Vowel Harmony, Opacity, and Finite-State OT

Samuel R. Bowman
The University of Chicago

July 22, 2011

Abstract: In this paper I propose cyclic and stratal extensions to Finite-State Optimality Theory as treatments for vowel harmony and for opaque phenomena more generally. I develop a model of grammar and an accompanying set of constraints within a software model built on the open-source PyPhon toolkit—a computational implementation of OT's GEN and EVAL algorithms.

0 Constraint-based grammars and typology

Since their introduction in the early nineteen-nineties, constraint-based grammars have been widely used in generative phonology for their ability to describe complex patterns in a way that is simple, cognitively plausible, and easily generalized across languages.

In any constraint-based grammar, the great bulk of the generative labor is carried out by violable constraints. In an idealized picture, these constraints are clear and principled descriptions of phenomena that are widely dispreferred across languages, like DEP: *do not epenthesize segments*. The job of the language user is, for every underlying form, to find a surface form that best satisfies the constraints that matter to their language, and to do so directly, without a sequential derivation.

Under the associated richness of the base hypothesis (Prince and Smolensky, 1993/2004), constraint-based grammars are expected to be able to produce an acceptable output for any input, even if (as in crosslinguistic borrowing) that input flagrantly violates basic structural principles of the target language. Because of this, it is quite common for analyses to focus on inputs for which no output satisfies all of the constraints. For example, when the grammar of a language like Japanese, which heavily restricts consonant clusters, is presented with an input containing a cluster of obstruents, it cannot produce a phonotactically well-formed output without violating some constraint against insertion (like DEP, above), deletion, or substitution.

The two major constraint-based grammar frameworks—Optimality Theory (Prince and Smolensky, 1993/2004, OT) and Harmonic Grammar (Legendre et al., 1990, HG)—

diverge on what to do in the frequent cases where there is no single ideal candidate. HG and older related connectionist frameworks (e.g. Goldsmith and Larson, 1990) weight constraints and select the output whose weighted sum of violations is lowest. In OT, constraints are ranked, and the optimal output is the one which best satisfies the highest-ranked constraint. If the highest-ranked constraint is not violated or if multiple candidates violate that constraint equally, the optimal output is the candidate which minimally violates *the highest ranked constraint that is not equally satisfied by another output*.

Constraint-based grammars, especially OT, have often been presented alongside the assumption that there exists a universally available constraint set Con. Since all living languages continue to exist because they have been learned by children, children must be able to somehow access all the attested constraints, through either innate knowledge or a reliable mechanism of inference. It is generally assumed that most constraints can be ranked or weighted freely, and that by proposing set of constraints, one commits oneself to addressing the entire family of languages describable by rankings (or weightings) of that set. Proposed constraint sets are generally evaluated both on their ability to correctly and comprehensively generate attested patterns in a minimally ad-hoc fashion, and on their *inability* to generate implausible patterns.

However, the task of separating the implausible patterns from the merely unattested is quite fraught. Between the few thousand mutually-related languages that have been spoken in recent history and tightly constrained laboratory studies, we have only very limited information about what kinds linguistic phenomena can and cannot emerge. However, there is evidence (Bane and Riggle, 2008) that concise and well motivated constraint sets can produce OT factorial typologies (the sets of languages describabable by some constraints) which mirror real-world language distributions quite closely.

Imposing the strict restriction that typologies only represent attested phenomena is quite harsh, and can cost a framework a great deal of explanatory power. As an example, a remark by Chomsky in his Minimalist Program (1995) constituted a common early criticism of OT: He points out that in any reasonably comprehensive set of phonological OT constraints, ranking enough markedness constraints above all of the faithfulness constraints can produce the non-language wherein every input is realized as a single, predictable form (e.g. /lɪŋˈgʷɪstɪks/ → [ba])[1]. Setting aside the possibility that such a reductionist phonology may accurately capture a phase of early childhood learning, allowing such degenerate languages to appear in a phonological typology need not damn that typology. If a set of plausible constraints can produce a degenerate language, it is possible that that degenerate language can be learned and produced by human speakers and that it is degenerate for reasons external to the proposed constraints. The [ba] language would fail to be transmitted between speakers because its efficiency as a means of communication, which is not typically

[1] He then goes on to claim (in what must be a misunderstanding; see McCarthy and Prince, 1994) that the nature of optimization is such that *only* that language can be generated.

a focus of generative grammar, would be very poor. I intend to focus on analyses which provide well-motivated descriptions of attested phenomena, and attend only secondarily to the unattested languages which result.

For any given set of constraints and set of inputs, HG generates at least as many outputs as OT, and typically a good deal more: For a finite lexicon of input forms, any OT ranking can be captured by an HG weighting with sufficiently large spans between the weights (Bane and Riggle, 2009). In contrast, weightings allow for phenomena which cannot be captured by rankings. Multiple violated constraints can form what Pater et al. (2007) call a *gang*, and overwhelm a higher-weighted constraint's effect on the output score. This can result in an tableau for which the optimal output does not satisfy the most prominent constraint as well as some other candidate, a situation which is strictly impossible under OT.

These gang effects often predict typologies littered with unattested interactions, but they can be useful. They allow us to represent cumulative markedness effects, which have been reported in a number of domains, including phonotactic well-formedness judgements (Albright, 2008) and Japanese geminate devoicing (Pater, 2007). However, while none of what is described below is strictly incompatible with weighting, these additional *gang effect* candidates are not obviously useful for the task at hand, and HG rarely arises in the harmony literature, so I will focus on OT in this paper.

In any case, both OT and HG define algorithms for choosing optima (termed EVAL), and both of these algorithms can be implemented in artificial neural networks quite readily (Smolensky and Legendre, 2006), lending credibility to both as models of cognitive processes. This work will take advantage of the fact that EVAL need not stand alone: It can also serve as the still-implementable core of a more complex grammar.

1 Computing constraints and generating contenders

Constraint-based grammar, as it is commonly discussed, contains a significant black box. The function that produces the output for a given input and a given set of ranked (or weighted) constraints must produce an output that is preferable to *any other possible string of segments*. Even if the grammar were restricted to considering outputs consisting of segments that are licit in the language, this set would still be infinite, and so the function needs some substantial tricks up its sleeve if it is to find an optimal output in the fraction of a second that it takes a speaker to pronounce a word.

This limits the kind of work that can be done by hand within OT. If the size of a constraint set is small, or the constraints in it are neatly restricted and non-interacting, then the optimal output for some ranking and input can generally be found by hand, but this rapidly becomes difficult as either the number or the complexity of the constraints grows.

In many constrained cases, however, there are tricks available to allow one to generate complete OT typologies computationally in finite and reasonable time. Various limitations on OT have been proposed to facilitate this, including limitations both on both candidates and constraints. From its inception, much work on OT has focused on formulating the theory in sufficient detail to be computationally implementable, and restricting it sufficiently to render it tractable.

I follow the approach particular approach stemming from Ellison (1994), Eisner (1997), Karttunen (1998), Frank and Satta (1998), and Smith and Eisner (1999) and assume that the entire evaluation process can be represented as a finite-state machine. This requires the modest assumtion that the constraints themselves also be finite-state, and allows me to use Riggle's (2004, 2009) algorithms to directly generate exactly those candidates which can win under some ranking (or optionally, some weighting) of a set of constraints.

These Finite-State OT grammars model optimization as the search for the shortest path through a graph. The graph is a finite-state transducer that transforms linguistic input–output pairs into sets of violation counts, and it can be built through the composition of graphs representing individual constraints. This formalism guarantees the computability of generation and evaluation, while preserving all of the fundamental assumptions of core Prince and Smolensky (1993/2004) OT, including that of an unlimited field of candidates. However, the approach imposes some real restrictions on the power of the grammar: It cannot represent some constraints, including the widely used (but often criticized) quadratic alignment constraints (McCarthy and Prince, 1993, Eisner, 1997, Biró, 2003) and the multi-segment base-reduplicant faithfulness constraints used in some treatments of reduplication (McCarthy and Prince, 1995).

Unfortunately, these algorithms can quickly exhaust any computing resources when presented with large enough constraint sets. Especially when building grammars of the sort needed to generate complete typologies rather than single languages, both the run time required to convert a set of constraints into a single finite-state machine and the size of that finite-state machine increase very rapidly with the size and complexity of the constraint set[2]. For the constraint sets discussed below, this conversion takes between seconds and hours, and I thus abbreviate the tableaux by removing from consideration constraints which are not plausibly relevant to the phenomena at hand.

My immediate material goal is to rigorously investigate the typology and complexity of a basic model of vowel harmony. In order to have a completely precise picture of the consequences of my proposed constraint set, I will build on Riggle and Bane's (2011) PyPhon, a toolkit which implements the algorithms I have been describing

[2]Machine size—the appropriate measure of constraint complexity for this purpose—is largely a function of how much the constraint cares about the environment surrounding a potential violator. Thus, AgreeV, which needs to know about adjacent vowels, is more complex than any *[αf, βg], or even *[αf, βg]&IO-Ident constraint, both of which care only about a single violating segment.

for Finite-State OT and HG. The preexisting tools take as input a description of the constraints and symbols of a model, as well as a list of test inputs, and any ranking or weighting restrictions. For each input, MakeTableaux generates all of the contenders available under the specified framework. From this, Generate can produce a typology of all possible languages distinguishable by the given inputs, and a specification of the ranking conditions necessary to produce each language. I include the PyPhon-compatible specifications of both the constraints and symbols of my model in the appendices.

It should be noted that in this work, as in PyPhon, a language is defined as a set of I-O mappings. Thus, one language can be described by multiple (or, in some cases, all) rankings of some constraints, but if two rankings differ in their output for at least one of the *provided* inputs, then those rankings are classified as belonging to different languages. This places a burden on the grammar writer—which I attempt to meet—to test any set of constraints with a broad enough range of inputs to ensure that all of the possible language classes generable by those constraints become visible.

As it stands at the time of writing, PyPhon is only capable of generating typologies from traditional single-round evaluations. To straightforwardly account for vowel harmony, this will turn out not to be enough.

2 Opacity and learnability

Opaque phonological patterns are—to loosely paraphrase Kiparsky (1976)—those which are active in derivation, but which are not consistently true on the surface. They are typically (though not necessarily) framed within serial theories as cases where a later process modifies the output of an earlier process, rendering that earlier process invisible, and making it difficult for the learner to correctly generalize about when the earlier process should apply.

Opacity is a reliable source of argument among the authors of constraint-based grammars. Canonical OT (i.e., Kager, 1999) conspicuously fails to account for many classic cases of opacity, and straightforward additions to the theory that address these phenomena—like sympathy theory (McCarthy, 1999)—tend to introduce problems of their own. Sympathy constraints, used in parts of Baković's analysis, posit the existence of a two-stage derivation within each application of Eval. In the first evaluation a *sympathetic* candidate is chosen which best satisfies a designated selector constraint, and then that sympathetic candidate is added to the second derivation as a secondary input, to which candidates can be *sympathetic* (faithful), and the selector constraint is removed from its privileged position. Both OT's elegant typological predictions and its relatively straightforward framing of the learning problem crumble with the addition of sympathy constraints (Kiparsky, 2000), motivating attempts to find a better way to handle opaque phenomena.

Harmony systems have been a fertile source of proposals about linguistic structure largely because generalizations about harmony can be so hard to capture elegantly. In this sense, harmony is opaque. Stem control, blocking, and transparency have been explained under numerous frameworks, many of which require considerable hidden structure. Many proposals like Walker (2001) have used a hidden correspondence tier or a related autosegmental structure (Goldsmith, 1976), which must be established before harmony-producing constraints or rules can apply.

Baković (2000) provides a comprehensive analysis of harmony systems with a focus on descriptive simplicity and successfully eschews autosegmental representation and underspecified inputs as unnecessary. However, he provides substantial evidence that a well-motivated analysis requires *some* additional tools, and uses cyclic evaluation (which I adopt), targeted constraints (Wilson, 2001), and sympathy constraints (McCarthy, 1999).

Kiparsky and Pajusalu (2003) provide the only alternate analysis of vowel harmony which manages to largely avoid problematic amounts of hidden structure. They use what they call a Generalized Marked Harmony constraint, a *non-local* conjunction of a local agreement constraint and a markedness constraint, which penalizes a contender if for each of the two constraints, there is some segment in the contender which violates that constraint. Such a constraint can be translated into a finite-state representation, but at a substantial cost: The need to handle partial violations of constraints grows the complexity of EVAL quite quickly. My analysis addresses these problems differently, with an eye to maximal ease of implementation.

Cyclic and stratal evaluation, both of which I will introduce in the coming sections, are steps towards addressing opaque phenomena within constraint-based grammars. They are sufficient for describing vowel harmony using only simple finite-state constraints, and can be implemented using existing Finite-State OT algorithms.

3 A constraint-based model of vowel harmony

3.1 Vowel harmony

The basic principle of harmony is quite simple: In languages with harmony for some feature f, all segments in a word must surface with the same value of f.

Vowel harmony is harmony that requires featural agreement only between vowels. This can be analyzed either as harmony which bypasses consonants in its effects, or harmony for features which are not contrastive in consonants. Although harmony constraints are not often labeled as such[3], many do implicitly assume the former analysis. I take this approach, but explicitly mark all vowel-specific constraints as such.

Of course, were harmony as simple as uniform agreement across a word, it would not be as problematic or interesting of a case study as it is. In most languages

[3] This is visible in Baković's (2000) example 59/p.45 for *[+VOI] and suggested in example 89/p.78 for AGR[RD].

nominally displaying vowel harmony, morphological boundaries, inventory control and specially marked affixes conspire to mask or complicate many instances of the phenomenon. These conflicts are a clear case of derivational opacity, and I will review and account for a number of such cases below. For a general survey of vowel and consonant harmony, see Archangeli and Pulleyblank (2007).

3.2 Basic constraints

I borrow four fundamental constraint families from Baković (2000): AGREE, IO-IDENT, *[αf, βg] and the local conjunction *[αf, βg]&IO-IDENT.

The two most fundamental constraints are formulated as follows:

AGREEV-f:

> Assigns one mark to any pair of vowels in the input word which are not interrupted by any other intervening vowels, and which differ in their specification for f.

Agreement constraints are the backbone of harmony, penalizing words whose vowels are not uniformly specified for a given feature. I describe here a family of constraints following the same structure, but penalizing disagreements for any single vowel feature. No known language has enough contrastive vowel pairs for all of these constraints to be active at once, but many languages show harmony for one or two features, including HIGH, BACK, ROUND, ATR, and NASAL.

This constraint family is among the simplest proposed for vowel harmony. It requires no hidden structure: There is no additional tier of representation to mark which segments harmonize, nor is there any specially marked head from which harmony spreads.

The constraint family described here acts strictly locally, imposing one violation for each change in value between *adjacent* vowels. The constraints thus penalize a word with a medial disharmonic vowel twice as heavily as one with a peripheral disharmonic vowel: AGREEV-BACK would assign two violations to the invented example [bibebib**u**be] (BACK: $- - - * + * -$), and only one to [bibebebib**u**] (BACK: $- - - - * +$), which has the same number of vowels. It is partnered with a faithfulness constraint:

IO-IDENTV-f:

> Assigns one mark to any vowel whose specification for f in the output differs from its specification in the input.

The IO-IDENT family of faithfulness constraints allows us to limit the application of harmony by penalizing exactly the feature-value changes that harmony requires. I follow Baković's well-argued assertion that input symbols should be fully specified for all features. This bans underspecified affixes, and allows any segment to be potentially disharmonic, causing a clash between violations of AGREE and IO-IDENT.

For the tableaux I will present here, I am restricting my attention to harmony that occurs through feature value changes. This should not be a significant limitation in practice: I am aware of no harmony system wherein disharmonic affixes are repaired by any other means. However, assuming that the purportedly universal deletion constraint MAX is available, the typology should include deletion-driven harmony, wherein vowels at risk of being disharmonic can be dropped from a word. For the purposes at hand, I will only consider grammars wherein MAX is held to be inviolable, or equivalently, ranked above all the relevant markedness constraints, including ARGEEV-f.

We can create a simple harmony system based on only these two constraints, each instantiated for the same feature. A disharmonic input can only have two possible outputs under such a grammar. Either the entire word harmonizes with the feature value that occurs more often in the word, or no spreading occurs. The only exception to this occurs when the number of segments with each value of the harmonizing feature is equal (i.e., IPA [yi], ROUND: +−). In the absence of any other constraint, there will be two tied contenders, each harmonizing to one of the two values. Such a grammar is demonstrated here:

/yiyyi/	AGREEV-ROUND	IO-IDENTV-ROUND
a. yiyyi	***	
b. yyyyy		**

Here, the two simple types of contender are neatly sorted into two languages. In one language (the one in which *a* wins), IDENTV dominates AGREEV, and inputs are faithfully preserved. In the other (the one in which *b* wins), the ranking is reversed, and harmony occurs. It is worth noting that this simplistic harmony system exhibits a property that Baković claims is unattested: majority rule. Since no constraint in this simplistic model can decide between the two values of the harmonic feature f, the number of violations of IDENT decides, and the value which spreads across the word is the one represented by the majority of the input segments.

The contents of all of the tableaux presented in this paper are generated within PyPhon, and are complete in the sense defined in Riggle (2008): They contain all of the candidates which are potentially optimal under some ranking of the constraints shown. Where I am using a tableau simply to demonstrate the range of contenders available for some input (as above), I will follow the convention of separating all constraint columns with dashed vertical lines. Where I am showing the conditions under which a particular phenomenon emerges, I will order the constraints by ranking, and separate crucially ranked pairs of constraints with solid lines. Inputs and candidates in sample tableaux follow the IPA, though precise feature values are specified in Appendix A. Constraint definitions follow in Appendix B.

3.3 Choosing the harmonic feature value

I accept Baković's claim that the simple majority rule system shown above is unattested, and that in every language with harmony one of two systems is active in selecting the harmonic feature value. These two systems which I intend to account for are dominant-recessive harmony and stem-controlled harmony. In languages like Kalenjin and Nez Perce with dominant-recessive harmony, any segment bearing the pre-specified dominant polarity value of a feature will spread that value as far as it can. In contrast, languages like Turkish show cyclic stem-control, wherein as each affix is attached to the stem, it takes on (if possible) the harmonic feature value of the stem to create a new stem for the next more peripheral affix. This creates a spread of harmonic values out from the root and through each affix in order.

Majority rule emerges naturally with the presence of a local agreement constraint, and I do not stipulate any mechanism to rule it out. While its occurrence is unsettling, it rarely survives in competition with other possible methods of harmonic value selection: The vast majority of specific input sets and constraint sets that I have tested either cannot generate majority rule or generate only for a very limited set of rankings and inputs.

3.3.1 Dominance and re-pairing

Baković explains dominance using a one-directional faithfulness constraint—one which penalizes a change of a feature polarity α into its opposite $-\alpha$ in a certain context, but does not penalize a change in the other direction (i.e., $-\alpha \rightarrow \alpha$). This can be treated as the local conjunction (after Smolensky, 1993) of a complex markedness constraint and a simple faithfulness constraint: *[αf, βg]&IO-IDENTV-f.

If such a constraint dominates the simple IO faithfulness constraints, then an input vowel specified [βg] will force adjacent vowels to harmonize to [-αf]. Local conjunction theory impels us to admit the constituent constraint *[αf, βg] to our inventory as well, and appropriately so: It is necessary for inventory control, and offers a simple account for opaque vowels. The new constraints are formulated as follows:

*[αf, βg]-V:

> Assigns one mark to any vowel with the polarity α for feature f and β for feature g.

*[αf, βg]-V&IO-IDENTV-f:

Assigns one mark to any vowel with the output polarity α for feature f and β for feature g and the input polarity $-\alpha$ for feature f.

With this new conjoined constraint instantiated as *[+Round, -High]-V&IO-IdentV-Round, the correct ranking can allow a single /a/ ([-High, -Round]) to force the rest of the word to harmonize to [-Round]. /o/ ([-High, +Round]), however, cannot force harmony in this way, since its unfaithful [-Round] counterpart is not penalized.

Re-pairing also emerges organically with the addition of this constraint. Re-pairing is the well-attested phenomenon (Baković, 2000) wherein harmony interacts with these conjoined feature markedness constraints to change a segment into another segment which differs in more than just the harmonic feature. It can be seen in the examples that follow.

Local conjunction theory (as laid out in Smolensky, 1995) proposes that any such conjoined constraint must universally outrank its constituent constraints (i.e., $A\&B \gg \{A, B\}$). This new constraint is simple enough that there may be other plausible framings of it[4], but this ranking restriction turns out to be somewhat useful. The candidates that it rules out are, in every example that I could find, unattested. I present these candidates in parentheses in the sample tableaux that follow.

/iayyy/	Agr-Rd	IO-Hi	IO-Rd	*[+Rd, -Hi]&...
a. iaiii			***	
b. yyyyy		*	**	
c. (yoyyy)			**	*
d. iayyy	*			

Figure 1: A complete tableau for /iayyy/. The abbreviated constraint is *[+Round, -High]&IO-IdentV-Round. Where I abbreviate constraints in this family, the first feature specified in the bracketed markedness constraint is the feature that the conjoined faithfulness constraint applies to.

In *a*, /a/ forces harmony, despite minority status, demonstrating dominance-based harmony. The other three candidates show (*b*) majority rule with re-pairing, (*c*) majority rule, and (*d*) faithfulness. In contrast, the below tableau has no dominant segments, and so shows only majority rule and faithfulness:

[4]See, for example, Rubach's (2003) IdentC[-Back].

/uoiii/	Agr-Rd	IO-Hi	IO-Rd	*[+Rd, -Hi]&...
a. iaiii			**	
b. uoiii	*			

Figure 2: A complete tableau for the analogous /uoiii/. /o/ cannot force harmony.

Before I proceed with any further examples, I must raise some representational issues:

Kalenjin uses vowel duration as a contrastive feature. I choose to represent long vowels here computationally as adjacent pairs of identical vowel symbols, rather than using a length feature and establishing separate long vowel symbols or implementing a lengthening symbol as in the IPA. Abstractly, this is equivalent to treating all long as splittable "false geminates" as in Hayes (1986), but Hayes's analysis is couched in autosegmental representations, and the difference between false and true geminates within OT if no phonological processes attend to length directly. The analysis presented here is agnostic: While long vowels to incur two faithfulness violations where a short vowels only incur one, these added violations do not significantly change the typology. The constraints presented here cannot split or shorten such a vowel pair, and the extra violations can only affect the harmonic feature value under unattested majority rule rankings.

For simplicity of presentation, I omit violation profiles in my example tableaux for constraints which are not necessary for the success of at least one contender. In the immediately following tableaux this includes the implicitly present but low-ranked *[−Low, −ATR]. I generally consider all of the constraints which are plausibly applicable in a given tableau, and then show those that turn out to be necessary, but as is standard practice, my tableau do not consider of all of the constraints allowed by my proposed analysis: Enumerating every possible constraint from each family (i.e., ArgeeV-High, ArgeeV-Low, ArgeeV-Back, ArgeeV-ATR, ArgeeV-Round...) would be computationally taxing and yield bafflingly large tableaux.

Baković (2000) cites Kalenjin as a prominent example of dominant-recessive harmony for the feature ATR, and provides real-world examples of the phenomenon (p. 111). I use the constraint *[−Low, −ATR] to account for the across-the-board dominance of [−Low, +ATR] segments, and find that the model behaves as expected: It generates a language wherein harmony only occurs in the presence of such a vowel.

/kɪ+a#kɛr#/	Agr-ATR	IO-Bk	IO-Hi	IO-Lo	(1)	IO-ATR
a. ☞ kɪakɛr						

(1): *[-Low, -ATR]-V&IO-IdentV-Low

Figure 3: A complete tableau for Kalenjin /kɪ+a#kɛr#/.

In /kɪ+a#kɛr#/[5] ("I shut it"), no dominant segment is present, and so there is no conflict between faithfulness and harmony. The one contender shown is thus trivially optimal.

/kɪ+a#keːr#ɪn/	AGR-ATR	IO-BK	IO-HI	IO-LO	(1)	IO-ATR
a. ☞ kiɐkeːrin						***
b. (kɪakɛːrɪn)					**	**
c. kɪakaːrɪn		**		**		**
d. kɪakeːrɪn	**					

Figure 4: A complete tableau for Kalenjin /kɪ+a#keːr#ɪn/.

In the input /kɪ+a#keːr#ɪn/ ("I saw you"), /e/ ([-LOW, +ATR]) forces harmony under the same ranking, as expected.

3.3.2 Stem control

Stem-controlled harmony is not as straightforward as dominant-recessive harmony, and Baković's proposal—which I implement—requires a change to the basic architecture of the grammar: the introduction of cyclic evaluation. All of the major alternatives require at least as bold proposals, and cyclic approaches like the one at hand have been repeatedly proposed for cyclic evaluation from phenomena like stress assignment (Duanmu, 1997, Kiparsky, 2000) and chain shifts in consonant realization (Rubach, 2003).

In cyclically evaluated affixation, each time an affix is added to a word, the result is passed to EVAL. When a word has an equal number of affixes on each side, no analysis that I have found provides a principled way of choosing which to add first, and the data at hand are similarly agnostic. I add equally peripheral affixes simultaneously: Under cyclic evaluation, a root, one prefix and three suffixes would be introduced to EVAL in the following order: 1#1#1 + 2 + 3.

This analysis (unlike Baković's—see below) is strictly *sequentially local* in that no information is carried between cycles of evaluation besides the winning candidate itself. This means that constraints cannot make reference to the true underlying form of the root or of any previously affixed morphemes—only the output of the previous round. Furthermore, constraints cannot see any morphological boundaries besides those delimiting the affix which was just added.

For the fundamental constraint of stem control, Baković proposes the following family:

[5] I use "#" to represent the boundaries of the stem and "+" to represent affix-affix morpheme boundaries.

SA-IdentV-f:

> Assigns one mark to each segment in the output which differs in its value of f from the stem of affixation (i.e., either the output of the previous cycle, or the word root in the first cycle).

SA-Ident (S for "stem", A for "affixed form", after Baković) penalizes unfaithful mappings in the stem of affixation while allowing them in the newly added affix. This constraint can force each affix to harmonize with the root (which is typically internally harmonic), but not the reverse, regardless of the relative segment counts. However, since it is violable, dominance is still possible. If SA-Ident is not highly ranked, a *dominant* vowel in any section of the input can still force harmony once it is introduced.

It is worth noting that a constraint very similar to SA-Ident-f appears in Kiparsky and Pajusalu (2003) as IdentRoot-f. However, that analysis does not provide examples with multiple rounds of affixation, and so the distinction between the root morpheme (as their constraint title might suggest) and the stem of affixation (as in SA-Ident) as the domain for faithfulness is not made. Inkelas's (To Appear) Faith$_s$ constraints are also loosely equivalent if we adopt her convention that stem segments are marked with a higher confidence or strength then affix segments.

Under Baković's model, the constraints proposed thus far require us to introduce an additional demand on Eval. Since his analysis contains both SA and IO identity constraints, a correspondence relationship must be maintained between each stem segment and both its true underlying form and its previous output form. This additional mechanism may not be empirically necessary. I propose an analysis in which all correspondence is to the single input string of the current round. I define the input to each round as the concatenation of the previous cycle's output (the stem) and the underlying form of the new affix, and the resulting system can capture both identity constraints without looking outside the input and output strings of a single tableau. SA-Ident is violated only in the stem, and IO-Ident can be violated by any segment. SA-Ident and IO-Ident are both defined just as they were, only with the stipulation that the input that IO-Ident refers to need not be the underlying form.

This proposal, at least abstractly, differs from Baković's in its potential coverage, but no example surfaces in Baković's lengthy analysis that would distinguish the two approaches. Such a counterexample would need to be a tableau with three properties: The stem of affixation must differ from the underlying form, this difference must cause a further difference in the number of violations that at least one IO-Ident-family constraint assigns, and an IO-Ident constraint meeting that criteria must be ranked highly enough to potentially influence the output.

This modified proposal also greatly simplifies the generation of typologies. I do this by running my existing contender generation algorithm cyclicly. I first generate a typology of all the contender outputs for the stem. Then, for each contender, I add the next affix (or prefix-suffix pair), and evaluate again. Since I hold constraint rankings constant between cycles, the true contenders for the system are only those candidates for which some specific ranking or set of rankings can generate both the output and all the intermediate forms that led to it. Because of this, at each stage of contender generation after the first, we only consider candidates which are optimal under the ranking conditions that produced the input. In tracking this, we use PyPhon's elementary ranking condition sets (ERC sets, read [ɝk], proposed in Prince, 2002), which provide a way to concisely represent the full set of constraint rankings compatible with a specific optimal output.

In the following example, I create a typology around the hypothetical vowel sequence #y#a+i (Round: + − −). In the first round, the algorithm evaluates the stem with the first affix attached:

Cyclic Eval (Cycle 1 of 2):

/#y#a/	Agr-Rd	SA-Rd	IO-Hi	IO-Lo	IO-Rd	*[+Rd, -Hi]	(2)
a. yu			*	*	*		
b. ya	*						
c. ia		*			*		
d. yo				*	*	*	*

(2): *[-Round, -High]&IO-Ident-Round

Figure 5: A complete tableau for the stem and first affix #y#a.

In this round, we evaluate each contender from the previous round with an additional affix. Since this added affix is the only one remaining, the output of this round is the output of the entire system:

Cyclic Eval (Cycle 2 of 2):

/#yu#i/	Agr-Rd	IO-Rd	SA-Rd	IO-Hi	IO-Lo	*[+Rd, -Hi]	(2)
a. ☞ yuy					*		
b. yui	*						

Figure 6: A complete tableau for the stem–affix pair /#yu#i/ (from figure 5 candidate *a*) under the ranking conditions that yielded candidate *a* as the winner. Since it cannot win under these conditions, candidate *b* here is not a contender in this tableau, even though it can win under some ranking of these constraints.

The above tableau demonstrates stem control with re-pairing (since the output contains [u] rather than the more faithful but more marked [o]). Starting with other lines from figure 5, we get the following: /#ya#i/ maps to [yai] in the faithful output, /#ia#i/ maps to [iai] for pure stem control, and /#yo#i/ maps to [yoy] for dominance, which remains available in cyclic evaluation.

3.4 Opaque vowels

In a potentially confusing collision of terms, *opacity* refers to two very relevant but genetically unrelated phenomena. In the immediate context of harmony systems, *opaque vowels* are those which prevent the spread of the harmonic feature into more peripheral segments. Except in morphologically conditioned cases (which I do not address here), opaque vowels have no harmonic parter—a vowel which differsonly in its the harmonic feature value—in the surface inventory. Since richness of the base impels us to admit a hypothetical partner in input forms, the analysis must force vowels of either underlying feature polarity to take on the opaque value. Fortunately, opacity of this sort emerges naturally under stem control where a constraint of the form *[αf, βg]-V outranks Agree. The following sequence of tableaux demonstrate opacity as implemented with these constraints:

Cyclic Eval (Cycle 1 of 2):

i#y#o	*[+Rd, -Hi]	IO-Bk	SA-Rd	IO-Hi	Agr-Rd	IO-Lo	IO-Rd
a. iyo	*				*		
b. iie		*	*				**
c. ☞ yya					*	*	**
d. iya					**	*	*
e. iyu				*	*		
f. yyu				*			*
g. iye		*			**		*
h. iia			*			*	**
i. yye		*			*		**
j. yyo	*						*

Cyclic Eval (Cycle 2 of 2):

#yya#i	*[+Rd, -Hi]	IO-Bk	SA-Rd	IO-Hi	Agr-Rd	IO-Lo	IO-Rd
a. ☞ yyai					*		

Figure 7: Tableaux for the hypothetical input /i#y#o+i/ (round: −++−), including all contenders in the first round. In the second round, there is only one contender under the restricted set of rankings compatible with [yya] winning in the first.

Here [+Round] spreads from the stem to the prefix, but the marked /o/ blocks it from reaching the final suffx.

Opacity can be easily shown in real-world data using these constraints. The following represents the output of familiar constraints acting on Turkish, following Baković's example 86 (p. 76):

Cyclic EVAL (Cycle 1 of 2):

/#kul#lor/	SA-RD	IO-BK	IO-HI	*[+RD, -HI]	AGR-RD	IO-RD
a. kullur			*			
b. ☞ kullar					*	**
c. kıllar	*					****
d. kullor				*		

Cyclic EVAL (Cycle 2 of 2):

/#kullar#u/	SA-RD	IO-BK	IO-HI	*[+RD, -HI]	AGR-RD	IO-RD
a. kullaru					**	
b. ☞ kulları					*	**

Figure 8: Tableaux for Turkish /#kul#lor+u/ (Turkish orthography) showing stem control in cyclic evaluation.

3.5 Transparent vowels

Transparent vowels, like opaque ones, do not participate in harmony. They also, like opaque vowels, do not have harmonic partners. Unlike opaque vowels however, they allow the spread of harmony to continue past them, even when they themselves are disharmonic. If we hold the rest of our assumptions fixed, transparent vowels require us to either drop the assumption of strict locality for harmony, or else introduce a hidden correspondence tier from which transparent vowels can be removed.

There is, however, a more radical alternative which allows us to keep these simple constraints: Assume that transparent vowels *do* participate in harmony, but merge to a single surface form before being pronounced. Such an analysis would require abandoning the idea that each language consists of only a single OT grammar.

In LPM-OT (Kiparsky, 2000, a development of earlier stratal OT; short for Lexical Phonology and Morphology) there exists a universally defined set of sequentially evaluated grammatical levels, each with its own distinct language specific ranking (or weighting) of the constraints. In the original formulation, there are three levels: root optimization, lexical optimization, and postlexical optimization. I assume that roots are already harmonic in the cases under consideration, and so my analysis does not use this first tier. The other two tiers form a nice basis for an analysis of harmony compatible with transparent vowels. Lexical-tier evaluation is already hypothesized to be cyclic, and so transparency emerges from our existing constraints if harmony occurs on the lexical tier as before, and transparent vowels are corrected by *[αf, βg]-V constraints in an otherwise inactive postlexical grammar.

It is in fact possible to represent such a grammar using only a single ranking, but at the cost of some new hidden structure. Differences in postlexical and lexical rankings can thus be explained simply by making two copies of each constraint, ranking half for postlexical derivations, and half for lexical ones, and combining the two rankings. Each input, then, would have to be marked such that it is uniquely acted on by the constraints appropriate for its grammatical tier. However, this would add a significant computational load not found in a true multi-tier grammar. Beyond this, Inkelas and Zoll (2007) argue that differently ranked cophonologies are independently motivated, and may be preferable to these indexed constraints for their empirical coverage and formal simplicity as well. For these reasons, I develop my analysis using a two-tiered grammar with a single set of constraints that are ranked separately on each tier.

Since words with transparent vowels can select suffixes of either harmonic feature value in a way that cannot be predicted based on the information in their surface forms, any analysis of transparent vowels must hypothesize some hidden information in the lexicon which specifies this choice. This can be done in a number of ways, including through a phonetically null vowel feature that interacts with harmony or through phonologically active flags on certain morphemes. I choose what I believe to be the simplest option: allowing phonemes in the inventory that bear the harmonic feature value opposite to the transparent vowels, participate in word specification, and merge to their transparent counterparts on the surface. If we accept the richness of the base hypothesis wholesale, then the inventory-narrowing constraints that neutralize this contrast are necessary anyway. This approach readily predicts transparent vowels' lack of harmonic parters, is also the only one of these which is compatible with an straightforward strictly-local agreement constraint.

There is also phonetic evidence which strongly suggests a neutralized contrast, at least in one language. Benus and Gafos (2007) found that Hungarian transparent vowels which select for back affixes are realized slightly farther back than ones that select for front affixes. This effect holds even when no affixes are present, ruling out coarticulation and suggesting incomplete phonological neutralization.

I will demonstrate this approach with data from Hungarian. Hungarian shows stem-controlled Back harmony, and has three transparent vowels: the front vowels [i], [iː], and [eː]. I assume the existence of these vowels' back counterparts: [ɯ], [ɯː], and [ɤː].

As previously, I represent consonants (which I continue to assume to be harmonically irrelevant) orthographically. I use the inventory and precise feature values specified in Kornai (1987), though I use the symbol /ɑː/ in place of his /aː/ in order to correctly match IPA feature specifications. Unlike in the Kalenjin example above, I do not treat Hungarian long vowels as splittable geminates. I follow his analysis, and let "á" and "é" differ from their short siblings in more than just duration, mapping "a" and "e" to /ɔ/ and /ɛ/, and "á" and "é" to /ɑː/ and /eː/, respectively. Since this asymmetry appears to be phonologically significant, I treat Hungarian long vowels as distinct segments (different symbols in the PyPhon tableaux), marked with the feature [+Long]. In light of this, Hungarian requires one new constraint: *[+Back, +Round, -Low]-V penalizes the specified feature values, and is synonymous with *[ɯ, ɯː, ɤː] for the inventory at hand.

The following examples are drawn from Gafos and Benus (2006). I follow Ringen and Vago's (1998) observation that the suffix used in the following examples corresponds to the freestanding word "*nek*", and thus that its suffix form is also specified with an /ɛ/. I begin with a simple case:

Cyclic Eval (Cycle 1 of 1):

/#ciːm#nɛk/	SA-Bk	Agr-Bk	IO-Bk	IO-Lg	IO-Lo	IO-Rd	(3)
a. ☞ ciːmnɛk							

(3): *[+Back, -Round, -Low]-V

Postlexical Eval:

/ciːmnɛk/	(3)	SA-Bk	IO-Lg	IO-Lo	IO-Rd	IO-Bk	Agr-Bk
a. ☞ ciːmnɛk							

Figure 9: Tableaux for *címnek* [ciːmnɛk]: "address", dative.

Since *címnek* contains no affixes that do not immediately border the stem, it passes through cyclic evaluation only once before reaching postlexical evaluation. Since it violates no constraints, it is trivially optimal no matter what the ranking, and passes through both evaluations unchanged.

The following case shows the behavior of this system with an input containing one of the hypothesized vowels, /ɯ/:

Cyclic EVAL (Cycle 1 of 1):

/#sɯːp#nɛk/	SA-BK	AGR-BK	IO-BK	IO-LG	IO-LO	IO-RD	(3)
a. sɑːpnɑːk			*	*	*		
b. sɑːpnɯk			*		**		*
c. sɑːpnɛk		*			*		
d. sɯːpnɑːk			*	*			*
e. siːpnɛk	*		*				
f. ☞ sɯːpnɔk			*			*	*
g. sɑːpnɔk			*		*	*	
h. suːpnɛk		*				*	
i. sɯːpnɯk			*		*		**
j. suːpnɔk			*			**	
k. suːpnɑːk			*	*		*	
l. sɯːpnɛk		*					*

Postlexical EVAL :

/sɯːpnɔk/	(3)	SA-BK	IO-LG	IO-LO	IO-RD	IO-BK	AGR-BK
a. ☞ siːpnɔk						*	*
b. sɑːpnɔk				*			
c. sɯːpnɔk	*						
d. suːpnɔk					*		
e. siːpnøːk				*		**	

Figure 10: Tableaux for *sípnak* [siːpnɔk]: “whistle”, dative.

Here, the highly ranked AGREE and SA-IDENT constraints force stem-controlled harmony in the first (cyclic) tableau. This yields a marked /ɯ/, which is eliminated with the help of the highly ranked markedness constraint in the second (postlexical) tableau, and replaced with its unmarked [-BACK] equivalent, making the affix appear disharmonic.

In the last tableaux, I demonstrate the effect of the hypothesized /ɤ/, and show transparency interacting with multiple other vowels:

Cyclic EVAL (Cycle 1 of 1):

/#kaːvɤ#nɛk/	SA-BK	AGR-BK	IO-BK	IO-LG	IO-LO	IO-RD	(3)
a. kaːvaːnɯk			*		**		*
b. kaːvaːnɛk		*			*		
c. kaːvaːnɔk			*		*	*	
d. kaːvøːnɛk		*				*	
e. kaːvaːnaːk			*	*	*		
f. kaːvɤːnaːk			*	*			*
g. ☞ kaːvɤːnɔk			*			*	*
h. keveːnɛk	**		**	*			
i. kaːvɤːnɯk			*		*		**
j. kaːvøːnɔk			*			**	
k. kaːvøːnaːk			*	*		*	
l. kaːvɤːnɛk		*					*
m. keːveːnɛk	**		**		*		
n. kaːveːnɛk	*	*	*				

Postlexical EVAL:

kaːvɤːnɔk	(3)	SA-BK	IO-LG	IO-LO	IO-RD	IO-BK	AGR-BK
a. kaːvaːnɔk				*			
b. kɛveːnɔk			*			**	*
c. kaːvɤːnɔk	*						
d. kaːvøːnɔk					*		
e. ☞ kaːveːnɔk						*	**

Figure 11: Tableaux for *kávénak* [kaːveːnɔk]: "coffee", dative.

As in the previous tableaux, the highly ranked AGREE and SA-IDENT constraints force stem-controlled harmony in the first tableau, then transform the resultting hypothesized vowel to its harmonic partner in the second tableau, causing the affix to appear to harmonize exclusively with the more distanct of the two vowels in the root.

3.6 Remaining issues in vowel harmony

Baković also describes numerous cases of morphologically conditioned opacity, wherein specific morphemes are opaque to harmony despite lacking any characteristically opaque feature combinations. While membership of this class of opaque morphemes is fairly arbitrary, they are common enough that a complete model of harmony should contain a mechanism to allow specially flagged segments to block spreading.

I have also not yet considered the (likely related) existence of disharmonic words in some languages which generally show harmony. It appears plausible that such examples may require some degree of morphological marking, or even a distinct (likely loanword targeted) cophonoloogy, and so I will leave this issue aside for now.

4 Opacity and learnability

I have shown in this paper that cyclic and stratal evaluation allow Finite-State OT to account for vowel harmony with simple constraints. Since harmony is littered with opaque interactions otherwise inaccessible to Finite-State OT, these extensions clearly yield a theory with a greater generative power.

While the choice of Finite-State OT did guide this analysis, it is certainly not the case that the value of an LPM-OT-style model emerges only with the stipulation of a finite-state grammar. LPM-OT, as it was originally proposed and in many of its subsequent uses, had no such explicit requirement (though most common constraints are finite-state, including those most often used in LPM-OT). Likewise, numerous elements of vowel harmony—especially transparency—remain difficult to account for under any formulation of OT which is sufficiently well-specified to guarantee computability.

Opaque phenomena are difficult to learn as a simple consequence of the definition of opacity, and so the challenge of developing a method to quickly learn a correct grammar in the face of opacity may be difficult even given a model (like this one) capable of describing such a grammar. I have focused in this paper on developing a theory of grammar that can capture all of the phenomena at hand, but I do not mean to disregard the importance of finding an adequate method to learn such grammars, and will briefly consider the learnability of each proposal here.

Since simple cyclic evaluation allows for intermediate forms which affect the output but differ from both the input and the output, it does allow for some weak opacity. Its capacity is significantly limited by the restriction that rankings be consistent through each application of EVAL, and it provides the learner with a fairly easy learning task. They may use information from any single cycle of evaluation to infer the nature and ranking of the constraints, and so once they have reconstructed the intermediate forms, the learning task is no harder than it is in classical OT. The underlying roots and affixes that the learner must infer under any framework provide ample data with which to go about this reconstruction. With more complex models come greater problems in learning, but there exist ample phenomena which cannot be easily explained even under a cyclic analysis.

Expanding into the use of a stratal grammar yields a needed increase in generative power, but that generative power brings with it the potential for a larger and potentially less realistic typology and a greater challenge to the learner. In OT, the problem of language acquisition is essentially a search through the space of all possible rankings of some constraints—excepting the not immediately relevant problem of identifying the constraints to begin with. This ranking search becomes more complex in the stratal case. In classical and cyclic OT, input-output pairs (to the extent that they can be inferred by the learner) allow a learner to converge a single adequate ranking quite quickly. In stratal OT, though, the learner must simultaneously search for two or more rankings with no more available data.

In general, any learning theory attempting to converge on a sufficient grammar based on some data will have a harder time when the number of possible grammars compatible with any subset of the data is larger. The total number of distinguishable grammars available to a theory for some fixed set of data is thus an indirect but viable measure of its learnability, and one that is relatively independent of any one learning theory.

Abstractly, the maximum number of possible unique grammars for n constraints is $n!$ under classical OT and $(n!)^k$ in stratal OT with k tiers (here $k = 2$). In practice, though, the increase in descriptive power that comes from the addition of a second tier is vastly less than this would suggest. Ultimately, the actual number of languages that emerge under stratal evaluation is relatively small, though each may emerge under a fairly diverse set of ranking pairs.

Since intermediate forms are not available to the learner, and there is no reason to assume that intermediate forms are correctly reconstructed when an incorrect reconstruction yields the same results, I measure generative power based (as before) on the number of distinct languages *defined by input–output pairs* that each theory can describe, ignoring the newly introduced intermediate forms. For example, a two-tier grammar (pair of sets of ranking conditions) yielding the input–intermediate–output sequence /a/→/b/→[c], and another two-tier grammar yielding /a/→/d/→[c] would both be counted as belonging to the single /a/→[c] language, as long as the two grammars cannot be better distinguished by other inputs. Defined this way, a selection of 12 Turkish input forms yields 14 distinct single-tier languages with the constraints presented above for that language, and 18 distinct two-tier languages. Hungarian is more prolific: With 9 inputs, it yields 84 single-tier languages, and 559 two-tier languages—a considerable jump, but far less than the square of the single-tier count.

A number of factors explain this relatively modest increase. Of particular importance is that any grammar that can be described by a single-tier ranking has at least two—and typically more—corresponding two-tier grammars which yield the same outputs, and thus count as belonging to the same language. For example, a language with dominance-based vowel harmony and no transparent vowels could be implemented in at least three ways: A cyclic grammar could implement harmony, which would be preserved by a faithful postlexical grammar; A cyclic grammar could preserve the input forms verbatim, letting a postlexical grammar handle harmony; Or both tiers could implement harmony, with the second requiring no change, since its input would already be harmonic.

In a work of this size it is not feasible to try to connect this proposal with any one account of language acquisition under OT. There are many approaches in the literature (among others: Tesar and Smolensky, 1993, Eisner, 2000, Goldwater and Johnson, 2003), each with its own assumptions about the nature and source of constraints, and on the accessibility and acquisition of underlying (and intermediate) forms. However, the observation that the total generative power increases relatively only moderately with a the addition of cyclic and stratal evaluation suggests that it is a viable approach to vowel harmony, and to opacity more generally in constraint-based grammars.

Appendix A: PyPhon compatible inventories

Note: To simplify the inventory specifications, consonants are not distinguished in the models presented here. Consonants presented in tableaux are corrected from the generic "C" to their underlying values. Since no constraint penalizes any consonant, this correction reproduces the behavior of an otherwise identical consonant-aware model.

Shared Inventory:

`C: +cons, +seg` (generic consonant)
`#: +marker, +stem` (root morpheme boundary, "#")
`.: +marker, -root` (affix–affix boundary, "+")

Inventory for Turkish and unspecified examples:

`i: +high, -low, -back, -round, -cons, +seg` (/i/, "i")
`I: +high, -low, +back, -round, -cons, +seg` (/ɯ/, "ı")
`a: -high, +low, +back, -round, -cons, +seg` (/ä/, "a")
`e: -high, -low, -back, -round, -cons, +seg` (/e/, "e")
`y: +high, -low, -back, +round, -cons, +seg` (/y/, "ü")
`u: +high, -low, +back, +round, -cons, +seg` (/e/, "e")
`O: -high, -low, -back, +round, -cons, +seg` (/ø/, "ö")
`o: -high, -low, +back, +round, -cons, +seg` (/o/, "o")

Partial inventory for Kalenjin:

`i: +high, -low, -back, +atr, -cons, +seg` (/i/)
`I: +high, -low, -back, -atr, -cons, +seg` (/ɪ/)
`a: -high, +low, +back, -atr, -cons, +seg` (/a/)
`A: -high, +low, +back, +atr, -cons, +seg` (/ɐ/)
`e: -high, -low, -back, +atr, -cons, +seg` (/e/)
`E: -high, -low, -back, -atr, -cons, +seg` (/ɛ/)

Inventory for Hungarian:

`e: -back, -high, +low, -round, -cons, -long` (/ɛ/, "e")
`E: -back, -high, -low, -round, -cons, +long` (/eː/, "é")
`q: -back, -high, -low, +round, -cons, -long` (/ø/, "ö")
`Q: -back, -high, -low, +round, -cons, +long` (/øː/, "ő")
`y: -back, +high, -low, +round, -cons, -long` (/y/, "ü")
`Y: -back, +high, -low, +round, -cons, +long` (/yː/, "ű")
`i: -back, +high, -low, -round, -cons, -long` (/i/, "i")

```
I: -back, +high, -low, -round, -cons, +long   (/iː/, “í”)
o: +back, -high, -low, +round, -cons, -long   (/o/, “o”)
O: +back, -high, -low, +round, -cons, +long   (/oː/, “ó”)
a: +back, -high, +low, +round, -cons, -long   (/ɔ/, “a”)
A: +back, -high, +low, -round, -cons, +long   (/ɑː/, “á”)
u: +back, +high, -low, +round, -cons, -long   (/u/, “u”)
U: +back, +high, -low, +round, -cons, +long   (/uː/, “ű”)
w: +back, +high, -low, -round, -cons, -long   (/ɯ/, does not surface)
W: +back, +high, -low, -round, -cons, +long   (/ɯː/, does not surface)
R: +back, -high, -low, -round, -cons, +long   (/ɤː/, does not surface)
```

Appendix B: PyPhon compatible constraints

```
AgreeV-Round:
[-cons, -round]([+cons]|[+marker])*_[-cons, +round]_
[-cons, +round]([+cons]|[+marker])*_[-cons, -round]_

IO-IdentV-Round:
_[-cons, +round]:[-cons, -round]_
_[-cons, -round]:[-cons, +round]_

*[+Round, -High]:
_[-cons, +round, -high]_

*[+Round, -High]&IO-IdentV-Round:
_[-cons, -round, -high]:[-cons, +round, -high]_

SA-IdentV-Round:
\#@*_[-cons, -round]:[-cons, +round]_@*\#
\#@*_[-cons, +round]:[-cons, -round]_@*\#

Structure (filter):
([+cons]:[+cons]|[-cons]:[-cons]|\#:\#|\.:\.)*
```

References

Adam Albright. From clusters to words: grammatical models of nonce word acceptability. Handout presented at the 82nd Annual Meeting of the Linguistic Society of America, 2008.

Diana Archangeli and Douglas Pulleyblank. Harmony. In *The Cambridge Handbook of Phonology*. Cambridge University Press, 2007.

Eric Baković. *Harmony, Dominance and Control*. PhD thesis, UC San Diego, 2000.

Max Bane and Jason Riggle. Three correlates of the typological frequency of quantity-insensitive stress systems. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*. Rutgers Optimality Archive # 966, 2008.

Max Bane and Jason Riggle. The typological consequences of weighted constraints. In *Proceedings of the 45th Annual Meeting of the Chicago Linguistics Society*. Rutgers Optimality Archive # 1068, 2009.

Max Bane and Jason Riggle. pyphon 1.5. software package. http://code.google.com/p/clml/, 2011.

Stefan Benus and Adamantios Gafos. Articulatory characteristics of hungarian transparent vowels. *Journal of Phonetics*, 35(3):271–300, 2007.

Tamás Biró. Quadratic alignment constraints and finite state optimality theory. In *Proceedings of the Workshop on Finite-State Methods in Natural Language Processing*, pages 119–126, 2003.

Noam Chomsky. *The Minimalist Program*. MIT Press, 1995.

San Duanmu. Recursive constraint evaluation in optimality theory: evidence from cyclic compounds in shanghai. *Natural Language & Linguistic Theory*, 15(3):465–507, 1997.

Jason Eisner. Efficient generation in primitive optimality theory. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 313–320. Association for Computational Linguistics, 1997.

Jason Eisner. Easy and hard constraint ranking in optimality theory. In *Finite-state phonology: Proceedings of the 5th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 22–33. Association for Computational Linguistics, 2000.

T. Mark Ellison. Phonological derivation in optimality theory. In *Proceedings of the 15th Conference on Computational Linguistics*, volume 2, pages 1007–1013. Association for Computational Linguistics, 1994.

Robert Frank and Giorgio Satta. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics*, 24(2):307–315, 1998.

Adamantios Gafos and Stefan Benus. Dynamics of phonological cognition. *Cognitive science*, 30(5):905–943, 2006.

John Goldsmith. *Autosegmental Phonology*. Garland Publishing, 1976.

John Goldsmith and Gary Larson. Local modeling and syllabification. In *Papers from the 26th Annual Regional Meeting of the Chicago Linguistics Society: Parasession on the Syllable in Phonetics and Phonology*, pages 129–141, 1990.

Sharon Goldwater and Mark Johnson. Learning ot constraint rankings using a maximum entropy model. In *Proceedings of the Stockholm workshop on variation within Optimality Theory*, pages 111–120. Citeseer, 2003.

Bruce Hayes. Assimilation as spreading in toba batak. *Linguistic Inquiry*, 17(3): 467–499, 1986.

Sharon Inkelas. Where did the derived environment condition go? In *Proceedings from the 47th Annual Meeting of the Chicago Linguistic Society*, To Appear.

Sharon Inkelas and Cheryl Zoll. Is grammar dependence real? a comparison between cophonological and indexed constraint approaches to morphologically conditioned phonology. *Linguistics*, 45(1):133–171, 2007.

René Kager. *Optimality Theory.* Cambridge University Press, 1999.

Lauri Karttunen. The proper treatment of optimality in computational phonology. In *The Proceedings of International Workshop on Finite-State Methods in Natural Language Processing*, pages 1–12, 1998.

Paul Kiparsky. Abstractness, opacity, and global rules. In A. Koutsoudas, editor, *The application and ordering of grammatical rules.* Mouton, 1976.

Paul Kiparsky. Opacity and cyclicity. *The Linguistic Review*, 17(351–365), 2000.

Paul Kiparsky and Karl Pajusalu. Towards a typology of disharmony. *The Linguistic Review*, 20(2-4):217–241, 2003.

András Kornai. Hungarian vowel harmony. In *Proceedings of the 6th West Coast Conference on Formal Linguistics*, pages 147–61, 1987.

Geraldine Legendre, Yoshiro Miyata, and Paul Smolensky. Harmonic grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. Technical report, University of Colorado, 1990.

John McCarthy. Sympathy and phonological opacity. *Phonology*, 16(3):331–399, 1999.

John McCarthy and Alan Prince. *Generalized alignment.* Rutgers University Center for Cognitive Science, 1993.

John McCarthy and Alan Prince. The emergence of the unmarked: Optimality in prosodic morphology. In *Proceedings of the North East Linguistic Society*, volume 24, pages 333–379, 1994.

John McCarthy and Alan Prince. Faithfulness and reduplicative identity. *University of Massachusetts occasional papers in linguistics*, 18:249–384, 1995.

Joe Pater. Cumulative ill-formedness in typological and experimental data. In *Handout from the Conference on Experimental Approaches to Optimality Theory.* University of Michigan, 2007.

Joe Pater, Karen Jesney, and Anne-Michelle Tessier. Phonological acquisition as weighted constraint interaction. *Proceedings of the 2nd Conference on Generative Approaches to Language Acquisition North America (GALANA)*, pages 339–350, 2007.

Alan Prince. Arguing optimality. *Papers in Optimality Theory II*, pages 269–304, 2002.

Alan Prince and Paul Smolensky. *Optimality Theory: Constraint Interaction in Generative Grammar.* MIT Press, 1993/2004.

Jason Riggle. *Generation, Recognition, and Learning in Finite State Optimality Theory.* PhD thesis, University of California, Los Angeles, 2004.

Jason Riggle. Counting rankings. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, page 28. Association for Computational Linguistics, 2008.

Jason Riggle. Generating contenders. Unpublished ms., 2009.

Catherine Ringen and Robert Vago. Hungarian vowel harmony in optimality theory. *Phonology*, 15(3):393–416, 1998.

Jerzy Rubach. Duke-of-york derivations in polish. *Linguistic Inquiry*, 34(4):601–629, 2003.

Noah A. Smith and Jason Eisner. Doing ot in a straitjacket. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 486–493, 1999.

Paul Smolensky. Harmony, markedness, and phonological activity. Handout from The Rutgers Optimality Workshop, 1993.

Paul Smolensky. On the structure of the constraint component con of ug. Handout of talk at UCLA, 1995.

Paul Smolensky and Geraldine Legendre. *The Harmonic Mind*. MIT Press, 2006.

Bruce Tesar and Paul Smolensky. Learnability in optimality theory. Technical report, Johns Hopkins University, 1993.

Rachel Walker. Consonantal correspondence. In *Proceedings of the Workshop on the Lexicon in Phonetics and Phonology*. Rutgers Optimality Archive # 379, 2001.

Colin Wilson. Consonant cluster neutralisation and targeted constraints. *Phonology*, 18(1):147–197, 2001.