Phonology **32** (2015). Supplementary materials

Saltation and the P-map

Bruce Hayes
University of California, Los Angeles

James White
University College London

Supplementary materials

Tableaux for the model incorporating geminate inputs

1 Purpose

The paper gives a simplified analysis, focusing on the pattern of saltation for single-consonant inputs. Here we incorporate the geminate inputs /apːa/ and /abːa/. For the latter, there are two possible outputs, [abːa] and [aβa].We first analyse each with separate grammars (in order to obtain diagnostic tableaux); these were obtained using the Constraint Demotion algorithm in OTSoft 2.3.3. We then combine the two grammars into a single grammar that generates free variation (§4), and lastly check that this grammar performs correctly (§5).

2 Grammar generating the output [aβa] from /abːa/

(1)

Stratum	*Ranking*
1	*Map(b, β), *Map(pː, β)
2	*Map(pː, b), *V[−cont]V
3	*Map(pː, bː), *β, *Map(pː, p), *Map(bː, β)
4	*V[−voice]V, *Map(bː, b), *Map(p, β)
5	*Map(p, b)
6	*b

(2)

	/apa/	*Map(b,β)	*Map(p,β)	*Map(p,b)	*V[−cont]V	*Map(p,b;)	*β	*Map(p,p)	*Map(b,β)	*V[−vce]V	*Map(b,b)	*Map(p,β)	*Map(p,b)	*b
a.	i. apa				*!					*				
	ii. aba				*!								*	*
	☞ iii. aβa						*					*		
b.	/aba/													
	i. apa				*					*!			*	
	☞ ii. aba				*									*
	iii. aβa	*!					*							
c.	/aβa/													
	i. apa				*!					*		*		
	ii. aba	*!			*									*
	☞ iii. aβa						*							

(3)

	/pa/	*Map(b,β)	*Map(p,β)	*Map(p,b)	*V[−cont]V	*Map(p,b;)	*β	*Map(p,p)	*Map(b,β)	*V[−vce]V	*Map(b,b)	*Map(p,β)	*Map(p,b)	*b
	☞ i. pa													
	ii. ba												*!	*
	iii. βa						*!					*		
b.	/ba/													
	i. pa												*!	
	☞ ii. ba													*
	iii. βa	*!					*							
c.	/βa/													
	☞ i. pa											*		
	ii. ba	*!												*
	iii. βa						*!							

(4) a.

/ap:a/	*Map(b,β)	*Map(p,β)	*Map(p,b)	*V[−cont]V	*Map(p,b:)	*β	*Map(p,p)	*Map(b,β)	*V[−vce]V	*Map(b,b)	*Map(p,β)	*Map(p,b)	*b
☞ i. ap:a				*					*				
ii. apa				*			*!		*				
iii. ab:a				*	*!								*
iv. aba			*!	*									*
v. aβa		*!				*							
b. /ab:a/													
i. ab:a				*!									*
ii. aba				*!						*			*
☞ iii. aβa						*		*					

The tableaux in (5) are not real data; rather, they illustrate made-up tableaux fed into the software so that it would properly enforce P-map-compliant rankings for *Map constraints whenever contradictory data did not exist.

(5) a.

/EnforceP-map/	*Map(b,β)	*Map(p,β)	*Map(p,b)	*V[−cont]V	*Map(p,b:)	*β	*Map(p,p)	*Map(b,β)	*V[−vce]V	*Map(b,b)	*Map(p,β)	*Map(p,b)	*b
☞ Winner												*	
Loser											*!		
b. /EnforceP-map/													
☞ Winner										*			
Loser			*!										
c. /EnforceP-map/													
☞ Winner					*								
Loser			*!										
d. /EnforceP-map/													
☞ Winner												*	
Loser			*!										
e. /EnforceP-map/													
☞ Winner							*						
Loser			*!										
f. /EnforceP-map/													
☞ Winner			*										
Loser		*!											

g.

/EnforceP-map/	*Map(b,β)	*Map(p:,β)	*Map(p:,b)	*V[−cont]V	*Map(p:,b:)	*β	*Map(p:,p)	*Map(b:,β)	*V[−vce]V	*Map(b:,b)	*Map(p,β)	*Map(p,b)	*b
☞ Winner							*						
Loser		*!											

h.

/EnforceP-map/													
☞ Winner								*					
Loser		*!											

i.

/EnforceP-map/													
☞ Winner										*			
Loser								*!					

3 Grammar generating the output [abːa] from /abːa/

(6)

Stratum	*Ranking*
1	*Map(b, β), *Map(pː, β)
2	*Map(pː, b), *Map(bː, β)
3	*V[−cont]V, *Map(pː, p), *Map(bː, b), *Map(pː, bː)
4	*β, *V[−voice]V
5	*Map(p, β)
6	*Map(p, b)
7	*b

(7) a.

/apa/	*Map(b,β)	*Map(p:,β)	*Map(p:,b)	*Map(b:,β)	*V[−cont]V	*Map(p:,p)	*Map(b:,b)	*Map(p:,b:)	*β	*V[−vce]V	*Map(p,β)	*Map(p,b)	*b
i. apa					*!					*			
ii. aba					*!							*	*
☞ iii. aβa									*		*		

b.

/aba/													
i. apa					*					*!		*	
☞ ii. aba					*								*
iii. aβa	*!								*				

c.

/aβa/													
i. apa					*!					*	*		
ii. aba	*!				*								*
☞ iii. aβa									*				

(8)

	/pa/	*Map(b,β)	*Map(p,β)	*Map(p,b)	*Map(b,β)	*V[−cont]V	*Map(p,p)	*Map(b,b)	*Map(p,b)	*β	*V[−vce]V	*Map(p,β)	*Map(p,b)	*b
a.	☞ i. pa													
	ii. ba												*!	*
	iii. βa									*!		*		
b.	/ba/													
	i. pa												*!	
	☞ ii. ba													*
	iii. βa	*!								*				
c.	/βa/													
	☞ i. pa											*		
	ii. ba	*!												*
	iii. βa									*!				

(9)

	/apːa/	*Map(b,β)	*Map(pː,β)	*Map(pː,b)	*Map(bː,β)	*V[−cont]V	*Map(pː,p)	*Map(bː,b)	*Map(pː,bː)	*β	*V[−vce]V	*Map(p,β)	*Map(p,b)	*b
a.	☞ i. apːa					*					*			
	ii. apa					*	*!				*			
	iii. abːa					*			*!					*
	iv. aba			*!		*								*
	v. aβa		*!							*				
b.	/abːa/													
	i. abːa					*								*
	ii. aba					*		*!						*
	☞ iii. aβa				*!					*				

Like (5), the tableaux in (10) are not real data; they again illustrate made-up tableaux fed into the software so that it would properly enforce P-map-compliant rankings for *Map constraints whenever contradictory data did not exist.

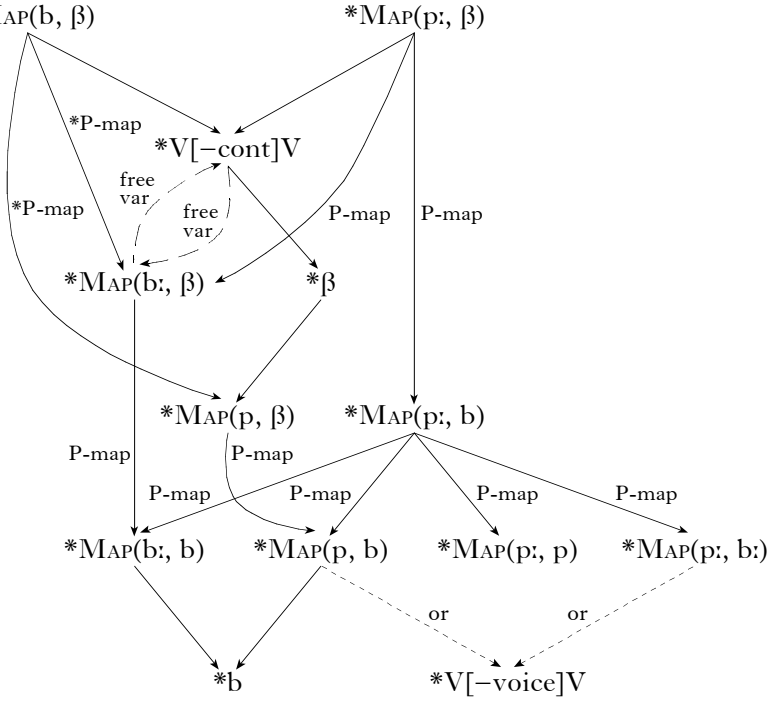
(10)

	/EnforceP-map/	*Map(b,β)	*Map(p,β)	*Map(p,b)	*Map(b,β)	*V[−cont]V	*Map(p,p)	*Map(b,b)	*Map(p,b.)	*β	*V[−vce]V	*Map(p,β)	*Map(p,b)	*b
a.	☞ Winner												*	
	Loser											*!		
b.	/EnforceP-map/													
	☞ Winner							*						
	Loser			*!										
c.	/EnforceP-map/													
	☞ Winner								*					
	Loser			*!										
d.	/EnforceP-map/													
	☞ Winner												*	
	Loser			*!										
e.	/EnforceP-map/													
	☞ Winner						*							
	Loser			*!										
f.	/EnforceP-map/													
	☞ Winner			*										
	Loser		*!											
g.	/EnforceP-map/													
	☞ Winner						*							
	Loser		*!											
h.	/EnforceP-map/													
	☞ Winner				*									
	Loser		*!											
i.	/EnforceP-map/													
	☞ Winner							*						
	Loser				*!									

4 Hasse diagram incorporating both possible outputs for /ab:a/

We produced the diagram in (11) by running the Fusional Reduction algorithm (Brasoveanu & Prince 2011) on both of our input files, then collating the pairwise rankings obtained. There is only one case of required differential ranking, shown by the dashed arrows labelled 'free var'. This free ranking obtains the observed free variation between [ab:a] and [aβa] for /ab:a/.

(11)



5 Making sure the final grammar works

To verify that this grammar is adequate, we used rankings to create an analogous grammar in the framework of Stochastic OT (Boersma 1998, Boersma & Hayes 2001). Specifically, for any two constraints that are in a ranking relationship in the Hasse diagram, there is a ranking value difference of at least 20. Moreover, where there is free ranking (one single pair of constraints), the ranking values are the same. The values in Table I satisfy these criteria.

constraint	ranking value
*V[−cont]V	100
*V[−voice]V	20
*β	80
*b	20
*Map(p, β)	60
*Map(p, b)	40
*Map(b, β)	120
*Map(pː, p)	40
*Map(pː, bː)	40
*Map(pː, b)	60
*Map(pː, β)	120
*Map(bː, b)	40
*Map(bː, β)	100

Table I
An adequate final grammar

Using OTSoft 2.3.3, we tested this grammar by running it on 100,000 inputs for each candidate. For the free variation pair /abːa/ → [abːa, aβa], this generated 50,094 [abːa] outputs and 49,906 [aβa] outputs. For all other inputs, the correct output was generated 100,000 times. We conclude that our grammar generates the intended output forms correctly.

ADDITIONAL REFERENCES

Boersma, Paul (1998). *Functional phonology: formalizing the interactions between articulatory and perceptual drives*. PhD dissertation, University of Amsterdam.

Boersma, Paul & Bruce Hayes (2001). Empirical tests of the Gradual Learning Algorithm. *LI* **32**. 45–86.