Arguing Optimality

Alan Prince

Rutgers University, New Brunswick
10/22/02

Abstract

Ranking and optimality are based on pairwise comparisons between a desired optimum and its competitors. The 'comparative tableau' identifies the elements that figure in the ranking arguments derived from such comparisons. Precisely keyed to the logic of the theory, comparative representation makes it straightforward to present and evaluate ranking claims; to find redundancies, non-obvious consequences, and contradictions in sets of ranking arguments; to employ the constraint demotion algorithms; to efficiently determine universal suboptimal status; and to assess the explanatory role of individual constraints in particular analyses.

This paper revises Prince 2000:ROA-376, itself a revision of Prince 1998:ROA-288.

Acknowledgment

Thanks to John McCarthy for many useful comments and suggestions, and to Ed Keer, Jane Grimshaw, Bruce Tesar, Vieri Samek-Lodovici, Paul Smolensky, Z.Q. Li, Ron Artstein, Nicole Nelson, Ken Safir, Graham Horwood, Alexandra Zepter, and the Rutgers Optimality Research Group (RORG) for helpful discussion. The author retains responsibility for errors.

1. Optimality and Comparison

Optimality is relative success, not perfection. The flaws of an optimal form provide a kind of yardstick against which each suboptimum is measured and found wanting; beyond that, constraint violations have no significance.

The crucial comparisons run pairwise. A form is optimal if — paraphrasing Grimshaw's concise formulation (Grimshaw 1997) — in each of its pairwise competitions, it better-satisfies the highest-ranking constraint which distinguishes it from its competitor.[1]

---

[1] To see the need for the pairwise restriction, note that "the highest-ranking constraint on which the competitors differ" denotes at best one constraint, but when we are talking about more than one suboptimal competitor, different constraints may be decisive in each case: desired optimum ω may differ decisively from candidate a_1 on C_1 and from candidate a_2 on C_2, and so on.

Grammars are built to meet the requirements set by such two-way competitions. A ranking argument compares desired optimum ω and desired suboptimum *z* over the entire set of constraints. An argument pitting ω against *z* is informative only if ω loses to *z* on some constraints. If ω also betters *z* on some other constraints,[2] we can correctly infer that at least one of the constraints preferring ω must dominate *all* of the constraints preferring *z*.

To be able to reason efficiently and correctly with ranking arguments, we need a data structure that explicitly represents their logic. Working from the observations just made, we can see that we must know precisely these things:

- of each *candidate*, whether it
  - is a desired optimum, or
  - competes with a desired optimum (is a desired suboptimum).

- of each *constraint*, whether it
  - prefers the desired optimum; or
  - prefers the desired suboptimum; or
  - is neutral between them.

The following array implements these notions:

(1) **Comparative Tableau**: ω *vs. z*

	A	B	C	D	E	F	G
ω ~ z		L	W		W	L	L

Notation:

ω ~ *z* ω competes with *z*; and ω is the desired optimum.
W the constraint prefers the desired optimum (‘prefers the ***W***inner’).
L the constraint prefers the desired suboptimum (‘prefers the ***L***oser).
blank the constraint does not distinguish the candidates

It is immediately clear from (1) that either C or E must dominate all of {B,F,G} if ω is to win.

Typographical note. We will *not* assume generally that the order of constraints in a tableau reflects an established domination order, because we will be predominantly interested in determining which rankings the tabulated information supports. We will explicitly note when left-to-right constraint listing reflects a domination order. On occasion, to emphatically announce an assumption of non-ranking, we will use vertical dotted lines to separate unranked columns as in (1).

[2] If ω loses to *z* somewhere but never betters *z* on any other constraint, then ω cannot be optimal; it is ‘harmonically bounded’ (Samek-Lodovici 1992, Prince & Smolensky 1993, Samek-Lodovici & Prince 1999).

By ‘desired optimum’ and ‘desired suboptimum’ we refer to the first and to the second position in the comparative pair. A typical use of a tableau is to test the success of a ranking hypothesis, when the ‘desired optimum’ is truly desired to be optimal in order to validate an analysis. However, the very same tableau structure can be used to show that a certain candidate occupying the first position is not or cannot be optimal. And the tableau can be used to *test* for the properties and predictions of a ranking, or even those of an un- or partially ranked constraint set, when curiosity rather than desire is the primary motive: the logic is the same. We continue with the language of intent, but it should be kept in mind that at bottom we are simply interested in unambiguously signaling which member of pair is judged better than the other on a constraint-by-constraint basis.

Because the ‘comparative tableau’ identifies and represents just those elements that figure in the logic of ranking arguments, it provides the basis for understanding any optimality-theoretic analysis. As a data-structure, the comparative tableau makes it straightforward to present and assess ranking claims (§3-5, Appendix); to find redundancies, nonobvious consequences, and contradictions in sets of ranking arguments (§6); to employ the constraint demotion ranking algorithms (§8-9); and to efficiently determine universal suboptimal status (§7). Because of this, the comparative tableau supports the kind of fine-grained analysis that can determine how the constraints interact to produce the various effects that emerge from the grammar (§10).

2. Data Tableaux

The internal logic of the theory turns on comparison, but the direct encounter with linguistic forms takes place in terms of constraint violation. Determining optimality requires two assessments: one which organizes the constraint violation data, and one which reveals the *relative* violation status of competitors.

The familiar violation tableau has proved to be an valuable tool for calculating constraint interactions, but it does not adequately represent both the absolute and the comparative functions, even with standard current methods of annotation.

The essential difficulty is not far to seek. Examine a typical annotated data tableau, with constraints presented in a domination order:

(2) **Annotated data tableau**

	C_1	C_2	C_3
ω	*		***
a	*	*!	**
b	**!	*	***

What crucial ranking relations are determined by these competitions? To find the answer, further computations are required, because the annotation marks only the occasion of global victory by the optimum [!], given this one ranking. Yet knowledge of the optimum’s *defeat* is just as essential to a ranking argument. And knowledge of *all* local victories and defeats is required to determine the range of rankings compatible with these data.

In detail: it is clear from the exclamatory annotations that C_1 is responsible for *b*'s demise in the face of ω, and C_2 for *a*'s. But no sign indicates that C_3 *must be* subordinated in the ranking, and that C_2 need not be. Nor is there an indication that it is candidate *a* that forces subordination of C_3, while candidate *b* is indifferent. It is precisely among these unannotated properties where mistakes and oversights thrive, and where we must look for the patterns of explanation that the theory provides.

The comparative tableau, in accord with the logic of ranking, marks the optimum's competitive successes (W generalizing "!"), as well as its failures, where the rival succeeds (L).

(3) **Comparative tableau version** of data tableau (2)

	C_1	C_2	C_3
ω ~ a		W	L
ω ~ b	W	W	

Ranking theory says that each L must be preceded by a W, so necessarily $C_2 >> C_3$. But the [ω~*b*] row has no L's in it; no particular ranking is required for ω to win against *b*, which is harmonically bounded by ω.

A close approach to comparison within the star-marking format is obtained by cancelling out shared violations (Prince & Smolensky 1993/2002, ex. (192), 2002: p.142). The absolute number of violations never need be *counted*; the relevant comparisons turn only on the *difference* between the desired optimum's violation profile and that of each competitor. Cancelling out shared violations indicates this difference; winning and losing occurs when one candidate's tally is driven to zero and the other's is not. Here is cancellation at work on a simple three-constraint tableau:

(4) **Cancellation**

**	*	****
***	*	**

→

		**
*		

Cancellation, of course, is not typically used in presentation, because it can be misread as distorting the violation claims. More compellingly, cancellation *cannot* be in general be carried out across an entire tableau when there are three or more rows. Let us return to our example (2).

(5) **Cancellation-irreducible data tableau**

	C_1	C_2	C_3
ω	*		***
a	*	*	**
b	**	*	***

Cancellation must be done pairwise, against the desired optimum. In this case, the ω row cannot be given a unique valuation. In the [ω~*a*] comparison, C_3 keeps a star (3−2=1):

CANCELED	C_1	C_2	C_3
ω			*
a		*	

But in the [ω~*b*] comparison, the C_3 cell in row ω goes blank (3−3=0).

CANCELED	C_1	C_2	C_3
ω			
b	*	*	

Furthermore, even under cancellation, each competitor's results must still be measured against the desired optimum's: although any remaining uncanceled mark spells loss in the competition, a blank cell is locally ambiguous. (For example, in (4) above, the blank cells in the first and third columns mean victory, but in the second column, merely equality and no-decision.) Pairwise cancellation can be a useful computational step on the route to evaluation, but in itself it is intrinsically unsuitable for multi-comparison collections and it does not provide a row-local representation of all the crucial information. To achieve this formally, we need to go beyond cancellation to the *subtraction* of one row from another, working with a three-way distinction — positive, negative, zero: W,L, *blank* — as opposed to the two-way distinction of cancellation — zero *vs*. positive.

To deal with legacy tableau-ware, it is useful to have a conversion patch. (The same annotational technique will allow us to advance from a data tableau to the comparative tableau that we need for analysis.) The key is simply to ignore the optimum's row: it is, after all, merely the standard of measure; the real interest lies in demonstrating that the other candidates fail against it. One can then simply inscribe the losers' cells according to the comparative recipe,[3] in order to demarcate the implicit comparative structure. Here is our example tableau (2), thus massaged.

(6) **Annotating a data tableau**

	C_1	C_2	C_3
ω	*		***
a	*	* W	** L
b	** W	* W	***

To read off the comparative structure, simply ignore the ω-row and the stars.

In fact, there is no typographical difficulty in retaining all the information, should that prove desirable; here we render the mark-count numerically in aid of visual parsing.

(7) **All information displayed**

	C_1	C_2	C_3
ω	1		3
~ a	$_1$	$_1$ W	$_2$ L
~ b	$_2$ W	$_1$ W	$_3$

We observe finally that the comparative tableau eliminates the need for various auxiliary flourishes intended to guide the eye to those essential aspects of the argument that derive from (but are not explicitly present in) the array of violation data. The index W subsumes the role of '!'. In a tableau with columns in domination order, a successful optimum is marked unmistakably by the occurrence of W in the leftmost non-blank cell in the row: there is no need for *shading* to set it off. And like those constraints living in perpetual shadow, any constraint that *never assesses* a leftmost W does no work in the analysis: its inertness can be readily spotted. Another emphasis-evoking situation occurs when a tentative constraint hierarchy selects the wrong form. Once again the bare comparative tableau tells the tale: the first-encountered index in the left-to-right sweep will be L, sufficient to indicate — without further adornment — that the second term of the [ω~*a*] comparison is the undesired winner.

---

[3] Translation involves a twist in perspective: the relevant data tableau rows are about the suboptimum, giving its flaws, but in the comparative tableau they are most easily read as being about the optimum (it W*ins* or L*oses*), or about the pair. Experience has shown that this difficulty can be overcome by the heuristic of using suboptimum-centered *sotto voce* mnemonics for W and L during the process of annotation: such as **W***orse* for W and, say, **L***ess* (marks than the desired optimum) for L. These are extensionally equivalent to the pair-centered or optimum-centered interpretations of W and L. (The suboptimum has more marks = the suboptimum is **W***orse* = the optimum **W***ins*; and similarly for L.)

3. A typology of relations

Constraint conflict arises exactly when both competitors appear as winners on different constraints:

(8) **Conflict**

W ~ L	C_1	C_2
ω ~ a	L	W

Here candidate *a* wins on C_1, as marked by its index L, and ω wins on C_2 , which bears its index,W. This is precisely the configuration in which *constraint conflict* is found: a row with both W and L indices marked in it. Tableau (8) supports the ranking argument C_2>>C_1, because the form ω is the desired optimum. Tableau (9) supports the opposite conclusion:

(9) **Conflict**

	C_1	C_2
ω ~ a	W	L

In other possible win/lose/draw configurations, there is obviously no difficulty in adjudicating the competition, no conflict, and no ranking argument.

(10)

	C_1	C_2
ω ~ a		W

	C_1	C_2
ω ~ a		

	C_1	C_2
ω ~ a	W	

	C_1	C_2
ω ~ a		L

	C_1	C_2
ω ~ a	L	

If a row includes only L, then the desired optimum can never win under any ranking: constraint domination cannot redeem its loss. Similarly, if a row contains only W, the desired *sub*optimum is a universal loser, and no ranking can render it optimal.

The comparative tableau classifies constraints as well as arguments. Constraints separate into two varieties: those assessing W or L, and those indifferent to the comparison(s) at hand. This allows us to answer a fundamental question of analytical tactics: in constructing a ranking argument, which constraints may we legitimately *ignore*?

The answer is that we may omit from discussion all and only those constraints assessing *blank*: these are the ones that do not enter into the comparative evaluation. If — with an excessive concern for conciseness, or through mere oversight, or because we prefer the expositional strategy of withholding key information — we ignore a constraint that assesses L, we have lost track of a constraint that *must be dominated* and our argument is dangerously incomplete. If we ignore a constraint that assesses W, we have (as long as some other constraint assesses L) produced a ranking argument that is too strong and may be literally false. For example, if in the case like (9), there is an additional out-of-sight constraint C_3 that assigns W, we can no longer conclude that $C_1 >> C_2$ is necessary. We can only be certain that $C_1 >> C_2$ **or** $C_3 >> C_2$, an assertion that is entirely consistent with finding out that $C_2 >> C_1$.

4. Practicum

To illustrate the different character of data and comparative tableaux, let us examine arguments drawn from Lardil phonology, using a slightly modified version of Prince & Smolensky 1993:122/2002:134, ex. (184). Epenthetic segments are outline-fonted; deletion sites are heuristically marked with '■'.

Of the faithfulness constraints, Dep-C militates against the insertion of consonants, Dep-V against the insertion of vowels, and Max against the deletion of segments (C or V). Free-V is in essence an antifaithfulness constraint (Alderete 1999, Horwood 1999) demanding that certain final vowels be unparsed (deleted). Align wants stem-final vowels to be present and prosodic-constituent-final in the output. Lex≈Pr wants lexical words to be prosodic words. *Gem forbids geminate consonants. We write 'ţ' for the retroflex coronal stop, and 'ŗ' for the retroflex rhotic.

(11)

/wiţe/	*Gem	Lex≈Pr	Dep-V	Free-V	Align	Dep-C	Max
.wi.ţe				*			
.wiţ.■		*			*		*
.wi.ţ■a.			*		*		*
.wiţ■.ţa.	*		*		*	*	*

Converting, we have:

(12)

/wiţe/	*Gem	Lex≈Pr	Dep-V	Free-V	Align	Dep-C	Max
.wi.ţe				1			
.wiţ.■		$_1$ W		$_0$ L	$_1$ W		$_1$ W
.wi.ţ■a.			$_1$ W	$_0$ L	$_1$ W		$_1$ W
.wiţ■.ţa.	$_1$ W		$_1$ W	$_0$ L	$_1$ W	$_1$ W	$_1$ W

And in purely comparative form:

(13)

/wiţe/ →	*GEM	LEX≈PR	DEP-V	FREE-V	ALIGN	DEP-C	MAX
.wi.ţe. ~ .wiţ.■		W		L	W		W
.wi.ţe. ~ .wi.ţ■a.			W	L	W		W
.wi.ţe. ~ .wiţ■.ţa.	W		W	L	W	W	W

The comparative tableau makes it abundantly clear that the suboptimal competitors' only advantage lies on the constraint FREE-V. It is also clear that, if the constraints ALIGN, DEP-C, and MAX are ranked below FREE-V, as indeed they are, then the constraints LEX≈PR and DEP-V must be carefully positioned to eliminate competitors. Let us demonstrate how an exact ranking argument can be constructed in practice, given the type of information presented here.

Imagine that we know from prior analysis that FREE-V>>Align>> DEP-C >> MAX, *i.e.* that the last four columns of the tableau are in correct ranking order. We re-write the tableau to reflect this, using a slightly condensed variant of style shown above. The desired input-output relation is indicated in the upper left-hand box, and only the competing outputs are listed in the evaluation rows, without repetition of the desired output. We use the ad-hoc notation of a double line to fence off the already-ranked constraints from those whose ranking has not yet established.

(14)

/wiţe/ → .wi.ţe.	*GEM	LEX≈PR	DEP-V	FREE-V	ALIGN	DEP-C	MAX
a. ~ .wiţ.■		W		L	W		W
b. ~ .wi.ţ■a.			W	L	W		W
c. ~ .wiţ■.ţa.	W		W	L	W	W	W

We still have to determine the ranking of LEX≈PR, DEP-V, and *GEM relative to each other and to the ranked constraints. Turning to *wiţe* for illumination, we obtain the comparative data shown in (14). The already-established rankings involving FREE-V tell us that none of the W's in the last three columns can be used to establish the success of *wiţe* in the comparisons at hand. Therefore, we deduce from rows (14)a,b that LEX≈PR and DEP-V must dominate FREE-V. Row (14)c turns out to be uninformative — it yields only that **either** *GEM **or** DEP-V must dominate Free-V, a requirement that is weaker than, and entailed by, the argument from row (14)b. The ranking analysis could proceed without this candidate.

These considerations have not yet fixed a ranking between LEX≈PR and DEP-V. To settle this matter, we examine a further relevant datum.

(15)

/ŗelk/ →	*GEM	LEX≈PR	DEP-V	FREE-V	ALIGN	DEP-C	MAX
[.ŗel.ka.] ~ .ŗel.■		W	L				W

The optimality of the epenthetic candidate *.ŗel.ka.* establishes that LEX≈PR >> DEP-V. The ranking of *GEM must await further data.

Discussion. The desired suboptimum .ŗel.■ wins (earns L) on DEP-V because it has no epenthesis and the optimum does. The optimum wins on LEX≈PR because, as indicated, its bisyllabicity allows it to be a prosodic word, while the monosyllabic suboptimum is not and cannot be a prosodic word, due to high-ranked but unmentioned FTBIN, which demands bisyllabicity/bimoraicity of feet. Note that the suboptimal competitor in (15) satisfies FTBIN by virtue of having *no* feet, and hence none that are non-binary. The desired optimum satisfies FtBin more straightforwardly — by being bisyllabic. Since the competitors do equally well with respect to binarity requirements, FTBIN assesses *blank* comparatively, and may be legitimately omitted.

To illustrate what we have gained, we display a data tableau dealing with the same competition:

(16)

/ŗelk/	*GEM	LEX≈PR	DEP-V	FREE-V	ALIGN	DEP-C	MAX
[.ŗel.ka.]			*	*	*		
.ŗel.■		*		*	*		*

We have simplified the discussion in two respects. First, by not including comparisons with other candidates, e.g. **ŗelk*, which must be disposed of to make *ŗel.ka* optimal; this omission does not render the argument in (15) incorrect, only incomplete. The second, more serious simplification is the failure to survey *all* the constraints involved. As noted above, a correct ranking argument must include every constraint that distinguishes the compared candidates (assigns W or L). Should there be some other unmentioned constraint that prefers the desired optimum *ŗelka* and disprefers *ŗel*, and which is rankable above DEP-V, we could not safely conclude that LEX≈PR is doing the work. Prince & Smolensky 1993/2002:§7, at least, cite no such constraint; the reader might wish to examine that work for a fuller account of the Lardil hierarchy.

5. The data tableau: some virtues

Whenever absolute violation data is under scrutiny, the data tableau retains its usefulness. Ed Keer notes that an analyst will often need to try out different constraint definitions, and their absolute violation structure is what is directly manipulated. (But the comparative sense of things is equally essential to the constraint-definer, and should perhaps take priority, lest the temptation to posit intricate, fully satisfiable descriptive constraints be acceded to; on such issues, see McCarthy 2002a:§1.4.4.) Keer observes further that the data tableau takes a flat view of the candidate set, treating each candidate in isolation, whereas the comparative tableau is biased toward the desired optimum, and this can influence usage. John McCarthy notes that local conjunction effects (Smolensky 1995), which depend on absolute violation data, are computed from the data tableau:

(17) **Constraint Conjunction**

	$C_1\&C_2$	C_1	C_2
a		*	
b	*	*	*

Candidate *b*'s violation of the conjoined constraint $C_1\&C_2$ is entailed by its violations of C_1 and of C_2 individually. In this case, the comparative tableau hides the key violation of constraint C_1:

(18) **Conjunct contribution rendered opaque**

	$C_1\&C_2$	C_1	C_2
a ~ b	W		W

The same comparative cell for C_1 would have been obtained if both candidates had *succeeded* on the constraint, in which case $C_1\&C_2$ would not be violated by candidate *b*. Note, however, that once the violation structure of $C_1\&C_2$ has been ascertained, it behaves exactly like any other constraint, and the comparative representation functions perfectly to reckon its behavior.

The comparative tableau also hides certain distinctions among suboptimal candidates. The act of comparison with a desired optimum divides the candidate world into three parts: those better than (L-marked), those worse than (W-marked), and those the same as (*blank*) the desired optimum. Lost are any further distinctions in the W- and L-sections, which may be relevant to other competitions. Suppose two candidates *a,b* compare equivalently to the desired optimum ω on some constraint C, in the sense that either ω≻*a,b* or *a,b*≻ω on C (writing '≻' for 'better than'). Comparisons [ω~*a*] and [ω~*b*] are oblivious to any further distinction in C between *a* and *b,* although these would appear if the two were set off against each other in [*a*~*b*]. We cannot establish from an ω-centered tableau that *a* harmonically bounds b, although it is clear when ω itself does.

As a simple instantiation of this remark, consider the following two-constraint system:

(19)

	C_1	C_2
ω	0	2
a	1	0
b	2	1

If we map out the order relations inherent in the violation data, we see that candidate *a* harmonically bounds candidate *b*. But taking ω as the desired optimum results in *a* and *b* being classed together in the comparative reckoning:

(20)

	C_1	C_2
ω	0	2
~ a	$_1$ W	$_0$ L
~ b	$_2$ W	$_1$ L

Observe, however, that only one of the rows [ω~a] and [ω~b] is needed for purposes of constructing ranking arguments. This reflects, in small, the general situation: if a set of ranking arguments is non-redundant, then its associated set of candidates is free of harmonic bounding (Prince 2002:§6). In the case at hand, we may make the argument on the basis of either [ω~a] or [ω~b], and within each of the associated candidate sets {ω,a} and {ω,b, harmonic bounding is absent. This means that although comparative tableaux do not display harmonic bounding relations among suboptima, we are nonetheless able to ensure, given only comparative data, that a candidate set is free of harmonic bounding among its members: we need only require that the set of comparisons based on the candidate set be redundancy-free. A redundant ranking requirement is precisely one that is entailed by others: and, as we will now see, entailments can be computed easily from comparative tableaux.

6. Row manipulations: entailment and inconsistency of arguments

Each tableau row encodes a restriction on ranking that we will call an 'elementary ranking condition' or ERC. The exact content of an ERC is this: *at least one* constraint assessing W (preferring the targeted optimum) must dominate *all* constraints assessing L (those preferring the targeted suboptimum). The entire tableau imposes a set of such conditions, all of which must be simultaneously satisfied by any ranking that successfully generates the desired optima.

Terminological note. A 'tableau row', as the term is used here, is simply a record of how one competing pair performs on every constraint. No prior ranking restrictions are presupposed.

In any collection of such conditions, there may be patterns of relatedness and logical dependency that will determine the shape and outcome of the analysis. If one row (or set of rows) logically entails another row, there is no need to produce the superfluous entailed ranking argument or the suboptimal candidate that goes with it. Concomitantly, a row or set of rows may entail a restriction that is not locally represented as a tableau row in the data at hand: a logical consequence of the information already gathered, perhaps non-obvious, which must be respected. And if one row is inconsistent with others, important conclusions follow about the constraint set — it doesn't work! The comparative tableau makes it easy to discover such patterns of entailment and contradiction. (The discussion here summarizes material from Prince 2002.)

Two basic conditions arise, one keyed to the disjunctive logic of W, the other to the conjunctive logic of L. Let's consider the W-situation first.

(21)

W ~ L	C_1	C_2	C_3
ω ~ a	W		L
ω ~ b	W	W	L

The argument from [ω~*a*] is logically the stronger: it asserts that $C_1>>C_3$, but [ω~*b*] tells us only that ***either*** $C_1>>C_2$ ***or*** $C_1>>C_3$. Propositional calculus says $p \rightarrow p \vee q$; this is an instance.

In short, the ERC associated with [ω~*a*] *entails* the ERC associated with [ω~ *b*]. This abstract example mirrors the concrete case already observed in ex. (14), where (14)b entails (14)c, repeated for convenience here:

(22)

/wiţe/ → .wi.ţe.	*GEM	LEX≈PR	DEP-V	FREE-V	ALIGN	DEP-C	MAX
(14)b. ~ .wi.ţ■a.			W	L	W		W
(14)c. ~ .wiţ■.ţa.	Ⓦ		W	L	W	Ⓦ	W

We can see that ERC (14)c is merely a twice-weakened version of (14)b, and therefore redundant.

With L, we have the following:

(23)

W ~ L	C_1	C_2	C_3
ω ~ a	W		L
ω ~ b	W	L	L

Now the entailment relation is reversed: from [ω~*b*] we learn that ***both*** $C_1>>C_2$ ***and*** $C_1>>C_3$ are required to hold, from which we may safely conclude that [ω~a] = $C_1>>C_3$ holds, because $p \wedge q \rightarrow p$. We will look at an ecologically-valid example shortly.

The general relation between one tableau row and another emerges straightforwardly from these observations. If the W's of row α — the constraints assessing W — are a subset of the W's of row β **and,** at the same time, the L's of β are a subset of the L's of α, then we may conclude that α entails β.

(24) **Row entailment**. If α and β are elementary ranking conditions, corresponding to individual rows in a tableau, and if W(X) and L(X) denote the sets of constraints preferring respectively the desired optimum and the desired suboptimum of ERC X, then $\alpha \rightarrow \beta$ if $W(\alpha) \subseteq W(\beta)$ and $L(\beta)] \subseteq L(\alpha)$.

Proof. See Prince 2002:§1, Proposition 1.1.

We can strengthen the ‘if’ to ‘if and only if’ by limiting ourselves to ‘nontrivial’ ERCs — those in which both W and L are present. Among trivial ERCs, those containing no L’s are satisfied by any ranking, and therefore entailed by any other ERC, regardless of its W,L-structure; and, those containing L but no W are unsatisfiable, and therefore entail everything.

The row entailment condition may be broken down into two ‘rules of inference’ that allow us to create entailed rows from those already on hand.

(25) **W-extension**. Let R be a tableau-row. If R′ can be derived from R by filling a blank cell with W, then R′ is entailed by R.

In other words, adding more W’s produces a weaker, more disjunctive argument. (Observe that W-extension applies validly to trivial rows.)

(26) **L-retraction**. Let R be a tableau-row. If R′ can be derived from R by replacing an L cell with *blank*, then R′ is entailed by R.

Removing L’s weakens an argument by withdrawing conjuncts, which are asserted. (Again, L-retraction applies validly to trivial rows.)

These rules of inference may be applied repeatedly, and in any combination, to produce valid entailments.

In example (21) above, we see that [ω~*a*]→[ω~*b*] by W-extension. In example (23), we have [ω~*b*]→[ω~*a*] by L-retraction. In each case, the entailed argument merely asserts a weakened version of its source, adding a querulous disjunct or taking away an assertive conjunct. Discussion can proceed without such redundant arguments.

Note finally that W-extension and L-retraction apply to any rows whatsoever, in isolation, without presupposition about their contents or how they figure in the larger scheme of things.

In using W-extension and L-retraction, it is important to realize that the set of ranking conditions is being expanded, and previously-established arguments are not replaced or erased. Suppose, for example, that we face a situation like this, where the ranking order A>>B>>C has already been determined.

(27) **A>>B>>C assumed**

	A	B	C
ω ~ z	L	W	L

This is a calamity: the desired optimum ω loses to *z* on the highest-ranked constraint that distinguishes them. It is not improved by invoking L-retraction to arrive at this more pleasing configuration:

(28) **A>>B>>C assumed**

	A	B	C
ω ~ z		W	L

The propositional content of the tableau rows is this: (27) B>>{A,C} and (28) B>>C. Clearly, the first implies the second, as promised. But B>>{A,C} contradicts the prior ranking arguments that force A>>B. This fatal problem with the ERC B>>{A,C} cannot be alleviated by spinning out its implications.

The more elaborate logic involving several rows is daunting in the abstract. Each ERC is a disjunction of conjunctions; a collection of them amounts to a conjunction of disjunctions of conjunctions, with the prospect of much chopping of logic needed to unknot the Gordian tangle.

But we need just one more principle of row manipulation to find the ERCs entailed by a set of ERCs: a way of making two rows into a third by combining corresponding cells. In previous versions of this paper, the operation was called 'summation' and represented by '+'. The operation turns out to be equivalent to one called 'fusion' in the literature on relevance logics (Anderson & Belnap 1975), when W,L, *blank* are recognized as functioning like the system of three truth-values — T, F, and a third — for the logic RM3 (Prince 2002:§ 7). We will therefore switch to the standard nomenclature and notation for the row-combining operation. In present terms, the principles of fusion are these, writing *e* for 'blank'; X for any of {*e*, W, L}; and X∘Y for the fusion of X and Y.

(29) **Fusion of cell entries**

X∘L = L∘X = L	dominance of L (*cf.* dominance of F in logical conjunction)
X∘*e* = *e*∘X = X	*e* is identity
X∘X = X	idempotence

In short: L drags everything down to its own level; *e* is transparent; and like fuses to like. Rows are fused by constraint column. The utility of fusion can be seen in the following example:

(30) **Row fusion**

	A	B	C
α	W	L	
β		W	L
α∘β	W	L	L

The fusional ERC α∘β states that A>>B and A>>C.. This condition is not explicit in rows α and β individually, but follows from them by transitivity of '>>'. Further, from α∘β it follows that A>>C, because $p \wedge q \rightarrow q$; this is L-retraction. Once again, we have found a ranking condition that is implicit in the conjunction of ERCs α and β, but not explicitly represented.

This example shows a key property of fusion: *conjunction entails fusion.* Any ERC obtained by fusing a set of ERCs represents a logical consequence of the conjunction of ERCs in that set. In the example, $\alpha \wedge \beta$ clearly entails $\alpha \circ \beta$, and we are guaranteed that this is always the case, no matter what α and β may be. Furthermore, by transitivity of implication, any L-retract of $\alpha \circ \beta$ is also consequence of $\alpha \wedge \beta$.

Remarkably, any nontrivial ERC at all that follows from (the conjunction of) a set of ERCs also follows from the fusion of some subset via W-extension and L-retraction (Prince 2002:§2). Thus fusion reduces the problem of dealing with sets of ERCs to that of finding the consequences of a single (fusional) ERC, and those consequences are easily found by W-extension and L-retraction.

Fusion is particularly useful in dealing with ERCs that impose disjunctive ranking conditions. Consider the following ERC set, with no prior ranking assumed, adapted from a real-life encounter with Latin prosody:

(31)

	A:FtBin	B:Dep-μ	C:NonF(F′)	D:AL(F′,R)	E:Parse-σ	F:PkProm
α	W	L				W
β		W	L	W	L	L
γ			W	L	W	

Must FtBin dominate PkProm? Perhaps the answer does not leap from the tableau. To approach through classical logic, one must examine this expression for $\alpha \wedge \beta \wedge \gamma$:

$$[(A \gg B) \vee (F \gg B)] \wedge [((B \gg C) \wedge (B \gg E) \wedge (B \gg F)) \vee ((D \gg C) \wedge (D \gg E) \wedge (D \gg F))] \wedge [(C \gg D) \vee (E \gg D)].$$

But the question is equivalent to asking whether the following ERC δ is a consequence of $\{\alpha,\beta,\gamma\}$:

(32)

	A:FtBin	B:Dep-μ	C:NonF(F′)	D:AL(F′,R)	E:Parse-σ	F:PkProm
δ	W					L

If (32) is entailed by (31), it can be gotten from a fusion of some subset of $\{\alpha,\beta,\gamma\}$. In this case, fusing the lot will work:

(33)

	A:FtBin	B:Dep-μ	C:NonF(F′)	D:AL(F′,R)	E:Parse-σ	F:PkProm
α	W	L				W
β		W	L	W	L	L
γ			W	L	W	
α∘β∘γ	**W**	**L**	**L**	**L**	**L**	**L**

From this it is transparently clear that FtBin>>PkProm: we may obtain ERC δ by L-retraction.

Continuing with this example, let systematically use our logical resources to find the entire ranking structure implied by the tableau (31).

• From α∘β∘γ we immediately get A>>B. (Formally, we can L-retract from C,D,E,F.)

• From β∘γ, shown below, we get B>>{C, D, E,F). So we have A>>B>>{C,D,E,F}

(34)

	A:FtBin	B:Dep-μ	C:NonF(F′)	D:AL(F′,R)	E:Parse-σ	F:PkProm
β		W	L	W	L	L
γ			W	L	W	
β∘γ		**W**	**L**	**L**	**L**	**L**

• From γ, we have C>>D ***or*** E>>D. We can't do any better with these data, so we're done. The restrictions imposed by {α,β,γ} can be portrayed as in (35), where 'X above Y' means X>>Y necessarily and {X,Y} means that X and Y may be ranked in either domination order:

(35)

```
A                  – or–              A
|                                     |
B                                     B
|   \                                 |   \
C   {E,F}                             E   {C,F}
|                                     |
D                                     D
```

This example shows that fusion is a powerful tool for combining ERCs so as to eliminate disjunctions (multiple occurrences of W in an ERC) wherever possible. L-retraction then allows us infer specific pairwise ranking relations between constraints in cases where multiple L's result.

Fusion is not a method of deriving new constraints from old ones, like local conjunction or the other methods of boolean constraint combination introduced in Crowhurst & Hewitt 1997. It is a way of operating with ranking arguments so that their logical consequences are brought out and expressed in the standard format. It produces ranking conditions from ranking conditions: ERCs from ERCs. Because a fusion is entailed by the set of its components, we are guaranteed that the fusion holds true of any ranking that accords with those components. In the case just discussed, we know that any ranking satisfying α,β, and γ will also satisfy α∘β∘γ and β∘γ (as well as α∘γ, which happens to be uninformative in this instance). The fusions α∘β∘γ and β∘γ turn out to give a better guide to the required rankings than the original α, β, γ, in which various 'confounds' obscure the systematic import of the data.

A further and related use of fusion is shown in the following, where all non-blank cells fuse to L. (Call this 'fusion to L^+', since the resulting row contains at least one L and no W's.)

(36) **Fusion to L^+**

	A	B	C
α		W	L
β		L	W
α∘β		L	L

A ranking argument with the structure of α∘β cannot be satisfied: the lack of W indicates that its implicit desired *sub*optimum would always win. This means that ranking arguments α and β, which jointly imply this conclusion, cannot be jointly satisfied and must therefore be contradictory. And so they are: α says B>>C and β counters with C>>B.

A general criterion for the inconsistency of tableaux emerges: if any collection of rows in a tableau fuses to L^+, the tableau cannot support a consistent ranking. (Note that fusion is associative and commutative so that order of fusion doesn't matter.)

(37) **Inconsistency of Ranking Arguments**. A set of elementary ranking conditions is inconsistent iff its comparative tableau contains a set of rows that fuses to L^+.

Remark. This is Proposition 2.4, Prince 2002:11.

Let us now put this result to use.

7. Finding Losers

It is often useful or necessary to know whether a given candidate can be optimal under *some* ranking: linguistically possible, under a certain hypothesis about the nature of the constraint set. A candidate that is not optimal under any ranking is a *loser*, embodying structures predicted to be humanly impossible. Naively one might imagine that establishing loser status must involve a painstaking search of the far reaches of the ranking space, since a loser must fail on every single ranking of the constraint set. Here we will see that univeral loserdom is quite easy to determine, via a simple pencil-and-paper operation, or (with luck) mere inspection, given the relevant set of competitors, using the comparative tableau structure. (As we will see in §8, the procedure connects directly with the Recursive Constraint Demotion (RCD) algorithm of Tesar 1995, Tesar & Smolensky 1998, 2000; indeed, we are following the very path that leads Samek-Lodovici & Prince 1999 to their rendition of RCD.)

In the simplest case, loser status is immediately evident in a comparison between a loser and a candidate that harmonically bounds it:

(38) **Candidate *z* is harmonically bounded by *b*** over the constraint set $\{C_1\ C_2\ C_3\ C_4\ C_5\}$.

	C_1	C_2	C_3	C_4	C_5
z ~ b		L		L	

Candidate *z* is placed in the 'desired optimum' slot here: but desire is thwarted, because no constraint prefers *z* even though some constraints do prefer the competition. In general, harmonic bounding occurs when one candidate (above: *b*), is always at least as good as its rival (*z*) and is strictly better somewhere (C_2 and C_4). No ranking of the constraints can make *z* optimal.

More complex cases occur when access to optimal status is blocked by a confederation of several candidates. This situation is analyzed in detail in Samek-Lodovici & Prince 1999; here, we review a typical example. Suppose we have constraints $\{C_1\ C_2\ C_3\ C_4\}$ which impose the following order structures on a candidate set $\{a,b,c,d,z\}$.

(39)

$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$
a,c,d,z	c,d	a	b
|	|	|	|
b	b,z	b,z	a,d,z
	|	|	|
	a	d,c	c

These order relations can be modeled by the following assignment of violations (among others):
(40)

	C_1	C_2	C_3	C_4
z		*	*	*
a		**		*
b	*	*	*	
c			**	**
d			**	*

It is not immediately obvious that *z* loses on every ranking. The following procedure establishes it: first, construct the comparative tableau using *z* as the targeted optimum:

(41)

W ~ L	C_1	C_2	C_3	C_4
z ~ a		W	L	
z ~ b	W			L
z ~ c		L	W	W
z ~ d		L	W	

At this point, the failure of *z* can be spotted at a glance: rows [*z~a*] and [*z~d*] fuse to L^+. Of course, intricate multi-row arrangements may also fuse to L^+ in other cases, and we need a method for finding them that is more reliable than the primate visual system.

As a point of departure, let us observe that it is easy to tell when a row *cannot* participate in a collection that fuses to L^+: it will have in it a W that is matched in other rows only by W or *e* (blank), never by L. (The only way a W can participate in an L-tending fusion is to be matched somewhere to an L in its column.) Therefore we can seek out such *free* W's, and eliminate the rows they occur in. The result will typically be a smaller sub-tableau of the original. We then repeat the procedure until we have either eliminated the entire tableau, or reached an irreducible sub-tableau. In the latter case, the sub-tableau will be inconsistent, and so will any larger tableau that contains it.

We can define a function R that accomplishes this procedure.

(42) **Tableau Reduction**. Let T be a comparative tableau. R(T) is the sub-tableau of T obtained by removing all rows containing a free W.

(43) **Def. Free W.** In a tableau, a cell-entry W is termed *free*, iff there is no L in its column.

Let's apply the function R to tableau (41). Observe that only row [*z~b*] contains a free W, in the C_1 column We mark elimination by shading and line-crossing.

(44) R(T)

W ~ L	C_1	C_2	C_3	C_4
z ~ a		W	L	
~~z ~ b~~	~~W~~			~~L~~
z ~ c		L	W	ⓦ
z ~ d		L	W	

As a result of the row-removal procedure, the new tableau R(T) has one new free W (circled), in the row [*z~c*]. We may therefore reapply R nontrivially, eliminating this row.

(45) R(R(T))

W ~ L	C_1	C_2	C_3	C_4
z ~ a		W	L	
~~z ~ b~~	~~W~~			~~L~~
~~z ~ c~~		~~L~~	~~W~~	~~W~~
z ~ d		L	W	

At this point we are stuck: no row of the remaining tableau R(R(T)) has a free W. Further application of R will yield nothing new: $R^3(T)=R^2(T)$. The remaining rows must therefore fuse to L, and the tableau is inconsistent.

Candidates *a* and *d* form what Samek-Lodovici & Prince call a *Bounding Set*: each defeats *z* somewhere, and whenever one is worsted by *z*, another member of the set comes to the rescue, besting *z*. Samek-Lodovici & Prince show that every loser is associated with a Bounding Set; this result lies behind the correctness of the procedure outlined here.

8. RCD

Recursive Constraint Demotion is an algorithm guaranteed to produce a ranking that satisfies a consistent set of ranking arguments, and guaranteed to fail with an inconsistent set (Tesar 1995, Tesar & Smolensky 1998, 200). The fundamental observations are two in number. First: if a constraint never prefers a suboptimum — never assesses an L; only assesses free W and *e*— then it may stand at the top of the ranking. (Only constraints that disprefer the desired optimum, assess L, need ever be subordinated.) Second, if we remove such top-ranked constraints from consideration, along with the ranking arguments (comparative data; tableau rows; candidates) they dispose of, we are faced again with the same sort of problem, but reduced in size. We want to impose a ranking on a smaller set of constraints, based on a smaller set of ranking arguments. We therefore repeat the procedure, and if in the end we successfully rank all constraints, we have accomplished our goal. Put in terms of comparative tableau, the RCD procedure can be described as follows.

(46) **RCD**. Given a tableau T with some unranked constraints,

[1] Locate all constraint columns lacking L; *i.e.*, containing only free W and *e*.

[2] Place these constraints in a stratum ranked just beneath all previously ranked strata.

[3] Remove from consideration all constraints just ranked (tableau columns), as well as the ranking arguments they satisfy (tableau rows).

[4] Go to [1], applying it to the remaining rows and columns.

The procedure produces a 'stratified hierarchy' of constraints, in which members of the same stratum do not conflict and are ranked as as high as possible. Any linearization that respects stratum order will produced a totally-ordered ranking of the constraints.

Here's a brief illustration. Consider the following set of constraints and ranking arguments:

(47)

W ~ L	C_1	C_2	C_3	C_4
z ~ a		L	W	
z ~ b	L			ⓦ
z ~ c	W	W	L	

Step [1] of RCD leads us to identify C_4 as the only rankable constraint; no others assess free W's. Let us now rank it [step 2], and partition it off from the others [step 3]. Notice that the argument [*z~b*] is now satisfied: some constraint preferring *z* (C_4) dominates all those preferring *b* (just C_1).

(48) $C_4 \gg \{C_1\ C_2\ C_3\}$

W ~ L	C_4	C_1	C_2	C_3
z ~ a			L	W
z ~ b	W	L		
z ~ c		ⓦ	W	L

We now devote our attention to what remains: the sub-tableau [C_1 C_2 C_3] with rows [*z~a*], [*z~c*]. Step [1] identifies C_1 as the only rankable constraint in [C_1 C_2 C_3]. Observe that the free W of C_1 will satisfy the [*z~c*] row. Ranking C_1 below C_4 produces the following shrunken system:

(49) $C_4 \gg C_1 \gg \{C_2\ C_3\}$

W ~ L	C_4	C_1	C_2	C_3
z ~ a			L	ⓦ
z ~ b	W	L		
z ~ c		W	W	L

It now remains only to rank C_2 and C_3 . Proceeding with rigid formality, we execute step [1], observing that C_3 supplies the only free W: we rank it [step 2], and remove from consideration the row it subdues [step 3]. We then stand back to observe that nothing remains to be done. The resulting hierarchy can be tabulated in familiar fashion:

(50) $C_4 \gg C_1 \gg C_3 \gg C_2$

W ~ L	C_4	C_1	C_3	C_2
z ~ a			W	L
z ~ b	W	L		
z ~ c		W	L	W

Each row now begins with W, the sign of a well-regulated tableau in which all conflicts are successfully resolved in favor of the targeted optimum.

Observe finally that the inconsistency-detection procedure of §7 is a stripped-down version of RCD in which no attempt is made to remember the rankings that are discovered as the algorithm proceeds.

9. RCD Practicum

To illustrate the procedure just outlined, let us apply RCD to a basic-syllable-structure example slightly modified from Tesar & Smolensky 1998. For simplicity, we consider no candidates with complex onsets, nuclei, or codas. The target language is strictly CV on the surface, *via* insertion and deletion of C.

We limit ourselves to the most basic and familiar of syllable-structure constraints and faithfulness constraints:

(51) **Constraints**:

Markedness: ONSET
NOCODA

Faithfulness: DEP-C } *No insertion*
DEP-V
MAX-C } *No deletion*

The anti-insertion constraint DEP-C approximates Tesar & Smolensky's FILL^{Ons}. The anti-insertion constraint DEP-V corresponds to FILL^{Nuc} and the anti-deletion constraint MAX replaces their PARSE.

(52) **Data tableau** for /opek/ → .ʔo.pe.

/opek/	DEP-C	DEP-V	MAX	NOCODA	ONSET
☞ .ʔo.pe.■	1		1		
.o.pek.				1	1
■.pe.ki.		1	1		
■.pe.■			2		

We heuristically indicate deletion sites by ■,■ and place epenthetic elements in an outline font; we also abandon the tally-count of stars for the Arabic numerals.

Transmuting into comparative form, we have:

(53)

/opek/	DEP-C	DEP-V	MAX	NOCODA	ONSET
→ .ʔo.pe.■	1		1		
~ .o.pek.	$_0$ L		$_0$ L	$_1$ W	$_1$ W
~ ■.pe.ki.	$_0$ L	$_1$ W	$_1$		
~ ■.pe.■	$_0$ L		$_2$ W		

In purely comparative terms, this comes out as the following:

(54) **Comparative Tableau**

/opek/ → .ʔo.pe.■	DEP-C	DEP-V	MAX	NOCODA	ONSET
~ .o.pek.	L		L	Ⓦ	Ⓦ
~ ■.pe.ki.	L	Ⓦ			
~ ■.pe.■	L		W		

Let us now run the algorithm. The first pass identifies NOCODA, ONSET, and DEP-V as providers of free W. These enter the top ranking stratum: they do not conflict and cannot be crucially ranked among themselves.

(55) {NoCoda, Onset, Dep-V} >> {Dep-C, Max}

/opek/ → .ʔo.pe.■	NoCoda	Onset	Dep-V	Dep-C	Max
~ .o.pek.	W	W		L	L
~ ■.pe.ki.			W	L	
~ ■.pe.■				L	Ⓦ

It remains only to rank the last two faithfulness constraints and occlude the rows eliminated:

(56) **Final Step**

/opek/ → .ʔo.pe.■	NoCoda	Onset	Dep-V	Max	Dep-C
~ .o.pek.	W	W		L	L
~ ■.pe.ki.			W		L
~ ■.pe.■				W	L

More transparently, we have attained the following ranking:

(57) {NoCoda, Onset, Dep-V} >> Dep-C >> Max

/opek/ →	NoCoda	Onset	Dep-V	Max	Dep-C
.ʔo.pe.■ ~ o.pek.	W	W		L	L
.ʔo.pe.■ ~ ■.pe.ki.			W		L
.ʔo.pe.■ ~ ■.pe.■				W	L

Violations of NoCoda and Onset are avoided by deletion and ʔ-insertion respectively, as illustrated by the first row. When deletion is matched by deletion, V-insertion is avoided in favor of ʔ-insertion, as in row 2. And ʔ-insertion rather than V-deletion preserves the requirements of Onset (row 3).

10. Credit where Credit is Due

It is the cause, it is the cause, my soul.
Let me not name it to you, you chaste stars!
— *Othello*, V.ii

Optimality Theory exists to develop explanation-from-principle in the linguistic realm. The constraints relevant to a certain phenomenal domain give a subtheory of that domain, and the set of possible rankings of all constraints predicts how the subtheories will interact. Within phonology, this represents a perspective that grew in force and ambition through the 1980's, rejecting (on one hand) the notion of phonology as a lumbering descriptive apparatus of vast reach and little grasp, and (on the other) the notion that phonology is the artifactual product of functional 'laws' whose vagueness and inherent contradictions support 'reasonable expectations' and 'intuitively satisfying' accounts rather than predictions.

Like any other explicit system for managing relations between principles, Optimality Theory can be put to a variety of uses in the hands of practitioners, and these uses need not reflect any particular set of larger goals. If Optimality Theory is to be pursued in the context of explanatory ambitions, it is essential to be able to assess the patterns of explanation to which it gives rise. Here we indicate some directions of analysis.

Consider first the simple familiar case of consonants deleting to avoid NoCoda violations under the basic syllable structure theory. One might hear it said that "NoCoda drives the alternation." An examination of the key comparisons shows that this locution, though partly informative, needs refinement.

(58) **Deletion under the Basic Syllable Structure Theory**

/CVC/	NoCoda	Dep	Max	Remark
a. CV.■ ~ CVC.	W		L	Reject faithful candidate
b. CV.■ ~ CV.C𝕍		W	L	Delete, not Insert

NoCoda drives the alternation in the sense that it defines the target structure (it provides a 'postcondition', in the sense of Prince & Smolensky 1993/2002:§10). As can be seen immediately in tableau (58), if it were subordinated to Max, the desired, deleting candidate 'CV.■' would no longer win. However, the domination of Dep over Max is equally necessary, as the tableau makes clear. In this case, the domination relation chooses the particular breach of faithfulness that permits the desired target to be realized. Were this domination relation reversed, the deleting candidate would lose to an epenthetic candidate, under the basic theory.

This example illustrates the simplest kind of ranking requirement, which arises when a single W occurs with one (or more) L's, as in row (58)a and in row (58)b. Here it is transparently justified to attribute the optimum's bad performance on the L-assessing constraint(s) to the benefit obtained on the W-assessor. This is the kind of elementary situation one has in mind when speaking of 'driving'. The further import of the example is that several such W-assessors may independently be involved in this 'driving' relation with respect to a single L-assessor. Each W-assessor is responsible for a particular aspect of the input's behavior with respect to the L-assessor, and all aspects must be under control simultaneously for the desired behavior to happen. Since each W-assessor, and its ranking, is required if the desired optimum is to win, we can call this the *conjunctive* regime.

A more tantalizing case arises when multiple W's occur in the same row.

(59)

	A	B	C
ω ~ z	W	W	L

Suppose this ERC gives the totality of restrictions on the relations between constraints A,B, and C that are required for ω to be optimal. (Other constraints may fall under restrictions as well, due to other competitions involving ω.) Then either or both of A and B will dominate C in the various grammars that accord with [ω~z]. (As usual, we take a grammar to be a total order on the constraint set.) The driving relation will differ locally in these grammars. In A>>C>>B, the constraint A will be responsible for ω's performance on C, and in B>>C>>A it will be B that is responsible. The most interesting cases are A>>B>>C and B>>A>>C, in which *both* W-assessors dominate the L-assessor. (All these cases can also arise when they are independent reasons forcing such rankings.)

Suppose we have A>>B>>C in a particular grammar. The temptation is to simply declare that A is the responsible party governing ω's performance on C, since A is the highest-ranked constraint on which ω and *z* differ. But ω *would still win* even if A were subordinated to C. (If the ranking A>>B>>C is forced by independent considerations, then subordinating A will produce other optima elsewhere, while retaining ω.) Thus, B still plays a back-up role, which emerges in the factorial typology. B's role falls short of being crucial in the grammar at hand, though, since if it is subordinated to C or even simply omitted, the outcome remains the same. In this case, constraint B — and its ranking above C— is significant in placing the grammar within a set of related grammars in which ω is the predicted outcome. This is a disjunctive regime, and B retains force in explaining the goodness of ω.

To fix terminology, let us say that the lone W-assessor of nontrivial ERCs like those in (58) plays a *conjunctive* role in explaining the failure of the desired optimum on the L-assessors that accompany it; that a W-assessor among other W-assessors plays a *disjunctive* role when they can be ranked above all L-assessors; and that a disjunctive W-assessor plays a *subordinate* role when it is ranked below another W-assessor but above the L-assessor(s).[4]

These abstract considerations come directly into play in comparing divergent theories of the same (or closely related) phenomena. Recent work on the analysis of Tagalog *um*-infixation provides an instructive example. One line of analysis (Prince and Smolensky 1993/2002:§4.1) takes syllable structure markedness to be the basis of infixation (developing an insight from Anderson 1972). This work idealizes glottal stop-initial words as phonologically vowel-initial. McCarthy (2002b), in contrast, takes the glottal stop phonology seriously and rebuilds the analysis, taking into account as well the wider range of facts brought forth in Orgun & Sprouse 1999 (for detailed exploration of this

---

[4] The issues here are formally analogous to those arising in the analysis of causal structure. Take '>>' to be temporal precedence, the 'constraints' to be events, and an ERC connected with α to be a statement of the form 'at least one W-event must precede all L-events in order for outcome α to happen.' A world is a total ordering of events; the 'factorial typology' is the set of all such worlds. A set of ERCs connected to outcomes $\{\alpha_i\}$ delineates those worlds in which all the α_i occur together. We can then ask about the sense in which one event 'causes' another, and the configurations that arise will be like those sketched above, in which we ask, in essence, which constraints 'cause' the (increased) violation of which others. In a situation with the formal structure of (58), both W-events must precede the L event or the outcome ω will not occur, although there is no need for a particular temporal order among the W-events. In a complex situation like (59), outcome ω will occur in a world where A precedes B and B precedes C, but ω will also occur wherever B precedes C, regardless of the disposition of A. Thus B might not be taken to be the immediate cause of C in a world where A happens first, but B has status in the general theory as a potential cause.

and related infixation phenomena, see Klein 2002). We may then ask: what role, if any, is retained by syllable-structure considerations in the new analysis?

If we naively expect to find unitary driving principles, we might be tempted to say that the syllable markedness approach sees NoCoda as the key player in infixation, citing an example like this:

um+sulat → s**u.m**u.lat *avoiding* **um.**sulat. 'write, actor focus'

McCarthy shows that Dep-C is a prime mover, so we'd say:

um+sulat → **s**umu.lat *avoiding* **ʔ**umsulat

Proceeding in this fashion, we might expect to find that, in the Dep-C analysis, NoCoda has been dismissed from consideration.

To move to a sounder assessment, let us first examine some key comparisons as they play out under the Dep-C analysis (DCA). In the interest of focus, we remove from sight all constraints not relevant to our immediate concerns.

(60) **DCA**

/um+sulat/	Ons	Dep-C	NoCoda	*Complex	Al×seg[um]	Contig
→ s**u.m**u.lat			1		1	1
a. ~ **um**.su.lat	$_1$ W		$_2$ W		$_0$ L	$_0$ L
b. ~ ʔ**um**.su.lat		$_1$ W	$_2$ W		$_1$	$_0$ L

Remarks:

Ranking. In addition to the ranking restrictions that follow from (60), we have from McCarthy Dep-C>>NoCoda, via a transitivity argument involving MParse, not discussed here. We also have NoCoda>>*Complex, as in (62) below.

Candidates. We examine only the contrast between infixing and non-infixing outcomes, and do not attempt to deal with the various possible sites of infixation.

Constraints. The constraint *Complex forbids consonant sequences inside syllables. The constraint Al×seg[um] declares the the affix -um- should be at the left edge of the category it forms, and specifically lumps together in one violation class all those candidates in which -um- lies as much as one segment away from the edge. (McCarthy's treatment of infixation takes place in the context of a general argument about the nature of Alignment violation, which we will not even glance at here.) The constraint Contig requires that underlying contiguity relations within morphemes be preserved; infixation disrupts these (see Horwood 2002 for recent proposals).

What *drives* infixation here? Tableau (60) shows that the relevant configurations fall into the disjunctive regime, involving multiple W's, as in the abstract example (59).

In particular:

- The faithful candidate **umsulat* is dismissed by either Ons or NoCoda.[5]
- The epenthesized prefixing candidate *ʔumsulat* is dismissed by Dep-C, which is backed up by lower-ranked NoCoda.

At this point, it is useful to compare the DCA with the glottal-stop-less syllable markedness grammar of Prince & Smolensky.

(61) **Syllable Markedness Grammar**

/um+sulat/	Ons	Dep-C	NoCoda	*Complex	Align	Contig
→ s**u.m**u.lat			1		1	1
~ **um**.su.lat	$_{1}$ W		$_{2}$ W		$_{0}$ L	$_{0}$ L

This comparative row is isomorphic to its cognate (60)a in the DCA tableau. The two analyses, then, agree completely on the proposition that syllable-structure markedness eliminates *umsulat. They also agree that NoCoda plays a disjunctive or even subordinate role in driving infixation. Thus, even from the syllable markedness perspective, the example *sumulat* cannot be put forward as definitive evidence for the claim that NoCoda plays a conjunctive role in the analysis.

To achieve the direct confrontation with NoCoda, one must turn to the behavior of forms like /um+gradwet/ 'graduate$_{\text{Verb}}$, actor-focus' (cf. Prince & Smolensky 1993/2002:36).

(62) **DCA**

/um+gradwet/	Ons	Dep-C	NoCoda	*Complex	Al×seg[um]	Contig
→ gr**u.m**ad.wet			2	1	1	1
a. ~ **um**.grad.wet.	$_{1}$ W		$_{3}$ W	$_{1}$	$_{0}$ L	$_{0}$ L
b. ~ ʔ**um**.grad.wet.		$_{1}$ W	$_{3}$ W	$_{1}$	$_{1}$	$_{0}$ L
c. ~ g**um**.rad.wet			$_{3}$ **W**	$_{0}$ **L**	$_{1}$	$_{1}$

Here we observe in row (c) that the *only* constraint preferring the desired optimum is NoCoda. (McCarthy's revision of alignment theory does not distinguish between the two infixational candidates *gr**um**adwet* and *g**um**radwet*.) Thus, even in the DCA, the constraint NoCoda retains a conjunctive role in the analysis. Once again, the relevant tableau row (c) is isomorphic to one that occurs in the syllable markedness analysis (as may be inferred from the blankness of the Dep-C cell).

[5] Since Ons is not crucially dominated, while NoCoda is clearly dominated (by the various relevant faithfulness constraints, at least), it may be that a direct ranking relation can be established between them, in which case there would be no grammars where NoCoda plays the leading role in explaining the sumulat~umsulat contrast.

In both, the conjunctive role of NOCODA is to choose between infixational candidates (*grumadwet~gumradwet*). And in both, NOCODA plays a disjunctive role in promoting infixation over prefixation (as in *sumulat~umsulat*), in a competition to which Dep-C is irrelevant.

In sum, the Dep-C analysis incorporates rather than supersedes the syllable markedness analysis. In the cases reviewed here, it's a kind of proper super-analysis, dealing with new material unexamined in the earlier work, but accepting the syllable markedness explanations for the shared competitions.

A final observation. The form *grumadwet* is drawn from French 1988. Other speakers have remarked at various times that the form *gumradwet* is also acceptable. As noted in McCarthy 2002b, accepting this form requires re-ranking NOCODA and *COMPLEX.

(63)

/um+gradwet/	ONS	DEP-C	***COMPLEX**	**NOCODA**	AL×SEG[UM]	CONTIG
→ g**um**.rad.wet				3	1	1
a. ~ **um**.grad.wet.	1 W		1 W	3	0 L	0 L
b. ~ ʔ**um**.grad.wet.		1 W	1 W	3	1	0 L
c. ~ gr**u.m**ad.wet			1 **W**	2 **L**	1	1

In this speech variety, *COMPLEX ascends into the driver's seat, displacing NOCODA from its conjunctive role.

We conclude by observing that the various roles played by NOCODA, ONS, *COMPLEX, and DEP-C are rendered accessible only in the comparative tableau representation. To assess the role of a constraint, we must be able to see the roles of all the other constraints. To assess the import of a comparison, we must be able to distinguish the properties in which the competitors differ from those properties in which they are the same (the latter being those that are being properly 'controlled for'). Parsing the patterns of explanation implicit in optimality theoretic analysis rests on the logic of optimum-*vs.*-suboptimum comparison. The comparative tableau presents the structure required for investigating and manipulating that logic.

Appendix. Efficacy of Rundancy Reduction

Here we rephrase an argument from Prince & Smolensky 1993/2002, ch. 8, p. 146-150, showing how a constellated 13-row data tableau reduces to 4 transparent comparative rows, through W-extension and row-fusion.

First we give the base data tableau in (64); then, its reduction as (65). NB: The constraints are not arranged in domination order. The challenge is to find the ranking conditions under which the desired optimum ω wins against candidates (a)-(k), and then to show that these accord with conditions previously determined. The reader might like to try this from (64) alone.

(64) **Basic data**

/tα/→	ONS	*M/□	*P/□	*M/α	*M/*t*	*P/*t*	*P/α	NOCODA
ω ☞					*		*	
a .t́α.	*			*		*		*
b .t□́α.			*	*	*			*
c .t́.ά.	**					*	*	
d .t́.□ά.	*	*				*	*	
e .t́.α□́.	*		*	*		*		
f .□t́.ά.	*	*				*	*	
g .□t́.□ά.		**				*	*	
h .□t́.α□́.		*	*	*		*		
i .t□́.ά.	*		*		*		*	
j .t□́.□ά.		*	*		*		*	
k .t□́.α□́.			**	*	*			

(65) **Nonredundant comparative form**

/tα/ →	ONS	*M/□	*P/□	*M/α	*M/*t*	*P/*t*	*P/α	NOCODA
a **.tά.** ~ .t́α.	**W**			**W**[mh]	**L**[mh]	**W**[ph]	**L**[ph]	**W**
c **.tά.** ~ .t́.ά.	**W**[1]				**L**[1]	**W**[1]		
g **.tά.** ~ .□t́.□ά.		**W**[2]			**L**[2]	**W**[2]		
k **.tά.** ~ .t□́.α□́.			**W**[3]	**W**[3]			**L**[3]	

Discussion:

From the universal Peak and Margin hierarchies, we have

[mh] *M/α>>*M/*t* and

[ph] *P/*t* >>*P/α,

solving row (a).

From the Possible Onset Condition (Prince & Smolensky 1993/2002:) we have

[1] *P/*t* or ONS>>*M/*t*, solving row (c), *and*

[2] *P/*t* or *M/□ >>*M/*t*, solving row (g).

From the Possible Peak Condition (p. 144), we have

[3] *M/α or *P/□ >> *P/α, solving row (k). QED.

To achieve this reduction, we first remodel the data tableau (64) into its comparative form:

(66)

	/*t*α/ → **.tά.**	ONS	*M/□	*P/□	*M/α	*M/*t*	*P/*t*	*P/α	NOCODA
a	~ .t́α..	W			W	L	W	L	W
b	~ .t□́α.			W[k]	W[k]			L[k]	Ⓦ
c	~ .t́.ά.	**W**				**L**	**W**		
d	~.t́.□ά.	W[c]	Ⓦ			L[c]	W[c]		
e	~ .t́.α□́.	W[c]		W[k]	W[k]	L[c]	W[c]	L[k]	
f	~ .□t́.ά.	Ⓦ	W[g]			L[g]	W[g]		
g	~ .□t́.□ά.		**W**			**L**	**W**		
h	~ .□t́.α□́.		W[g]	W[k]	W[k]	L[g]	W[g]	L[k]	
i	~ .t□́.ά.	W		W					
j	~ .t□́.□ά.		W	W					
k	~ .t□́.α□́.			**W**	**W**			**L**	

At this point, the following remarks may be made:

1. Rows (i) and (j) are harmonically bounded by the desired optimum (no L, no ranking).
2. Row (c) implies row (d) by W-extension.
3. Row (g) implies row (f) by W-extension.
4. Row (k) implies row (b) by W-extension
5. (e) = (c) ∘ (k)
6. (h) = (g) ∘ (k)

From this it follows that the only independent, informative rows are (a), (c), (g), and (k).

References

ROA = Rutgers Optimality Archive, http://roa.rutgers.edu.

Alderete, John. 1999. *Morphologically Governed Accent in Optimality Theory*. Ph.D. dissertation, University of Massachusetts at Amherst. ROA-309.

Anderson, Stephen. 1972. On nasalization in Sundanese. *Linguistic Inquiry* 3, 253-268.

Crowhurst, Megan and Mark Hewitt. 1997. Boolean Operations and Constraint Interaction in Optimality Theory. ROA-229.

French, Koleen Matsuda. 1988. *Insights into Tagalog reduplication. Infixation and stress from non-linear phonology.* MA. Thesis, University of Texas Arlington. Summer Institute of Linguistics and the University of Texas at Arlington Publications in Linguistics 84.

Grimshaw, Jane. 1997. Projection, Heads, and Optimality. *Linguistic Inquiry* 28.4, 373-422. ROA-68.

Horwood, Graham. 1999. Anti-faithfulness and Subtractive Morphology. Ms. Rutgers University. ROA-466 (2001).

Horwood, Graham. 2002. Precedence faithfulness governs morpheme position. ROA-527.

Klein, Thomas. 2002. Infixation and segmental constraint effects: UM and IN in Tagalog, Chamorro, and Toba Batak ROA-535.

McCarthy, John. 2002a. *A Thematic Guide to Optimality Theory*. CUP.

McCarthy, John. 2002b. Against Gradience. ROA-510.

Orgun, C. Orhan and Sprouse, Ronald.1999., From MParse to control: Deriving ungrammaticality. *Phonology* 16, 191-220. Also ROA-244.

Prince, Alan. 1998. A Proposal for the Reformation of Tableaux. ROA-288.

Prince, Alan. 2000. Comparative Tableaux. ROA-376.

Prince, Alan. 2002. Entailed Ranking Arguments. ROA-500.

Prince, Alan & Paul Smolensky. 1993/2002. *Optimality Theory: Constraint Interaction in Generative Grammar.* RuCCS TR-2, Rutgers Center for Cognitive Science. CU-CS-696-93, University of Colorado Department of Computer Science. ROA-537.

Samek-Lodovici, Vieri. 1992. Universal Constraints and Morphological Gemination: a Crosslinguistic Study. Ms. Brandeis University. See Samek-Lodovici 1996.

Samek-Lodovici, Vieri. 1996. A unified analysis of cross-linguistic morphological gemination. ROA-149.

Samek-Lodovici, Vieri & Alan Prince. 1999. Optima. ROA-363.

Smolensky, Paul. 1995. On the structure of the constraint component Con of UG. ROA-86.

Tesar, Bruce. 1995. *Computational Optimality Theory*. Ph. D. Dissertation, University of Colorado. ROA-90.

Tesar, Bruce and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29.2, p. 229-268. See also ROA-155, ROA-156.

Tesar, Bruce and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.