Fixed Ranking and the 'Too Many Solutions' Problem

Paul de Lacy
University of Cambridge

CASTL Kick-Off Conference
Friday, October 3, 2003

• *I would like to thank the members of CASTL and organizers of the conference for inviting me to speak here* ***and*** *most of all for giving me the last position on Friday − I'm hoping that by now you're all too exhausted to object to anything that I say.*
• *As a quick preamble to this talk, I think I was asked to speak here because two of the conference's topics are 'universal scales in phonology' and 'segment strength and positional neutralization'. I wrote about both these issues in my doctoral dissertation (which is on the ROA), and I'm still writing about them − in a book for Cambridge University Press, which will be out in mid 2005 [I'm sorry, but I had to plug that − please buy it − I have a massive student loan]. Anyway, I feel a twinge of conscience because I'm not going to talk about scales or neutralization today, at least not as the main topic. Mainly because I'm sick of hearing myself talk about it.*

1m 30

1. Too Many Solutions

What I am going to talk about is a major problem facing Optimality Theory, which I'm going to call the 'Too Many Solutions Problem'.

This problem occurs when a restriction is only ever repaired by a proper subset of possible repairs. For example, a ban on voiced coda obstruents could in principle be repaired by ***deletion*** *of the coda obstruent,* ***epenthesis*** *of a vowel (thus turning the obstruent into an onset), or* ***neutralization*** *to [-voice]. However, Lombardi (1995) has claimed that only neutralization ever happens.*

Other examples are given in (2). In (2b), bans on feature value disagreement are always dealt with by assimilation, or in Trubetzkoy's terminology 'assimilative neutralization'. Although I'm not going to talk about tone today, I think tone holds the same sort of problems: namely that tonal conditions can do whatever they want to tones, but they never mess with tonal sponsors − Tone-bearing units.

(1) The *Too Many Solutions* Problem
• Some problems are only ever resolved by a proper subset of possible solutions.
References: Lombardi (1995), Wilson (2000), Pater (2001)

(2) Examples
(a) Ban on voiced codas: resolved by neutralization; never by deletion or epenthesis (Lombardi 1995)
/ab/ → [ap]; ☠[a], ☠[a.bi]

(b) Requirement that nearby vowels agree in nasality: resolved by harmony; never by deletion (Pater 2001/2003)
/mãwa/ → [mãwã]; ☠[mãw], ☠[mãã], etc.
(c) Tonal conditions can be resolved by changes in *tone* (change, shift, deletion, epenthesis), but never by changes in their sponsors.
/ábá/ → [ábà] (by OCP-H); ☠[áb], ☠[ábì̱á]

To add to our problems, I want to point out that there are two further problems that are related to the 'Too Many Solutions' problem.

I've been using the term 'solution' to mean, roughly, 'process'. But, as (3) points out, there are restrictions even within processes. For example, Wilson has explored the typology of consonant deletion, and argued that cluster simplification only ever results in one of the possible outputs: the rightmost consonant always remains.

(3) <u>The *Too many subsolutions* problem</u>
Deletion in /C_1C_2/ always produces [C_2], never [C_1] (roughly speaking, see below) (Wilson 2000)

Yet another closely related problem is the 'Too many blockers' problem. As we all know, phonological processes can be prevented from happening if they will inadvertantly violate some other constraint. However, some processes <u>never</u> get blocked by certain types of constraints. In general – and I'll refine this a little later – prosodically motivated processes never get blocked by conditions on subsegmental features. For example, syncope is motivated by constraints on foot form. Importantly, syncope cannot be blocked by – for example – a requirement that adjacent obstruents agree in voicing.

(4) <u>The *Too many blockers* Problem</u>
Many prosodic processes are never blocked by feature-based conditions
e.g. AGREE[voice] never blocks syncope:
/abiga/ → [abga], but /apiga/ → [apiga] because AGREE[voice] blocks *[apga].
(cf *prosodic conditions*, which can block syncope)

So, why bother talking about the problem, and what justifies calling it a 'major problem for OT'? Well, to be accurate, it's a problem facing any constraint based theory that decouples problems from solutions/constraints from repairs. This is ironic, because one of the major achievements of constraint-based theories is decoupling problems from solutions; rule-based theories fail to provide a formal link between cases where the same problem is resolved by many solutions.

(5) <u>Who faces this problem?</u>
• Potentially any theory that decouples problem from solution (i.e. almost all constraint-based theories)

In contrast, such 'heterogeneity of process, homogeneity of target' cases are what OT does well. However, the Too Many Solutions problem underscores that OT in its current conception does it <u>*too*</u> *well. We're faced with the reverse of the opacity problem: when it comes to 'heterogeneity of process, homogeneity of target', Rule-based theories are too restrictive, and OT is too powerful.*

By the way, there are three major works on this topic that I know of: Lombardi's paper, which focuses on Place and Voice; Colin Wilson's dissertation (2000), and Joe Pater's manuscript (2003). I'll be mentioning these works a lot.

I have two aims today. One is to identify the right empirical generizations about the Too Many Solutions problem and the related problems. The other is to explore a particular type of solution to the problem that hasn't been discussed in great deal for these particular problems: i.e. fixed ranking.

(6) <u>Issues</u>
(a) What generalizations lie behind these problems?
(i) Not hitherto addressed in a general way.
(b) How to do deal with them.

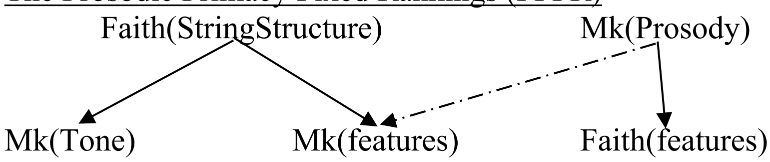
To give a brief preview of the results of my aims, I'm going to argue that the generalizations are (roughly) those in (7a). To generalize, constraints on subsegmental features are 'weak' – they can't affect prosodic structure or string structure – by 'string structure' I mean the number of elements in the segmental string and their order. All the problems go away once this observation is expressed correctly.

(7) <u>Aims</u>
(a) To identify the right generalizations [i.e. answer (6a)]
• Feature conditions cannot affect string structure or prosody
• Tone conditions cannot affect string structure; can affect prosody
• Sonority and prosodic conditions can affect string structure and prosody

I'm going to try to express this observation by using a very traditional method – through imposing fixed rankings between classes of constraints.
The fixed rankings I'm going to argue for are in (8).

(b) To provide a solution [i.e. answer (6b)]
☞ Universally fixed rankings
‼ Important precursor: Pater 2003 ‼

(8) <u>The Prosodic Primacy Fixed Rankings (PPFR)</u>



(i) Note: Faith(StringStructure) »» Mk(features) builds on Pater's fixed ranking of Faith(StringStructure) » Assimilation markedness constraints

(9) Structure
§2 The generalizations
§3 The proposal
§4 Case Studies & Predictions
§5 Alternatives and Problems
§6 Summary

8m

2. Generalizations

Let's move on to section 2.

My aim in this section is to make some 'pretheoretical' generalizations about the issue that lies behind the Too Many Solutions problems and its friends.

I should point out right from the start that this is where I and previous work differ fundamentally.

The previous approaches – Wilson, Lombardi, and Pater – effectively imply that we cannot predict whether a particular type of problem will be able to provoke a particular type of solution. These are because the theories – especially Lombardi's and Wilson's – are at least presented as 'local'. Lombardi's theory, for example, claims that different problems may be expressed by fundamentally different constraints, and that constraint form is at the root of their differences; but there is no way to predict for a particular problem what form its constraints will take. Similarly, it isn't clear in Wilson's theory which constraints are Targeted (maybe they all are, maybe not).

In the extreme is Joe Pater, who makes this point so well I've called it 'Pater's Pessimism', in (10).

The quotes in (10) will make more sense as we proceed. The gist of them is that it's hopeless to try and make generalizations about types of constraints and hope that that's all we have to say. In (10b), Pater basically says that we have to take everything on a case by case basis; he says that the solution to the Too Many solutions problem is to impose fixed rankings, but you can do that on a case by case basis.

(10) Pater's (semi-)Pessimism (Pater 2003: 16)
(a) "This observation [i.e. re. Fixed Ranking] however, is unlikely to form the basis of a workable account of possible repairs, because there are instances in which the typology of possible repairs for a given constraint often crosscuts the segmental/featural divide."
(b)"… not only are there broad schemas [like the ones proposed] … but there are also more parochial fixed rankings between particular markedness and faithfulness constraints."

Contrary to previous proposals, I want to suggest that we can make predictions about which type of constraints provoke which types of solutions. The predictions rely on recognizing certain distinctions in constraints. The distinctions – laid out in (11) – are very traditional, so I'm sure they hold no surprises.

I think there has always been at least an intuition in phonology that the categories in (11) are independent; this was formally expressed in autosegmental phonology by putting these things – well, at least tone, prosodic structure, and subsegmental features – on different planes.

(11) Representational Categories
(a) String Structure (= the number of segments in a candidate and their order)
(b) Sonority
(c) Prosodic Structure
(e) Tone
(f) Features (= subsegmental features)
❶ A very traditional division, except for the 'sonority-valency' distinction, which is usually packaged in one.
❷ cf 'autosegmental planes' of the 1980's.

(12) Related constraints
(a) StrStruc: MAX, DEP, LINEARITY, CONTIGUITY (M&P 1995)
(b) Sonority: sonority-distance constraints
(c) Prosodic Structure: ONSET, NOCODA, FTBIN, etc.
(d) Tone: OCP(H), etc.
(e) Features: *+voice, AGREE[voice], etc.

In (13) I have attempted to summarize the generalizations I'm going to discuss in greater detail in a moment. The table pairs up types of conditions with whether they can affect certain types of structure.

As a reminder, I use the term 'string' here to refer to the fundamental properties of a candidate's root nodes. In other words – how many root nodes there are and their order.

(13) Direct Influence Generalizations

	Structure				
Constraints	String	Prosody	Tone	Sonority	Features
Prosodic	✓	✓	✓	✓	✓
Tonal	×	✓	✓	×	×
Sonority	✓	✓	×	✓	✓
Featural	×	×	×	✓	✓

Now, I'm not going to talk about all of the influences expressed in the table; I am only going to pick out the major ones. And they are that:
(1) prosodic constraints can cause deletion and epenthesis; they can influence sonority, and interfere with tones, and – indirectly at least – cause changes in features.

(2) in contrast, featural constraints can affect segmental sonority, but they can't do anything else. Importantly, they can't provoke deletion, epenthesis, or influence prosodic structure.
(3) I'll briefly mention tone at some point.

13m 30

2.1 Prosody and Sonority

Section 2.1 will hopefully clarify what I'm trying to say. There's evidence that prosodic and sonority conditions can do anything.

(14) Prosody and sonority: The heavyweights
Q: What responses can prosodic and sonority conditions force?
A: Anything.

*I'm sure in these examples, I'm not telling you anything new, so I'll move over them quickly. We all know that prosodic constraints like ONSET, NOCODA, *COMPLEX can cause deletion, epenthesis, and feature change, as in the examples in (15).*

(15) Prosody Example
(a) **Deletion** in Ladahki (Koshal 1979) through *COMPLEX
/bcu/ → [cu] 'ten' (cf /ŋə+bcu/ → [ŋəb.cu] '50')
/ʈuk+bcu/ → [ʈuk.cu] '60', *[ʈub.cu]
(b) **Epenthesis** in Harar Oromo (Owens 1985) through *COMPLEX
/dubr+ti/ → [dubarti:] 'girl-nom' (cf [dubr-a])
(c) **Feature change/neutralization**: glide formation through ONSET
/iuniti/ → [juniti] 'unit' (in Niuafo'ou – de Lacy 2000)

Conditions on segmental sonority can do the same thing. By the way, a 'sonority constraint' is one that refers to the sonority of a segment in a particular prosodic position, like Prince & Smolensky's (1993) Nucleus and Margin sonority constraints, or the combination of heads, non-heads, and sonority I've argued for, and Kenstowicz (1994), and Crosswhite (1999) have argued for. In addition, there are constraints on syllable contact and sonority distance.

As with prosodic constraints, sonority constraints can cause deletion, epenthesis, and feature change as you can see in (16).
It is worth pointing out that sonority conditions can also block the formation of prosodic structure. For example, a ban on high sonority onsets in stressed syllables blocks the action of ONSET in Niuafo'ou.

(16) Sonority example
(a) **Deletion** of high vowels in Arabic (Abu-Mansour 1987, Gouskova 2003)
/kibir-at/ → [kibrat] – due to *NUC/{i,u}
(b) **Epenthesis** to avoid 'syllable contact'
Nasal+Sonorant clusters provoke epenthesis in Balantak (Pater 2003)
/mµŋ+memel/ → [meŋememel] 'to cool'

/mµŋ+roŋor/ → [moŋoroŋor] 'to hear'
cf /mµŋ+bala/ → [mambala] 'to fence'

(c) **Feature change**

(i) Nasalization in Dakota codas
/xap/ → [xam] (cf [xa.p-a]) 'to be stripped'
/RED-ʔot-a/ → [ʔon.ʔo.ta] 'be many'

(ii) Vowel sonorization/reduction
(many – Crosswhite 2000?)

(iii) Desonorization in Kazakh (Davis 1998)
/murin-lar/ → [murin-dar] 'noses'

(d) Blocks **prosodic structure**:
blocks glide-formation in Niuafo'ou stressed syllables (de Lacy??)
/wafu/ → [u.á.fu] 'wharf', *[wá.fu], cf [ju.ní.ti] 'unit'

15m

2.2 Features

This leads us to featural conditions. I'm going to stick my neck out and say that no featural condition can force deletion or epenthesis. On top of that, no featural condition can force a change in prosodic structure. Featural conditions cannot even block prosodically-motivated changes. The only thing that featural conditions can do is influence segmental sonority and features.

(17) Features: The featherweight
Q: What responses can featural conditions trigger?
A: (a) Not deletion, epenthesis, etc.
(b) No changes in prosodic structure
(c) Changes in sonority.
(d) Cannot block prosodically-motivated processes

On the sonority front, feature changes like assimilation or harmony can cause segments to change their sonority level. In Korean, for example, stops become nasals under the influence of other nasals; in Dakota, a ban on adjacent coronals forces liquids to become stops.

(18) Force sonority **change**
(a) Voice assimilation; manner assimilation
Korean nasal assim – Cho 1999:88)
/kak+mok/ → [kaŋ.mok] 'stick'
/nap+nita/ → [nam.nita] 'sprout'
(b) Dakota dissimilation: OCP(coronal) forces /l n/ → [k]
/red-zula/ → [zukzula] 'full, filled'
/red-zuta/ → [zukzuta] 'be upright'

In (19) there's an interesting case. In Dakota, codas must be highly sonorous – preferably nasals. However, there is no velar nasal, so underlying /k/ is blocked from *nasalizing by the featural constraint *ŋ. Therefore, a featural condition *ŋ blocks a sonority constraint *coda≤obstruent.*

(19) Can **block** sonority neutralization
(a) Dakota nasalization
*ŋ blocks Dakota coda nasalization (cf examples above)
/ʃok/ → [ʃog] 'thick, solid' (cf [ʃok-a])
(b) Vowel neutralization
IDENT[round] blocks neutralization of /o u/ to [ə] in Blugarian
[rábutə] 'work' cf [rəbótnik] 'worker' (Crosswhite 1999)

So, to summarize, prosodic and sonority constraints can provoke any response. Tone constraints can provoke changes in prosodic structure, but not string structure.
Featural conditions can't provoke changes in anything but sonority.

17m

3. Proposal

So, assuming all these are right, how do we account for them?
In section 3, I make the claim that the generalizations in section 2 fall out from certain universally fixed rankings.
Of course, invoking universally fixed rankings is not a new idea; it's been around right from the start of OT. Most of the time, the fixed ranking has been between constraints derived from the same harmonic scale, though.
Ironically, I've argued that the ranking between constraints from the same harmonic scale cannot be fixed.
However, that doesn't mean that Fixed Ranking is a bad idea. I should admit, though, that it's rare to impose a fixed ranking between markedness and faithfulness constraints, as I'm about to do.

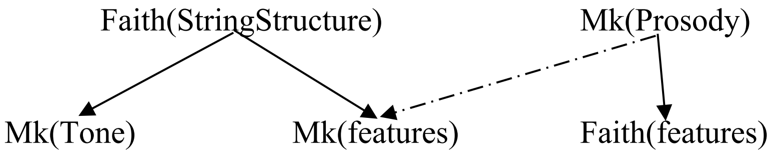
(20) Leading Idea: Universally Fixed Rankings
• Fairly standard response to cases of non-interaction:
Impose a fixed ranking between classes of constraints.
• However, somewhat rare to impose rankings between markedness and faithfulness constraints, or between classes defined on 'features' vs 'prosody'
(i) cf Pater (2001/3), who has has StringStructure faithfulness outranking all assimilation constraints (see below)
(ii) See Pater (2003§3.2) for discussion.
• Other work:
Golston (Syntax » Phonology)
M&P1993 (M » P), PM1, GA
and many others.
Note: I exclude fixed rankings between constraints derived from the same harmonic scale (Prince 1997, de Lacy 2002a)

The fixed rankings I'm going to argue for are given in (21).

The real losers here are constraints that refer to features. These are universally outranked by all faithfulness constraints on string structure – in other words, MAX, DEP, UNIFORMITY, INTEGRITY, and so on. The only faithfulness constraint that doesn't universally outrank feature markedness constraints are feature faithfulness constraints – IDENT, in other words.

Similarly, all prosodic markedness constraints universally outrank all featural markedness constraints. Prosodic markedness constraints are things like ONSET, NOCODA, and so on.

(21) <u>The Prosodic Primacy Fixed Rankings (PPFR)</u>



(a) Faith(StringStructure) = {MAX, DEP, LIN, CONTIG, UNIF, INTEG} (M&P'95)
(b) Mk(Prosody) = all constraints that refer to prosodic elements only (e.g. ONSET, NOCODA; not *CODA/+voice, *not* *ONS≤{sonority level})
(c) Mk(features) = all constraints that refer to a feature
(e.g. *+voice, *CODA/+voice; *not* prosody-sonority constraints)
(d) Mk(Tone) = all constraints that refer to tones.

*Some markedness constraints sort of stand on the border. For example, is *coda/+voice a featural or prosodic constraint? Well, I say that a constraint is only truly prosodic if it doesn't refer to anything but elements from the prosodic hierarchy. This leaves out *coda/+voice, and constraints on prosodic positions and sonority, like P&S's nucleus and margin sonority constraints.*

Now, will these few rankings solve the Too Many Solutions problem and everything else I've just raised?

(22) *Today's primary focus*: Prosody-Features interaction

20m

4. Evidence

Now, section 4. I'm going to focus only on prosodic and featural constraints. By the way, (23) introduces some new symbols. I'll mention these in more detail when we get to them.

Let's move on to section 4.1. I'm going to focus on evidence for the fixed rankings, which are laid out linearly in (24).

(23) <u>Symbols</u>
α «» β = α and β are freely permutable (α » β or β » α)
α »» β = α universally outranks β (i.e. α » β, never β » α)

☞ = winner and attested
☠ = winner and not attested
☹ = perpetual loser

(24) Fixed Rankings: Evidence
(a) Faith(StrStruc) »» Mk(features)
(b) Mk(Prosody) »» Faith(features)
(c) Faith(StrStruc) »» Mk(Tone)

(25) Free Ranking: Evidence (for the interesting cases)
(a) Faith(StrStruc) «» Faith(features)
(b) Mk(Prosody) «» Mk(features)
(c) Sonority-related rankings

4.1 Faith(StrStruc) »» Mk(Features)

4.1.1 Too Many Subsolutions

Let's start in section 4.1.1 with the fixed ranking of string-structure faithfulness over featural markedness constraints, and relate it to the Too Many Subsolutions problem, and therefore the consonant deletion problem raised by Wilson. Wilson's problem was that when deletion applies to intervocalic consonant clusters, its always the first one that gets deleted, and the second that remains.

(26) The C_1C_2 problem (McCarthy 1997, Wilson 2000)
/VC_1C_2V/ → [VC_2V], ☠[VC_1V]
• but featural markedness could favour [VC_1V]!
e.g. /atka/ → [ata] because of *DORSAL
• Further point: /$VC_1C_2C_3V$/ → [$VC_1.C_3V$], *[$VC_2.C_3V$]

(27) Example: Ladahki (Koshal 1979)
(a) /bcu/ → [cu] 'ten' (cf /ŋə+bcu/ → [ŋəb.cu] '50')
(b) /ʈuk+bcu/ → [ʈuk.cu] '60', *[ʈub.cu]

The problem, Wilson points out, is that it's easy in standard OT to get the wrong result. To show this, we need to identify the faithfulness constraint that's in conflict here – it's in (28) – OI-contig-σ effectively penalizes deletion of the leftmost consonant in a cluster because the result of such deletion would be two underlyingly nonadjacent things in the same syllable.

(28) Faithfulness
OI-CONTIG-σ "If (a) *x* is adjacent to *y* in the output
and (b) *x* and *y* are in the same syllable
then *x′* is adjacent to *y′* in the input."
(after Lamontagne 1996, Pater 1997)
(i) This constraint favours /atka/ → [a.ka] over [a.ta]
(ii) There is no faithfulness constraint that favours the opposite.

*When OI-CONTIG-σ is ranked as in (29), the wrong result happens. This is because NOCODA does not specify which consonant to delete, and the first thing to come along that makes a decision is a featural markedness constraint − *dorsal.*

(29) OT without PPFR

/atka/	NOCODA	MAX	*DORSAL	OI-CONTIG-σ
(a) at.ka	*!		*	
(b) a.ka		*	*!	
☠ (c) a.ta		*		*

By the way, the skull and crossbones indicates a possible winner that is never attested.

So, what's wrong with tableau (29)? Well, my take on this is in (30). Suppose a prosodic constraint motivates a process, and that process could happen in several different ways. Well, the choice of what to do is never influenced by featural concerns. It doesn't matter that the first consonant might be featurally excellent − prosodic processes are blind to featural likes or dislikes.

(30) Generalization
If a prosodic constraint can be satisfied by *n* possible resolutions, the winner is never chosen by a featural constraint.

*I guess it's not exactly obvious that the fixed ranking of string-faithfulness over featural markedness gets this generalization. However, (31) illustrates this point. OI-CONTIG-σ is a string faithfulness constraint − it regulates the segmental string's adjacency relations. Therefore, it must outrank *dorsal, and all other such featural constraints.*

In more general terms, faithfulness will always decide which of the two consonants will survive. Because faithfulness outranks featural markedness, feature markedness will never be able to have an effect, therefore.

*The tableau in (31) omdocates that OI-contig-σ always outranks *dorsal. In contrast, NOCODA can be ranked anywhere. As you can see, all possible permutations − after the Fixed Ranking is considered − result in either the faithful candidate or deletion of C_1, never C_2.*

(31) OT with PPFR: Faith(StrStruc) »»» Mk(features) = OI-CONTIG-σ » *DORSAL

/atka/	OI-CONTIG-σ	*DORSAL	NOCODA
☞ (a) at.ka		*	*
☞ (b) a.ka		*	
☹ (c) a.ta	*!		

*Notice that a fixed ranking between string faithfulness and featural markedness is the only way to go here. We can't say that string faithfulness universally outranks feature faithfulness, because feature faithfulness is irrelevant − all candidates satisfy all IDENT constraints. It's also irrelevant to say that prosodic markedness outranks featural markedness. NOCODA could outrank *DORSAL, and we could still get /ata/ winning, as in tableau (29).*

Now, how do we make this result watertight? Well, there are two conditions that hold in order for the Fixed Ranking to get the right result. First of all, for every *prosodically motivated process that admits of several outcomes, there must be some set of string-faithfulness constraints that winnow the relevant candidates down to one tin terms of string structure. After all, featural markedness can only be influential if string-faithfulness constraints leave them options – when we consider all stringStruc-Faithfulness constraints, this is not a difficult condition to require.*

The second condition has to do with featural faithfulness. You will note that I haven't imposed any fixed rankings involving string-faithfulness and featural faithfulness constraints, like IDENT*[voice] or* IDENT*[Place].*

The ranking of featural faithfulness constraints doesn't matter for Wilson's problem as long as no featural faithfulness constraint disfavours deletion of a particular feature. This is exactly what MAX[feature] *constraints do. For example,* MAX*(dorsal) could effectively keep dorsals around no matter whether they're the first or second consonant in the cluster. Personally, I think they don't exist – as I've argued in de Lacy (2002a) – or they are all outranked by string faithfulness constraints.*

(32) Featural Faithfulness

- Could featural *faithfulness* also influence the outcome?

A1: No, if IDENT[f] is the only type of featural faithfulness constraint (de Lacy 2002:xx?, others?Keer?)

From /akta/, IDENT[dorsal] is not violated by either [ata] or [aka].

A2: Yes, if MAX[f] exists.

|| NOCODA » MAX, MAX[dorsal] » OI-CONTIG-σ || will produce /akta/→☠[aka]; [ata] is blocked by MAX[dorsal].

Well, so far so good. It's worth noting, by the way, that this approach makes a few other fun predictions.

One is that C_1 should win in certain conditions. One is with sonority. Sonority constraints should be able to favour the leftmost consonant because some constraints can be ranked anywhere. The other is that you could get coalescence resulting in apparent preservation of C_1. Both of these things happen (contrary to what Wilson (2000) says). I'm not going to dwell on them now, though you're welcome to ask me about them.

(33) Further Predictions

(a) OI-CONTIG-σ favours /ʈuk+bcu/ → [ʈuk.cu]

(b) *Sonority* constraints should be able to influence the choice!

e.g. Sanskrit [<u>cu</u>-ʃcut] cf [<u>ʃu</u>-ʃru:ṣa] cf [a-<u>ti</u>-trasam]

i.e. || *MAR/*sonority* » OI-CONTIG-σ ||

(see Gnanadesikan 1995)

(c) /VC_1C_2V/ should be able to coalesce to [$VC_{1,2}V$]

If coalescence is total, we may get what looks like surface [VC_1V]

e.g. Chipewyan: /Vt+nV/ → [VtV]

(cf /Vt+xV/ → [VkV])

(i) cf Wilson 2000:61)

27m

4.1.2 Too Many Solutions

As it is one of the core cases, I should point out that − in section 4.1.2 − the too many solutions problem for voiced codas can be explained by the same ranking.

As a reminder, Lombardi (1995) claims that a ban on voiced codas only ever results in neutralization. Pater (2003) claims the same thing for assimilation and harmony constraints.

(34) The ranking accounts for why featural conditions can only trigger neutralization.
(a) *coda/[+voice] and neutralization only (Lombardi 1995)
(b) Pater (2001/3): Suppose that [w] blocks nasal spread, then why can't AGREE[nasal] motivate deletion of [w]? /mãwo/ → [mãõ]
(or "Why does AGREE[nasal] only ever motivate assimilation?")

To generalize here, the problem that Lombardi raises is that a featural markedness constraint can force violation of a string faithfulness constraint like MAX and DEP in standard OT. We can see this in tableau (35), where anything can win.

(35) OT with free ranking

	/ab/	*coda/+voice	IDENT[voice]	MAX	DEP
☞	(a) ab	*			
☞	(b) ap		*		
☠	(c) a			*	
☠	(d) a.bi				*

*Of course, markedness constraints can only force a violation of a faithfulness constraint if they outrank them. Since MAX and DEP universally outrank *coda/+voice, though, the only remaining option is neutralization.*

(36) OT with PPFR
• F(StrStruc) » M(f)
Therefore ‖ MAX, DEP » *CODA/+voice ‖

	/ab/	MAX	DEP	*coda/+voice	IDENT[voice]
☞	(a) ab			*	
☞	(b) ap				*
	(c) a	*!			
	(d) a.bi		*!		

Again, the string-faithfulness constraints have winnowed down the options. By the time featural markedness has any say in the matter, the structure of the string has been determined.

But again, how do we know that this result doesn't just work for this case and this set of constraints?

Well, let's consider what it would take for the result to be rendered invalid.

We need a constraint that can both outrank and block MAX and DEP. In other words, we need a constraint to rule out [ap] and [ab] before we get to MAX and DEP.

But what constraint can do that? Not a featural markedness one, for ranking reasons. Not a featural faithfulness one either – featural faithfulness will simply favour the faithful candidate. We're left with prosodic markedness constraints, which <u>can</u> promote deletion (e.g. NOCODA). But if these constraints motivate deletion or epenthesis, the deletion/epenthesis process is no longer featurally motivated. Therefore, features can't motivate deletion.

(37) <u>How do we know this result is general?</u>

- Fixed Ranking harmonic bound: α is an FR harmonic bound for β if
    - (a) For all constraints C s.t. C(β>α), C is universally outranked by some constraint K s.t. K(α>β)
    - (b) There is no constraint Q s.t. Q(β>α) and Q may outrank K.
- MAX, DEP favour [ab], [ap] over [a], [abi]
- Is there any constraint that outranks MAX, DEP and favours [a] or [abi]?
    - (i) Featural markedness constraints are out (due to fixed ranking).
    - (ii) A featural *faithfulness* constraint (e.g. IDENT[voice]) may block [ap], but it cannot block the fully faithful form.
    - (iii) Another type of markedness constraint (e.g. prosodic) may eliminate [ab] (e.g. NOCODA), but then the deletion process would be a prosodically motivated one.

30m

4.1.3 Too many blockers

The fixed ranking has one final implication: it means that a featural markedness constraint cannot block a prosodic process.

(38) <u>The effect of this ranking:</u>
A featural markedness condition cannot block a prosodic process

Hypothetical examples of what shouldn't happen are easy to make up. For example, suppose a language has a metrically-driven syncope process. Suppose that in the course of syncope, a voiceless and voiced obstruent happen to become adjacent, as in (39). Such a structure will violate AGREE[voice], and its featural faithfulness counterpart IDENT[voice].

(39) <u>Hypothetical Example</u>
(a) /bada/ → SYNCOPE [bdá]
(b) but AGREE[voice] and IDENT[voice] block it in
/pada/ → [padá], *[pdá]

- Compare with cases like Macushi Carib, where syncope is only blocked by *prosodic* conditions (i.e. a ban on complex onsets).

(40) <u>OT with Free Ranking</u>

	/pada/	AGREE[voice]	IDENT[voice]	SYNCOPE	MAX
☠	(a) padá			*	
	(b) pdá	*!			*
	(c) bdá		*!		*

In OT with Free Ranking, it's easy to use Agree and Ident to block syncope.

For the condition on voicing to block syncope, AGREE[voice] and IDENT[voice] both need to block off SYNCOPE's escape. However, they need to work together.

Unfortunately, MAX blocks that particular route. MAX must outrank AGREE[voice], so the best ranking one could hope for is in (41).

(41) <u>OT with PPFR</u>

• Mk(π) » Mk(f) means that SYNCOPE » AGREE[voice]

	/pada/	SYNCOPE	MAX	AGREE[voice]	IDENT[voice]
☹	(a) padá	*!			
☞	(b) pdá		*	*	
☞	(c) bdá		*		*

Again, stringstructure-faithfulness winnows down the canddiates that are left.

33m

4.2 Faith(StrStruc) «» Faith(features)

To finish our discusion of string-faithfulness, we need to look at what it is <u>not</u> in a fixed rankng with. It might seem obvious that Faith(StrStruc) should universally outrank Faith(features), but there is evidence that this isn't so.

If there was such a fixed ranking, it would make a strong prediction: if a prosodically-motivated processes could be resolved by neutralization, it will be.

(42) <u>Suppose that Faith(StrStruc) »» Faith(features)</u>

This fixed ranking predicts that:

"if a prosodic process can be resolved by neutralization, it will be."

• This is false.

Why? Well, if MAX and DEP always outrank IDENT[f], the prediction is that neutralization is always a better option than deletion or epenthesis, even for prosodic processes. Now this is just too strong a prediction: We want this to be true for featural processes, but prosodic processes must be allowed to do <u>anything.</u>

For example, we know that ONSET can be resolved by deletion, epenthesis, and feature change, as in (43). But if IDENT[f] is lower ranked than MAX or DEP, then violation of IDENT[f] should always be the best option – in other words, onset should always be solved by neutralization.

(43) ONSET can be resolved by

(a) feature change: /ia/ → [ja] (see Rosenthall 1994)

(b) epenthesis: /ia/ → [tita] (Asheninca Campa)

(c) deletion: /iaka/ → [ka]

• If MAX and DEP always outranked IDENT[f], then feature change should always take place!

(44) Ponapean (Rehg & Sohl 1981)

(a) Coronal + Non-Coronal: Assimilation

/nan/

[nam + par] 'in trade wind season'

[naŋ + kɛp] 'inlet'

(b) Labial + Non-Labial: Epenthesis

[lim + a + tip] 'slices, chips'

[lim + a + ka] 'rows, lines'

Ther's a more extended example in (45).

There's a further reason: in some cases, MAX and DEP are blocked by featural faithfulness. In Ponapean, famously, faithfulness to labial blocks assimilation, resulting in epenthesis.

In other words, epenthesis happens even though neutralization is an option.

(45) Analysis

π, IDENT[labial] » DEP » IDENT[coronal] (Goodman 1995, de Lacy 2002:ch.7)

(46) (a) Faith(features) » Faith(StrStr)

/lim-tip/	π	IDENT[labial]	DEP	IDENT[coronal]
(a) limtip	*!			
(b) lintip		*!		
☞ (c) limatip			*	

(b) Faith(StrStr) » Faith(features)

/nan-par/	π	IDENT[labial]	DEP	IDENT[coronal]
(a) nanpar	*!			
☞ (b) nampar				*
(c) nanapar			*!	

In short, the fact that prosodic processes can result in deletion and epenthesis even when neutralization is an option, and the fact that featural faithfulness can block string-faithfulness constraints shows that there is no fixed ranking between featural faithfulness and string faithfulness.

36m

4.3 Mk(Prosody) »» Faith(Features)

Now let's quickly turn to the other fixed rankings.

There is evidence that Mk(prosody) universally outranks feature faithfulness. The evidence comes from the lack of languages with feature-driven stress (de Lacy 2002a).

The example's a little convoluted. However, suppose a language has vowel reduction. Suppose also that IDENT[round] blocks reduction of round vowels, as Crosswhite (1999) shows. Well, if TROCHEE is at the bottom of the pile, the net result could be a roundness-driven stress movement. YOu can see this in tableau (48). All the prosodically optimal options are blocked either for faithfulness or sonority reasonss. The net effect is that stress moves to where it's neede to preserve the round vowel.

This is probably unattested. Stress only moves for sonority, tone, or prosodic structure, never for less marked features, or to preserve features. Unless we haven't found such a language yet.

(47) Problematic Implications? Indirect influence

Featural faithfulness could *indirectly* block a prosodic process.

- IDENT[round] prevents a sonority process: reduction of /u/ to [ə]
- This forces a violation of a prosodic constraint

(48) *Featural faithfulness and stress shift*

/pato/	IDENT[round]	*ŏ/fullV	TROCHEE
(a) (páto)		*!	
(b) (pátə)	*!		
☞ (c) (pətó)			*

(i) Roundness-driven stress movement with reduction! [!Unattested?!(?)]

(49) *Featural faithfulness and stress shift*

/pato/	TROCHEE	IDENT[round]	*ŏ/fullV
☞ (a) (páto)			*
☞ (b) (pátə)		*	
(c) (pətó)	*!		

(i) Roundness-driven stress movement with reduction! [!Unattested

*The constraint *ŏ/fullVs is a sonority constraint, by the way, not a featural constraint. We know that it can outrank TROCHEE because it causes stress shift in some laguages (de Lacy 2002: ch.4). Therefore, the only option open to us is having TROCHEE universally outrank IDENT[round].*

The Fixed Rankign has another result which I won't discuss here for time reasons – it's in section 4.3.1

38m

4.3.1 Positional Faithfulness and the Fixed Ranking

(50) Positional faithfulness and contrastive syllabification (Wilson 2000:xxx?)

- positional faithfulness + featural constraints can force contrastive syllabification.

• Broader generalization:
Featural faithfulness cannot influence the outcome of a prosodic process

(51) OT without PPFR

/aba/	*+voice	onset-IDENT[voice]	ONSET
☞ (a) a.ba	*		
☹ (b) ab.a	*		*
☞ (c) a.pa		*	
☠ **(d) ap.a**			*

(i) cf /apa/ → [a.pa]; therefore, surface contrastive [a.pa] and [ap.a]

(52) OT with PPFR
• MK(π) »»» MK(F) = ONSET » *[+VOICE]

/aba/	ONSET	*[+voice]	onset-IDENT[voice]
☞ (a) a.ba		*	
☹ (b) ab.a	*	*	
☞ (c) a.pa			*
☹ (d) ap.a	*		

(i) ☹ perpetual loser

4.4 Mk(Prosody) »»» Mk(Features)

Depending on your theoretical predilections we may need a final fixed ranking, in section 4.4. Why do we need this fixed ranking? Well, without it, feature-based constraints could force changes in prosodic structure.

*For example, suppose a constraint mentions a prosodic constituent and a feature. There are many potential examples, as long as you believe in such constraints. Suppose AGREE[vd] operates within the syllable only. /ætrækt/ will come out as [ət.rækt], not *[ə.tr̥ækt], while /ædrɛs/ → [ə.drɛs].*

MAX and DEP are irrelevant here. Only ONSET matters, and its clear that with AGREE[voice]$_\sigma$ outranking ONSET, features can influence prosodification.

Suppose you have a constraint against unaspirated stops foot initially. An easy way to avoid violating this constraint is to syllabify an underlyingly unaspirated stop into the preceding syllable's coda: /ati:/ → [at(í:)] while /at^{h}i/ → [a(t^{h}i:)].

If we have ONSET universally outranking these constraints, then the problem doesn't arise.

(53) Mk(Prosody) »»» Mk(Features)
predicts that feature conditions cannot force a change in prosodic structure.

(54) *Markedness Examples*
A domain bounded featural markedness constraint can be satisfied by moving the domain boundary.
(a) e.g. *[$_{Ft}$C$^{[-aspirated]}$ means that /ati:/ → [at(í:)] while /at^{h}i:/ → [a(t^{h}i:)]
(b) AGREE[voice]$_\sigma$ means that /atra/ → [at.ra] while /adra/ → [a.dra]
(c) AGREE[nasal]$_{Ft}$ means that /ãha/ → [(ã)ha] while /ãhã/ → [(ãhã)]

(55) If Mk(prosody) »» Mk(features), this problem doesn't arrive.
(a') ONSET » *$[_{Ft}C^{[-aspirated]}$, so /ati:/ → [a(ti:)] or [a(t^hi:)]
(b') NOCODA » AGREE$[voice]_{\sigma}$, so /atra/ → [a.tra]
(c') FTBIN » AGREE$[nasal]_{Ft}$ means that /ãha/ → [(ãhã)]

Of course, this all depends on how you want to deal with domain-reference. If there are no domain-referring markedness constraints – if its' done by some careful mixture of positional faithfulness and context-free markedness, for example – then this fixed ranking will be unnecessary.

40m 30

5. Place of Articulation

Before summing up, I have to say something about Place of Articulation. As Lombardi observes, Place of Articulation seems to behave differently from voice.

In Ponapean, as we've seen, the requirement that adjacent consonants agree in PoA forces epenthesis – a change in string structure. In Attic Greek, the same condition forces deletion.

(56) The proposals in here are based on de Lacy (2003)

(57) Place of Articulation
Can featural conditions on PoA change string structure?
e.g. Ponapean: AGREE[PoA] and IDENT[labial] trigger epenthesis
/lim-tip/ → [limatip]
e.g. Attic Greek: AGREE[PoA] triggers deletion
/RED+anut+k+a/ → [ɛːnuka] 'I have accomplished' (Steriade 1988:217)

So, it looks like a featural condition forces a change in string structure.

However, I'm going to stick my neck out here and say that not all PoA markedness constraints can trigger deletion and epenthesis. Only the ones that require homorganicity can.

(58) Generalizations
(a) PoA constraints *do not* trigger deletion/epenthesis
e.g. *dorsal doesn't trigger deletion
(b) All PoA-related changes are due to a requirement of ***homorganicity***.
(c) There's evidence that homorganic consonants are *prosodically* distinct from heterorganic ones (lenition, stress).

I'd like to suggest that homorganicity – agreement in PoA – is actually a prosodic requirement. There is evidence that homorganic clusters are 'prosodically visible' in a way that heterorganic ones are not. For example, homorganic clusters often act like geminates in lenition, according to Kirchner (200?). More interesting is the fact *that some languages treat syllables with a homorganic coda as heavy for stress, while ones with heterorganic codas are light.*

If we suppose homorganic clusters have a distinct prosodic structure, then we could suppose that a constraint that requires that prosodic structure is responsible for homorganicity.

Since homorganicity is therefore a prosodic requirement, we should expect it to trigger deletion and epenthesis, just like other prosodic requirements.

(59) Implication 1

HOMORGANICITY should trigger deletion and epenthesis

(i) This is roughly equivalent to CODACOND (Ito 1986);
see suggestions by Lombardi and Pater (2003:xx?) that CODACOND is really what's responsible.

We should also expect a slew of other things. We should expect that featural faithfulness constraints can never block homorganicity from happening.

We could also expect some subtler things, such as Implication 3.

(60) Implication 2

As Mk(Prosody) »» Faith(features), featural faithfulness can never block HOMORGANIC (see Ponapean)

(61) Implication 3

Since HOMORGANIC makes no distinction between different PoAs, if one PoA undergoes deletion/epenthesis/neutralization, then all undergo some change.

e.g. There is no language like Ponapean, but where dorsals don't undergo change.

(62) Interfering with Implications 2 & 3

• AGREE[PoA] could also be a featural requirement!

So, we should expect languages like Catalan and Chukchi, where assimilation is *blocked* by featural concerns (de Lacy 2002:ch.7)

However, subtleties aside, I'd like to suggest that homorganicity is not a counter-proposal to the present proposals, once we recognize that it is a prosodic requirement.

45

6. Alternatives

I would love to talk about alternative approaches, but I've been talking 90-to-the-dozen, and we're running out of time. The point section 6 makes is that we can't rely on constraint form alone to solve the too Many Solutions problem. But I'll leave this section for your later amusement.

================

So far I've been advocating fixed ranking as a solution to all our problems. But are there alternatives?

Two have been proposed. One is Wilson's theory of Targeted Constraints. I don't have time to discuss that theory here and now. I will only point out that Wilson's theory is constraint-specific – there is no general procedure for determining whether a constraint is targeted or not. Consequently, Target Constraint theory does not predict which processes will provoke which repairs. In that sense it is less predictive than the present one. Anyway, discussing Wilson's theory will take us far too long; I'll satisfy myself by mentioning the critiques in Pater's (2003) paper and McCarthy's paper.

The alternative I will briefly touch on is Lombardi's approach. Lombardi's approach is very smart in that it uses constraint form to deal with the Too Many Solutions problem.

(63) <u>Can constraint form solve the Too Many Solutions problem?</u>
Lombardi (1995): *Yes* (at least for the ban on voiced codas)

There's a way to schematize this whole approach to the Too Many Solutions problem, as in (64). Suppose you have a process, and it can only be resolved by one method. Well, there are three ways you can go about massaging your constraints' form to guarantee this.

One way is to make sure that whatever markedness constraint motivates your process is only satisfied by the one solution you want. For example, suppose you want to ensure that a ban on voiced codas is only satisfied by neutralization, then you would have to define your markedness constraint to ban not only forms like [ab], but [abi] and [a] as well.

To cut a long story short, this is very difficult to do in standard OT. This is especially clear for /ab/→[a] – this will have to violate the markedness constraint that bans [ab], but only when it derives from input /ab/. Without carefully interweaving some faithfulness condition into the constraint, it's really hard to see how this could be possible.

At the other end of the scale, the other solution is to make sure that all your faithfulness constraints favour just the one solution. So, for a ban on voiced codas, you need to make sure that there is no faithfulness constraint that favours [ap] over [a] or [abi]. So, there's no ident[voice], for example. There's also no max[voice], because [abi] beats [ap] that way. To cut another long story short, the pure faithfulness approach poses just as many problems as the pure markedness approach. It's extremely difficult to see how [abi] is less faithful than [ap] for every faithfulness constraint, from /api/.

What we're left with is a mixed solution, which is actually Lombardi's solution.

(64) <u>Schematically</u>
Suppose that a process can be resolved by solution S_1,
but never S_2 or S_3 (e.g. *coda/+vd allows neutralization, but not deletion or epenthesis).
(a) Every featural 'process' is an interaction of Mk(f) and Faith(f)
(b) Mk(f) must disfavour the faithful candidate (F)
(c) Faith(f) favours the faithful candidate
(d) Either:
(i) *Pure Markedness solution*

(I) Mk(f) favours S_1 over S_2 & S_3 & F
(II) There is no Mk that favours non-S_1 over S_1

(ii) *Pure Faithfulness solution*
(I) Mk(f) favours all solutions over F
(II) Faith(f) favours solutions F & S_1 over all other solutions (more accurately: for every not-S_1 S_x, there is a faithfulness constraint F s.t. $F(S_1 > S_x)$)
(III) There is no constraint that favours non-S_1 over S_1

(iii) *Pure Hybrid solution (or 'passing the buck' solution)*
(I) Mk(f) favours S_1 & S_2 over S_3 & F
(II) Faith(f) favours S_3 & F over S_1 & S_2
(III) some other constraint favours S_1 over S_2
(IV) no other constraint favours S_2 over S_1

6.1 Pure Markedness and Faithfulness

(65) Is the Pure Markedness solution viable?
• Not in standard OT (cf Wilson 2000)
• For all featural constraints C_F, C_F would have to favour neutralization over deletion, epenthesis, and faithfulness.
☒ Rules out most (all?) featural markedness constraints so far proposed.
*coda/[+voice] [ap] > [a], [a.bi], [ap]
*[αF] must not be solvable by deletion.
• One big problem is how to produce neutralization without forcing deletion – how can ∅ not be neutralization?
• The other is how in domain-bounded constraints, movement outside the domain (through 'reprosodification' or epenthesis) cannot be a solution.
e.g. ASSIM[f] cannot be solved by deletion or epenthesis
• Suppose we had a constraint that favoured $C^{-vd}]_\sigma > \varnothing, {}_\sigma[C^{+vd}, C^{+vd}]_\sigma$
This would predict a language where codas are required (CODA!)
• Wilson's solution: /ab/→[ap] is noncomparable to /ap/→{[abi], [a]}
• Final issue: *why neutralization?* Why is it neutralization that is always possible? How does this follow from any restriction on constraint form?

(66) Is the Pure Faithfulness solution viable?
• Not in standard OT.
• For *coda/+voice: this means that for every faithfulness constraint that assigns a violation to /ab/→[ap], it also assigns a violation to /ab/→{[a], [abi]}. How can /ab/→[ab<u>i</u>] violate IDENT[voice]/MAX[voice]?

6.2 Hybrid Solution

Lombardi's solution puts half the burden on markedness and half on faithfulness.

(67) The Hybrid Solution
• This is Lombardi's (1995) solution

- e.g. Mk(f) = *+voice
  favours neutralization and deletion over epenthesis and F
  /ab/ → *+voice([ap] & [a] > [ab] & [abi])
- e.g. Faith(f) = MAX[voice]
  favours epenthesis and F over neutralization and deletion
  /ab/ → MAX[voice]([ab] & [abi] > [ap] & [a])
- "some other constraint favours S_1 over S_2"
  i.e. MAX, which favours [ap] over [a].
- "no other constraint favours S_2 over S_1"
  e.g. rules out IDENT[voice], which favours [a] over [ap].
- Also rules out all markedness constraints of the form $*\pi/f$:
  these constraints can be resolved by moving π; since this doesn't happen, and there are no faithfulness constraints to prosodic structure, it must be the case that there are no markedness constraints that favour f in prosodic position π_1 over f in prosodic position π_2.

*You can see this in the tableau. *+voice rules out the faithful and epenthetic forms. The faithfulness constraint MAX[voice], on the other hand, is equally violated by the remaining solutions – neutralization and deletion. Then MAX comes along to favour the neutralization candidate.*

Crucial to this solution is that there is no faithfulness constraint that favours deletion over neutralization – no IDENT[voice], in other words.

(68)

	/ab/	*+voice	MAX[voice]	MAX
☞	(a) ab	*!		
	(b) ap		*	
	(c) a.bi	*!		
	(d) a		*	*!

(69) Is the Hybrid solution viable?

- Does it rule out markedness constraints that refer to a domain?

• • i.e. $*[\alpha f...\beta f]_\delta$ = AGREE$[f]_\delta$

Potential Solutions: $[\alpha f..\alpha f]_\delta$ (neutralization)
$[\alpha f]_\delta$ (deletion)
$[\alpha f\ldots]_\delta\ \beta f$ (domain alteration *or* epenthesis)

- How can Mk rule out $[[\alpha f\ldots]_\delta\ \beta f]$?
  e.g. How does Mk rule out /apga/ →[apiga] without ruling out /apiga/→[apiga]?
- Therefore, Faith must rule out epenthesis = the same problem as the Pure Faithfulness solution: how can IDENT[voice]/MAX[voice] rule out [apiga]?

(70)

	/abka/	AGREE[voice]	MAX[voice]	DEP	MAX
☞	(a) abka	*			
☞	(b) apka		*		
☠	(c) abika			*	
	(d) aka		*		*

What's the problem here? Well, Markedness fails to eliminate epenthesis, and there is some faithfulness constraint that favours epenthesis over neutralization – i.e. MAX[voice].

The Hybridized solution works for voice neutralization. However, it is less clear how it works for constraints that impose a locality or domain requirement. For example, suppose in (69) we have a constraint that bans disagreement within a domain; or equivalently, requires agreement. The challenge is how to eliminate candidates that attempt to resolve this constraint by changing the domain. For example, one way to avoid AGREE[voice] – a requirement that <u>adjacent</u> obstruents agree in voicing – is to epenthesize a vowel. We want to make sure the epenthetic candidate loses. So, under a hybrid solution, we have two choices. The epenthetic candidate is ruled out by the markedness constraint – which is clearly wrong, because that defeats the purpose of specifying a domain, or by the faithfulness constraint. Now, this means that every faithfulness constraint that bans neutralization must also ban epenthesis. This means that MAX[voice] cannot exist because MAX[voice] is violated by neutralization but not epenthesis. So we have a contradiction.

(71) <u>Result</u>: Constraint Form is not a comprehensive solution
(though it can provide *some* solutions: e.g. de Lacy (2002§2.4.2)

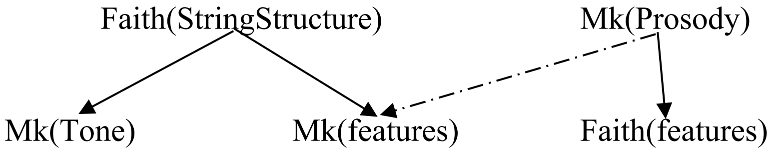
The result is that we don't have a comprehensive solution. We obviously can get <u>partial</u> solutions, but they don't seem to work out once we start considering other things.

In short, all we have left without fundamentally changing EVAL – as Wilson does – is fixed ranking. QED.

7. Summary and Conclusions

To conclude, I've proposed that there are several fixed rankings, repeated again in (72).

(72) <u>The Prosodic Primacy Fixed Rankings (PPFR)</u>



(73) <u>Predictions</u>
(a) No feature constraint can trigger deletion or epenthesis
(b) All prosodic constraints should be able to trigger deletion, epenthesis, and neutralization (structural descriptions permitting).
(c) Prosodic processes can never be blocked by a featural markedness condition.
(d) Feature identity should be able to influence the outcome of a prosodic process.

The fixed rankings make several predictions, with the main ones detailed in (73).

SO, my aim to day was to explore whether a tool we have almost for free − Universally Fixed Ranking − can provide a restrictive way to account for a vast problem − vast in its empirical implications. Whether it does or not depends on the empirical facts.

*On that front, there are several nagging worries. I've found some languages where PoA markedness constraints like *dorsal seem to result in deletion, not neutralizaiton. Coalescence is also a bugbear − there are some cases where it looks like featural faithfulness blocks coalescence.*

However, it remains to be seen whether these cases can be explained away, or will lead to a refinement of the fixed rankings, or will simply debunk the whole approach.

Any suggestions you have along these lines would be appreciated!

Thanks for listening!

50m

Appendix 1: Tone

Prosodic constraints are therefore 'heavyweights' in the competitive world of phonology – they can motivate anything.

In contrast, – and I'm sticking my neck out a bit here – tonal conditions seem to be more limited.

(74) Tone: the middleweight
Q: What responses can featural conditions trigger?
A: (a) No change in valency
(b) Change in prosodic structure

There are very well known things that tonal conditions, like the OCP, can do. The well known conditions all involve tonal conditions affecting tones. I haven't bothered to put examples in (75) – there are plenty that show that tonal conditions can trigger tone spread, dissimilation, epenthesis, and deletion.

(75) Tone: No change in valency
(a) OCP can trigger (Odden 2000)
- Tone spread, shift: /u-hí-lima/ → [uhílíma]
- Tone dissimilation: /kú-dá/ → [kúdà]
- Tone epenthesis: /bá/ → [bǎ]
- Tone deletion: /bâdá/ → [bàdá]

(b) Cannot cause
(i) Sponsor Deletion: /bádá/ → [bád]
(ii) Sponsor Epenthesis: /bádá/ → [bádìá]

(c) ?Cannot block?
(i) Deletion: /bádiá/→ [bádiá], while /bádia/ → [báda]
(ii) Epenthesis: /bád/ → [bád], while /bad/ → [badá]?

In contrast, I haven't been able to find good cases where tonal conditions like the OCP can cause tone sponsors (Tone-bearing units) to delete or epenthesize. It's easy to imagine situations where this could happen, though. An OCP on high tones could easily motivate deletion.

I also not sure about cases where the OCP blocks prosodically-motivated deletion and epenthesis.

I am certainly willing to be enlightened on this issue, though.

In fact, I should say that I'm willing to be enlightened on any of the broad empirical claims I'm making. All of the generalizations I'm making and others have made may well turn out to be wrong. We know so little about phonological processes, after all.

(76) Change in prosodic structure
Tone-driven stress (de Lacy 2002b)

Unlike features, by the way, tones can cause a change in prosodic structure.

4.1.2 The Indirect Influence of Featural Faithfulness

Unfortunately, in section 4.1.2, the Wilsonian problem persists. We have seen how featural conditions cannot directly influence the output of a prosodic process. However, can they indirectly influence the outcome?

For example, what if a featural markedness constraint forces a change in C_2. Could this effectively gang up with other featural markedness constraints to prefer C_1?

For example, suppose there is a ban on [ki] clusters that makes /k/ turn into [c]. Is it possible for input /atki/ to be realized as [ati], so avoiding the unfaithfulness that comes with [aci] and the markedness that comes with [aki]?

(77) Indirect Influence
- Could a featural faithfulness constraint block a featural process, which thereby favours deletion of C_2?
- e.g. Suppose *ki results in /ki/→[t͡ʃi].
- Could /atki/ → [ati] because [aki] is ruled out by *ki and [at͡ʃi] is ruled out by IDENT[dorsal]?

*Tableau (78) gives the ranking that is needed for such blocking. Obviously it won't work because we've had to invoke the featural markedness constraint *ki over OI-CONTIG-σ.*

(78) The ranking needed…

	/atki/	NOCODA	MAX	*ki	IDENT[dorsal]	OI-CONTIG-σ
	(a) at.ki	*!				
	(b) aki		*	*!		
☞	(c) ati		*			*
	(d) at͡ʃi		*		*!	

(i) Ranking ruled out by Faith(StrStr) »» Mk(features)
Therefore || OI-CONTIG-σ » *ki || and [at͡ʃi] wins!

The point here is that for a featural process to effectively block a certain prosodic outcome, you need both markedness and faithfulness constraints to outrank all string faithfulness constraints. Only if you have both markedness and faithfulness constraints active can you block every possible escape hatch, and force the featurally desirable outcome.

In general, then, there is no way to indirectly influence the outcome of a prosodically motivated process.

(79) *Prediction*
/VC_1C_2V/ cannot be realized as [$V.C_1V$] by an indirect featural condition.

(80) Assimilation and PPR
- AGREE[nasal] 'Adjacent obstruents agree in [voice]

(a) OT without PPFR

/apga/	AGREE[voice]	IDENT[voice]	MAX	DEP
☞ (a) apga	*			
☞ (b) abga		*		
☠ (c) aga			*	
☠ (d) ap<u>i</u>ga				*

(b) OT with PPFR

/apga/	MAX	DEP	AGREE[voice]	IDENT[voice]
☞ (a) apga			*	
☞ (b) abga				*
☹ (c) aga	*!			
☹ (d) ap<u>i</u>ga		*!		

Questions

Q: You haven't explained *why*? Why are the fixed rankings there?
A: (i) Makes learning easier, but I don't think that's the motivation. After all, there are many other rankings that aren't fixed but would make learning easier.

(ii) As a formalist, I have to say evolution. Not for survival, though, but as an epiphenomenon of the way the lg faculty has been cobbled together.

Q: Why not see manner assimilation as 'sonority assimilation'?
A: I guess because sonority usually dissimilates, as in sonority-distance effects.

5.3 Further Limitations: sonority and stress

Why not *σ́/[labial]?

A1: π » Mk(*f*)

So, ALLFTL » *σ́/[labial]

A2: But then why not *σ́/[labial] » IDENT[labial] → elimination of labials in *x*?

Must be that *σ́/[labial] doesn't exist.

i.e. *σ́/f doesn't cause *anything* – no neutralization, deletion, etc.

NC

(81) Counter-example: *NC̥

Featural condition, but provokes coalescence (though never epenthesis)

- Indonesian

/məN+pilih/ → [məm$_{1,2}$ilih],

cf /məN-bəli/ → [məmbəli]

Perhaps deletion is really: /məN+pilih/ → [mə.^{m}pilih] → [mə.milih] because [^{m}p] is banned.

cf /məN+bəli/ → [mə.^{m}bəli]

|| NOCODA » UNIFORMITY » *^{m}p » IDENT[^{m}p→m] ||

[There is one final issue: can the rank of Prosodic markedness rank freely with featural faithfulness? I think so. There's an example in (82). In this case, NOCODA forces a coda nasal to coalesce with the preceding segment. However, coalescence is blocked when it would lose a marked feature – labial.

(82) Faith(f) can block a prosodic process
IDENT[labial] blocks NOCODA
Chickasaw:
/cholhkan-a-n/ → [cholhkanã] 'spider-object'
cf [apa-ta-m] 'eat-question-past'

This can be modeled straightforwardly with IDENT[labial] outranking NOCODA, which outranks UNIFORMITY. However, it's crucial that IDENT[labial] outrank NOCODA.
[If someone asks!!
*How do you know it's not the markedness constraints' fault? Why not *Vn?*
Answer: this predicts that there could be neutralization to labials. E.g. /aŋ/ → [am], /an/ → [am]. This doesn't happen.]

(83) (a)

	$/a_1m_2/$	IDENT[labial]	NOCODA	UNIFORMITY
☞	(a) a_1m_2		*	
	(b) $ã_1$	*!		*

(b)

	$/a_1n_2/$	IDENT[labial]	NOCODA	UNIFORMITY
	(a) a_1n_2		*!	
☞	(b) $ã_1$			*

]

4.4 Tone

(84) What can tonal conditions do? (Odden 2000)
- Tone spread, shift: /u-hí-lima/ → [uhílíma]
- Tone dissimilation: /kú-dá/ → [kúdà]
- Tone epenthesis: /bá/ → [bǎ]
- Tone deletion: /bâdá/ → [bàdá]

(85) Cannot cause a change in sponsors (TBUs)
- Sponsor deletion: /ábá/ → [áb]
- Sponsor epenthesis: /ábá/ → [ábi̱á]
- Sponsor metathesis: /ábàì/ → [ábìá]

(i) Is contour-driven lengthening an exception? i.e. /bâ/ → [báà]???

(86) OT without PPFR

	/ábá/	OCP-T	IDENT-T	DEP
☞	(a) ábá	*		
☞	(b) ábà		*	
☠	(c) ábi̱á			*

(87) OT with PPFR: Faith(StrStruc) » Mk(Tone) = DEP » OCP-T

/ábá/	DEP	OCP-T	IDENT-T
☞ (a) ábá		*	
☞ (b) ábà			*
☹ (c) ábi̠á	*!		

4.4.1 Mk(π) and Mk(tone)

(88) Tonal conditions can directly affect prosodic structure
• Tone-driven stress (de Lacy 2002 and references cited therein)
e.g. *σ́/L » ALIGNFTL

(89) Tonal conditions could indirectly cause a change in valency:
i.e. • lengthening under stress
• tone displaces stress
• Therefore, tone inadvertantly causes a change in valency.
Ranking: *σ́/L » ALIGNFTL
WEIGHT-TO-STRESS » DEP

(90) Could a tonal condition ever block a prosodic process?
e.g. could OCP(H) block /ábìdá/ from undergoing syncope to [ábdá]?

4.5 Sonority

(91) Constraints on sonority can
(a) force feature change
e.g. Vowel reduction
(b) force deletion/epenthesis
e.g. sonority-driven syncope (Gouskova 2003)
(c) force change in prosodic structure
e.g. *ONS≥liquid forces syllabification into codas
(d) block prosodic processes
e.g. Niuafo'ou glide formation: *ONS/glide blocks ONSET

(92) Constraints on sonority can be *blocked* by feature conditions
(a) IDENT[labial] prevents vowel reduction (Crosswhite 1999, de Lacy 2002:xx?).
(b) *ŋ blocks Dakota nasalization

(93) Dakota coda stops→nasals
Dakota aims to have high sonority codas.
(a) It aims to have nasals:
/xap/ → [xam] (cf [xap-a]) 'to be stripped'
/RED-ʔot-a/ → [ʔonʔota] 'be many'
(b) But there is no [ŋ], so /k/ → [g]
/ʃok/ → [ʃog] 'thick, solid' (cf [ʃok-a])

(94)

/tak/	*ŋ	*coda<*nasal*	*coda<+*vd stop*	IDENT[f], IDENT[son]
(a) tak		*	*!	
☞ (b) tag		*		*
(c) taŋ	*!			*

Counter-examples

Siuslawan: k-deletion
Koine Greek: word-final /t/ deletion: /onomat/ → [onoma], cf /arab/ → [arab]
Catalan: /n/ deletion; but this is complicated!

Chickasaw: /Vn/ → [Ṽ], /Vm/ → [Vm] = Faith(f) » UNIFORMITY

Kui: *kp triggers metathesis

Glide formation: /ea/ is blocked going to [ja] because IDENT[high] » ONSET