# Comparative Tableaux

Alan Prince
Rutgers University, New Brunswick
2/12/2000

Abstract

Ranking and optimality are based on pairwise comparisons between a desired optimum and its competitors. The 'comparative tableau' identifies and represents the elements that figure in the logic of ranking arguments, a prerequisite for the understanding of any optimality-theoretic analysis. As a data-structure, the comparative tableau makes it straightforward to present and assess ranking claims; to find redundancies and contradictions in sets of ranking arguments; to employ the constraint demotion ranking algorithms; and to efficiently determine universal suboptimal status.
This paper revises and replaces Prince 1998/ROA-288.

Acknowledgment

Thanks to John McCarthy, 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 comments and discussion. The author claims responsibility for errors and oversights.

## 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 on which the two competitors differ.[1]

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

---

[1] To see the need for the pairwise restriction, note that "the highest-ranked constraint on which the competitors differ" denotes at best one constraint, but when we are talking about more than one suboptimum, different constraints may be decisive in each case: desired optimum $\omega$ may differ from $a_1$ only on $C_1$ and $a_2$ only on $C_2$ , for example.

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

To be able to reason efficiently and correctly with ranking arguments, we need a data structure that explicitly represents their constituent notions. Working from the observations just made, we can see that we must know of each candidate whether it *is a desired optimum* or *competes with a desired optimum (is a desired* sub*optimum)*. Of each constraint, we must know whether it *prefers the desired optimum*; or *prefers the desired suboptimum*; or *is neutral between them*.

The following array implements exactly 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.

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 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 signaling when one member of pair is better than the other.

## 2. Data Tableaux

The internal logic of the theory turns on comparison, but the direct encounter with linguistic forms (as it is usually understood) 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 invaluable 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:

(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 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* 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 nothing 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.

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

| | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| 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 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 1993:130). 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 computes 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; but in [*ω*~*b*], it goes blank. Pairwise cancellation is a necessary computational step on the route to evaluation, but in itself it is intrinsically unsuitable for multi-comparison collections.

To deal with legacy tableau-ware, it is useful to have a conversion patch. 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:

(6) Patched 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.[4]

[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, or the pair.This difficulty can be overcome by the heuristic of assuming suboptimum-centered mnemonics for *W* and *L* during annotation: such as *W*orse and, say, *L*ivelier (than the desired optimum). These are extensionally equivalent to the pair-centered or optimum-centered interpretations of (1).

[4] This should not be taken as an endorsement of the mega-tableau method of exposition, which aims to justify an entire multi-constraint ranking hierarchy in one swoop by massing data into a single multi-row tableau. Nothing can replace careful analysis of the ranking structure of a proposed grammar or eliminate the need to show that the hoped-for optima are indeed optimal over the whole candidate set.

We observe finally that the comparative tableau eliminates the need for various auxiliary annotations intended to guide the eye to 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 "!", and because the optimum's success on a comparison is marked unmistakably by the occurrence of *W* as the leftmost non-blank cell in the row, there is no need for shading to set it off. Like those in perpetual shade, any constraint that never assesses a leftmost *W* does no work in the analysis and its inertness can be readily spotted. Another situation that evokes typographical emphasis occurs when a tentative constraint hierarchy selects the wrong form. Once again the comparative tableau tells the tale quite visibly: the first-encountered index in the left-to-right sweep will be *L*, sufficient to indicate that the second term of the [$\omega$~*a*] comparison is the undesired winner.

## 3. A typology of relations

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

(7) 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 $\omega$ 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 (7) supports the ranking argument $C_2 >> C_1$, because the form $\omega$ is the desired optimum. Tableau (8) supports the opposite conclusion:

(8) 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.

(9)

| | $C_1$ | $C_2$ |
|---|---|---|
| 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.

## 4. Practicum

To illustrate the different character of data and comparative tableaux, we analyze a candidate set, slightly modified from Prince & Smolensky 1993:122, in both forms. Epenthesized segments are capitalized and 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 (cf. 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.

(10)

| /wiţe/ | LEX≈PR | DEP-V | FREE-V | ALIGN | DEP-C | MAX |
|---|---|---|---|---|---|---|
| .wi.ţe | | | * | | | |
| .wiţ.■ | * | | | * | | * |
| .wiţ■.ŢA. | | * | | * | * | * |
| .wi.ţ■A. | | * | | * | | * |

Comparativized, we have:

(11)

| /wiţe/ → | LEX≈PR | DEP-V | FREE-V | ALIGN | DEP-C | MAX |
|---|---|---|---|---|---|---|
| .wi.ţe. ~ .wiţ.■ | W | | L | W | | W |
| .wi.ţe. ~ .wiţ■.ŢA. | | W | L | W | W | W |
| .wi.ţe. ~ .wi.ţ■A. | | W | L | 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, given the ranking at hand, the constraints LEX≈PR and DEP-V do all the work in eliminating competitors.

The notation used here repeats the desired optimum in every row, a convenience that some might wish to forgo. It can easily be eliminated from the rows, as long as the focus of the competition, the desired optimum, remains clear. Here we indicate the desired optimal *map* in the upper left-hand box, and list only the competing outputs below it.

(12)

| /wițe/ → .wi.țe. | LEX≈PR | DEP-V | FREE-V | ALIGN | DEP-C | MAX |
|---|---|---|---|---|---|---|
| ~ .wiț.■ | W | | L | W | | W |
| ~ .wiț■.Ț𝔸. | | W | L | W | W | W |
| ~ .wi.ț■𝔸. | | W | L | W | | W |

Stepping back from the full ranking displayed here, let us demonstrate how a ranking argument might be constructed in practice, given the type of information presented in a comparative tableau.

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. Assume that we still have to determine the ranking of LEX≈PR and DEP-V relative to each other and to the already ranked constraints. Turning to *wițe* for illumination, we obtain the comparative data shown in (12). The already-established dominance of FREE-V tells us that none of the *W*'s in the last three columns can be used to establish the success of *wițe* in the comparisons at hand. Therefore, we deduce that LEX≈PR and DEP-V must dominate FREE-V.

These considerations have not yet fixed a ranking between LEX≈PR and DEP-V. To complete this mini-argument, let us examine a further relevant datum:

(13)

| /ŗelk/ → | 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.

Note that the suboptimum wins (*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 be a prosodic word where the monosyllabic suboptimum is not and cannot be due to high-ranked but unmentioned FT-BIN.

For purposes of comparison, we display a data tableau dealing with the same competition:

(14)

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

We have simplified 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 (13) incorrect, only incomplete. The second, more serious simplification is the failure to survey *all* the constraints involved. 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 *ɟelka* and disprefers *ɟel*, we cannot safely conclude that LEX≈PR is doing the work. (This kind of failure also lies behind a misstep occasionally encountered in the literature — the putative factorial typology constructed on a proper subset of the relevant constraints, without concern for the effects of the others.) The reader might wish to examine Prince & Smolensky 1993:ch. 7 for a fuller account of the Lardil hierarchy. The purpose here has been to illustrate in miniature how ranking arguments play out over comparative tableaux.

## 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. Keer notes 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 easily read off the data tableau:

(15) 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 violation of $C_1$ and of $C_2$ individually. In this case, the comparative tableau hides the key violation of constraint $C_1$:

(16) Conjunct contribution rendered opaque

| | $C_1$&$C_2$ | $C_1$ | $C_1$ |
|---|---|---|---|
| a~b | W | | W |

The same comparative tableau would be obtained if both candidates had *succeeded* on $C_1$.

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.

## 6. Row manipulations: entailment and inconsistency of arguments

The tableau row encodes an *elementary ranking condition*: at least one constraint assessing *W* (preferring the targeted optimum) must dominate all constraints assessing *L* (those preferring the targeted suboptimum). 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 logically entails another, there is no need to produce the superfluous entailed ranking argument or the suboptimal candidate that goes with it; discussion should be simplified by omission of the redundant. 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 redundancy and contradiction. (The discussion here summarizes material from Prince 2000.)

Two 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.

(17)

| W ~ L | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| ω ~ a | W | | L |
| ω ~ b | W | W | L |

The argument from row [*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 informs us that $p \rightarrow p \vee q$; this is an instance.

With *L*, we have the following:

(18)

| W ~ L | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| ω ~ a | W | | L |
| ω ~ b | W | L | L |

Now the situation 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 $C_1 >> C_3$. holds, because $p \wedge q \rightarrow p$.

The general relation between one tableau row and another is given by this condition:

(19) **Row entailment**. If $A_1$ and $A_2$ are elementary ranking conditions, corresponding to individual rows in a tableau, and if $W(A_i)$ and $L(A_i)$ are the sets of constraints preferring respectively the desired optimum and the desired suboptimum of $A_i$, then $A_1 \rightarrow A_2$ iff $W(A_1) \subseteq W(A_2)$ and $L(A_1) \supseteq L(A_2)$.

*Pf.* See Prince 2000.

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.

(20) **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.

(21) **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.

In example (17) above, we see that [*ω~a*]➝[*ω~b*] by *W*-extension. In example (18), we have [*ω~b*]➝[*ω~a*] by *L*-retraction. In each case, the entailed argument merely asserts a watered-down 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.

For more elaborate logic involving several rows, it turns out that we need just one more principle of row manipulation, a way of combining two rows into a third by "summing" their corresponding entries. The principles of summation are these, writing *e* for 'blank':

(22) Summation of cell entries.

$X+L = L+X = L$
$X+e = e+X = X$
$X+X = X$

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

(23) Row summation

| | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| $A_1$ | W | L | |
| $A_2$ | | W | L |
| $A_1$+$A_2$ | W | L | L |

Observe that $A_1$+$A_2$ contains the information that $C_1$>>$\mathbf{C_3}$ must hold, which is not explicit in rows $A_1$ and $A_2$ individually, but follows from them by transitivity of '>>'. Indeed, *all* elementary ranking conditions that follow by virtue of the properties of '>>' also follow by row summation (Prince 2000). It is worth noting that summation 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 simply a way of operating with rows so that their logical consequences are brought out and expressed in the standard format.

A further related use of summation, perhaps even more useful, is shown in the following, where all non-blank cells sum to *L*.

(24) Summing to *L*

| | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| $A_1$ | | W | L |
| $A_2$ | | L | W |
| $A_1$+$A_2$ | | L | L |

A ranking argument with the structure of $A_1$+$A_2$ cannot be satisfied: the lack of *W* indicates that its implicit desired *sub*optimum would always win. This means that ranking arguments $A_1$ and $A_2$, which jointly imply this conclusion, cannot be jointly satisfied and must therefore be contradictory. And so they are: $A_1$ says $C_2$>>$C_3$ and $A_2$ counters with $C_3$>>$C_2$ .

A general technique for demonstrating the inconsistency of tableaux emerges: if any collection of rows in a tableau sums to *L*, the tableau cannot support a consistent ranking. (Note that summation is associative and commutative so that order of summation doesn't matter.)

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

*Remark*. This follows from propositions in Samek-Lodovici & Prince 1999 and Prince 2000.

Let us now put this result to immediate 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. 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 §7, the procedure connects directly with the Recursive Constraint Demotion (RCD) algorithm; indeed, we are following the very path that leads Samek-Lodovici & Prince 1999 to their version of RCD.)

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

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

| W ~ L | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|---|---|---|---|---|---|
| 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, the bound (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*}.

(27)

| $\mathbf{C_1}$ | $\mathbf{C_2}$ | $\mathbf{C_3}$ | $\mathbf{C_4}$ |
|---|---|---|---|
| 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):

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| 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:

(28)

| 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*] sum to *L*. Of course, more complex multi-row arrangements may also sum 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 sums 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 sum is to be matched somewhere to *L*.) 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.

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

Let's apply the function R to tableau (28). Observe that only row [*z*~*b*] contains a free *W*, in the $C_1$ column We mark elimination by shading and line-crossing.

(30) 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 | |

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.

(31) 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 sum 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). 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.

(32) 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’, 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 abstract illustration. Consider the following set of constraints and ranking arguments:

(33)

| W ~ L | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| z ~ a | | L | W | |
| z ~ b | L | | | W |
| 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 $C_1$).

(34) $C_4 >> \{C_1\ C_2\ C_3\}$

| W ~ L | $C_4$ | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|---|
| z ~ a | | | L | W |
| z ~ b | W | L | | |
| z ~ c | | W | 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:

(35) $C_4 >> C_1 >> \{C_2\ C_3\}$

| W ~ L | $C_4$ | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|---|
| z ~ a | | | L | W |
| z ~ b | W | L | | |
| z ~ c | | W | 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], remove from consideration the row it subdues [step 3], and stand back to observe that nothing remains to be done. The resulting hierarchy can be tabulated in familiar fashion:

(36) $C_4 >> C_1 >> C_3 >> C_2$

| 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.

## 9. RCD Practicum

We conclude with an application of 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:

**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.

**Data Tableau** for the forms to be considered:

(37) Data tableau

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

We heuristically indicate deletion sites by ■,■ and place epenthetic elements in an outline font.

(38) Comparative Tableau

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

(39) {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 | W |

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

(40) 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:

(41) {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).

# Appendix. Efficacy of Rundancy Reduction.

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

First we give the base data tableau in (42); then, its reduction as (43). Double lines indicate rankability: the challenge is to find the conditions under which the desired optimum *ω* wins and show that these accord with ranking conditions previously determined. The reader might like to try this from (42) alone.

(42) Basic data

| | /tα/ → | ONS | *M/□ | *P/□ | *M/α | *M/*t* | *P/*t* | *P/α | NOCODA |
|---|---|---|---|---|---|---|---|---|---|
| ω | ☞ *.tά.* | | | | | * | | * | |
| 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□.* | | | ** | * | * | | | |

(43) 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] | |

From the universal Peak and Margin hierarchies, we have *M/α>>*M/*t* and *P/*t* >>*P/α, solving row (a).
From the Possible Onset Condition (p. 144) 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 star tableau (42) into its comparative form:

(44)

| | /tɑ/ → | ONS | *M/□ | *P/□ | *M/α | *M/*t* | *P/*t* | *P/α | NOCODA |
|---|---|---|---|---|---|---|---|---|---|
| **a** | ***.tά.*** ~ *.t́α.* | W | | | W | L | W | L | W |
| b | ~ *.t□́α.* | | | W[k] | W[k] | | | L[k] | W |
| **c** | ~ *.t́.ά.* | **W** | | | | **L** | **W** | | |
| d | ~*.t́.□ά.* | W[c] | W | | | L[c] | W[c] | | |
| e | ~ *.t́.α□́.* | W[c] | | W[k] | W[k] | L[c] | W[c] | L[k] | |
| f | ~ *.□t́.ά.* | W | 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) show harmonic bounding 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 in the tableau are (a), (c), (g), and (k).

## References


Alderete, John. 1999. *Morphologically Governed Accent in Optimality Theory*. Ph.D. dissertation, University of Massachusetts at Amherst. ROA-309. Rutgers Optimality Archive, http://ruccs.rutgers.edu/roa.html.

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

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

Horwood, Graham. 1999. Anti-faithfulness and Subtractive Morphology. Ms. Rutgers University.

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

Prince, Alan. 2000. Entailed Ranking Arguments. Ms. Rutgers University.

Prince, Alan & Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. RuCCS TR-2, Rutgers Center for Cognitive Science. To appear, MIT Press.

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

Samek-Lodovici, Vieri & Alan Prince. 1999. Optima. Ms. University of London & Rutgers University, New Brunswick. 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.