

## Error-Driven Learning in Harmonic Serialism\*

Anne-Michelle Tessier

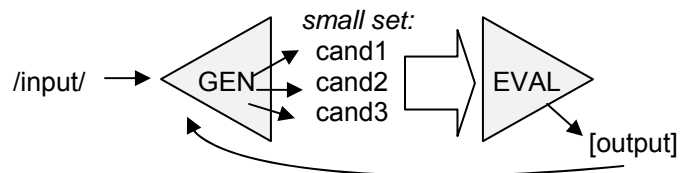
University of Michigan/University of Alberta  
tessiera@umich.edu, amtessier@ualberta.ca

### Introduction

**Harmonic Serialism** - Prince and Smolensky 1993/2004: 79; McCarthy 2000, 2007, 2008ab, 2009a; Pruitt 2010; Kimper 2010; Elfner, to appear

A **constraint-based** but **derivational** view of phonology and its typology

1) An HS schematic:



**Theoretical Optimism:** Harmonic Serialism has some interesting and desirable properties for accurately capturing phonologies without overgenerating (for a classified bibliography, see McCarthy 2009a)

**Assumption:** if Harmonic Serialism is a good theory of phonology, it must be associable with a good theory of learning  
- see also *Staubs and Pater (to appear) on HS learning*

#### **Stumbling Block for Learning:**

- derivations happen *step-by-step*, driven by one M constraint at a time  
- if error [outputs] drive learning, the learner only sees the *end result*

So how can learners figure out the steps in between?

#### **Hidden Rankings**

- HS sometimes requires rankings  $M1 \gg M2$  *not* revealed by errors!

\* Thanks especially to Karen Jesney for crucial improvements and insights, and to Michael Becker, Andries Coetzee, Brett Hyde, Emily Elfner, Matt Goldrick, John McCarthy, Joe Pater, Kathryn Pruitt, Matt Wolf, the MIT Phonology Circle, particularly Adam Albright, Ed Flemming and Igor Yanovich, and the UMichigan PhonDi Group. All errors are sadly mine.

### Roadmap of the Talk

#### **One: Some bare bones of HS, enough to get by in this talk**

Crucial properties:

- a restricted GEN function, creating a **(quite) finite candidate set**
- a single ranking per language
- **EVAL applies serially to /input/...**
- **converging finally on [output]** whose local harmony is maximized

#### **Two: Using errors (as in Tesar and Smolensky 1998, 2000) to learn phonotactics (as in Prince and Tesar, 2004; Hayes, 2004)**

- learning just the attested surface forms, as restrictively as possible
- errors made by current grammar used as learning data
- small proposal: in HS, errors must be processed slightly differently

#### **Three: The learning challenge of hidden rankings**

- in OT,  $M1 \gg M2$  rankings are pretty trivial to learn
- in HS they can be completely surprising
- the extent of the problem: *examining errors is not enough*

#### **Four: The Proposal for Learning Hidden Rankings**

- learn phonotactics first, with  $\{M1, M2\}$  left unranked
- then use *winners* – i.e. stored, observed surface forms
  - examine the violation profiles of their *failed loser candidates*
  - ... these can iteratively reveal hidden rankings

#### **Five: Implications, Imbrolios, Discussion**

#### **Take Home Message:**

- HS makes phonotactic learning much more complicated
- but it makes every input's candidate space much less complicated
- so this trade-off may be a key to successful HS learning

## I. Harmonic Serialism in a Nutshell<sup>1</sup>

### HS properties shared with classic OT (Prince and Smolensky 1993/2004):

- basic architecture: /any input/ --> ranked constraints --> [optimal output]
- (basic) constraint types: markedness/structural and faithfulness
- EVAL: same method of choosing optima via strict domination

### HS properties shared with derivational theories

- changes between /input/ --> [output] are small and gradual  
i.e. HS has a much more restricted GEN
- HS loops through multiple I→O iterations in the same mapping  
i.e. HS is less parallel than classic OT

### HS mappings work like this:

- feed any /input1/ to the GEN function
- GEN returns a **finite set of output candidates**: the fully faithful candidate, plus all the candidates that are 'one unfaithful mapping away' from the input (the set of such unfaithful changes being the subject of theoretical and empirical inquiry: see McCarthy, 2009b)

| 2) /εgz/      | * COMPCODA | DEP | MAX |
|---------------|------------|-----|-----|
| ☞ εg          |            |     | *   |
| εgz           | *!         |     |     |
| gεgz ... etc. | *!         | *   |     |

- EVAL uses the constraint ranking to map from /input/ → [output1]
- the winning [output1] is now taken as /input2/
- GEN reapplies, and then EVAL reapplies

| 3) /εg/     | * COMPCODA | DEP | MAX |
|-------------|------------|-----|-----|
| ☞ εg        |            |     |     |
| ε           |            |     | *!  |
| εgz ... etc | *!         | *   |     |

- the last cycle is one where /input/ → [fully faithful candidate]
- and that FFC is the [winner]

**Note:** Within a language, the ranking always stays the same

- like in classic OT, the HS constraint ranking defines the language
- different outputs win on different iterations because the limited GEN means different inputs give rise to different candidate sets

## II. Learning from errors in HS

Start with a maximally restrictive grammar: {M} >> {F}  
(in OT: Smolensky (1996) and many many others)

4) Toy Initial State: ONSET, \*COMPCODA >> MAX, DEP

Then attempt to reproduce an observed form with the current grammar, and analyze anything mapped unfaithfully as an error:

5) ERC vectors, following Prince (2002) and others

| input  | winner ~ loser | *COMPCODA | ONSET | MAX | DEP |
|--------|----------------|-----------|-------|-----|-----|
| /tost/ | [tost] ~ [tos] | L         | e     | W   | e   |

Every error is stored in a Support table, which grows over time

**To learn** (Prince and Tesar, 2004; Hayes, 2004; Tessier, 2009):

Take all errors in the Support and build a new ranking from scratch:

- o one that resolves these errors,
- o but otherwise remains as restrictive as possible

### A rather simplified version of this learning algorithm:

6) Build each stratum as follows:

Step 1: Install all M constraints that prefer no Losers in the current stratum  
If any errors resolved, remove from Support and start next stratum

Step 2: If Step 1 has done nothing, install as few of the Faith constraints which prefer Winners as possible, so as to free up Ms constraint(s) at the next stratum

If any errors resolved, remove from Support and start next stratum

Continue cycling through Steps 1 and 2 until no more M constraints:

Step 3: Install all remaining F constraints and then **terminate**.

### Turn over to see learning at work for this error:

<sup>1</sup> A practical textbook-style intro: [http://works.bepress.com/john\\_j\\_mccarthy/108/](http://works.bepress.com/john_j_mccarthy/108/)

7) Learning from this error:

| input  | winner ~ loser | *COMP CODA | ONSET | MAX | DEP |
|--------|----------------|------------|-------|-----|-----|
| /tost/ | [tost] ~ [tos] | L          | e     | W   | e   |

- Stratum 1: Step One installs **ONSET** (since it hasn't made any errors)...  
 Stratum 2: Step Two installs **MAX** (since it will resolve the errors)...  
*Error is now resolved, so is ignored...*  
 Stratum 3: Step One install **\*COMP CODA** (since its error is gone...)  
 Stratum 4: Step Three installs **DEP** and terminate.

8) New Grammar: ONSET >> **MAX** >> \*COMP CODA >> DEP

### 2.1 HS phonotactic learning: how to process errors?

How does the learner learn from errors with multiple // → [O] mappings?

9) Suppose  $\mathcal{H}_1$ : \*COMP CODA >> ONSET >> {MAX, DEP}

| 10a) /εgz/ | * COMP CODA | ONSET | MAX | DEP |
|------------|-------------|-------|-----|-----|
| ☞ εg       |             | *     | *   |     |
| εgz        | *!          | *     |     |     |
| gεgz       | *!          |       |     | *   |
| etc...     |             |       |     |     |

| 10b) /εg/ | * COMP CODA | ONSET | MAX | DEP |
|-----------|-------------|-------|-----|-----|
| ☞ gεg     |             |       |     | *   |
| εg        |             | *!    |     |     |
| εgz       | *!          | *     |     |     |
| etc...    |             |       |     |     |

| 10c) /gεg/ | * COMP CODA | ONSET | MAX | DEP |
|------------|-------------|-------|-----|-----|
| ☞ gεg      |             |       |     | *   |
| εg         |             | *!    | *   |     |
| gεgz       | *!          | *     |     | *   |
| etc...     |             |       |     |     |

11) Full derivation: /εgz/ → εg → [gεg]

**Question:** How do you store this error as a winner-loser pair?  
 Recall that the HS grammar never compares εgz ~ gεg directly!

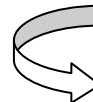
**Proposal:** Simply use the first step in the derivation.<sup>2</sup>

| 12) input | winner ~ loser | *COMP CODA | ONSET | MAX | DEP |
|-----------|----------------|------------|-------|-----|-----|
| /εgz/     | [εgz] ~ [εg]   | L          | e     | W   | e   |

This error will teach the learner something small, but right, about the target:

13)  $\mathcal{H}_{-2}$  learned from (12): ONSET >> **MAX** >> \*COMP CODA >> DEP

| 14a) /εgz/ | ONSET | MAX | * COMP CODA | DEP |
|------------|-------|-----|-------------|-----|
| εg         | *!    | *   |             |     |
| εgz        | *!    |     | *           |     |
| ☞ gεgz     |       |     | *           | *   |



| 14b) /gεgz/ | ONSET | MAX | * COMP CODA | DEP |
|-------------|-------|-----|-------------|-----|
| gεg         |       | *!  | *           |     |
| εgz         | *!    | *   | *           |     |
| ☞ gεgz      |       |     | *           | *   |

With  $\mathcal{H}_{-2}$ : /εgz/ → [gεgz] – one step closer to the target

### Summary of progress

Relying on 'first step' errors,

- the HS learner can determine phonotactics slowly but effectively
- for those marked structures highlighted by errors

### The problem to come

What about marked structures that the learner never sees?

- In OT, a restrictive learner will rule them out from  $H_{-initial}$
- In HS, more learning is necessary... and errors are no help

<sup>2</sup> There are other ways to do this; some might be better, but some are, I think, worse. Certainly there are ways to get more out of such errors, but Tessier (2011) suggests that's maybe not such a good idea...

### III. Hidden M rankings as the challenge for Error-driven HS learning

#### 3.1 The target grammar: feeding among two well-attested processes

Process one: Stops lenite to fricatives intervocalically

15a) ex: Tiberian Hebrew /p, t, k/ → [f, θ, x] / V\_V  
- see Gurevich (2004), summarized especially in Kaplan (2010)

**M1:** \*INTERVOCALIC STOPS, or  
**\*V-STOP-V**

Process two: Fricative allophony -- palatals before [i], velars elsewhere

15b) ex: Greek (Kazazis 1972) /xi/ → [çi], but elsewhere [x], \*[ç]  
/yi/ → [ji], but elsewhere [Y], \*[j]

**M2:** \*[VELAR FRICATIVE] [HIGH V] or  
**\*[xi]** ... and here **\*[xi] >> \*PALATAL FRICATIVE, \*[ç]**

Hypothetical language: both processes in a feeding order (Kiparsky 1968)

16) a) /max/ → [max]      b) /maxi/ → [maçi]      neutralization  
/mak/ → [mak]      c) /makil/ →

Process Interaction in Harmonic Serialism:

Driving (16c)'s mapping: /maki/ → maxi → [maçi]  
**... requires a ranking of M1 >> M2:**

17) **\*V-STOP-V >> \*xi >> FAITH**

| 18) Harmonic Improvement on /maki/ | *V-STOP-V | *xi | IDENT [CONTIN] | IDENT [PLACE] <sup>3</sup> |
|------------------------------------|-----------|-----|----------------|----------------------------|
| ma.ki<br>... less harmonic than    | *!        |     |                |                            |
| ma.xi<br>... less harmonic than    |           | *!  | *              |                            |
| ma.çi                              |           |     |                | *                          |

<sup>3</sup> This constraint name is a cover name for a proper theory of featural place faithfulness, but the choice of faith constraints does not affect this learning result.

The opposite HS ranking, M2 >> M1, would **block any change in /k/**

| 19) Step One: /maki/ | *xi  | *V-STOP-V | IDENT [CONTIN] | IDENT [PLACE] |
|----------------------|--|-----------|----------------|---------------|
| ma.ki                |  | *         |                |               |
| ma.xi                | *!   |           | *              |               |
| ma.çi                | <b>not in the candidate set for /maki/</b> |           |                |               |

Comparison: in OT, M1 and M2 don't need to be ranked wrt each other:

20) OT ranking: **{\*V-STOP-V, \*xi} >> \*ç >> FAITH**  
correct drives: /maki/ → [maçi]

#### 3.2 The learning problem with hidden M rankings

Assumptions

- an initial state is a partial ordering: {all M} >> {F}
- {all M} means a total ordering M1 >> M2 >> M3..., chosen randomly each time grammar is used
- just acquiring phonotactics – via the 'Identity Map'

21)  $\mathcal{H}_{\text{initial}}$ : **{\*V-STOP-V, \*xi, \*ç} >> {ID-CONT, ID-PLACE}**

22) Observed kinds of forms:  
**[mak] [max] [maçi] [çik]**

Phonotactic errors made by (21) on forms in (22) will just reveal allophony:

| 23) input | winner ~ loser | *V-STOP-V | *xi      | *ç       | ID[CONT] | ID[PLACE] |
|-----------|----------------|-----------|----------|----------|----------|-----------|
| /çil/     | çi ~ xi        |           | <b>W</b> | <b>L</b> |          | <b>W</b>  |

24)  $\mathcal{H}_{\text{new}}$  learned from (22):  
**{\*V-STOP-V, \*xi} >> \*ç >> {ID-CONT, ID-PLACE}**

The grammar in (2) maps all of (21) faithfully – *no more errors or learning*.

**Successful properties of the final phonotactic grammar:**

- \*xi >> \*ç >> ID[PLACE] = correct allophony: /xi/ → [çi], otherwise [x], \*[ç]
- \*V-STOP-V >> ID[CONT] = correct lenition: /aka/ → [axa]

**Failures of this phonotactic grammar:**

-stop lenition and fricative allophony do not feed:

24) Language in (23), at the end of phonotactic learning:

/baxi/ → [baçi]      but: /baki/ → [baki] OR [baçi]  
/baçi ↗

**Upshot: hidden ranking not learned**

- grammar vacillates between [baki] and [baçi]
- learner (incorrectly) believes that [Vki] is the unique legal intervocalic stop

Is this final state even a language we want in the typology?

- maybe, maybe not – see section 5.
- but if the language has NO intervocalic stops...
- learning (23)-(24) is a failure of restrictiveness

**Broader upshot: errors cannot be relied on to reveal hidden rankings.**

Side question: Will alternations help?

Short answer: if they occur, they might; but even so they are not guaranteed.

[Longer answer: even if there ARE alternations, the learner still has hard work to do – because they will need to establish a UR that maps to a winner in more than one step – which means setting up the winner is rather harder than setting up a loser back in (12). This is an independent problem that HS learning has to resolve.<sup>4</sup>]

For more on alternations, see the Appendix, and/or discussion period questions, although I frankly know little on the topic as of yet.

<sup>4</sup> Bermudez-Otero (2003) describes the acquisition of alternations in stratal OT – where the grammar uses multiple derivations (with every stratum's ranking different) but EVAL remains parallel as in classic OT – and there, alternations do indeed help...

**IV. Proposal: Learning Hidden Rankings from Winners**

**The order of the approach**

- (i) first, learner acquires F >> M and unhidden M >> M rankings, using errors
  - (ii) then learner determines remaining hidden M rankings – proposal below
  - (iii) still later, alternations and unfaithful URs are set up
- ... this is a proposal about step (ii)

Starting with (23) again, the end state of phonotactic learning:

23)  $\mathcal{H}_{\text{phono}}$  { \*V-STOP-V, \*XI } >> \*ç >> { ID-CONT, ID-PLACE }

The learner's unanswered question:

**Does restrictiveness require a ranking of {M1, M2}?**

here: { \*V-STOP-V, \*XI }

**Step One: Examine Failed Candidates and Construct ERCs**

Look for winners – stored observed forms -- whose failed candidate set includes losers to which M1 or M2 assign a W and the other assigns an e:

| 25) /input1/ | *V-STOP-V | *XI | *ç | IDENT [CONT] | IDENT [VELAR] |
|--------------|-----------|-----|----|--------------|---------------|
| ☞ winner1    |           |     | ?  |              |               |
| loser2       | *!        |     | ?  | ?            | ?             |

| 26) /input3/ | *V-STOP-V | *XI  | *ç | IDENT [CONT] | IDENT [VELAR] |
|--------------|-----------|------|----|--------------|---------------|
| ☞ winner3    | *         | **   | ?  |              |               |
| loser4       | *         | ***! | ?  | ?            | ?             |

With these losers we build 'constructed ERCs':

For (25): - take /loser2/ as the input  
- construct the ERC: /loser2/ → [winner1]~[loser2]

In this case, words like [baçi] have both kinds of losers in their candidate set:

27) /baçi/'s candidate set: baçi, **baçi**, **baxi**, paçi, baHi... (\*\*baki)

| 28) /baçi/     | *V-STOP-V | *XI | *Ç | IDENT<br>[CONT] | IDENT<br>[VELAR] |
|----------------|-----------|-----|----|-----------------|------------------|
| a) <b>baci</b> | *!        |     |    | *               |                  |
| ↻ baçi         |           |     | *  |                 |                  |
| b) <b>baxi</b> |           | *!  |    |                 | *                |

From which we can construct two ERCs, taking each **loser** as the winner:

| 29) input | winner~loser | *V-STOP-V | *XI | *Ç | ID<br>[CONT] | ID<br>[VELAR] |
|-----------|--------------|-----------|-----|----|--------------|---------------|
| a) /baci/ | baçi ~ baci  | W         |     |    | L            |               |
| b) /baxi/ | baçi ~ baxi  |           | W   | L  |              | L             |

(Recall that \*\*baki is not in the candidate set for this winner [baçi]!)

These constructed ERCs in (29) don't yet teach us anything NEW

- the current grammar already prefers the winners to the losers
- (29a) has simply created evidence for an initial state M >> F rankings

### Step Two: Building the complement of the constructed ERCs <sup>5</sup>

For each constructed ERC (as in 29):

- take the 'loser' as an input, and build its candidate set (30)
- again, consider this loser as the **winner**
- and find any losers with the opposite violation profile  
e.g. faring worse on M2 but not M1 -- as in 31)

30a) /baxi/'s candidate set: [baxi] [baçi], [baki], [bayi], [bahi] etc...

b) /baci/'s candidate set: [baci], [baçi], [baki], [baçi], [bahi] etc...

<sup>5</sup> There is also an additional step – checking to make sure that these losers are in fact mapped in the phonotactic grammar to these winners... that is, checking that /baxi/ doesn't actually map to [baxu] or something else...

Reasoning from (30a)

- we built an ERC from the loser [baxi] because it fared worse on \*XI (28c)
- is there a loser for winner [baxi] which fares worse on \*V-STOP-V? **Yes:**

| 31) /baxi/  | *V-STOP-V | *XI | *Ç | ID[CONT] | ID[VELAR] |
|-------------|-----------|-----|----|----------|-----------|
| <b>baxi</b> | *!        |     |    | *        |           |
| ↻ baxi      |           | *   |    |          |           |

So we have a third constructed ERC:

| 32) input | winner~loser | *V-STOP-V | *XI | *Ç | ID<br>[CONT] | ID<br>[VELAR] |
|-----------|--------------|-----------|-----|----|--------------|---------------|
| a) /baxi/ | baxi ~ baxi  | W         | L   | e  | L            | e             |

[In this case, 30b won't produce any new ERCs, because to get from /baci/ to a form that violates \*XI requires violating *both* IDENT constraints...]

### Step Three: Add the set of constructed ERCs to the Support and learn

33) Combining the pre-existing Support and the constructed ERCs

|       | /input/ | winner~loser | *V-STOP-V | *XI | *Ç | ID<br>[CONT] | ID<br>[VELAR] |
|-------|---------|--------------|-----------|-----|----|--------------|---------------|
| (22)  | /çi/    | çi ~ xi      |           | W   | L  |              | W             |
| (29a) | /baci/  | baçi ~ baci  | W         |     |    | L            |               |
| (29b) | /baxi/  | baçi ~ baxi  |           | W   | L  |              | L             |
| (32)  | /baxi/  | baxi ~ baxi  | W         | L   |    | L            |               |

Stratum 1: Step One installs \*V-STOP-V

errors (29a) and (31) resolved, can be ignored

Stratum 2: Step One installs \*XI

errors (22) and (29b) resolved, can be ignored

Stratum 3: Step One installs \*Ç

Stratum 4: Step Three installs all remaining Faith ... and terminate.

34)  $\mathcal{H}_{\text{final}}$  \*V-STOP-V >> \*XI >> \*Ç >> {ID-CONT, ID-PLACE}

**Success – the hidden ranking. Now – to assess.**

## V. Insufficient Discussion

Three broader questions about this proposal:

Q1. *What mistakes could it make?*

- will it ever assign a ranking that is less restrictive than the target, which errors could then never overcome?

Q2. *What is the role of 'negative evidence'?*

- reasoning back from observed winners to failed losers with certain violation profiles
- what's the cost of such a search?
- what sacrifice is this to the error-driven endeavour?

Q3. *Should the learner simply be biased to prefer hidden rankings?*

With respect to Q3 – here is the opposite, incorrect ranking from (19):

| 19) Step One:<br>/maki/ | *XI  | *V-STOP-V | IDENT<br>[CONTIN] | IDENT<br>[PLACE] |
|-------------------------|--|-----------|-------------------|------------------|
| ma.ki                   |  | *         |                   |                  |
| ma.xi                   | *!   |           | *                 |                  |
| ma.çi                   | <b>not in the candidate set for /maki/</b> |           |                   |                  |

If this ranking were learned definitely, we'd have this language:

35) Language with (19)'s ranking

/baxi/ → [baçi]      but: /baki/ → [baki]  
 /baçi/ ↗

*Generalization:*

- the only intervocalic stops are those which, if lenited, would also palatalize
- maybe it's just as well to bias the learner away from such languages?

**But: here is an example where both rankings seem warranted:**

McCarthy (2008a) on cluster simplification:

Q: why are word-medial clusters /VC<sub>1</sub>C<sub>2</sub>V/ usually reduced to VC<sub>2</sub>V/

A: because deletion takes two HS steps:

- first coda debuccalization      /pat.ka/ → [paH.ka]
- then placeless-C deletion      /paH.ka/ → [pa.ka]
- ... and /pat.ka/ → \*[pat.Ha] is not harmonically improving

36) *Harmonic improvement on /patka/*

| /pat.ka/                         | CODA<br>CONDITION | HAVE<br>PLACE | MAX<br>[PLACE] | MAX |
|----------------------------------|-------------------|---------------|----------------|-----|
| pat.ka<br>... less harmonic than | *!                |               |                |     |
| paH.ka<br>... less harmonic than |                   | *!            | *              |     |
| pa.ka                            |                   |               | (*)            | *   |

37) The hidden ranking: CODA COND >> HAVE PLACE

*required to debuccalize before deleting*

*Result:*              \*patka              \*paH.ka              [pa.ka]

38) The reverse ranking: HAVE PLACE >> CODA COND

*blocks debuccalization to begin with...*

*Result:*              [pat.ka]              \*paH.ka              [pa.ka]

**Upshot: some hidden rankings are reversible, without pathology...**

- in fact, local blocking is something HS is in part designed to predict
- but some such patterns are desirable, while others might be pathological
- but which ones?
- and is this a question for the builder of CON? or for the learning algorithm?

## VI. Tentative Conclusions – are also the Take Home Message

- HS makes phonotactic learning much more complicated
- but it makes every input's candidate space much less complicated
- so this trade-off may be a key to successful HS learning

**Thank you very much.**

## References

- Bermúdez-Otero, Ricardo (2003). The acquisition of phonological opacity. In Jennifer Spenader, Anders Eriksson & Östen Dahl (eds.), *Variation within Optimality Theory: Proceedings of the Stockholm Workshop on 'Variation within Optimality Theory'*, 25-36. Stockholm: Department of Linguistics, Stockholm University.
- Emily Elfner. To appear. Stress-epenthesis interactions in Harmonic Serialism. In John McCarthy and Joe Pater (eds.) *Harmonic Grammar and Harmonic Serialism*. Equinox Press.  
[[http://www.people.umass.edu/eelfner/elfner\\_2011\\_Stress-epenthesisHS.pdf](http://www.people.umass.edu/eelfner/elfner_2011_Stress-epenthesisHS.pdf)]
- Gurevich, Naomi. 2004. *Lenition and Contrast: The Functional Consequences of Certain Phonologically Conditioned Sound Changes*. Outstanding Dissertations in Linguistics. Garland Publishing, New York, NY
- Hayes, Bruce. 2004. Phonological Acquisition in Optimality Theory: the early stages. In Kager, Rene, Joe Pater & Wim Zonneveld, (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.
- Jesney, Karen 2008/To appear. In Suzi Lima, Kevin Mullin & Brian Smith (eds.), *Proceedings of the 39th Meeting of the North East Linguistics Society (NELS 39)*. Amherst, MA: GLSA
- Kazazis, Kostas. 1972. Some remarks on palatalization in Modern Greek. *Research on Language & Social Interaction* 5(2): 330–332.
- Kaplan, Abby. 2010. *Phonology shaped by phonetics: The case of intervocalic lenition*. Ph.D. Dissertation, UC Santa Cruz.  
[<http://home.utah.edu/~u0703432/CV/dissertation.pdf>]
- Kimper, Wendell. 2011. Locality and globality in phonological variation. *Natural Language and Linguistic Theory* 29(2): 423-465.
- Kiparsky, Paul. 1968. Linguistic universals and linguistic change. In Emmon Bach and Robert Harms (eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehart & Winston: 170-202
- McCarthy, John J. 2000. Harmonic Serialism and Parallelism. In M. Hirotani, A. Coetzee, N. Hall and J.-Y. (eds.) *Proceedings of NELS 30*. Amherst: GLSA: 501-524.
- McCarthy, John J. 2007. Restraint of analysis. In Sylvia Blaho, Patrik Bye, and Martin Krämer (eds.) *Freedom of Analysis*. Berlin: Mouton de Gruyter: 203-231.
- McCarthy, John J. 2008a. The gradual path to cluster simplification. *Phonology* 25:271-319.
- McCarthy, John J. 2008b. The serial interaction of stress and syncope. *Natural Language & Linguistic Theory* (26)3: 499-546.
- McCarthy, John J. 2009a. Classified Bibliography of Works on OT with Candidate Chains (OT-CC) and Harmonic Serialism (HS). Ms, UMass Amherst. [ [http://works.bepress.com/john\\_j\\_mccarthy/102](http://works.bepress.com/john_j_mccarthy/102)]
- McCarthy, John J. 2009b. Studying Gen. *Journal of the Phonetic Society of Japan* 13(2).
- Prince, Alan and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Oxford, UK: Blackwell.
- Prince, Alan and Bruce Tesar 2004. Learning Phonotactic Distributions. In R. Kager, W. Zonneveld, J. Pater, (eds.) *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, UK: CUP.
- Pruitt, Kathryn. 2010. Serialism and locality in constraint-based metrical parsing. *Phonology* 27(3): 481-526. Smolensky, P. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry* 27: 720–31.
- Staubs, Robert and Joe Pater. (to appear). Learning serial constraint-based grammars. To appear in J. J. McCarthy and J. Pater (eds.) *Harmonic Grammar and Harmonic Serialism*, Equinox.  
[<http://people.umass.edu/pater/staubs-pater-serial-learning.pdf>]
- Tessier, Anne-Michelle. 2009. Frequency of Violation and Constraint-based Phonological Learning. *Lingua* 119 (1): 6-38.
- Tessier, Anne-Michelle. 2011. 'Modeling gradual learning in serial and parallel phonological grammars.' Poster presented at the *Child Phonology Conference*, June 2011, University of York, York, England.  
[[http://www.ualberta.ca/~annemich/Tessier2011\\_ICPCPoster.pdf](http://www.ualberta.ca/~annemich/Tessier2011_ICPCPoster.pdf)]

### Appendix: Alternations that reveal hidden rankings?<sup>6</sup>

Suppose the language from sections 3-4 had revealing alternations:

- |    |               |                    |
|----|---------------|--------------------|
| 1) | a) Bare Stem: | b) Derived Form    |
|    | /mad/ → [mad] | /mad + i/ → [madi] |
|    | /max/ → [max] | /max + i/ → [maçi] |
|    | /mak/ → [mak] | /mak + i/ → [maçi] |

If we have worked out the correct URs for 1b), could they teach us the hidden ranking, just making errors with the phonotactic grammar?

With an input /maki/, the variable behaviour of the grammar will be observed:

- 2) /maki/ → [maçi] OR \*[maki]

But when looking at the error version, the learner now has to decide how to represent this *winner*.

- 3) /mak/ → [maçi]?

*This winner is not in this input's candidate set!* So what can we do?  
In effect, we need to know the middle step.

We could invent a mechanism to create that middle step – reasoning about faith violations to create intermediate members on the chain:

- 4) /maki/ → ?? → [maçi]

If we knew to go via /maxi/, we would learn the hidden ranking fine:

| 5) input  | winner~ loser | *V-STOP-V | *XI | *Ç | ID [CONT] | ID [PLACE] |
|-----------|---------------|-----------|-----|----|-----------|------------|
| /mak + i/ | maxi ~ maki   | W         | L   |    | W         |            |
| /maxi     | maçi ~ maxi   |           | W   | L  |           | W          |

These two errors will teach the learner everything, just as in (32)

<sup>6</sup> Thanks especially to Igor Yanovich for pressing me on this point, and helpful ensuing discussion.

- 6) Correct Result: \*V-STOP-V >> \*XI >> \*Ç >> ID[CONT], ID[PLACE]

This search for an intermediate candidate is definitely not hopeless

- because of the finite candidate set
- and it's probably rather simpler than the one needed in sections 3 and 4
- *but you can't guarantee the relevant alternation will occur*

### Upshot: alternations cannot be relied on to reveal hidden rankings

And since we need a mechanism for phonotactic hidden rankings anyway, we needn't equip the learner with *two* similar devices for finding crucial intermediate losers

The present proposal could help the learner find these hidden rankings before alternations raise the problem of building a winner in (3) here

*Of course:*

- *real alternations still need to be learnable*
- *so the problem of building winners, if not like (3) then how, must be solved*

Future, future work.