	kh	kt	kp’	kʧ	kt^h	km
h	hp	hs’	hl	h∅	hʧ^h	hp^h	hn	hk	hʧ’	ht’	hk^h	hs	hk’	hh	ht	hp’	hʧ	ht^h	hm
ks	ksp	kss’	ksl	ks∅	ksʧ^h	ksp^h	ksn	ksk	ksʧ’	kst’	ksk^h	kss	ksk’	ksh	kst	ksp’	ksʧ	kst^h	ksm
lh	lhp	lhs’	lhl	lh∅	lhʧ^h	lhp^h	lhn	lhk	lhʧ’	lht’	lhk^h	lhs	lhk’	lhh	lht	lhp’	lhʧ	lht^h	lhm
k^h	k^{h}p	k^{h}s’	k^{h}l	k^h∅	k^{h}ʧ^h	k^{h}p^h	k^{h}n	k^{h}k	k^{h}ʧ’	k^{h}t’	k^{h}k^h	k^{h}s	k^{h}k’	k^{h}h	k^{h}t	k^{h}p’	k^{h}ʧ	k^{h}t^h	k^{h}m
ʧ^h	ʧ^{h}p	ʧ^{h}s’	ʧ^{h}l	ʧ^h∅	ʧ^{h}ʧ^h	ʧ^{h}p^h	ʧ^{h}n	ʧ^{h}k	ʧ^{h}ʧ’	ʧ^{h}t’	ʧ^{h}k^h	ʧ^{h}s	ʧ^{h}k’	ʧ^{h}h	ʧ^{h}t	ʧ^{h}p’	ʧ^{h}ʧ	ʧ^{h}t^h	ʧ^{h}m
s	sp	ss’	sl	s∅	sʧ^h	sp^h	sn	sk	sʧ’	st’	sk^h	ss	sk’	sh	st	sp’	sʧ	st^h	sm
t	tp	ts’	tl	t∅	tʧ^h	tp^h	tn	tk	tʧ’	tt’	tk^h	ts	tk’	th	tt	tp’	tʧ	tt^h	tm
nʧ	nʧp	nʧs’	nʧl	nʧ∅	nʧʧ^h	nʧp^h	nʧn	nʧk	nʧʧ’	nʧt’	nʧk^h	nʧs	nʧk’	nʧh	nʧt	lnʧp’	nʧʧ	nʧt^h	nʧm
lp^h	lp^{h}p	lp^{h}s’	lp^{h}l	lp^h∅	lp^{h}ʧ^h	lp^{h}p^h	lp^{h}n	lp^{h}k	lp^{h}ʧ’	lp^{h}t’	lp^{h}k^h	lp^{h}s	lp^{h}k’	lp^{h}h	lp^{h}t	lp^{h}p’	lp^{h}ʧ	lp^{h}t^h	lp^{h}m
lt^h	lt^{h}p	lt^{h}s’	lt^{h}l	lt^h∅	lt^{h}ʧ^h	lt^{h}p^h	lt^{h}n	lt^{h}k	lt^{h}ʧ’	lt^{h}t’	lt^{h}k^h	lt^{h}s	lt^{h}k’	lt^{h}h	lt^{h}t	lt^{h}p’	lt^{h}ʧ	lt^{h}t^h	lt^{h}m
l	lp	ls’	ll	l∅	lʧ^h	lp^h	ln	lk	lʧ’	lt’	lk^h	ls	lk’	lh	lt	lp’	lʧ	lt^h	lm
∅	∅p	∅s’	∅l	∅∅	∅ʧ^h	∅p^h	∅n	∅k	∅ʧ’	∅t’	∅k^h	∅s	∅k’	∅h	∅t	∅p’	∅ʧ	∅t^h	∅m
ps	psp	pss’	psl	ps∅	psʧ^h	psp^h	psn	psk	psʧ’	pst’	psk^h	pss	psk’	psh	pst	psp’	psʧ	pst^h	psm
ls	lsp	lss’	lsl	ls∅	lsʧ^h	lsp^h	lsn	lsk	lsʧ’	lst’	lsk^h	lss	lsk’	lsh	lst	lsp’	lsʧ	lst^h	lsm
s’	s’p	s’s’	s’l	s’∅	s’ʧ^h	s’p^h	s’n	s’k	s’ʧ’	s’t’	s’k^h	s’s	s’k’	s’h	s’t	s’p’	spʧ	s’t^h	s’m
k’	k’p	k’s’	k’l	k’∅	k’ʧ^h	k’p^h	k’n	k’k	k’ʧ’	k’t’	k’k^h	k’s	k’k’	k’h	k’t	k’p’	k’ʧ	k’t^h	k’m
t^h	t^{h}p	t^{h}s’	t^{h}l	t^h∅	t^{h}ʧ^h	t^{h}p^h	t^{h}n	t^{h}k	t^{h}ʧ’	t^{h}t’	t^{h}k^h	t^{h}s	t^{h}k’	t^{h}h	t^{h}t	t^{h}p’	t^{h}ʧ	t^{h}t^h	t^{h}m
ʧ	ʧp	ʧs’	ʧl	ʧ∅	ʧʧ^h	ʧp^h	ʧn	ʧk	ʧʧ’	ʧt’	ʧt^h	ʧs	ʧk’	ʧh	ʧt	ʧp’	ʧʧ	ʧt^h	ʧm
lp	lpp	lps’	lpl	lp∅	lpʧ^h	lpp^h	lpn	lpk	kpʧ’	lpt’	lpt^h	lps	lpk’	lph	lpt	lpp’	lpʧ	lpt^h	lpm
nh	nhp	nhs’	nhl	nh∅	nhʧ^h	nhp^h	nhn	nhk	nhʧ’	nht’	nht^h	nhs	nhk’	nhh	nht	nhp’	nhʧ	nht^h	nhm

- 'c'-series simulation: randomly shuffle segments; combine according to neutralization 'template' for rule
  - Template: specification of which coda-onset sequences neutralize with each other
  - Example from tensification rule: for each ordered pair in set B, first member preceded by anything from set A neutralizes with second member preceded by same thing

    ⇒ So, neutralizing pairs include {k+k, k+k'}, {t+k, t+k'}, {k+t, k+t'}, etc.

Figure 3: Template for tensification

{/A + B_1/, /A + B_2/}	A = {k, t, p, s, kt, nt, lk, lp, lt, pt, ks, ns, ls, ps} B = {⟨k, k'⟩, ⟨t, t'⟩, ⟨p, p'⟩, ⟨s, s'⟩, ⟨ʧ, ʧ'⟩, ⟨sh, s'h⟩}

  - In 'c'-series simulation: randomly assign new segments to sets A and B (and C and so on, if template has more sets); restrictions:
    - Identical segments across sets stay that way (i.e., every [k] in figure 3 becomes [s] in figure 4)
    - Segments that appear as codas in template must be possible codas, those as onsets must be possible onsets (i.e., every member of set A must be a possible coda)
  - Example given in figure 4

    ⇒ New neutralizing pairs include {s+s, s+k^h}, {n+s, n+k^h}, {s+n, s+s'h}

Figure 4: Example of new sets for tensification template in a 'c'-series simulation

A = {s, n, t, k, ns, ∅, ŋ, p, m, ps, ls, lt, pt, lk}

B = {⟨s, k^h⟩, ⟨n, s'h⟩, ⟨t, k^hh⟩, ⟨k, t^h⟩, ⟨t, ŋh⟩, ⟨mh, p^hh⟩}

- In theory, 'b'-series simulations phonologically more plausible than 'a'-series, 'c'-series more plausible than 'b'-series

3 Results

- For each rule, I give:
  - Description of rule
  - Template for rule
  - Homophony produced by rule
  - Results of three simulations
- Display of simulation results
  - Two graphs: homophones measure on left, weighted homophones on right
  - Density curves ($\sim$ smoothed histograms) show distribution of homophony in 1000 patterns run in each simulation
  - Three curves, one for each simulation ('a', 'b', 'c')
  - Vertical bar: actual level of homophony
  - Legend notes percentile rank of actual level in each simulation
- Note:
  - Rules apply in ordered fashion, each rule to output of previous rule
  - Possible onsets and codas at each stage differ accordingly
  - 'Total' homophones listed for each rule are total homophones in lexicon after rule applies

3.1 Rule 1: Resyllabification

- Resyllabify single coda consonant into following onsetless syllable or syllable with initial [h] (Sohn 1994, 164)
- Almost all simulations produce far less homophony than actual rule; most produce none at all!
- Small additional peak in histograms for ‘c’-series near actual pattern
  - ⇒ Suggests a single pair of neutralized coda-onset sequences responsible for most of homophony produced by resyllabification

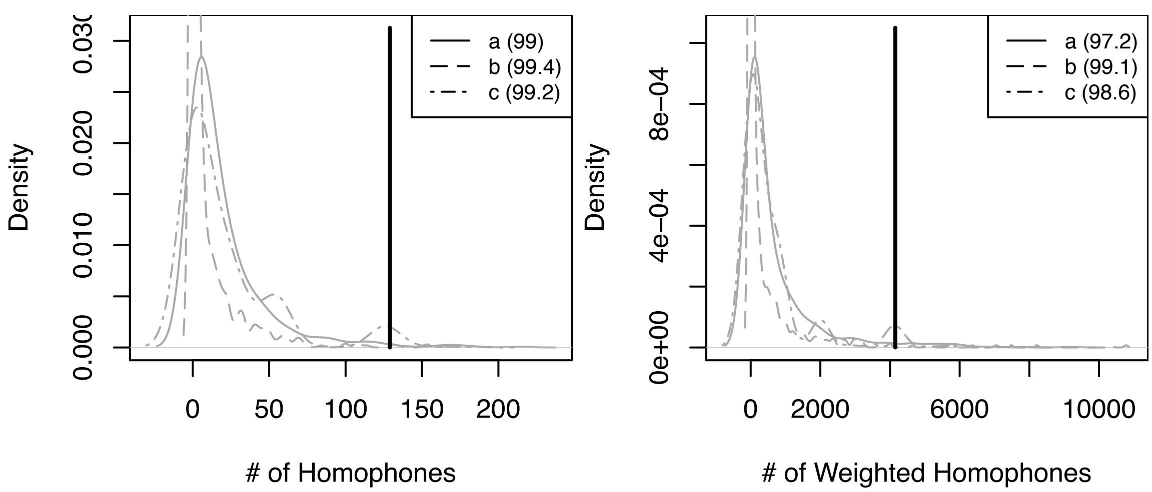
Figure 5: Homophony after resyllabification

Measure	Total	New
Homophones	6330	129
Weighted	295832	4151
Pairs	4417	109
Sets	2847	53

Figure 6: Template for resyllabification

	A = {k, n, t, l, m, p, s, ʧ, ʧ^{h}, k^{h}, t^{h}, p^{h}, h, k’, s’}
{/A + B/, /B + A/}	B = {∅}
{/C_1 + B/, /C_2 + C_3/}	C = {⟨ks, k, s⟩, ⟨nʧ, n, ʧ⟩, ⟨nh, n, h⟩, ⟨lk, l, k⟩, ⟨lm, l, m⟩, ⟨lp, l, p⟩, ⟨ls, l, s⟩, ⟨lt^{h}, l, t^{h}⟩, ⟨lp^{h}, l, p^{h}⟩, ⟨lh, l, h⟩, ⟨ps, p, s⟩}

Figure 7: Amount of homophony in simulation series 1



3.2 Rule 2: [h]-Aspiration

- Fuse plain noncontinuant obstruent with adjacent [h] into homorganic aspirated obstruent (Sohn 1994, 166)
- In ‘a’-series, most simulations yield less homophony than actual pattern: over half of ‘a’-series simulations yield none
- In ‘b’- and ‘c’-series, actual level of homophony in lower half of simulations
- Note that the more phonologically plausible the simulation, the more homophony it tends to produce (c > b > a)

Figure 8: Homophony after [h]-aspiration

Measure	**Total**	**New**
Homophones	6340	10
Weighted	296460	628
Pairs	4425	8
Sets	2851	4

Figure 9: Template for [h]-aspiration

{/A_1 + B/, /B + A_1/, /C + A_2/, /C + A_3/}

{/D + A_2/, /D + A_3/}

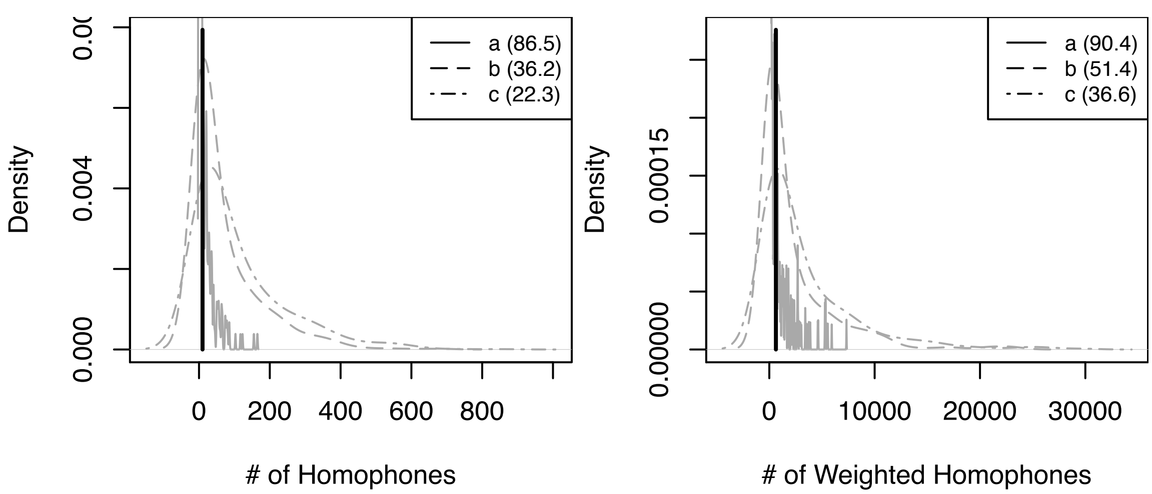
A = {⟨k, kh, k^h⟩, ⟨t, th, t^h⟩, ⟨p, ph, p^h⟩, ⟨ʧ, ʧh, $ʧ^h$⟩}

B = {h}

C = {∅}

D = {k, n, t, l, m, p, s, ŋ, ʧ, $ʧ^h$, k^h, t^h, p^h, k’, s’}

Figure 10: Amount of homophony in simulation series 2



3.3 Rule 3: Coda Neutralization

- Syllable-final obstruents become plain stops; /h/, /s/ → [t] (Sohn 1994, 165)
- Simulated patterns yield more homophony than actual pattern; less concentrated at low end of scale
- Exception: weighted homophones measure for ‘a’-series
- Note that percentile rank of actual level of homophony is generally greater for weighted homophones measure

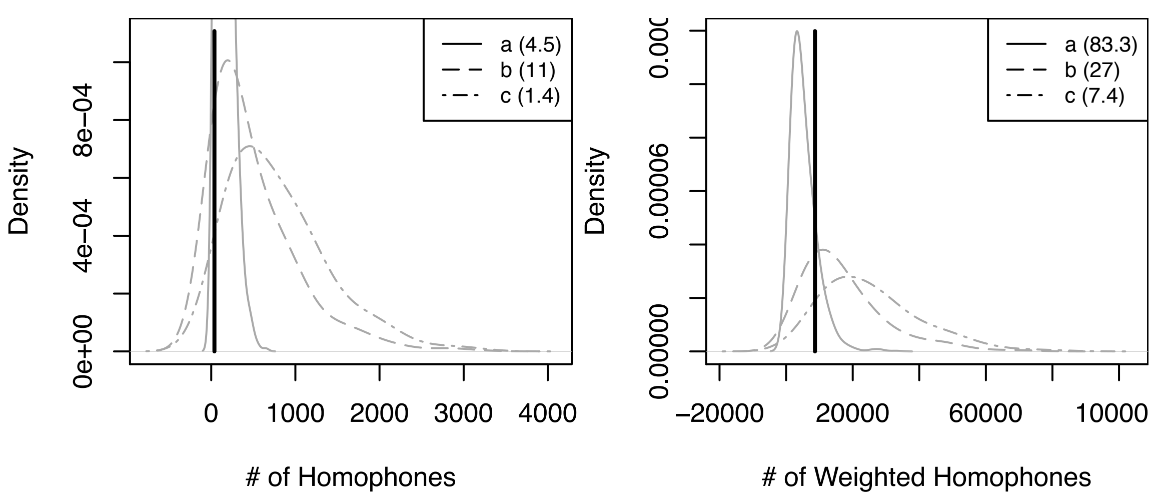
Figure 11: Homophony after coda neutralization

Measure	**Total**	**New**
Homophones	6378	38
Weighted	305133	8673
Pairs	4475	50
Sets	2863	12

Figure 12: Template for coda neutralization

	A = {⟨k, k’, k^{h}⟩, ⟨ls, lt^{h}, lh⟩}
{/A_1 + B/, /A_2 + B/, /A_3 + B}	B = {k, n, t, l, m, p, s, ŋ, ∅, ʧ, ʧ^{h}, k^{h}, t^{h}, p^{h}, k’, t’, p’, s’, ʧ’, nh, lh, mh, sh, ʧ^{h}h, k^{h}h, t^{h}h, p^{h}h, hh, k’h, s’h}
{/C_1 + B/, /C_2 + B/, /C_3 + B/, /C_4 + B/, /C_5 + B/, /C_6 + B/, /C_7 + B/}	C = {⟨t, s, ʧ, ʧ^{h}, t^{h}, h, s’⟩}
{/D_1 + B/, /D_2 + B/}	D = {⟨p, p^{h}⟩, ⟨nʧ, nh⟩, ⟨lp, lp^{h}⟩}

Figure 13: Amount of homophony in simulation series 3



3.4 Rule 4: Sibilation

- /t/ becomes [s] before [s] or [s'] (Sohn 1994, 165)
- Since all /s/s became [t]s in Coda Neutralization, this rule is non-neutralizing

3.5 Rule 5: Tensification

- Plain obstruents become tense after obstruents (Sohn 1994, 173)
- Most simulated patterns produce more homophony than actual rule
- Again, more plausible simulations tend to yield more homophony (c > b > a)

Figure 14: Homophony after tensification

Measure	**Total**	**New**
Homophones	6382	4
Weighted	305140	7
Pairs	4477	2
Sets	2865	2

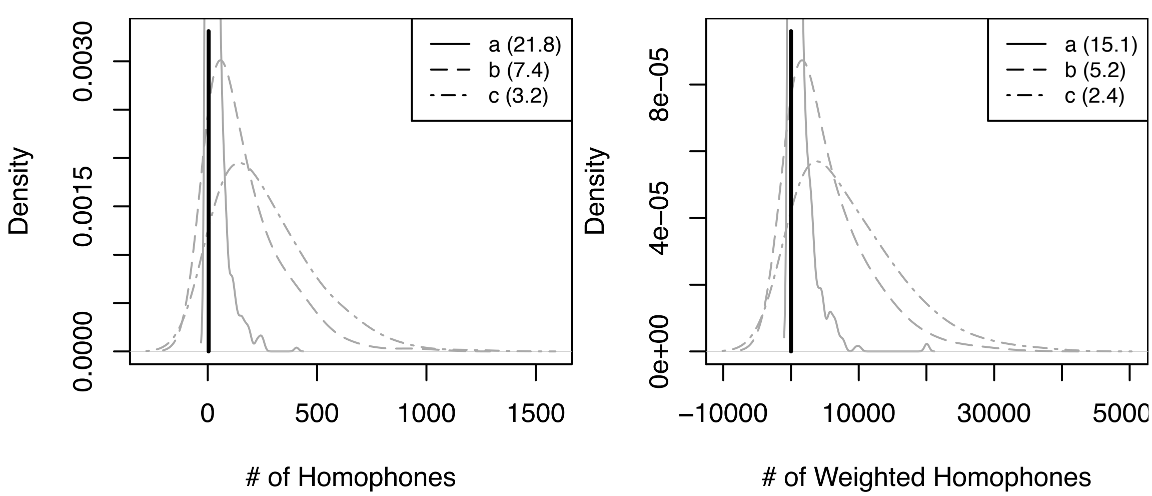
Figure 15: Template for tensification

{/A + B_1/, /A + B_2/}

A = {k, t, p, s, kt, nt, lk, lp, lt, pt, ks, ns, ls, ps}

B = {⟨k, k'⟩, ⟨t, t'⟩, ⟨p, p'⟩, ⟨s, s'⟩, ⟨ʧ, ʧ'⟩, ⟨sh, s'h⟩}

Figure 16: Amount of homophony in simulation series 5



3.6 Rule 6: Consonant Cluster Simplification

- Delete one consonant from all coda clusters (Sohn 1994, 170)
  - If first consonant in cluster is /l/ and second is non-coronal stop, delete /l/ (this simplifies the facts somewhat)
  - Otherwise, delete second consonant
- Most simulated patterns produce far more homophony than actual rule
- Again, note that percentile rank for actual rule is greater for weighted homophones measure

Figure 17: Homophony after consonant cluster simplification

Measure	Total	New
Homophones	6399	17
Weighted	308910	3770
Pairs	4499	22
Sets	2871	6

Figure 18: Template for simplification

{/A_1 + B/, /A_2 + B/, /A_3 + B/, /A_4 + B/}

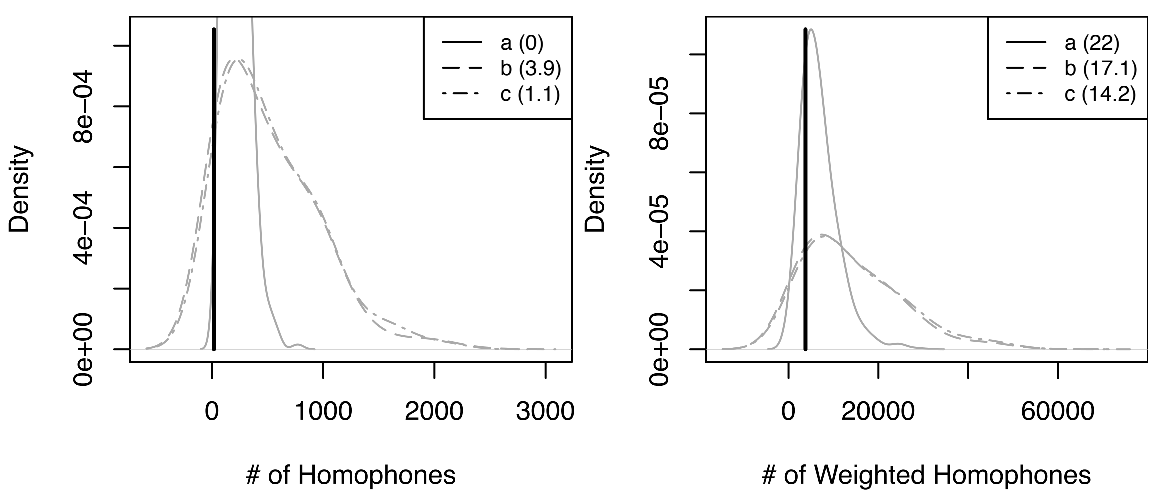
{/C_1 + B/, /C_2 + B/, /C_3 + B/}

A = {⟨k, kt, lk, ks⟩, ⟨l, lm, lt, ls⟩, ⟨p, lp, pt, ps⟩}

B = {k, n, t, l, m, p, s, ŋ, ∅, ʧ, $ʧ^h$, k^h, t^h, p^h, h, k', t', p', s', ʧ', nh, lh, mh, ŋh, sh, $ʧ^h$h, k^hh, t^hh, p^hh, hh, k'h, s'h}

C = {⟨n, nt, ns⟩}

Figure 19: Amount of homophony in simulation series 6



3.7 Rule 7: Decoronization

- /t/ assimilates in place to a following stop (Sohn 1994, 175)
- Decoronization creates practically no homophony (just one pair of words)
- Seems to be a lower limit: no simulation produced *no* homophones

Figure 20: Homophony after decoronization

Measure	**Total**	**New**
Homophones	6401	2
Weighted	308923	13
Pairs	4500	1
Sets	2872	1

Figure 21: Template for decoronization

$\{/A_1 + B/, /A_2 + B/\}$

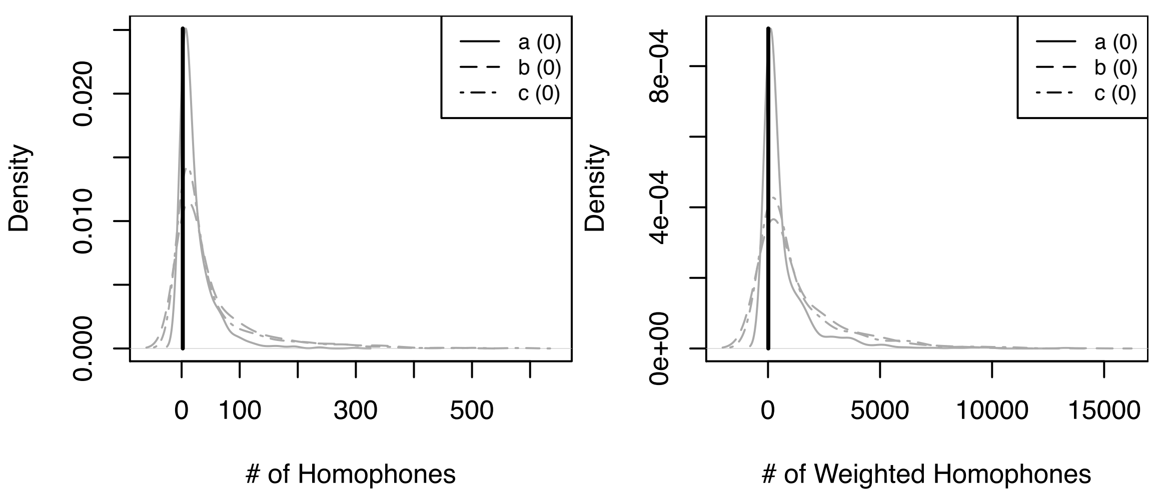
$\{/A_1 + C/, /A_3 + C/\}$

A = {⟨t, k, p⟩}

B = {k, ŋ, k^h, k', ŋh, k^hh, k'h}

C = {m, p, p^h, p', mh, p^hh}

Figure 22: Amount of homophony in simulation series 7



3.8 Rule 8: [h]-Weakening

- Delete /h/ between sonorants
- This rule creates more homophones than any other
- Actual level of homophony always in top half of simulations
- Recurrence of two patterns
  - Phonologically more plausible simulations yield more homophony
  - Percentile rank of actual homophony greater for weighted homophones measure

Figure 23: Homophony after [h]-weakening

Measure	**Total**	**New**
Homophones	6628	227
Weighted	319453	10530
Pairs	4677	177
Sets	2967	95

Figure 24: Template for [h]-weakening

{/A + C_1/, /A + C_2/}

{/B + C_1/, /B + C_2/}

{/B + D_1/, /B + D_2/}

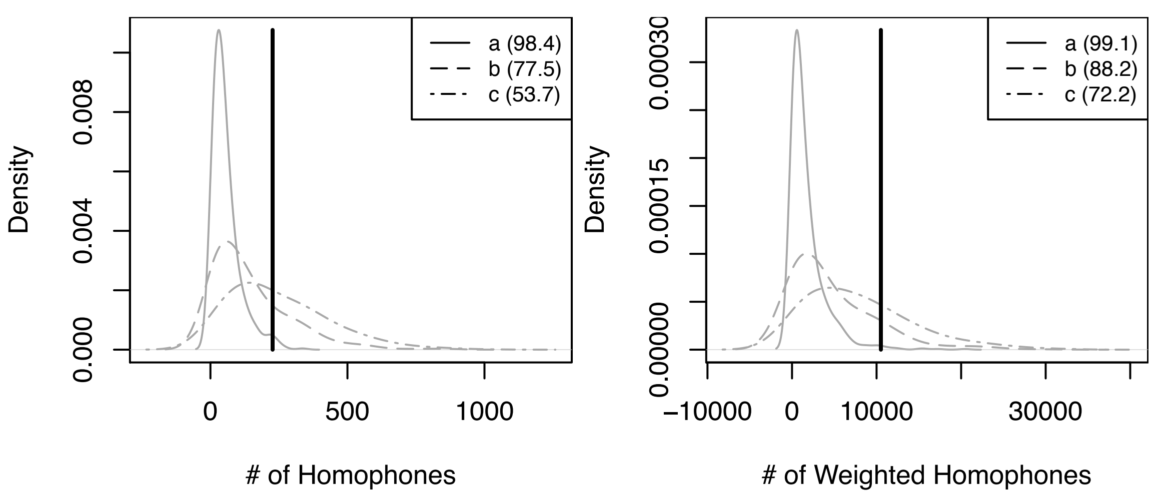
A = {k, t, p, s, n, l, m, ŋ}

B = {∅}

C = {⟨n, nh⟩, ⟨ŋ, ŋh⟩, ⟨l, lh⟩, ⟨m, mh⟩}

D = {⟨∅, h⟩}

Figure 25: Amount of homophony in simulation series 8



3.9 Rule 9: Pre-Tense/Aspirate Reduction

- Delete plain obstruents before homorganic tense or aspirated obstruents (Sohn 1994, 175)
- Actual level of homophony seems to fall in about the middle of simulated patterns
- Many simulations create no homophony
- Again, note greater percentile rank for actual level for weighted homophones measure

Figure 26: Homophony after pre-tense/aspirate reduction

Measure	**Total**	**New**
Homophones	6646	18
Weighted	319835	400
Pairs	4692	15
Sets	2975	8

Figure 27: Template for pre-tense/aspirate reduction

{/A + B/, /C + B/}

{/D + E/, /C + E/}

{/F + G/, /C + G/}

A = {k}

B = {k^h, k', k^hh, k'h}

C = {∅}

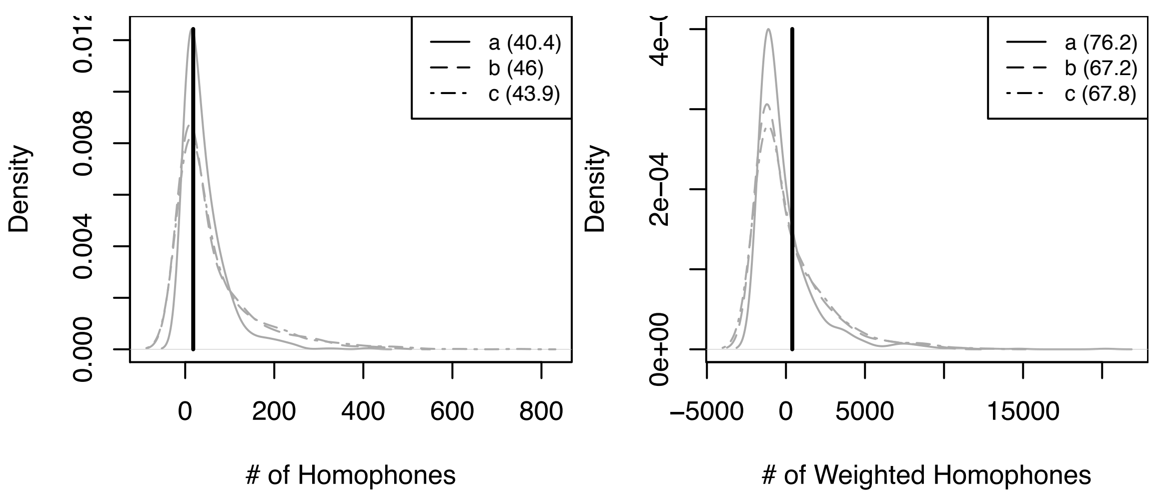
D = {t, s}

E = {$ʧ^h$, t^h, t', s', ʧ', $ʧ^h$h, t^hh, s'h}

F = {p}

G = {p^h, p', p^hh}

Figure 28: Amount of homophony in simulation series 9



4 Discussion

4.1 Does Korean Have Low Homophony?

- Rules 2, 3, 5, 6, and 7 produce less homophony than simulated patterns, as does overall pattern
- But rules 1, 8, and 9 produce more
- To assess overall results across simulations: linear mixed-effects model predicting percentile rank of actual level of homophony for each rule/measure
  - Predictors: measure of homophony, simulation series
  - Random effect of rule
- Two models: one for raw percentiles, one for arcsin-transformed percentiles

  ⇒ Similar results; only raw percentiles presented here

- Results
  - Intercept: 40

    ⇒ Below 50: suggests rules yield less homophony than the median, but not significantly different from 50

  - Weighted homophones measure associated with significantly greater percentiles than other measures ($p = .011$)
  - Percentile rank for 'b'-series simulations significantly less than for 'a'-series ($p = .0014$), for 'c'-series significantly less than for 'b'-series ($p = .0000014$)
- Tentative conclusion: actual level of homophony is indeed low
  - Low intercept in model
  - More rules yield less homophony than their simulations than yield more
  - Overall pattern has less homophony than simulations

4.2 Possible Mechanisms of Homophony Avoidance

- As discussed above, we don't want to build homophony avoidance (as opposed to neutralization avoidance) into formal phonological patterns
- These results suggest only a gradient avoidance of homophony
- How might this situation come about? Possibilities:

1. Given phonetic precursor to a rule, the less homophony the rule would create, the more likely the precursor is to be phonologized
2. Rules tend to neutralize contrasts that are already perceptually suboptimal; lexicon is already optimized to avoid homophones based on hard-to-perceive contrasts
3. Words in dense neighborhoods tend to resist alternation (Ussishkin and Wedel 2009)

4.3 Other Patterns in the Data

- The more phonologically plausible the rule, the more homophony it creates
  - If true, this trend might argue against explanation 2 above
  - Caveat: most simulated rules still not very plausible
  - Future research: how to better filter rules for phonological plausibility?
- Level of actual homophony looks less surprisingly low when homophones are weighted by frequency
  - In other words, few words are homophonized, but they tend to be especially frequent
  - Possible explanation: short words more likely both to be homophonized *and* to be frequent
    - ⇒ Unlikely: should be just as true of simulated patterns as of actual patterns
  - Looks like an anti-functional tendency

5 Conclusion

- Neutralizing alternations in Korean appear to produce less homophony than expected
- Thus, phonological rules may be sensitive to contrast among actual words, not just potential ones

References

Adam Albright and Yoonjung Kang. Predicting innovative alternations in Korean verb paradigms. In *Current Issues in Unity and Diversity of Languages: Collection of the Papers Selected from the CIL 18, Held at Korea University in Seoul*, pages 1–20, Seoul, Korea, 2009. Linguistic Society of Korea.

Edward S. Flemming. *Auditory Representations in Phonology.* Outstanding Dissertations in Linguistics. Routledge, New York, NY, 2002.

Naomi Gurevich. *Lenition and Contrast: The Functional Consequences of Certain Phonologically Conditioned Sound Changes*. Outstanding Dissertations in Linguistics. Garland Publishing, New York, NY, 2004.

Yongeun Lee. *Sub-Syllabic Constituency in Korean and English*. PhD thesis, Northwestern University, 2006.

Jaye Padgett. Contrast and post-velar fronting in Russian. *Natural Language and Linguistic Theory*, 21(1):39–87, 2003.

Jaye Padgett and Marija Tabain. Adaptive dispersion theory and phonological vowel reduction in Russian. *Phonetica*, 62:14–54, 2005.

Daniel Silverman. Neutralzation and anti-homophony in Korean. *Journal of Linguistics*, to appear.

Ho-Min Sohn. *Korean*. Descriptive Grammars. Routledge, New York, NY, 1994.

Adam Ussishkin and Andrew Wedel. Lexical access, effective contrast, and patterns in the lexicon. Ms., University of Arizona, 2009.