[k^ji]? Eek!

What happened last time…

Paco: What about the chain <ha:kimi:n, ha:k^jimi:n, ha:k^jmi:n, ha:kmi:n>? Depalatalization in final step improves markedness (and input-relative faithfulness).

Me: That chain isn't valid because gradualness has two clauses, and it fails the second one: (i) chains can't add more than one unfaithful mapping at a time, and (ii) chains can't take away unfaithful mappings. All faithfulness is input-relative.

Susie: Why isn't faithfulness, like harmony, determined relative to the immediately preceding member of the chain?

Me: Because I said so, that's why.

Chorus: No, no, no!

Adam: I think it works anyway, even if faithfulness is predecessor-relative instead of input-relative. Paco's chain <ha:kimi:n, ha:k^jimi:n, ha:k^jmi:n, ha:kmi:n> does worse on PREC(ID(back), MAX) than the intended winner <ha:kimi:n, ha:k^jimi:n, ha:k^jmi:n>.

Me: I need to think about that.

Gunnar: If palatalization is allophonic (Me: It is), then what about Richness of the Base. Won't /ha:k^jimi:n/ end up as *[ha:kmi:n]?

Me: I need to think about that too.

The Results of Cerebration

(1) Adam is correct. Clause (ii) of gradualness is not necessary here. We don't need to declare Paco's chain invalid because it's disfavored by PREC(ID(back), MAX) anyway. Compare:

<table>
<tr><th></th><th>PREC(ID(back), MAX)</th><th></th></tr>
<tr><td><ha:kimi:n, ha:k^{j}imi:n, ha:k^{j}mi:n>
<ID(back), MAX></td><td></td><td>Opaque winner.</td></tr>
<tr><td><ha:kimi:n, ha:kmi:n>
<MAX></td><td>*!</td><td rowspan="2">Transpa-rent losers.</td></tr>
<tr><td><ha:kimi:n, ha:k^{j}imi:n, ha:k^{j}mi:n, ha:kmi:n>
<ID(back), MAX, ID(back)></td><td>*!</td></tr>
</table>

Why? PREC(ID(back), MAX) says that every MAX violation is preceded by some ID(back) violation and not followed by any ID(back) violations. It assigns one violation for nonconformity with each of these requirements.

(2) Richness of the base presents no difficulties. Here's the range of logically possible inputs and what we'd like them to map to (assuming syncope):

/ki/ [k^j]
/ku/ [k]
/kji/ [k^j]
/kju/ [k]

The relevant part of the grammar is:

*ki, PREC(ID(back), MAX) >> *k^j >> ID(back)

The net effect of PREC(ID(back), MAX) >> *k^j is that the distribution of [k^j] is determined on pre-syncope representations. Syncope can bleed palatalization or feed depalatalization.

/ki/	*ki	PREC	*k^j	ID(back)
a. → <ki, k^ji, k^j> <ID, MAX>			*	*
b. <ki, k> <MAX>		*!		
c. <ki, k^ji, k^j, k> <ID, MAX, ID>		*!		

For /ku/, see pp. 27-28.

/k^ji/	*ki	PREC	*k^j	ID(back)
a. → <k^ji, k^j>			*	
b. <k^ji, k^j, k>		**!		*

/k^ju/	*ki	Prec	*k^j	Id(back)
a. → <k^ju, ku, k> <Id, Max>				*
b. <k^ju, k^j, k> <Max, Id>		**!		*
c. <k^ju, k^j> <Max>		*!	*	

OT-CC & Opacity, Reduced to the Essentials

In the final EVAL, after the candidate set is complete …

> Markedness constraints care about the *form* of the output, as usual. (Output = end of chain form (cf. Sara's question).)
>
> Faithfulness constraints care about *which* unfaithful mappings relate input to output, as usual.
>
> PREC constraints care about the *order* of the unfaithful mappings.

In OT, the "natural" order of the unfaithful mappings is the transparent one — the order that gives a result that best satisfies the markedness and faithfulness constraints as ranked in the language in question.

> E.g., Arabic <ktub, uktub, ?uktub> is the natural order because [?uktub] best satisfies the hierarchy ONSET, *COMPLEX-ONSET >> DEP.

If natural orders always won, we wouldn't need PREC constraints. (We might still want chains because of their other salubrious effects.)

PREC constraints are crucial when they favor nonnatural orders. They can produce a winner that would otherwise lose on markedness and/or faithfulness grounds. Crucial activity by PREC constraints signals a failure of pure markedness/ faithfulness OT to get the right result.

E.g., opaque <gabr, gabur> beats less marked <gabr, gabur, gibur> because no epenthesis mapping can precede any raising mapping according to PREC(ID(low), DEP).

E.g., opaque <ha:kimi:n, ha:k^{j}imi:n, ha:k^{j}mi:n> beats more faithful <ha:kimi:n, ha:kmi:n> because any syncope mapping should be preceded by some palatalization mapping according to PREC(ID(back), MAX).

The conditions on chain well-formedness are necessary elements of this scheme:

Harmonic improvement says that chains must monotonically increase in harmony, according to the language's constraint hierarchy. This means that PREC constraints can never force unfaithful mappings that are otherwise impossible in the language. If Arabic didn't have palatalization, PREC(ID(back), MAX) couldn't compel it.

Gradualness says that chains change by one unfaithful mapping at a time. This ensures that there is always *some* ordering (and therefore something for PREC to chew on) whenever more than one unfaithful mapping links input and output.

How to do analyses in OT-CC

Singaporean English (Mohanan, Anttila)

/his-z/ his@s (@=schwa)
*his@z

What's the opacity? *[his@z] is more faithful because it doesn't show the effects of voicing assimilation. Voicing assimilation in [his@s] seems gratuitous, since @ epenthesis kills two birds with one stone: it resolves the forbidden near-geminate cluster and it eliminates the need for voicing assimilation.

What are the relevant chains?

Even without a ranking, we know the language has two unfaithful mappings, devoicing and epenthesis, and we have a rough idea of where they occur. We can try every combination and ordering of them (brute force analysis):

*<hisz>	Faithful loser (bad cluster).
*<hisz, hiss>	Loser (bad cluster).
*<hisz, his@z>	Transparent loser.
<hisz, hiss, his@s>	Opaque winner.
**<hisz, his@z, his@s>	Invalid. Why?

Since we know that <hisz, hiss, his@s> is the winner, we can use it and the harmonic improvement requirement to make some deductions about the constraints and their ranking. (We can also do some of this using transparent phonology.)

hiss > hisz shows that AGREE(voice) >> ID(voice).

his@s > hiss shows that OCP >> DEP (Anttila).

Opacity problem: *<hisz, his@z> satisfies all constraints except DEP. <hisz, hiss, his@s> also violates ID(voice). Why?

To get opaque <hisz, hiss, his@s> to beat transparent *<hisz, his@z>,

<hisz, hiss, his@s>'s worst nonshared violation needs to be dominated. So we infer PREC >> ID(voice).

Which PREC is responsible?

Rewrite the chains as sequences of unfaithful mappings:

<hisz, hiss, his@s> = <ID(voice), DEP-V>

*<hisz, his@z> = <DEP-V>

The PREC constraint must prefer the chain where the DEP-V violation is preceded by an ID(voice) violation. So it's PREC(ID(voice), DEP-V). (Recall: PREC(A, B) says "For any B, there exists preceding A and there does not exist following A".)

	/his-z/	OCP	AGR(vce)	DEP-V	PREC(ID(vce), DEP-V)	ID(vce)
a. →	<hisz, hiss, his@s> <ID(voice), DEP-V>			*		*
b.	<hisz, his@z> <DEP-V>			*	*!	
c.	<hisz> < >	*!	*!			
d.	<hisz, hiss> <ID(voice)>	*!				*

Child 209 (Dinnsen & Farris)

Features assumed:

	[grooved]	[continuant]
s	+	+
th	-	+
t/d	-	-

Gradualness: change only one feature value at a time. /s/ -> [t] or /s/ -> [f] mappings involve two feature changes. (Local conjunction, comparative markedness, and sympathy all presuppose the same thing.)

/maws/	[mawth]	Id(grooved) violated.
/mawth/	[mawf]	Id(place) violated.

Candidate chains

mouse

*<maws>	
<maws, mawth>	Opaque winner.
*<maws, mawth, mawf>	Transparent loser.

mouth

*<mawth>	
<mawth, mawf>	Transparent winner.

Basic ranking (from Dan & Ashley's handout)

*s >> ID(grooved)
*th >> ID(place)
*s >> *th (needed for /maws/ -> [mawth] to improve)

How must the PREC constraint be ranked?

> It needs to dominate the opaque winner <maws, mawth>'s worst violation that's not shared with the transparent loser *<maws, mawth, mawf>. That's <maws, mawth>'s *th violation.

Which PREC constraint is responsible?

> First rewrite chains as sequences of unfaithful mappings:

mouse

<maws>	< >
<maws, mawth>	<ID(grooved)>
*<maws, mawth, mawf>	*<ID(grooved), ID(place)>

mouth

<mawth>	< >
<mawth, mawf>	<ID(place)>

PREC needs to favor the *mouse* candidate <maws, mawth> over *<maws, mawth, mawf> and not disfavor the *mouth* candidate <mawth, mawf> relative to <mawth> (since it dominates *th). PREC(ID(place), ID(grooved)) does the job.

	/maws	*s	Id(grv)	Prec(Id(plc), Id(grv))	*th	Id(plc)
a. →	<maws, mawth> <Id(grv)>		*	*	*	
b.	<maws, mawth, mawf> <Id(grv), Id(plc)>		*	**!		*
c.	<maws> < >	*!				

What does Prec(Id(place), Id(grooved)) say intuitively?

You can't change the place of an original /s/.

Stopping of initial fricatives by Child 209

/thum/	[pum]
/sun/	[tun]

(voicing ignored)

Additional basic ranking

*#FRIC >> ID(continuant)

What are the chains?

thumb

<thum>	< >
<thum, tum>	<ID(cont)>
<thum, fum, pum>	<ID(place), ID(cont)>

sun

<sun>	<>
<sun, thun>	<ID(grooved)>
<sun, thun, tun>	<ID(grooved), ID(cont)>
<sun, thun, fun>	<ID(grooved), ID(place)>
<sun, thun, fun, pun>	<ID(grooved), ID(place), ID(cont)>

(**<sun, tun> isn't gradual under the Dinnsen/Farris featural assumptions since it changes both [grooved] and [cont].)

The chains that most claim our attention are those that satisfy undominated *#FRIC:

*<thum, tum>	<ID(cont)>
vs.	
<thum, fum, pum>	<ID(place), ID(cont)>
<sun, thun, tun>	<ID(grooved), ID(cont)>
vs.	
*<sun, thun, fun, pun>	<ID(grooved), ID(place), ID(cont)>

How much is accounted for by the analysis we already have? Is an additional PREC constraint required? If so, what is it and how is it ranked?

The highest ranking constraint that distinguishes <sun, thun, tun> vs. *<sun, thun, fun, pun> is PREC(ID(place), ID(grooved)). ID(place) also distinguishes them. Both constraints have the right favoring relation.

<thum, fum, pum> vs. *<thum, tum> requires PREC(ID(place), ID(cont)). It has to dominate ID(place). It has to be dominated by PREC(ID(place), ID(grooved)) because it would otherwise wrongly favor *<sun, thun, fun, pun>

		*s	Id(grv)	Pr(Id(plc), Id(grv))	Pr(Id(plc), Id(cnt))	*th	Id(plc)
a. →	<thum, fum, pum> <Id(plc), Id(cnt)>						*
b.	<thum, tum> < Id(cnt)>				*!		
c. →	<sun, thun, tun> <Id(grv), Id(cnt)>		*	*	*		
d.	<sun, thun, fun, pun> <Id(grv), Id(plc), Id(cnt)>		*	**!			*

Relation of OT-CC to Sympathy Theory

OT-CC partly resembles but also supplants Sympathy.

Main elements of Sympathy:

Sympathetic candidate: most harmonic candidate that satisfies some designated faithfulness constraint, called the selector.

Sympathy constraint: requires winner to have all/some of sympathetic candidate's unfaithful mappings.

Relation to OT-CC:

Sympathetic candidate ~~ middle of the chain. More faithful in some respect than the ultimate output.

Sympathy constraint ~~ gradualness requirement plus PREC constraints.

Why reject Sympathy?

Argument on pp. 11-12. Sympathy has big problems with multiple opaque interactions in a language.