Optima

Vieri Samek-Lodovici & Alan Prince
ucljvsl@ucl.ac.uk prince@ruccs.rutgers.edu

Draft: 11/22/99

TABLE OF CONTENTS

ABSTRACT

Linguistic typology turns on the distinction between candidates that are optimal under some ranking and candidates that are never optimal under any ranking: this is the distinction between potential 'winners' and perpetual 'losers'. In this paper we develop necessary and sufficient conditions that decide the winner/loser status of any candidate without requiring that rankings be examined. To facilitate the discussion, we formulate Optimality Theory in a way that emphasizes its order-theoretic underpinnings.

Generalizing the familiar notion of harmonic bounding (Samek-Lodovici 1992, Prince & Smolensky 1993), we show that a candidate is a loser if and only if it has a non-null *bounding set* that meets two general conditions. Checking these conditions requires no reference to ranking at all; it is done on a constraint-by-constraint basis, and the only information needed is the relation on each constraint between the putative loser and the members of a proposed bounding set for it. Bounding sets are limited in size: they need be no larger than the constraint set, and will typically be much smaller. A set of candidates that does not satisfy the bounding set criteria can nonetheless certify a loser's status by providing, for each ranking, a candidate that is better than the loser; but any such set must contain a bounding set. The notion of bounding set thus yields a complete, ranking-free characterization of loser status.

How is a bounding set to be found? The pursuit of the bounding set leads, by recursive exclusion of nonbounds, to the construction of a 'favoring hierarchy' from the constraint set. The favoring hierarchy is, we show, equivalent to the 'target hierarchy' of Tesar 1995 and Tesar & Smolensky 1998, and its recursive definition parallels their Recursive Constraint Demotion (RCD) algorithm. A candidate is a winner if and only if it has a favoring hierarchy that exhausts the constraint set. An exhaustive favoring hierarchy leads to a ranking on which a candidate is guaranteed to win, if it wins on any ranking at all. For a loser, the construction of its favoring hierarchy leaves a residue of constraints that cannot be integrated into the hierarchy and a corresponding set of refractory candidates that cannot be eliminated in competition with the loser. From these residual candidates, a bounding set can be readily constructed; in addition, the maximal bounding set is identified. The size of the residual set of constraints also leads to a tighter upper bound on size of a loser's minimal bounding sets.

These results provide the analyst with new tools for handling the crucial winner/loser distinction. They affirm the theoretical centrality of RCD and its associated construct, the favoring hierarchy, which originally arose in the context of learning and computational issues, but here proves to be indispensable for understanding the core structure of the theory. The fully order-theoretic approach developed here also provides a new perspective on the key notions of bounding, evaluation, and optimality.

ACKNOWLEDGMENTS

The authors' last names are arranged in descending lexicographic order. They are grateful for support from NSF-SBR-9511891 (Jane Grimshaw, PI) and from the John Simon Guggenheim Memorial Foundation, respectively. Thanks are due to Bruce Tesar, Paul Smolensky, and the participants in the Rutgers Optimality Research Group for their helpful remarks, and to the audiences at the 2nd OT-Syntax Workshop at the University of Stuttgart and the SFB-Colloquium at the University of Konstanz for their valuable questions and comments.

Optima

Vieri Samek-Lodovici
University College, London
Alan Prince
Rutgers University, New Brunswick

0. Introduction and Overview

Every constraint set under OT divides candidates into two very different species:

- *WINNERS,* potential optima: candidates optimal under *some* ranking,
- *LOSERS*: all others, those *never* optimal under any ranking.

Intrinsic to the theory, the dichotomy between sometime-winners and perpetual-losers emerges whenever constraints and candidate sets are well-defined: rankings may be free, or limited by meta-conditions; the candidate set may be generated from a lexical representation, or defined in some other manner. The *winners* comprise the grammatical universe; the *losers* populate the complementary netherworld of the impossible.

Since arguments in favor of any hypothesis must at some point advert to the success of its predictions, the analyst must be able to characterize, for each candidate set, its winners and its losers. Less obviously, perhaps, all competitions for optimality may be conducted *between winners*. Because the crucial rankings in a grammar are completely determined by inter-winner relations, losers are grammatically inert. No ranking argument need ever refer to a *loser*.[1]

It follows that key aspects of the theory have a finitary cast. With a finite number of rankings, there can be only finitely many winners, or more precisely, grammatically-distinct classes of winners. (If two or more candidates share optimality on some ranking, they must perform equally well on all constraints, and are therefore indistinguishable by the constraint set.) Losers, by contrast, may come in infinite numbers, and often do: licit structures exist that contain any number of epentheses, adjunctions, and recursive expansions. To be optimal is to be better than *all* other grammatically-distinct members of the candidate set; yet if the winners are known, optimality with respect to some particular ranking can be conclusively demonstrated from the finite set of winner-winner comparisons.

If there is value to such finitude, then this result may be interpreted as supporting the basic, occamite strategy of the theory, which is to avoid stipulations in GEN as well as in CON, relying on interaction rather than declaration to explain limits on linguistic form. GEN must be allowed to produce large candidate sets — there being no good reason to set arbitrary numerical bounds on epentheses, adjunctions, and so on, when such bounds already follow from constraint interaction. The overall theory nevertheless retains a finitistic character, even in the abstract, since all the competitive action takes place among the set of potential winners and the legions of losers are irrelevant.

[1] Terminological note. In this paper, we will user the term ‘loser’ only to mean ‘candidate that never wins under any ranking’ and the term ‘winner’ to mean only ‘candidate optimal on some ranking’.

Learners and analysts, of course, must contend with candidate sets as Gen defines them, and losers do not declare their status openly. At first glance, it might seem difficult to demonstrate a candidate's universal suboptimality. A loser wins on *none* of the *n!* rankings of a freely-rankable constraint set of size *n*, and checking through such a mass of rankings is not generally practicable. But a much more direct path is provided by Tesar's Recursive Constraint Demotion (RCD) algorithm (Tesar 1995:74ff, Tesar & Smolensky 1995, 1996ab, 1998, in press). Given a set of pairwise competitions, each pitting a desired optimum against a desired suboptimum — a set of elementary ranking arguments, or 'mark-data pairs' in the parlance — RCD is guaranteed to find a ranking that will render all the desired optima better than their competitors, if such a ranking exists. If RCD fails to produce a successful ranking, then *no such ranking* is possible; this means that the procedure can be used to find losers effectively. For *n* constraints, RCD requires at most *n* passes to make its determination.

We will develop our attack on the problem from a different angle, though RCD will re-appear in the end. We seek a ranking-independent necessary-and-sufficient condition for loser status, one that depends only on the structure of individual constraints. We begin with the familiar notion of 'harmonic bounding' (Samek-Lodovici 1992; Prince & Smolensky 1993:176). A candidate is *harmonically bounded* if there is another candidate that is (a) at least as good on all constraints, and (b) better on at least one.[2] Although the definition makes no reference to the ranking of constraints, a harmonically bounded candidate can never win on any ranking: it is a *loser*. To see this, consider the competion between such a candidate and its bound. The constraints on which they tie cannot decide between them; but on all others, and there is at least one, the bound will win; with no constraints preferring the bounded candidate, ranking cannot come to its aid, and it is a lost cause. The bounding candidate need not be a winner itself, but its existence nevertheless dooms whatever it bounds (Prince & Smolensky 1993:95).

Harmonic bounding is a special case, in which the easily-ascertained behavior of a single candidate, the bound, certifies that another candidate is a loser. Generalizing, we will develop the notion of a 'bounding set', a set of candidates meeting two ranking-independent conditions analogous to those for simple harmonic bounding, which *collectively* stifle a loser. The empirical importance of collective bounding can be seen in a recent study by Tesar (1999). Examining a system of 10 prosodic constraints that yields a large number of distinct quantity-sensitive stress patterns, he finds a significant number of losers that are collectively bounded, growing as the forms increase in length. Examining his data, we find that for bisyllables and trisyllables, simple harmonic bounding is the norm for losers, but collective bounding becomes important for longer words. Among four-syllable forms, approximately 3/4 are losers in each candidate set, on average, and of these approximately 1/6 are collectively bounded. Among the five-syllable forms, nearly 5/8 are losers, on average, and of these nearly 1/5 are collectively bounded.

We will show that a candidate is a loser *if and only if* it has a non-null bounding set. (In classical harmonic bounding, the bounding set has just one element.) We will find that every

[2] The language comes from the theory of order (cf. e.g. Davey & Priestley 1990:27), not the theory of quantification: hence the regular, denominal participle: $[_V[_N$ bound$]]$*ed*. An element β is an upper *bound* for a set of elements S with an order $\geq$ on it, if for all $x \in S$, $\beta \geq x$. E.g. 10 is an upper bound for {1,2,3} and similarly 0 is a lower bound for set of nonnegative integers {0,1,2,...}, etc.

bounding set must contain a finite bounding set within it — for *n* constraints, no more than *n* candidates are needed to collectively bound a loser. Furthermore, when we examine the most general notion of a set of candidates witnessing the failure of a loser — a 'covering set' that contains, for each ranking, an element beating the loser — we will see that any such covering set must contain within it a *bounding set* as defined here. These results show that the bounding set construction completely answers the request for a finitistic, ranking-independent guarantee of loser status.

Verifying that a given collection of candidates is indeed a bounding set requires nothing more than checking it against each constraint. Exhibiting a non-null bounding set for a candidate is a quick and rigorous proof of its perpetual suboptimality. But how can such a bounding set be found? The obvious direct attack has power-set combinatorics: it is taxing, even for a relatively small candidate set, to check all one-element subsets, all two-element subsets, ... up to one short of the size of the candidate set at hand, or the number of constraints, whichever is smaller.

We therefore pursue a complementary line, seeking to *exclude* those candidates which cannot possibly be in the bounding set. The key observation is this: if the candidate in question is at least as good as any other candidate on a constraint, then any potential members of its bounding set must do equally well. All elements in the candidate set that do worse can be eliminated from consideration. (This follows from the definition of bounding set presented in §2.1 below.) It is terminologically useful to be able to refer concisely to the property of 'being the best'; let us say that a constraint 'favors' those members of a given set of candidates which have the fewest violations, those which perform as well as or better than any other members of the set under consideration. *Favoring* is highly relative and depends entirely on the constitution of the set of candidates at hand. For example, ONSET will favor even such a candidate as *.a.o.* if it is the only member of the set of candidates, or if all the other members of the set have even more onsetless syllables. Situations like this, where success is not the same as satisfaction, arise commonly when constraints are dominated.

To implement the strategy of exclusion, we recursively construct a 'favoring hierarchy' for the candidate whose winner/loser status we wish to assess. (Here we preview the construction; below we explore its details.) The first step is to collect all the constraints which favor the targeted candidate: these we set aside as the first rank or stratum of the Favoring Hierarchy. We then gather the set of candidates which are uniformly favored by *all* the constraints in this first stratum, thereby eliminating disfavored non-bounds; these are precisely the candidates that survive evaluation by the first stratum. We next turn our attention to the behavior of this new, reduced set of candidates with respect to the remaining constraints (should there be any). The procedure is recursively repeated. We collect those constraints that now favor the targeted candidate in the context of the reduced candidate set; these form a second stratum. Once more we cull the set of candidates, gathering those favored by all the constraints in the second stratum, which again excludes non-bounds. The procedure continues until either (a) all constraints have joined the Favoring Hierarchy, or (b) a dead-end is reached, in which none of the remaining constraints favors the targeted candidate. In the first case, the candidate is a winner, a potential optimum; in the second case, it is a loser and, by virtue of the successful recursive exclusion of non-bounds, the surviving, non-excluded candidates provide us with the material to construct the non-null bounding set we have been seeking. (Those surviving candidates that are better than the loser on some remaining constraint form a bounding set, as we will see below.)

The recursive construction of the favoring hierarchy derives from bounding set considerations and uses a selection procedure based on the order-structure imposed by constraints. Although Tesar's RCD is motivated by learning considerations and manipulates violation patterns rather than order relations, we will see in §3.2 below that it is precisely equivalent to our construction. The Favoring Hierarchy is exactly the stratified 'target' hierarchy produced by CD algorithms, with each constraint placed as high as it can be; and the steps of the recursive definition of the Favoring Hierarchy can be seen to parallel the steps of classic RCD (§3.2.2). These results confirm that the RCD procedure is more than a strategem, useful to learners, for efficiently cutting through the tangle of disjunctive possibilities entailed by a set of ranking arguments. The notions involved in RCD turn up unavoidably in analysis of issues central to the theory, even in its most abstract form, and the order-theoretic perspective we advocate here provides a conceptual vantage on RCD that illuminates its basic properties.

Because the claims offered here are not always obvious and typically cannot be justified by citing examples, it is necessary to demonstrate their validity. We take the occasion to develop Optimality Theory from the ground up in purely order-theoretic terms: we will think of a constraint in terms of the order it imposes on the candidate set. This notion is implicit in the violation-calculus; making it explicit will shed light on the structure of the theory and yield a useful perspective on bounding and on RCD. Working from the observation that active constraints *shrink* the candidate set, we will also view a constraint as a *function* from candidate sets to candidate sets. This will allow us to view the construction of constraint hierarchies as function composition (in this we are anticipated by Karttunen (1998)[3]), emphasizing the notional continuity of *constraint* and *constraint hierarchy*.

We begin (§1) by establishing that relations among the set of potential winners fully determine a ranking. We move on to state and explore the bounding set conditions (§2), and address its relation to the RCD after characterizing the notions of *constraint* and *optimality* (§3). We then lay the formal groundwork (§4) and establish our main propositions in detail (§5).

1. Winners and Ranking

A ranking argument compares a desired optimum against a competitor, yielding conditions which ensure that the desired optimum fares better than that competitor on the constraint hierarchy. *Every constraint preferring the suboptimal competitor must be dominated by some constraint preferring the desired optimum*. But if the desired optimum is to be optimal, it must survive comparison not just with one competitor but with *every* other candidate. Finding the conditions that guarantee a candidate's optimality will in general require reasoning from a number of pairwise competitions.

[3] Karttunen's broader argument must, however, be viewed with skepticism. Failing to distinguish *comparison* from *counting*, he advances the curious assertion that the only difference between rule-package serialism and OT is that OT requires unlimited counting. (OT requires no counting.) Karttunen also seems to be arguing that because rule-package serialism and OT both call on notions of *order*, they must be somehow equivalent — as if all theories using similar formal notions were indistinguishable.

None of these comparisons need be with *losers*, however, because of the following fact: a candidate is optimal on a given ranking if and only if on that ranking it is at least as good as the *potential winners* among its candidate set.

Optimal status straighforwardly implies survival against all other winners, because optimality implies success against everything. To see that the implication holds in the other direction, assume that on the given ranking, our desired optimum is at least as good as all candidates that are optimal on *some* ranking (‘survives against the winners’). But among that set of *winners* is a candidate ω that is optimal on the very ranking we are interested in. Since our desired optimum, by assumption, is as good as or better than any of the winners, it either *is* ω or does just as well as ω on the ranking at hand. (It cannot be strictly better than ω, because nothing is.) In any case, it is optimal. The following proposition records this finding:

(1)**Proposition. Determination of Ranking by Winners.**

Let Σ be a set of constraints, K a set of candidates. Let W(K,Σ) be the winners in K, the set of candidates that are optimal for some allowed ranking of Σ, and let $\omega \in$K.

Then ω is optimal for any ranking R of Σ iff ω is as good as or better than *every* element of W(K,Σ) on the ranking R.

Proof: Along the lines of the discussion in the text. See §5.

The utility of the proposition is quite general: it holds regardless of whether there are various extraneous conditions that limit the allowed rankings on the constraint set, and it holds over any constraint set, no matter how chosen. It also shows that, given a set of grammatically-distinct candidates, a finite number of comparisons — those with the other winners— is sufficient to establish a candidate’s optimal or suboptimal status.

We conclude by noting that only a very proper subset of relations among potential winners need be examined, in the usual case. At the outermost limit, if we limit attention to grammatically distinct candidates, so that we have one optimum per ranking, there can be at most $n!$ distinct winners with n constraints in Σ. But a totally ordered set of n constraints can be ranked by $n-1$ optimum-suboptimum comparisons, one for each ranking-adjacent pair. At the worst, if we proceed in an inefficient way that leads us to accumulate one comparison for each pairwise ranking relation among the constraints, we need only $n(n-1)/2$ such comparisons (cf. the data-complexity limits on the algorithms of Tesar 1995).[4]

[4] As an example, consider the following three constraint system. Each constraint recognizes 3 levels of violation; we use numerals to symbolize the 6 candidates we need to give distinct optimum to each of the 3! possible rankings. Constraint A: {1,2}>{3,4}>{5,6}. Constraint B: {3,6}>{1,5}>{2,4}; Constraint C: {4,5}>{2,6}>{1,3}. A moment’s calculation show that all candidates are winners. But if, for example, we want 1 to be optimal, we need merely note that $1 \succ 2$ (so B>>C) and $1 \succ 3$ (so A>>B), getting a total ranking with only 2 comparisons, the absolute minimum possible.

2. Harmonic Bounding and Beyond

In this section, we show how the notion of *harmonic bounding* can be generalized to provide a full, ranking-independent account of the conditions under which a candidate fails to be optimal under any ranking. The context of the inquiry, here as elsewhere, will be set quite broadly. We will think of a *set of candidates* as any collection of candidates, a *constraint set* as any collection of constraints. Linguistic theories and subtheories will impose considerable additional structure. For example, even the notion 'candidate' is quite complex in practice: a candidate in current phonology is not a form but a *mapping* between an input and an output, with correspondence relations of various kinds between them that are subject to evaluation. From our vantage, though, 'candidate' is an atomic, unanalyzed notion. Our goal is to achieve results that will apply to any system that uses the very general notions under analysis here.

2.1 Collective Harmonic Bounding

An important and easily detectable kind of loser is a candidate that is *harmonically bounded* by some other single candidate.

(2) **Harmonic Bounding**. A candidate $z \in K$ is *harmonically bounded* relative to a constraint set Σ if there exists a candidate $\beta \in K$ meeting two conditions:

- **Strictness**. β is strictly better than z on at least one constraint in Σ.
- **Weak Bounding**. β is at least as good as z on every constraint in Σ.

As noted above, no harmonically bounded candidate can ever be optimal. The bound need not even be a winner.

Terminological note. It is useful to agree on some language to refer to the key relations that show up repeatedly in bounding theory. Constraints and constraint hierarchies alike determine whether a candidate ω is better than z, which we can write $\omega > z$, or worse than z, which we can write $z > \omega$. If ω is neither better nor worse than z, then we cannot write '$\omega = z$' because this means that ω ***is*** z. We need to say, more clumsily, that ω is order-equivalent to z, which we notate as $\omega \approx z$. If ω is at least as good as z, we write $\omega \geq z$, meaning 'better than or order-equivalent to' z.

We will say that b is a 'bound' or 'weak bound' for z if $b \geq z$. If $b > z$ we will say that b is a *strict* bound for z.[5] For b>z, we will also sometimes simply say 'b beats z'.

Exemplification. The character of these abstract conditions can be usefully examined in concrete examples. Let us focus first on a simple case involving a three member candidate set $\{a,b,z\}$ and a

[5] This is an extension of the standard usage, which defines bounds for sets rather than for single elements. In Appendix B we show that, for our purposes, the notion of '*strict* bound for a set of elements' should be defined like this: b is a strict bound for S if b is a weak bound for all elements of S and a strict bound for at least one of them. For a unit set {z}, this gives the same result as the above definition of 'strict bound for an element'.

two-member constraint set. The entire factorial typology is presented below. Candidate *z* is harmonically bounded:

(3) Harmonic Bounding. Candidates *a* and *b* are potential winners, *z* is a loser.

$C_1>>C_2$	C_1	C_2
☞ a		* *
b	*	
z	*	*

$C_2>>C_1$	C_2	C_1
a	* *	
☞ b		*
z	*	*

Scrutiny of the tableaux reveals that the loser *z* is harmonically bounded by candidate *b*, which beats *z* on C_2 (thereby satisfying 'strictness') and shares the same number of violations as *z* on C_1 (meeting 'weak bounding').

The effect can be seen much more clearly if we pull out the order structure that is implicit in the data tableau (3). Each constraint induces a partial order on the candidate set, represented diagrammatically here with the better candidates above the worse ones, and violation-equivalent candidates at the same level. To emphasize that we are focusing on the order properties of the constraint C_i, we refer to it as C_i^.

(4) Constraints as orders on the candidate set, from ex. (3)

C_1^	**C_2^**
a	b
|	|
b,z	a, z

It is obvious from the order diagrams that $b \geq z$ on both constraints (weak bounding) and $b > z$ on C_2 (strictness). If we represent the competition between *b* and *z* in a comparative tableau (Prince 1998), the bounding property is even more obvious:

(5)

	C_1	C_2
b ~ z		b

Here, the cell-entries index the victor in the comparison, if there is one. Candidate *z* can never win against *b*, for there is no cell referring to *z*. Generally, if in a comparative tableau for *x~y*, there are occurences of *x* but none of *y*, then *y* can never win on any ranking.

Simple harmonic bounding is not the end of the story, for there are losers which are thwarted by a combination of competitors. Consider (6) below, which imposes a minor variation on our example. Candidates *a* and *b* are potential winners, and *z* is a loser, but *z* is harmonically bounded by neither *a* nor *b*.

(6) Collective Bounding

C_1>>C_2	C_1	C_2
☞ a		* *
b	**	
z	*	*

C_2>>C_1	C_2	C_1
a	* *	
☞ b		**
z	*	*

Once again the situation is considerable clarified when we pull out the order structure that follows from the patterns of violation:

(7) Order Structure implicit in ex. (6).

C_1^	**C_2**^
a	b
|	|
z	z
|	|
b	a

Candidate *z* is never in the topmost stratum of any constraint. Hence, whatever ranking is chosen, *z* is beaten right away at the highest ranked constraint.

The comparative tableaux tell the same story, albeit somewhat more indirectly than before:

(8)

	C_1	C_2
z ~ a	a	z
z ~ b	z	b

Notice that there is no *column* in which *z* is the only candidate that appears: candidate *z* is therefore not in the topmost order stratum of any constraint: no constraint *favors* z. This means that either C_1 or C_2, when in first position in the hierarchy, will eliminate candidate *z*. Notice that C_1 favors *a*, as can seen by comparing *a* with its competitors—

(9)

	C_1	C_2
a ~ b	a	b
a ~ z	a	z

Similarly, C_2 favors *b*. These affinities are perhaps even more clearly reflected in the order diagram (7), where *a* and *b* occupy the top strata of C_1 and C_2 , respectively.

From this example, it follows that the set of *winners* in K cannot be identified simply by removing from K every harmonically-bounded candidate. Losers like *z* in the case just examined are unaffected by this operation. Identifying losers takes more than identifying a single bounding candidate for each.

To extend the notion of harmonic bounding, we introduce the notion of *bounding set* for a candidate. Membership in the bounding set is determined by two conditions that parallel those for simple harmonic bounding.

(10) **Def. Bounding Set.** A set $B\subseteq K$ is a bounding set B(z) for $z\in K$ relative to a constraint set Σ, iff B has these properties:

- **Strictness**. Every member of B is better than z on at least one constraint in Σ.
- **Reciprocity**. If *z* is better than some member of B on a certain constraint $C\in\Sigma$, then some other member of B beats *z* on the constraint C.

Strictness is the same as for simple harmonic bounding. *Reciprocity* generalizes the weak bounding property; it ensures that members of the bounding set protect each other from being eliminated by the bounded candidate.[6]

When a bounding set contains a single element, we derive simple harmonic bounding. By Reciprocity, that one bound can never be bettered by the loser on any constraint, for there is no other member of B to save it. It follows that the bound must be at least as good as *z* on all constraints, which is exactly what Weak Bounding requires. For classical harmonic bounding, then, Reciprocity is satisfied, vacuously, by the nonoccurence of constraints on which the bound is beaten by the bounded.

Under collective harmonic bounding both properties hold non-vacuously, as illustrated by the example discussed above, here repeated for convenience:

[6] The term 'reciprocity' is intended to invoke the defining property of the social compact: 'you watch my back, I'll watch yours'. The relation of Reciprocity to weak bounding is examined in Appendix B.

If universally fixed ranking order is imposed on some constraints in Σ, Reciprocity must take account of it. Let $\mathbb{C}\subseteq\Sigma$ be a set of constraints among which ranking is fixed. Reciprocity comes out like this: for any $C_i\in\mathbb{C}$ such that *z* is better than *b* on C_i for some $b\in B(z)$, there is a constraint $C_j\in\mathbb{C}$ ranked no lower than C_i and an element $a\in B(z)$ such that *a* is better than *z* on C_j. We will not be further examining such refinements in the present paper, and ranking should be assumed to be free in the constraint sets under discussion.

(11) Order Structure of ex. (7). B(z) = {a,b}.

C_1^	**C_2**^
a	b
|	|
z	z
|	|
b	a

Candidate z has the bounding set B={*a*,*b*}. Strictness is met: *a* and *b* beat *z* on C_1 and C_2, respectively. Reciprocity is also satisfied: for example, *z* beats *b* on C_1 but is beaten in turn by *a*. Similarly, *z* beats *a* in C_2 but is beaten by *b*.

Although Strictness is obviously needed for bounding, it is perhaps not immediately clear that Reciprocity cannot be loosened or simplified to look more obviously like Weak Bounding. Suppose we simply tried to replace Reciprocity with a straightforward kind of weak bounding property:

(12) **Misleadingly Universalized Weak Bounding**.

For every $\beta\in B$, β is *at least as good as* z on every constraint ($\beta\geq z$).

This is too strong and fails immediately, since it is not even true of the bounding set {*a*,*b*} in ex. (11). A second, more plausible attempt might try to loosen the condition on the 'defending' member of the reciprocating dyad:

(13) **Pseudo-Reciprocity**.

If z beats $\beta\in B$ on a constraint C ($z>\beta$), there is a $\gamma\in B$ such that γ is *at least as good as* z ($\gamma\geq z$) on C.

But the following example shows that there are cases where a candidate has a pseudo-bounding set that meets Strictness and Pseudo-Reciprocity, yet is still a winner.

(14) Counterexample to Pseudo-Reciprocity

C_1^	**C_2**^	**C_3**^
a,b	a,c	b,c
|	|	|
c	b	a

According to the actual definition (10), no bounding set B(*a*) can be formed for *a* in the above system: candidate *c* is not eligible, because on C_1, *c* is strictly worse than *a*, but *a* is itself not strictly worse than some reciprocating partner for *c*, as required by Reciprocity. Similarly, candidate *b* is not eligible, because this pattern is repeated on C_2 in the relation between *a* and *b*. From this it follows that *a* is a potential winner, and in fact *a* wins on the two rankings with C_3 at the bottom.

Under Pseudo-Reciprocity, however, *a* would be putatively ‘bounded’ by the set B={*b*,*c*}. On constraints C_1 , *b* pseudo-reciprocates for the subordination of *c* and on C_2, *c* pseudo-reciprocates for the subordination of *b*.

The constraints in this example are highly symmetric, and the same arguments can be repeated with respect to attempts to find bounding sets for *b* and for *c*. In each case, Reciprocity informs us correctly that no nonnull bounding set exists; but Pseudo-Reciprocity delivers up pseudo-bounding sets for each: thus, each two-candidate set pseudo-bounds the remaining candidate. Such a high degree of symmetry is not required for demonstrating the incorrectness of Pseudo-Reciprocity; but in this case, Pseudo-Reciprocity leads to the charming result that every candidate is pseudo-bounded, so that there are predicted to be no winners at all!

A candidate *z* may have more than one bounding set, as can be shown by a constraint hierarchy with a single constraint C in it:

(15) Multiplicity of bounding sets

```
**C**^
a
|
b
|
z
```

The set B(z)={*a*,*b*} satisfies Strictness, because each member beats *z* on C. Reciprocity is satisfied vacuously. But the sets B′={a} and B′′={b} also satisfy Strictness and Reciprocity vacuously, and therefore both qualify as bounding sets for *z* as well.

Unlike B, though, the sets B′ and B′′ are *minimal*: no non-empty strict subset of theirs is itself a bounding set. Even minimality, then, does not guarantee uniqueness. The set B′′ also shows that the elements of a bounding set need not be winners: for *b* is a loser in (15) above with respect to Σ={C}, and yet it constitutes a minimal bounding set for *z*.

Note finally that the null set always qualifies as a bounding set, because it vacuously satisfies Strictness and Reciprocity. For example, in (15) above, the null set is the only available bounding set for the winner *a*. (This is, of course, nothing more than a definitional nicety, and a clause demanding non-nullity could be inserted in the requirements, if desired.)

The bounding set provides us with a powerful tool for identifying losers and winners in a ranking-independent fashion. In §5 below, we will demonstrate that every loser has a non-null bounding set.

(16) **Bounding Theorem**. For any constraint set Σ and candidate set K, a candidate *z* in K is suboptimal on every ranking R over Σ iff there is in K a non-empty bounding set B(z) for *z*.

$$z \notin W(K,\Sigma) \Leftrightarrow B(z) \neq \emptyset$$

The contrapositive version of the theorem provides a condition that identifies winners: a candidate *ω* is a winner if and only if its sole possible bounding set is null.

A further issue remains: suppose we have a set of candidates which has the property that for every ranking, some member of that set is better than a candidate *z*. Call this a 'covering set' for *z*, COV(*z*). The existence of COV(*z*) guarantees that *z* is a loser: it provides a witness to *z*'s failure on each ranking. Is this then a fundamentally different way of providing a set of elements that block *z*'s hopes for optimality? We will show in §5.2 that it is not: every covering set for *z* must contain within it a bounding set for *z*.

(17) **Covering Theorem**. Let COV(z)⊆K be such that for every ranking R of Σ, there is an element c∈COV(*z*), where c is *strictly better* than z on R. Then there is a non-empty set B(*z*)⊆COV(*z*), where B(*z*) is a bounding set for *z*.

As an illustration of the theorem, consider the example below. Here *a* wins under the ranking $C_1 >> C_2$, and *b* wins for $C_2 >> C_1$; candidate *z* loses on either ranking. The covering set COV(z)={a,b} provides a witness to *z*'s defeat under each ranking, yet it is not a bounding set, since Reciprocity fails on constraint C_2 , where *z*>*a* but has itself no strict bound. However, the covering set properly includes the bounding set B={b}, which satisfies both Strictness and Reciprocity.

```
(18)          C1^          C2^
              a            b, z
              |            |
              b            a
              |
              z
```

An interesting corollary of the Covering Theorem concerns the set of potential winners, which by definition constitutes a covering set relative to each loser, and therefore must include a bounding set for each of them. But the winner-set need not be itself a bounding set! Consider the example just shown in (18): the set {*a*,*b*} collects all the winners, but does not qualify as a bounding set for *z*.

2.2 A Bound on Bounding

How big must a bounding set be? If every constraint in Σ is strictly binary — divides the candidate set into just two classes— then any loser must be harmonically bounded in the classical sense, by a bounding set with *just one member*.

To see this, consider any bounding set B for an arbitrary loser *z*, and select from it a single element, call it *b*: we can show that {*b*} must also constitute a bounding set for *z*. Because *b*∈B, it must meet Strictness and beat *z* on some constraint; so Strictness is also satisfied for {*b*}. By contrast, the loser *z* cannot beat *b* on any constraint; for if it did, by Reciprocity there would have to be a γ∈B, with γ>z, and on that constraint we'd have the candidate order γ>*z*>*b*. This is impossible, because all constraints are assumed to be strictly binary. Therefore we have *b*≥*z* on all constraints: this is Weak Bounding, equivalent to Reciprocity on unit bounding sets. The set {*b*} meets both Strictness and Reciprocity, and is therefore a bounding set for *z*, as desired.

This result is interesting, but is of rather limited utility, since very few constraints are binary in the required sense. Observe that most “binary” constraints refer to some aspect of structure — for example, ONSET is binary *within syllables* — but candidates are composite and may contain multiple occurrences of the relevant structural configuration. Over a realistic candidate set, a typical constraint that is binary on structural units will be n-ary, even infinitary, in the order distinctions it imposes among candidates.

When we move even to ternary constraints, we can construct examples which need a bounding set that has the cardinality of the entire constraint set.

(19) Big Bounding Set

$\mathbf{C_1}$^	$\mathbf{C_2}$^	$\mathbf{C_3}$^	....	$\mathbf{C_n}$^
a_1	a_2	a_3	...	a_n
$\vert$	$\vert$	$\vert$	...	$\vert$
z	z	z		z
$\vert$	$\vert$	$\vert$	...	$\vert$
a_2,...a_n	a_1,...,a_n	a_1,..,a_n	...	a_1,...,a_{n-1}

Clearly, B(z)={a_1,...,a_n}, and no subset of B is a bounding set for z. (If any a_i is removed from B, then Reciprocity fails for the other members on C_i .)

But this is the limit: As we will demonstrate in §3.3, no more than *n* collective bounds are needed for a constraint set of *n* constraints.

3. The Favoring Hierarchy and the Residual Bounding Set

With the ranking-independent Strictness and Reciprocity conditions in hand, it is easy to check whether a collection of candidates qualifies as a bounding set, and this in turn permits an easy proof of loser status through the Bounding Theorem (16). But how are we to *find* the bounding set in the first place? It will hardly be efficient, in general, to sort through every 1-element subset of the candidate set, every 2-element set, up to the limits imposed by the candidate set or the number of constraints, checking each possibility for adherence to Strictness and Reciprocity.

The more plausible line of attack is indirect and follows from the character of the Reciprocity condition. Consider a constraint which favors a suspected loser. (Recall that a constraint is said to ‘favor’ a candidate if no other competitor does better.) To satisfy Reciprocity (vacuously), all elements of its bounding set must do equally well on that constraint and must also be favored by the constraint. Candidates that do less well, those not favored, can be eliminated from consideration as possible members of the bounding set.

Here’s an example, in which constraint C_1 shows the desired structure:

(20) Loser *z* favored on a constraint.

$$\begin{array}{ll} \mathbf{C_1}\wedge & \mathbf{C_2}\wedge \\ \mathrm{a,z} & \mathrm{a} \\ | & | \\ \mathrm{b} & \mathrm{b,z} \end{array}$$

A quick computation shows that *z* loses on all rankings, and must therefore have a non-null bounding set. Focusing on C_1, we see that by Reciprocity, candidate *b cannot* be in the bounding set for *z*: it is strictly bounded by *z* on C_1 and yet *z* itself is not strictly bounded there. Constraint C_1 favors *z*, and any potential members of *z*'s bounding set must also be similarly favored.

This observation leads to a powerful method for constructing a bounding set by elimination, outlined above in §1. For convenience, we repeat a concise characterization.

Gather all those constraints that *favor* the targeted candidate: by Reciprocity, its bounding set must be drawn from among those candidates that are likewise uniformly favored by these constraints. The very same reasoning can now be re-applied to the set of potential bounds thus identified, in the context of the remaining constraints. Pick out from among these constraints those which favor the targeted candidate within the set of potential bounds; collect those candidates that are also favored, reject those that are disfavored: as before, the bounding set must be constructed from the favored ones. Again the reasoning can reapply, recursively, until we reach one of two possible outcomes: either all constraints in the original constraint set have been accounted for, in which case the candidate under consideration has a null bounding set; or we reach a point where none of the residual constraints are favoring. This residue forms a 'disfavoring system' for the candidate, in which case we have a loser on our hands. At the same time, we have also computed the maximal bounding set for the targeted candidate: this consists of what remains of the recursively shrunken set of potential bounds, minus those which are not better than the target on some residual, nonfavoring constraint. (Including these would lead to a failure of Strictness.)

This procedure, which arises from bounding theory considerations, is equivalent to Recursive Constraint Demotion (RCD: Tesar 1995 *et seq.*), as we will show below. RCD was developed in the context of a specific learning problem: how to efficiently adduce a constraint ranking, given a constraint set and a collection of desired input-output pairings (optimal candidates). RCD, like other related constraint demotion algorithms, overcomes a computational complexity inherent in the notion of 'ranking argument'. Because each ranking argument can yield a disjunction of possible rankings, a collection of ranking arguments typically amounts to an unwieldy conjunction of disjunctions. Instead of attempting to untangle the skein of logical possibilities, Constraint Demotion seeks to home in, more-or-less directly, on a special class of rankings that must exist if the desired set of optima can be obtained by the constraint set at hand. Unlike certain other Constraint Demotion algorithms, RCD detects failure quite conspicuously: the procedure will halt when no ranking exists that is consistent with the set of assumed input-output pairs. It is this property that provides the link between grammar learning and the problems under investigation here. The centrality of RCD within the conceptual structure of OT is affirmed by its intimate connection with bounding theory.

In this section, we flesh out our account of the recursive favoring construction. We aim to provide enough formal development to ensure clarity, though not enough to preclude comprehension. We reserve many details of proof until §5. We develop a purely order-theoretic characterization of

OT, working explicitly from the kind of order structure that emerges from the violation calculus in classical accounts of the theory (e.g. Prince & Smolensky 1993: ch. 5). Formalizing the intuitive idea that each constraint in the hierarchy ‘shrinks’ the candidate set, we treat constraints as functions from candidate sets to candidate sets. With these tools, we develop three principal results, each resting on the previous:

[1] The output of a set of favoring constraints, when faced with a candidate set, is the *intersection* of the sets of candidates favored by each constraint (the Favoring Intersection Lemma, §4.2).

[2] A candidate is a winner iff it has a recursive favoring hierarchy that exhausts the set of constraints (the Winner/Loser Theorem, §5.1).

[3] The construction of a recursive favoring hierarchy leads us directly to a unique maximal bounding set as well as to a bound on the available minimal bounding sets (the Maximal Bounding Set and the Minimal Bounding Set Theorems, §5).

3.1 Background Assumptions: Constraints and Optimality

Here we introduce the basic ideas that support the discussion that follows. We start by briefly reviewing notions of order theory and go on to define ‘constraint’, ‘constraint hierarchy’, ‘ranking’, and ‘optimality’.

Order on a set. An order is a relation on a set; we are interested in ‘strict’ orders: irreflexive ($x \not> x$), asymmetric ($x>y \Rightarrow y \not> x$), and transitive ($x>y$ & $y>z \Rightarrow x>z$). The general order relation is often call ‘partial’ because it need not be specified for every pair. Elements not ordered with respect to each other are said to be ‘noncomparable’; if every pair is comparable, an order is said to be ‘linear’ or ‘total’. A totally ordered (sub)set is a ‘chain’.

We will write ⟨S;O⟩ for a set S with an order relation O on it. This explicit notation will be useful because each *constraint* imposes its own order on the entire universe of candidates, and therefore on any subset of that universe.

We distinguish between the order imposed by the constraint and the constraint as a functioning member of a constraint hierarchy. As we have done above, we write C^ for the order associated with the constraint C, and we use the ordered-set notation ⟨K;C^⟩ to refer to the set K as ordered by C^.

When specific elements of K are compared in the order C^, it is tempting to follow standard usage and write $x>_{C^\wedge} y$, marking the sign ‘>’ with a miniscule reminder of the particular order that it denotes. To avoid imperspicuities, though, we will always write (x>y;C^) for ‘*x* is greater than (better than) *y* in the C-order’.

The notion ‘maximal element’ in an order is central to OT: a maximal element is an element than which no other element is greater. (When all the elements in the set are comparable, it is an upper bound for the entire set.) We use the notation *max*⟨P;O⟩ to denote the set of maximal elements in the partially-ordered set P. When applied to constraints, *max*⟨K;C^⟩ identifies those candidates of K that *do best* on C, those which C orders in its top stratum: those favored by C. When K is unambiguously determined by the context we will shorten *max*⟨K;C⟩ into C^T, where T stands for ‘top stratum’.

(21) **Def. Maximal element**. $\forall x \in P,\ x \in \max\langle P;O\rangle \Leftrightarrow \neg\exists y \in P\ (y>x;O)$

Constraints and Order. At the outermost level of generality, an order imposed by constraint is any form of partial order on candidates in which every subset in a candidate set has a maximal element. This ensures that each possible set of candidates has a maximal element under every constraint, and allows us to define what it means to 'do best' on a constraint.[7]

Violation theory yields a restricted class of orders of this type. For any two candidates *x* and *y* in some candidate set, we have (x>y;C^) — 'x is better than y on C' — iff *x* violates C less than *y*. Two candidates violating a constraint C *to the same degree* will not be comparable with each other in the order C^, in the sense that neither is better (>) than the other. Violation theory entails, however, that they are *both* worse than any candidate violating C less than they do and *both* better than any other candidate violating C more. The order C^ thus constitutes a *stratified hierarchy* $S_1>S_2>...>S_n$, to use the terminology of Tesar 1995. The candidates in each stratum S_i are unordered with respect to each other, but ordered with respect to any candidate in a different stratum. The notion of a stratified hierarchy is perhaps the central notion of order in linguistics, reappearing in many different guises. The same concept arises, for example, in the theory of stress prominence — and when we deal with rankings produced by constraint demotion algorithms or the construction of a Favoring Hierarchy. The notion can be defined variously; here is one way of doing it:

(22) **Def. Stratified Hierarchy**. A stratified hierarchy is a partially ordered set $\langle P;>\rangle$ in which non-comparable elements share all order relations.

$$\forall a,b,x \in P,\ \neg(a>b \vee b>a) \Rightarrow (a>x \Leftrightarrow b>x)$$

Each set of non-comparable elements forms a stratum. For stratum-mates *a* and *b*, we will write $a \approx b$ for $\neg(a>b \vee b>a)$, because in the case of stratified hierarchies, '≈' is an equivalence relation, a fact of no little importance.[8]

Violations and order. To see how these notions play out in practice, consider the constraint C in (23) below. On the candidate set K={a, b, c, d}, C imposes the stratified hierarchy diagrammed on the right:

[7] To see the importance of this condition, consider the set of integers under ordinary '>'. There is no integer that 'does best' — is greatest — under this ordering.

[8] As noted above (§2.1, p.6), it will not do to say simply 'a=b', since '=' means 'is the same entity as'.

(23) Constraint violations and partial order imposed by C on K

	C violations
a	*
b	* * *
c	* * *
d	* * * * * *

C^: a > {b,c} > d

a
|
b, c
|
d

With respect to C, the following order relations emerge:

(i) *a* is better than *b*, *c*, *d* and constitutes C's top stratum

(ii) *b* and *c* are neither better nor worse than each other: *b*≈*c*. But they share all order relations: both are worse than *a* and better than *d*.

Constraint violations are, of course, relevant only comparatively. Consider how the constraint C′, with its different violation pattern, treats the same candidate set:

(24) Constraint violations and partial order imposed by C′ on K

	C′
a	
b	*
c	*
d	* *

C′^: a > {b,c} > d

a
|
b, c
|
d

The violations of C′ in (24) differ from those of C in (23), but the induced stratified order does not change at all: *a* is the most harmonic element and *d* the worst in both (23) and (24). The constraints C and C′ impose the same order on these candidates and are entirely equivalent over K. The order diagram represents this equivalence directly.[9]

Constraints as functions. The candidates favored by a constraint — the most harmonic ones, the upper bounds for the relevant candidate set, the maximal elements in the constraint's order — all lie in the top stratum of the order it imposes. The role of constraint in a hierarchy is to eliminate all other candidates from the candidate set it faces. A constraint can therefore be understood as a

[9] Diagram invariance is a good analyst's tool for detecting those cases where two differently defined constraints actually do the same work on a given candidate set..

function: given any set of candidates K, it returns the top stratum from K in the imposed order. We write this out below:

(25) **Def. Constraint.** A constraint C is a function $\mathcal{P}(U) \Rightarrow \mathcal{P}(U)$, from the power set of the universal set U of candidates into itself, such that for any set of candidates K⊆U, it returns the top stratum of K, max(K;C^) ⊆ K, consisting of all elements in K which are maximal relative to the order C^.

$$C: \mathcal{P}(U) \Rightarrow \mathcal{P}(U)$$
$$C(K) = \max(K;C\hat{})$$

Crucially, the value of C(K) varies with K. If C^ yields the order {a}>{b,c}>{d}, then we have:

C(K)={a} for K={a,b,c,d}, but
C(K)={b} for K={b,d}.

The latter follows because *b* beats *d*, (b>d;C^), and *a* and *c* are not part of the candidate set under consideration.

Ranking. A *ranking* on a set of constraints Σ is a total order R on that set, determining the order in which they are to be composed to form a hierarchy. A ranking is ⟨Σ;R⟩ in the ordered-set notation, but we will typically refer to a ranking simply as R or some subscripted variant, since the set of constraints under discussion will be clear. In a minor abuse of terminology, we will also refer via *ranking* to the constraint hierarchy that is so ranked.

We will use a concise sequential notation to describe rankings: for A>>B>>C, we will write [ABC]. It is convenient to work with variables ranging over continuous subsequences of a ranking. Thus, with H=[AB], we will write R=[**AB**C]=[**H**C]. Rankings may be null, i.e. contain no constraints, in which case we write R=Ø. (We identify the null ranking with the Identity function.) Throughout, we will use letters early in the alphabet for constraints, and letters such as G, H, J, R, X, Y as variables over rankings and subrankings

Constraint Hierarchy. A constraint hierarchy is simply the functional composition of constraints. A higher-ranked constraint lies *earlier* in the application order than a lower-ranked one. For a ranking R=[AB], we have R(K) = (B∘A)(K) = B(A(K)).[10] Intuitively, A gets first whack at initial candidate set K, and B selects from among the reduced set B(K). The following supplies a general characterization:

(26) **Def. Ranking Composition.** For any candidate set K⊆U, and any ranking R=[C_1..C_n] on a set of constraints Σ={C_1,...,C_n}, the function R(K) is constructed by composing the constraints in domination order, starting from the top:

$$R(K) = (C_n \circ C_{n-1} \circ \ldots \circ C_2 \circ C_1)(K) = C_n (C_{n-1} (.. (C_2 (C_1 (K))))).$$

[10] The reversal of left-right visual order between e.g. [ABC] and C∘B∘A is a nuisance, but the associated typography should always make it clear what we're dealing with: brackets will enclose constraints listed in domination order, and functional apparatus, typically parentheses, will mark the functional reading.

An equivalent definition can be formulated recursively:

(27) **Def. Ranking Composition (recursive).** For any candidate set $K \subseteq U$, and any ranking R of a constraint set Σ, the function R(K) is defined as follows:

If $R = \varnothing$, then $R(K)=K$
If $R=[CH]$, then $R(K) = (H \circ C)(K) = H(C(K))$.

Observe that when $R=[FG]$, we write $R(K) = [FG](K) = (G \circ F)(K) = G(F(K))$.

Optimality. With constraints as functions and constraint hierarchies as compositions of functions, *optimality* is easily defined: the set of optima is just R(K), the result of applying the composite function R to candidate set K.

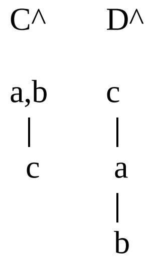
Def. **Optimal**. A candidate $\alpha \in K$ is *optimal* in some ranking R of a constraint set Σ iff $\alpha \in R(K)$.

Remark. A hierarchy is itself a *constraint*, as the term is defined in (25). A ranking R is a function from candidate sets to candidate sets; and, as we will show in §4.1 and in Appendix A, there is an associated order R^, induced by the orders native to the individual constraints of which it is composed (Prince & Smolensky's *harmonic ordering of forms*), and indeed, as expected, R(K)=max(K,R^). Thus a hierarchy is actually a favoring constraint for its optima.

Exemplification. To show how the functional conception works, let's examine the following two-constraint ranking R=[CD].

(28) Exemplary 2-constraint system

	C	D
☞ a		*
b		**!
c	*!	



The operations that identify the winners for R are the same whether we use the above definition or follow the equivalent violation-comparing procedure on the violation tableaux: first we collect the most harmonic candidates for C, i.e. C(K)= {*a,b*}. Then we evaluate D with respect to *this* set of candidates, and since *a* beats *b* in the order D^ , we find that *a* is optimal.

Crucially, the candidate set with respect to which D is evaluated — namely, C(K) — no longer contains *c*. Candidate *c* beats both *a* and *b* in the C^ order on K-in-its-entirety, but the evaluation is blind to this fact, because *c* is eliminated at the evaluation of C(K). This pattern of eliminations and restricted focus, familiar to all practitioners, is exactly what it means to have each constraint functionally applied to the value of the next-highest-ranked constraint. In our example, the relevant sequence of functional applications looks like this:

$$K = \{a,b,c\}$$
$$C(K) = \{a,b\}$$
$$D(C(K)) = D(\{a,b\}) = \{a\}$$

Unsurprisingly, constraint re-ranking *qua* change-in-order-of-composition can affect the set of winners. Consider the reversed ranking R'=[DC]:

$$[DC](K) = (C\circ D)(K) = C(D(K))$$
$$D(K)=\{c\}$$
$$C(D(K)) = C(\{c\}\}=\{c\}.$$

In general, the way distinct constraints C_1 and C_2 apply to some set K, of whatever provenance, need not be the same, and $C_1(K)$ and $C_2(K)$ could even be fully distinct sets, as in the example just cited. Rankings $R_1 = [...C_1C_2...]$ and $R_2 = [...C_2C_1...]$ will often, as is commonly observed in practice, serve up disjoint sets of optima.

3.2 Recursive Favoring and RCD

With the infrastructure in place, we now develop a way to identify the bounding set for a candidate: if the bounding set is null, we will have produced a (set of) rankings on which our candidate is optimal; but if the candidate is a loser, we will have found the maximal non-empty bounding set for it. We define a 'recursive favoring hierarchy' that progressively excludes candidates from the bounding set; we conclude the discussion by showing that this construction is an order-theoretic equivalent of RCD.

3.2.1 Favoring Hierarchies

The key notion is the *favoring constraint*, the OT analog of a constraint *satisfied* by a candidate in a non-OT Boolean theory. Here we restate the definition we have given above: a constraint favors those candidates that it orders into its top stratum, which are the maximal elements in the set of candidates the constraint is being applied to.

(29) **Def. Favoring Constraint.** For any candidate set K, a constraint F is a *favoring constraint* for α in K iff α is a maximal element of the ordered set $\langle K;F^\wedge\rangle$. Equivalently,

F is a *favoring constraint* for $\alpha\in K$ iff $\alpha\in F(K)$.

Let us write $\mathscr{F}(\alpha,K,\Sigma)$ for the set of *all* favoring constraints for α over a candidate set K and constraint set Σ. (We'll write $\mathscr{F}$ for short whenever K and Σ are clearly determined by context.)

The set $\mathscr{F}$ has the important property that α wins on *all* of its rankings. (The constraints in $\mathscr{F}$ do not conflict.) To see this, note that each $F_i\in\mathscr{F}$ returns a subset of K that includes α. No matter how these constraint-functions are composed, the favored candidate α will be returned by each functional application.

Any candidate not at the top of all constraints in $\mathscr{F}$ is inevitably a loser, because it is thrown out as soon as a constraint is evaluated on which it is not favored. It follows that the set of op tima for a favoring set $\mathscr{F}$ is just the *intersection* of the top strata of all the constraints in $\mathscr{F}$, collecting all candidates that share with α the top of each constraint in $\mathscr{F}$. Since this is a well-defined entity, invariant across the possible rankings $\mathscr{F}^{>>}$ of $\mathscr{F}$, we can notate it as $\mathscr{F}$(K). We will call its members the *co-winners* of α in $\mathscr{F}$(K).

Exemplification. The mini-system below illustrates the favoring property. In it, constraints C and D treat *ω* as maximal with respect to K={*a*,*b*,*c*,*ω*}. Both therefore qualify as favoring constraints for *ω*, forming together the favoring set $\mathscr{F}$={C, D}.

```
(30)      C^            D^
        a, b ω        b, c, ω
          |              |
          c              a
```

Candidate *ω* wins on any ranking of C, D with respect to K, because each ranking maps K into a subset of K that always includes *ω*. Similarly for candidate *c*.

The ranking R = [CD] eliminates *c* at C(K), and eliminates *a* on evaluation of D(C(K)), yielding the set of optima R(K)={*b*,*ω*}. The ranking [DC] eliminates *a* at D(K), *c* at C(D(K). Either way, we have the same outcome.

The intersection C(K)∩D(K)={*b*,*ω*} yields the optima for $\mathscr{F}$, with co-winners *a* and *b*.

These considerations lead to the following result:

(31) **Favoring Intersection Lemma.** Let $\mathscr{F}(\alpha,K,\Sigma) = \{F_1,..,F_n\}$ be a set of favoring constraints for α∈K. Then the set of candidates $\mathscr{F}$(K) that win on each and every ranking of $\mathscr{F}$ is given by the intersection of the sets max⟨K, F_i^⟩ for all *i*, that is, by $\cap_i F_i(K)$.

$$\mathscr{F}(K) = \cap_i F_i(K).$$

Pf. See (76), §4.1, p. 43.

Here's another way of putting the result: if the top order-strata in a set of constraints have a non-null intersection, then the optima for the constraint set are just the members of that intersection, and they all win over any and all rankings.

The favoring set has two desirable, complementary properties. First, the favored candidate will always lie among the optima on its favoring set. Second, any members of the bounding set for the favored candidate must also be found among the optima (by Reciprocity, as discussed above). In gathering together the constraints in the favoring set, we have not produced a ranking of the whole set that guarantees optimality, nor have discovered the contents of a bounding set; but we have taken a step toward both of those goals.

A further step can now be taken, by essentially repeating the first. Focusing on the set of co-winners from the first favoring stratum (a reduced version of the original candidate set) and the residual set of constraints that did not originally favor the targeted candidate (a reduced version of the original constraint set), we are faced with the same logic as before. Gather all the constraints from among this reduced constraint set that favor the targeted candidate; gather all the candidates from the reduced candidate set that survive evaluation by the new favoring set. This second-order favoring set will have the same key properties as the first: the targeted candidate will be among its co-winners, as will any members of its bounding set.

The process lends itself to recursive repetition. The outcome is *a favoring hierarchy*, in which each stratum consists of all the favoring constraints for the targeted candidate, determined with respect to the co-winners of the preceding stratum. The favoring hierarchy for a candidate α, $\mathcal{H}(\alpha) = \langle \mathscr{F}_1 ... \mathscr{F}_n \rangle$, where $\mathscr{F}_i$ is the ith favoring stratum, can be defined as follows:

(32) **Def. Favoring Hierarchy**. Let K be any set of candidates including α, and Σ any set of constraints. Let $\mathscr{F}(\alpha,K,\Sigma)$ be the set of favoring constraints F for α over Σ with respect to K. Then the *favoring hierarchy* $\mathcal{H}(\alpha)$ is a stratified hierarchy $\langle \mathscr{F}_1 .. \mathscr{F}_n \rangle$ where each favoring stratum $\mathscr{F}_i$ is a non-empty set of favoring constraints recursively defined as follows:

Base step:	Comments:
$K_1 = K$	
$\Sigma_1 = \Sigma$	
$\mathscr{F}_1 = \mathscr{F}(\alpha,K_1,\Sigma_1)$	1st fav. stratum = set of favoring constraints for α over K and Σ
Recursive step:	
$K_{i+1} = \mathscr{F}_i(K_i)$	Next candidate set = co-winners of current favoring stratum.
$\Sigma_{i+1} = \Sigma_i - \mathscr{F}_i$	Next constraint set = current set minus current favoring stratum
$\mathscr{F}_{i+1} = \mathscr{F}(\alpha,K_{i+1},\Sigma_{i+1})$	Next favoring stratum = favoring constraints for α over the new sets of candidates and constraints

Exemplification. To gain familiarity with the construction of favoring hierarchy via RCD as construed here, consider the constraint set Σ= {A,B,C,D,E} and candidate set K={*a,b,c,ω*}, with order properties shown below. Let us ask whether there exists a stratified hierarchy $\mathcal{H}(\omega)$ for *ω*.

(33)	**A^**	**B^**	**C^**	**D^**	**E^**
	a	b, c	a, c, ω	b, c, ω	a
	|	|	|	|	|
	ω	ω	b	a	c, ω
	|	|			|
	b, c	a			b

First, we find the favoring set for *ω* over the whole candidate set and the whole constraint set:

$$\mathscr{F}_1(\omega,K_1,\Sigma_1) = \{C, D\}$$

We then construct the next-generation candidate set from the winners for $\mathscr{F}_1$:

$$K_2=\mathscr{F}_1(K_1) = C(K_1)\cap D(K_1) = \{c,\omega\}$$

And we calculate the next-generation constraint set:

$$\Sigma_2 = \Sigma_1 - \mathscr{F}_1(\omega,K_1,\Sigma_1) = \{A,B,C,D,E\} - \{C,D\} = \{A,B,E\}$$

For the second-generation candidate set K_2 and for Σ_2 the second-generation constraint set, we are looking at the following:

(34)	**A^**	**B^**	**E^**
	ω	c	c, ω
	|	|	
	c	ω	

We can now construct the second stratum of the favoring hierarchy from the constraints that favor *ω* in this context:

$$\mathscr{F}_2 = \mathscr{F}(\omega,K_2,\Sigma_2) = \{A, E\}$$

Nothing but *ω* is left to be a member of the third generation candidate set:

$$K_3 = \mathscr{F}_2(K_2) = A(K_2) \cap E(K_2) = \{\omega\}$$

The third generation constraint set is simply B:

$$\Sigma_3 = \Sigma_2 - \mathscr{F}(\omega, K_2, \Sigma_2) = \{A,B,E\} - \{A,E\} = \{B\}$$

Since B over K_3 rather unsurprisingly favors K_3's only member, we have exhausted our original constraint set Σ and arrive at a complete, exhaustive favoring hierarchy for *ω*.

$$\mathcal{H}(\omega) = \langle\mathscr{F}_1\mathscr{F}_2\mathscr{F}_3\rangle = \langle\{C,D\}\{A,E\}\{B\}\rangle$$

Remark. The Favoring Intersection Lemma ensures that each favoring stratum $\mathscr{F}_i$ is itself a function from a set of candidates K to one of its possible subsets, namely that formed by all those members of K maximal in all constraints of $\mathscr{F}_i$. Therefore, we can extend to favoring strata all the properties associated with constraints, including that of functional composition. Besides indicating the co-winners for a single favoring stratum $\mathscr{F}$ over a set K as $\mathscr{F}(K)$, we can notate the co-winners for a ranked sequence of favoring strata $\langle\mathscr{F}_1...\mathscr{F}_n\rangle$ as $[\mathscr{F}_1\mathscr{F}_2\mathscr{F}_3](K)$, which will have the unique value $\mathscr{F}_3(\mathscr{F}_2(\mathscr{F}_1(K)))$ as per functional composition.

The example just examined can then be represented in the following three steps, each summarizing the evaluation of an additional favoring stratum for α. The last step also indicated the optimal candidates for the whole hierarchy.

(i) $[\mathscr{F}_1](K) = \mathscr{F}_1(K) = \{c, \omega\}$
(ii) $[\mathscr{F}_1\mathscr{F}_2](K) = \mathscr{F}_2(\mathscr{F}_1(K)) = \{\omega\}$
(iii) $[\mathscr{F}_1\mathscr{F}_2\mathscr{F}_3](K) = \mathscr{F}_3(\mathscr{F}_2(\mathscr{F}_1(K))) = \{\omega\}$

Remark. The construction of the favoring hierarchy may be fruitfully compared with the way winners are determined in a strictly Boolean constraint-satisfaction theory, one based on inviolability. There, a candidate is 'well-formed' iff it satisfies all constraints. From the order-theoretic point of view, each Boolean constraint divides the candidate set into two strata, top and bottom, satisfiers and violators. The well-formed candidates are arrived at by intersecting the tops of all constraints. If the intersection is empty, there is no output. But in OT, candidates that are not universally favored can get a reprieve: as long as there are *some* constraints whose intersected tops contain the candidate in question, we are allowed to proceed onward: we continue the evaluation over the rest of the constraints, restricting ourselves to the winners of the previous round.

The favoring hierarchy for a candidate decides its winner/loser status. For a winner, the favoring hierarchy contains all the constraints in the set. Any total ranking of those constraints that respects the stratum-order will be one in which the winner is optimal. For a loser, the favoring hierarchy will come up short, leaving out some constraints: the leftover constraints are those antagonists of the loser that cannot be resolved by ranking. These form a nonempty disfavoring system for the loser. The following theorem encodes this result:

(35) **Winner/Loser Theorem.** For a set of candidates K, a constraint set Σ, and candidate $\alpha \in K$, α is a winner over Σ iff there is a favoring hierarchy for α, $\mathcal{H}(\alpha) = \langle \mathscr{F}_1 .. \mathscr{F}_n \rangle$, exhausting Σ.

$$\alpha \in W(K, \Sigma) \Leftrightarrow \forall C \in \Sigma, C \in \mathscr{F}_i \text{ for some } \mathscr{F}_i \in \mathcal{H}(\alpha).$$

The theorem speaks of *winners*, but biconditionally, so that it provides by contraposition a necessary and sufficient condition for loser status as well. As promised, a candidate is a *loser* iff there is *no* exhaustive mapping of the constraint set into a favoring hierarchy for it. Notice that the meat of the theorem lies in the left-to-right implication. Going the other way, right-to-left, it's clear from the way an exhaustive favoring hierarchy is constructed that if such a thing exists, it will immediately give rise to a total ranking that makes the favored candidate optimal. More surprising, perhaps, a candidate that wins under any ranking whatever, an arbitrary winner, is guaranteed to have an exhaustive favoring hierarchy, a very specialized ranking (family of rankings) on which it wins.

3.2.2 Comparison with Classic RCD

The Winner/Loser Theorem is the analog of the correctness result for RCD (Tesar 1995, Tesar & Smolensky 1996ab, 1998, in press). Let us now pause to establish the relationship between the RCD work and the approach developed here; we resume our discussion in the next section.

RCD operates in two stages: a pre-processing stage, ‘mark cancellation’, prepares the raw violation-data for analysis; then a ‘recursive ranking’ stage produces the *target stratified hierarchy*, which we will see to be the same as our favoring hierarchy.

Mark Cancellation and Order Theory. RCD takes as input a set of statements of the form ‘a certain desired optimum must beat a certain suboptimal competitor’, $\omega_i \succ c_j$ for short. With this, we are given the constraint-violation profiles of the candidates. Each statement $\omega_i \succ c_j$, with its associated constraint-violation data, constitutes a elementary ranking argument, and is known as ‘mark-data pair’. Aggregated, these form the ‘mark-data pair list’. The goal is to find a stratified hierarchy that is consistent with the entire list, resolving all ranking arguments.

Each mark-data pair identifies the ‘loser-marks’, the list of constraint violations incurred by the would-be suboptimum in the comparison, as well as the ‘winner-marks’, the constraint violations of the desired optimum. (Here ‘marks’ = constraints violated, with multiplicity of violation recorded.) Since evaluation is strictly comparative, the absolute quantity of violations is only indirectly informative. All that matters is which candidate has more violations of a given constraint. To determine this, shared marks are eliminated from each competing pair. (This is the cancellation part of the Cancellation-Domination Lemma, Prince & Smolensky 1993:148.) In the post-cancellation data structure, the ‘loser-marks’ slot contains those constraints that *prefer the desired optimum*, those on which $\omega_i > c_i$. (These are known as the ‘uncanceled loser-marks’, i.e. those that survive cancelation.) Correspondingly, the ‘(uncanceled) winner-marks’ are precisely those constraints that perversely prefer the desired *sub*optimum, those on which $c_i > \omega_i$. (The winner-marks are those constraints that threaten the optimality of the desired optimum; they must be subordinated in the ranking.)

On the face of it, the post-cancellation mark-data pair list may not appear to closely resemble anything we have been talking about. But it encodes statements of the form ‘ω beats z on constraint C’, precisely the kind of *order*-information we have been working with: ($\omega > z;$C^). A mark-data pair based on a relation $\omega \succ z$ ‘desired optimum ω is better than z’ would have C listed in its ‘uncanceled loser-mark’ field. Conversely, if the desired outcome were instead $z \succ \omega$, with z as the desired optimum, this ω-preferring C would be listed in the ‘uncanceled winner-mark’ field, identified as a constraint that must be subordinated.

To use RCD to find out whether a candidate is optimal over a certain candidate set, we would examine a set of ranking arguments of the form $\omega \succ a_i$ for a fixed ω. If we collect all the forms involved, we arrive at the set $\{\omega, a_1,...a_n\}$, which is just the kind of candidate set we have focused on throughout our discussion. We also have considerable information on the order structure imposed by each constraint, determined from analysis of the raw violation data, encoded in the mark-data pair list. After mark cancellation, the mark-data pair list will classify, for each constraint, the order properties of all members of the candidate set with respect to ω, providing information as to whether they are better, worse, or at the same level as ω. This is a coarsened version of the order information that the constraint-order provides, but it is all we need to construct ω’s favoring hierarchy. And if we go through the candidate set testing various members for loserhood, we will eventually ferret out the entire order-structure imposed by each constraint.

Recursive Ranking and the Favoring Hierarchy. The first step of the RCD recursive ranking subprocedure gathers all constraints from the post-cancellation mark-data pair list which *assess no winner-marks*. These are precisely the constraints that do not prefer any threatening competitor to its correlated desired optimum: they either prefer the desired optimum, or they don't decide between the two. In the case at hand, where we're comparing ω to everything else in its candidate set, these are the constraints upon which, for all competitors c, either $\omega > c$ or $\omega \approx c$. Clearly, ω sits at the top of all such constraints: these are just the favoring constraints for ω. These form the first and highest stratum of constraints, in RCD as for us. Gathering these constraints and placing them in the first stratum of the hierarchy is step IIa of RCD (Tesar 1995: §4.2.2) and corresponds exactly to our statement in (32):

$$\mathscr{F}_1 = \mathscr{F}(\alpha, K_1, \Sigma_1)$$

The next step of RCD is to remove the constraints just gathered from the collection of constraints being ranked (step IIb). This corresponds to our definition of the next-generation constraint set:

$$\Sigma_{i+1} = \Sigma_i - \mathscr{F}(\alpha, K_i, \Sigma_i)$$

RCD (step IIc) goes on to remove from consideration all mark-data pairs which contain any marks assessed by the constraints just gathered. Since these constraints do not assess marks against (i.e. do not disprefer) ω, this move is targeted at selected ranking arguments $\omega \succ c_i$, where the competitor c_i is *not* at the top of the constraint. Left behind by this removal are those pairs $\omega \succ c_k$ where c_k is at the top of all the favoring constraints. In short, this step shrinks the relevant candidate set, just as in the definition of the recursive favoring hierarchy, to those candidates that are *favored* by the constraint-stratum just constructed. It corresponds to our definition of the next-generation set of candidates:

$$K_{i+1} = \mathscr{F}_i(K_i)$$

Finally (IId), RCD calls itself recursively, setting to working on *the remaining mark-data pairs*—that is, on the remaining candidates K_{i+1} and on the remaining constraints Σ_{i+1}. From these, it defines the next stratum as the constraints that assess no (uncanceled) marks against ω, i.e. those that favor ω in the context of the reduced candidate set. In our terms,

$$\mathscr{F}_{i+1} = \mathscr{F}(\alpha, K_{i+1}, \Sigma_{i+1}).$$

3.3 Favoring Hierarchies and the Maximal Bounding Set

The favoring hierarchy is the main tool for identifying bounding sets. The initial candidate set K contains all potential members of the bounding set for a candidate z. Each successive favoring stratum is guaranteed to eliminate from this set only non-bounds. This procedure of elimination turns out to be remarkably successful in leading directly to a bounding set of particular interest. When the construction of the favoring hierarchy is concluded, the set of remaining candidates, purged of those that are never better than z on some remaining constraint, constitutes the one and only *maximal bounding set* available for z in K. (If z alone remains, and all constraints are accounted for, then the bounding set is null.)

This result in turn permits us to establish an upper bound on the size of α's *minimal* bounding set(s): a bounding set never has to be larger than the number of remaining non-favoring constraints, those that must be left out of the favoring hierarchy.

These findings confirm once again the conceptual centrality of the favoring hierarchy and RCD within the theory of optimality.

Eliminating Reciprocity Violators. Let us begin by examining in detail how non-bounds are eliminated with each new round of recursive favoring.

Any candidate not at the top of all favoring constraints is guaranteed to lead to Reciprocity failure if included in a putative bounding set. For example, given the infinite candidate set K={α,*u*,*a*,*b*,...} and the favoring constraints for α shown below, no bounding set for α may have *u* as one of its members, because *u* has no reciprocating partner on constraint A relative to K, and therefore also relative to any of its subsets. Shrinking the candidate set to the intersection of the top-strata of the favoring constraints eliminates non-bounds like *u* and retains all existing potential bounds.

(36)

$$\begin{array}{ll} \textbf{A}^\wedge & \textbf{B}^\wedge \\ \alpha, a, b... & \alpha, u, a, b,... \\ | & \\ u & \end{array}$$

Besides leading to Reciprocity failure, candidates like *u* above cannot be used to help other candidates satisfy Reciprocity on other constraints. As the name reminds us, Reciprocity only holds between the *members* of a bounding set: since *u* is excluded from all bounding sets, it cannot constitute the reciprocating partner of any other bound. It follows that candidates like *u* are non-bounds independently of what favoring stratum they are in.

Suppose, for example, that A and B in (36) above constituted the n^{th} favoring stratum rather than the initial one: *u* would fail Reciprocity on A all the same, because no member of the current set of candidates strictly bounds α in the favoring stratum $\mathscr{F}_n$={A, B}. Crucially, any previously eliminated non-bound γ could not enter in a Reciprocity relation with *u* and rescue it, even if γ strictly bounded α in A, because γ itself is excluded from all bounding sets due to its own failure of Reciprocity.

Residual Constraints and Residual Candidates. The favoring hierarchy for a candidate α thus ensures that each successively-identified favoring stratum eliminates a set of non-bounds which need but fail to have reciprocating partners. When α is a loser, the favoring hierarchy may identify one or more favoring strata $\mathscr{F}_i$, but eventually hits a set of *residual constraints* none of which favors α, so that no additional favoring stratum can be formed. Likewise, the candidates surviving each favoring stratum — the co-winners of each favoring stratum $\mathscr{F}_i$— constitute the set of *residual candidates*, from which the maximal bounding set for α will be drawn. Both notions are defined below:

(37) **Def. Set of Residual Constraints.** For any constraint set Σ, and favoring hierarchy $\mathcal{H}(\alpha)$ over Σ, the set of residual constraints Res(Σ) is formed by all and only the constraints in Σ but not in $\mathcal{H}(\alpha)$:

$$\text{Res}(\Sigma) = \Sigma - \{C: C\in\mathcal{H}(\alpha)\}$$

(38) **Def. Set of Residual Candidates.** For any constraint set Σ, candidate set K, and favoring hierarchy $\mathcal{H}(\alpha)=\langle \mathscr{F}_1 \ldots \mathscr{F}_n \rangle$ for α in K, the set of residual candidates Res(K) is formed by all and only the candidates in K that co-win with α on each favoring stratum $\mathscr{F}_i$:

$$\mathrm{Res}(K)=[\mathscr{F}_1 .. \mathscr{F}_n](K)$$

Viewed from α's perspective, each residual constraint has the shape shown in (39) below, with at least some residual candidate β necessarily ordered above α, and, possibly but not necessarily, some other residual candidates γ and δ ordered with and below α.

(39) **C^**
β
|
α, γ
|
δ

The shape of the residual constraints guarantees that the entire set of residual candidates Res(K) always satisfies Reciprocity in Σ. Any member strictly worse than α on a residual constraint, such as δ above, is always rescued by a reciprocating member strictly better than α, i.e. by the necessarily present β. On all other constraints —those favoring α— all residual candidates share the same ordered-stratum of α, hence they are never strictly worse than α, thus satisfying Reciprocity vacuously.

Strictness Violators and the Maximal Bounding Set. We may now build *the maximal bounding set for α*, or $B^{Max}(\alpha)$, by simply collecting from the set of residual candidates all those that are strictly better than α on some residual constraint, i.e. the candidates that with respect to the schema in (39) above are in the β position on at least one of the residual constraints. $B^{Max}(\alpha)$ is guaranteed to satisfy Strictness by construction, since all the elements that are not strictly better than α somewhere — those elements that take either the γ or the δ position across *all* residual constraints— are never collected.

$B^{Max}(\alpha)$ also satisfies Reciprocity. This property already holds of the residual candidate set Res(K), of which $B^{Max}(\alpha)$ is a subset. Those elements in Res(K) that are left out of $B^{Max}(\alpha)$ could not in fact serve as reciprocating partners for any member of $B^{Max}(\alpha)$, because this would require them to be strictly better than α on some constraint, which by construction they are not. Therefore, Reciprocity remains satisfied even after their elimination, and holds of $B^{Max}(\alpha)$, the set which is precisely the result of eliminating them. It follows that $B^{Max}(\alpha)$ is a bounding set for α over Σ. Moreover, the way $B^{Max}(\alpha)$ is built guarantees its maximality and its uniqueness. Any eliminated candidate fails Strictness and could not be a member of any possible bounding set.

This result is encoded in the following definition and theorem, which is followed by an example illustrating the above discussion.

(40) **Def. Maximal Bounding Set.** For any constraint set Σ, candidate set K, and α in K, let $B^{Max}(\alpha)$ be formed by all and only those candidates in the set of residual candidates Res(K) that are strictly better than α on some constraint in the set of residual constraints Res(Σ):

$$B^{Max}(\alpha) = \{x : x\in Res(K) \text{ and } \exists C\in Res(\Sigma)\ (x>\alpha;C^\wedge)\}$$

(41) **Maximal Bounding Set Theorem.** For any constraint set Σ, candidate set K, and α in K, $B^{Max}(\alpha)$ constitutes the unique maximal bounding set for α relative to Σ and K.

Exemplification. Consider the system diagrammed below, with Σ={A,B,C} and K={*a,b,c,d,z*}. Let us seek the maximal bounding set for *z*. The first favoring stratum $\mathscr{F}_1$ for *z* includes only A. The co-winners for $\mathscr{F}_1$ are thus {*a,b,c z*}, with candidate *d* eliminated, because it would provoke Reciprocity failure on A, where no candidate is strictly better than *z*.

```
(42)   A^            B^           C^                $\mathscr{F}_1$ = {A}
       a,b,c,z       d             b                $\mathscr{F}_1$(K) = {a,b,c,z}
        |             |            |
       d             a             z, c, d
                      |            |
                     z            a
                      |
                    b, c
```

When we consider the remaining constraints relative to the residual candidates, we find the situation shown here below, with no constraint favoring *z*. The set of residual constraints set consists of B and C, and the set of residual candidates is equal to $\mathscr{F}_1$(K)**,** i.e. Res(K)={*a,b,c,z*}.

```
(43)    B^           C^              Res(Σ) = {B, C}
        a            b               Res(K) = {a,b,c,z}
        |            |               Maximal bounding set B^Max(z) = {a,b}
        z           z, c
        |            |
       b, c          a
```

As discussed above, the set of residual candidates satisfies Reciprocity relative to the original Σ: here *b* and *c* are strictly worse than *z* in B, but they are rescued there by *a*, which is also in the set. Symmetrically, *a* is strictly worse than *z* in C, where it is rescued by *b*, also in the set. Note that all members of the set {*a,b,c*} are order-equivalent to *z* in A, thus satisfying Reciprocity with respect to this constraint as well (vacuously). The same set, however, does not satisfy Strictness: candidate *c* is never strictly better than *z* on any of the constraints in Σ. Nor, of course, is *z* ever strictly better than itself. It follows that the set of residual candidates can never qualify as a bounding set.

Let us now construct B^{Max}(z) by selecting from Res(Σ) all and only the elements that are strictly better than *z* somewhere in Res(Σ), i.e. *a* and *b*. This set satisfies both Strictness and Reciprocity. Strictness is satisfied by *a* in B and by *b* in C. The challenge of Reciprocity is met by *a* rescuing the *z*-bettered *b* in B, and *b* rescuing the *z*-bettered *a* in C. A quick check shows also that set {*a*,*b*} constitutes the unique largest available bounding set for *z* relative to the original constraint set Σ.

Bounding sets for Winners. All above definitions and theorems apply to any favoring hierarchy $\mathcal{H}(\alpha)$, independent of whether α is a loser or a winner. When α is a winner, the favoring hierarchy eventually maps all constraints into favoring strata, leaving a null residue Res(Σ) = Ø . The set of residual candidates Res(K) will contain all and only those elements which appear with α at the top of all favoring constraints throughout the construction, and since in this case these exhaust the set of available constraints, the members of Res(K) are order-equivalent to α across the board. (They are therefore grammatically indistinguishable from α as far as Σ goes.) None of them is strictly better than α on any constraint, and they therefore are eliminated from B^{Max} as Strictness violators, which leaves us with a null maximal bounding set. The emptiness of the *maximal* bounding set, by a brief bout of set-theoretic reasoning, entails the emptiness of any of its subsets, confirming that winners associate with empty bounding sets (Bounding Theorem (16) §2.1).

Upper Bounds for Blocking Sets. The unique maximal bounding set sets an upper limit on the size of any other available bounding set B(α), including any minimal one. A more precise limit is given by the size of the set of residual constraints Res(Σ). We reach this result in three steps: first we show that any set S of constraints with α bound on each constraint allows for a canonical bounding set collecting a bound for each constraint whose size cannot exceed that of S. Then we use this result to identify an upper bound for minimal bounding sets in the size of the constraint set Σ, and finally we show that the residual constraint set Res(Σ) is itself a bound on minimal bounding sets, in fact the strictest possible bound.

We begin defining the notion of *disfavoring system* Δ(z), denoting a collection of constraints where *z* is strictly bounded by some other candidate. Clearly the set of residual constraints Res(Σ) for a favoring hierarchy $\mathcal{H}(\alpha)$ constitutes a disfavoring system for α.

(44) **Def. Disfavoring System.** For any set of candidates K, a set of constraints S is a *disfavoring system* Δ(z) for a candidate *z*∈K iff on each constraint of S some element α in K strictly bounds *z*:

$$\Delta(z) = \{C : C{\in}S \;\&\; \exists\alpha{\in}K, \alpha{>}z \text{ in } C\}$$

We may now show that any disfavoring system of cardinality *n* allows for a bounding set of at most *n* elements, and then generalize the result to constraint sets of any other type.

For any disfavoring system Δ(z) let us define a corresponding ‘blocking set’ B^{Δ} formed by picking for each constraint in Δ(z) an element α that strictly bounds *z* on that constraint; note that α is guaranteed to exist by the definition of disfavoring system.

(45) **Def. Blocking Set.** Let Δ(z) be a disfavoring system defined over some constraint set S and set of candidates K, then B^{Δ} is a blocking set for Δ(z) iff for each constraint C∈S the set B^{Δ} contains exactly one designated element α that strictly bounds *z* on C.

The defining property of a blocking set is that each member constitutes a designated strict bound for a specific constraint. This does not necessarily prevent a constraint from presenting two bounds. For example, the set B={*a,b*} in the example below is a blocking set because *a* counts as the designated strict bound for C_1 and *b* for C_2, even though both candidates strictly bound *z* on C_1 as well. In contrast, the set B′={*a,b,c*} is not a blocking set, because there is no way to identify each of its members as the designated strict bound of a distinct constraint. At least one constraint necessarily ends up with two strict bounds derived from it.

(46)	$\mathbf{C}_1{}^\wedge$	$\mathbf{C}_2{}^\wedge$	$\Delta(z) = \{C1, C2\}$
	a, b	a, c	A possible B^Δ : B={a, b}
	|	|	
	z, c	b	
		|	
		z	

Since more than one candidate may strictly bound *z* on a specific constraint, there may be more than one blocking set B^Δ for any disfavoring system Δ, and different blocking sets may be of different size depending on how many constraints select the same candidate as designated strict bound. In (47) below, for example, the sets {*a,b*} and {*b*} both constitute a blocking set for Δ(z), because both are built by picking just one element for each constraint that strict bounds *z*, except that in the case of {*b*} the selected element is the same for both constraints.

(47)	$\mathbf{C}_1{}^\wedge$	$\mathbf{C}_2{}^\wedge$	
	ab	b	$\Delta(z) = C_\Theta, C_\theta$
	|	|	$B^\Delta_\Theta = \{a,b\}$
	z	z	$B^\Delta_\theta = \{b\}$
		|	
		a	

For a disfavoring system Δ, any corresponding blocking set B^Δ constitutes a bounding set over Δ. A blocking set B^Δ obviously satisfies Strictness, because by definition each member of B^Δ is a strict bound for *z* on some constraint. Moreover, B^Δ satisfies Reciprocity, since for any member strictly bounded by *z* on some constraint of Δ, there is by definition another designated strict bound for *z* on that same constraint.

(48) **Blocking Set Theorem.** For any candidate set K and disfavoring system Δ(z), every corresponding blocking set B^Δ constitutes a bounding set B(z) over Δ(z).

$$\forall\Delta(z),\ B^\Delta = B(z)$$

The Blocking Set Theorem immediately yields the result that the size of a bounding set need be no larger than the cardinality of the constraint set. Any non-null bounding set B(z) implies the existence of a non-null disfavoring system Δ(z) formed by all those constraints where the members of B beat *z* to satisfy Strictness. Then the above theorem guarantees the existence of a corresponding Blocking Set B^Δ which qualifies as bounding set over Δ(z).

We can build B^{Δ} so that it constitutes a bounding set over the full constraint set Σ. All it takes is to pick as designated bound for each constraint in Δ(z) an element from the original B(z), so that B^{Δ} is one of its subsets. Constructed in this way, B $^{\Delta}$satisfies Strictness over Σ, because all members of the superset B(z) do so by definition, and it also satisfies Reciprocity, because on all constraints external to Δ(z) by definition all candidates in B(z) satisfy Reciprocity only vacuously, and this must then be true of its subset B^{Δ} as well.

The result is encoded in the theorem below. Since any candidate allows for a bounding set (since winners have empty bounding sets), we state the theorem in the most general terms:

(49) **First Bound on Bounding Theorem.** For any set of candidates K, constraint set Σ, and for any candidate α∈K, there is a bounding set B(α) over Σ whose cardinality does not exceed that of Σ.

$$\forall\Sigma,\ \forall K,\ \forall\alpha\in K,\ \ |B(\alpha)| \leq |\Sigma|$$

An even stricter limit is provided by the size of the set of residual constraints Res(Σ), which constitutes a disfavoring system for α of size *k*, with $k \leq n$. From any disfavoring system a *blocking set* may be constructed by collecting from each constraint in the system one candidate that is strictly better than α. Such a blocking set is necessarily a bounding set, according to the Blocking Set Theorem above. Since R(Σ) is a disfavoring system, there exists a blocking set B derived from it, with no more than *k* elements. By the theorem, B is a bounding set with respect to the constraints in Res(Σ). But B is also a bounding set with respect to *all* of Σ. The set B satisfies Strictness relative to Σ, since adding constraints can never jeopardize this property once it holds. The same holds for Reciprocity, since B consists of residual candidates, and residual candidates vacuously satisfy Reciprocity on any non-residual constraint, because they must be order-equivalent to α there. It follows that B constitutes a bounding set for α in Σ whose size at most equals that of Res(Σ). We record this result in the theorem below.

(50) **Second Bound on Bounding Theorem.** For any candidate set K, constraint set Σ, and candidate α∈K, there is a bounding set B over Σ whose cardinality does not exceed that of Res(Σ).

$$\forall\Sigma,\ \forall K,\ \forall\alpha\in K,\ \exists B(\alpha)\ |B(\alpha)|\ \leq |\mathrm{Res}(\Sigma)|$$

The bound provided by the size of Res(Σ) is also the strictest available one, in general, because a minimal bounding set may match it. One such case is shown in example (43) above, repeated below for convenience. The only blocking set available for *z* over the residual constraints B and C must include the two candidates *a* and *b*, reaching the cardinality of the residual constraint set itself.

(51)	**B^**	**C^**	Res(Σ) = {B, C}
	a	b	Res(K) = {a, b, c, z}
	|	|	Blocking set: $B^{Res(\Sigma)}(z)$ = {a, b}
	z	z, c	
	|	|	
	b, c	a	

Minimal bounding sets. Each minimal bounding set must necessarily be a subset of some blocking set defined over the set of residual constraints. Consequently, blocking sets bound minimal bounding sets in size. This leads to further limitations on the size of minimal bounding sets. Recall that there can easily be several blocking sets associated with the same disfavoring system, as long as some constraint has more than one candidate that is better than the targeted loser. According to the definition, only one such candidate may be chosen for a given blocking set, but either one is chooseable. Furthermore, a lucky choice might also do double duty, serving as the *z*-bettering candidate for some other constraint. This means that blocking sets from the same disfavoring system can be of different sizes as well as different compositions.

If B is a minimal bounding set for *z* in Σ, then B is a subset of the unique maximal bounding set B^{Max}, and hence its members must be drawn from the set of residual candidates that are strictly better than *z* on some residual constraint. Since B is minimal, it never needs more than one such bound per residual constraint, and is therefore a subset of some blocking set; the result is formalized in the theorem below.

(52) **Minimal Bounding Set Theorem.** For any candidate set K, constraint set Σ, and candidate α∈K, let B be a minimal bounding set for α over K and Σ, then B is a subset of some blocking set B′ over Res(Σ), drawn from Res(K).

A minimal bounding set need not be itself a blocking set. For example, given the residual constraints in (53) below, the only available blocking set for *z* is B={*a*,*b*}. The minimal bounding set, however, is B′={*a*}, which does not constitute a blocking set because it includes no candidate that is strictly better than *z* on constraint C_2. Yet B′satisfies both Strictness and Reciprocity (the latter only vacuously), and is thus a bounding set.

```
(53)    C1^            C2^          Blocking set: B={a, b}
         a              b           Minimal bounding set: B′={a}
         |              |
         z             z, a
         |
         b
```

This example illustrates how ‘weak bounding’ can be exploited even on residual constraints to limit the size of a bounding set. (As in the definition of simple harmonic bounding (2), we say that *a* ‘weakly bounds’ *b* on a constraint if *a* is at least as good as *b* on the constraint, maybe better, but not worse.) On all the other constraints, those that made it into the favoring hierarchy, *z* is weakly bounded by all residual candidates because there they must be order-equivalent to *z*. The definition of ‘blocking set’ requires that a strict bound for *z* be collected from constraint C_2. But for purposes of constructing a satisfactory bounding set, this is unnecessary, because *a*, whose strictness is guaranteed by C_1, weakly bounds *z* on this constraint, and there is no issue with reciprocity. (Choosing *a* here is thus consistent with the Strictness and Reciprocity conditions.) Because Strictness requires that each member of a bounding set be better than *z* on some residual constraint, it follows that each member of a minimal bounding set is also a possible member of a blocking set, ensuring the subset relation between minimal bounding sets and blocking sets.

3.4 Summary

The theorems examined in the previous sections emerge as closely related to each others, together casting light on the architecture of the order calculus intrinsic to harmonic optimization.

We started from the observation that optimization is intrinsically comparative, a property that suggests an infinite number of comparisons between a winner and the infinite losers. This intuition was proven wrong. The Ranking Determined by Winners Theorem tells us that all it takes to be optimal for some ranking is to beat all other winners, in principle freeing the learner and the analyst from comparing the winners with the infinite set of losers.

The issue is rather whether there is a finite and efficient way to know if a candidate is a winner or a loser. The answer is positive. The Winner/Loser Theorem, which recapitulates Tesar and Smolensky's RCD procedure and demonstrations, guarantees that we can tell apart a winner from a loser by simply building the associated favoring hierarchy $\mathcal{H}(\alpha)$, thus freeing ourselves from collecting them through the lengthy calculations of the *n*! available rankings of any *n*-sized Σ.

The importance of favoring hierarchies emerges equally strong when tackling the issue from the perspective of losers. The Bounding Theorem ties loser-status to the existence of one or more candidates collectively satisfying the two ranking-independent conditions defining bounding sets, Strictness and Reciprocity. But searching for bounding set members by eliminating non-bounds leads once again to favoring hierarchies, since the identification of successive favoring strata for α is the way to eliminate any potential bounding set member whose inclusion in the bounding set would cause Reciprocity failure. The Maximal Bounding Set Theorem then ensures that the same stratified hierarchy will yield a non-empty unique maximal bounding set B^{Max} for α when α is a loser. The set B^{Max} may in turn include one or more blocking bounding sets, each including one or more minimal bounding sets whose size never exceeds that of the set of residual constraints.

All these results rest on the reinterpretation of constraints as functions that return the maximal elements of a stratified hierarchy, and of optimization as functional composition; together, they form the base for the formal demonstrations of the next two sections.

4. The Fine Structure of Optimization

To set the stage for the main results, we establish here some fundamental properties of OT grammars. We start (§4.1) with the most basic properties of constraint hierarchies, which practitioners will not find surprising, and we move on to address the specific properties of favoring constraints and favoring hierarchies (§4.2). The results demonstrated here will provide the tools for proving the central theorems on winner/loser status in §5.

4.1 Basic Properties

Our first lemma concerns the recursive nature of evaluation: the generic evaluative problem faced by a ranking R when confronted by a candidate set K is replicated as we move down the hierarchy. Thus, evaluation of K over a ranking R=GH can be exactly recast as evaluation of the set of candidates G(K) over the subhierarchy H. In essence, this is a kind of 'compositionality' of ranking, which follows immediately from the representation of hierarchies as compositions of constraint functions.

(54) **Compositionality of Ranking**. Given a ranking R, and a partition R=GH, then the optima for R over the candidate set K are exactly the same as those for H over the candidate set G(K).

$$[GH](K) = H(G(K))$$

Pf. We have G = $[C_1...C_i]$ and H=$[C_{i+1}...C_n]$, for constraints $C_1,...,C_n$.

[1] By the definition of constraint hierarchy, G(K) = $C_i(..(C_1(K)))$.

[2] Likewise, H(G(K)) = $C_n(..(C_{i+1}($ G(K)))).

Substituting [1] into [2], we obtain, H(G(K)) = $C_n(..(C_{i+1}(C_i(..(C_1(K))))))$= R(K). □

We now show that some familiar properties of constraint hierarchies follow from our definitions: candidate sets shrink[11] as evaluation proceeds; multiple optima must do equally well on all constraints; relative harmony is a strict order; and optimal candidates are maximal in that order.

Shrinkage. *No constraint can ever restore a candidate that has fallen out of the candidate set*. In other words: for any sequence of constraints R, the set of survivors is monotonically non-increasing as each additional constraint is evaluated. In terms of functional composition, we want to show that, given R=[GH], the set of optima for the constraint function H∘G is included in the set returned by G. We first note an obvious property of all compositions of 'shrinking functions', those for which $f(K) \subseteq K$.

(55) **Shrinkage Property**. Let R be any ranking on any constraint set, K any set of candidates.

$$R(K) \subseteq K$$

Pf. If R contains only one C, then $C(K) \subseteq K$ for any set of candidates K, because C(K) is defined as max(K;C^)⊆K. Assume that the theorem holds up for hierarchies up to some length *n*, then consider a hierarchy R of length *n*+1. We have R=[HC] and $[HC](K) = C(H(K)) \subseteq H(K) \subseteq K$, the second inclusion following from the induction hypothesis, the first from our initial remark. □

This result, as André Nündel has put it, affirms only that the optimal candidate is *in* the candidate set. A more general shrinkage property for constraint hierarchies follows forthwith:

[11] We include both proper and improper subsetting under the rubric of 'shrinkage'.

(56) **Inclusion Lemma**: For any ranking R=GH, over any set of constraints Σ, and for any candidate set K,

$$[GH](K) \subseteq G(K)$$

Pf. $R(K)=[GH](K) = H(G(K))\subseteq G(K)$ by the shrinkage property, since G(K) is a set of candidates.

Terminological remark. It is useful to be able to refer to the G of a ranking [GH]. Order theory provides the right term: '(initial) section'. The (initial) section of an order is a contiguous subsequence starting at a designated extremum, for us the top.

(57) **Def. Initial section**. Let $\langle S;>\rangle$ be a totally ordered set. An *initial section* of $\langle S;>\rangle$ is a subset $I\subseteq S$ such that $\forall x\in I\ \forall y\in S\ (y>x \Rightarrow y\in I)$.

Example. Familiarity with the shrinkage property can be gained by working through the following example. In (58), the constraints are ranked left to right.

(58) $R=C_1 >> C_2 >> C_3$, K= {a, b, c, d}

C_1^	**C_2^**	**C_3^**
a, b, c	d	c, d
|	|	|
d	a, b	a
	|	|
	c	b

Let us evaluate various initial sections of the hierarchy, according to the recursive definition of ranking as function composition given in (26) above.

• For R=Ø, the null constraint-sequence, all candidates in K={a,b,c,d} are in the winning set.

•At R = $[C_1]$, candidate *d* is lost:

$C_1(\{a,b,c,d\}) = \{a,b,c\}$

Note: *d* is not maximal in C_1 relative to K.

•At R = $[C_1C_2]$, the set of surviving competitors is thinned to *a* and *b*:

$C_2\ (C_1(K)) = C_2(\ \{a, b, c\}) = \{a,b\}$.

Note: candidate *d* beats both *a* and *b* in C_2 over K, but it is *not in* $C_1(K)$. Having lost to them on C_1, *d* cannot be in $[C_1C_2](K)$, an instance of the Inclusion Lemma.

• At R = $[C_1C_2C_3]$, *a* emerges as the optimum:

$C_3((C_2\circ C_1)\ (K)) = C_3(\{a,b\}) = \{a\}$

> Note: Both *c* and *d* are eliminated in shorter initial sections of the ranking and they have no effect on the calculation at the C_3 level, despite beating *a* on C3 over all of K. But *a* is still in the running and beats *b*, pushing it out of the survivor set. Thus, the set of optima for this ranking — $[C_1C_2C_3\ ](K)$ — contains only *a*.

We can graphically illustrate the process of elimination as follows:

(59) C_1>>C_2>>C_3, K= {a, b, c, d}

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

This example also shows that being at the top of some constraint is not sufficient for being optimal under all rankings. It is, of course, a *necessary* condition for being optimal under *some* ranking (§4.2). For example, under the ranking C_2>>C_1>>C_3, the winner is *d*, and *c* wins under the ranking C_3>>C_1>>C_2.

But topmost status in a constraint is still not *sufficient* for being optimal under *some* ranking, i.e. for being a *winner*. For example, in (60) below, although it is at the top of C_1, *b* is never optimal under any ranking of C_1 and C_2.

(60)

$\mathbf{C_1}$^	$\mathbf{C_2}$^
a,b	a
	|
	b

Equivalence. These examples illustrate a further important property of survivor sets: all candidates that survive the evaluation of a constraint C somewhere in the hierarchy are always *equivalently ordered* on any constraint that precedes C in the ranking sequence. All such candidates must be equivalently ordered on C itself, and by the Inclusion Lemma it follows that they are equivalently ordered by all preceding constraints. These observations are recorded in the Equivalence Lemma below:

(61) **Equivalence Lemma.** Let R be a ranking on a set of constraints Σ . Let x,y∈K.

If *x* and *y* are both optimal in K for R, they are order-equivalent on every constraint in Σ.

$$\forall C\in\Sigma,\ \ x,y\in R(K) \Rightarrow (x\approx y;C^\wedge)$$

Pf. [1] Observe first that if *x,y* are both optimal on some R, then they are order-equivalent on the *last* constraint C in R=HC. This follows because R(K) = C(H(K)): since *x,y*∈R(K), we have by definition of C as function that both are maximal elements in C^ over H(K), and therefore neither x>y nor y>x on C^. But this means that they are order-equivalent in C^ over *any* set of candidates: (x≈y;C^).

[2] Now, consider any other non-null initial section $R^i \neq R$; it must also have a *last* constraint C_k, to which the preceding remark will apply. By the Inclusion Lemma, $x,y \in R(K) \Rightarrow x,y \in R^i(K)$. And by what we have just noted in [1], we have, for $R^i = HC_k$, $x,y \in [HC_k](K) \Rightarrow (x \approx y; C_k{}^\wedge)$. So if $x,y \in R(K)$, we have $(x \approx y; C_k{}^\wedge)$ for any C_k in Σ, since R^i is arbitrary. □

The Equivalence Lemma places no conditions on the size or shape of R, and R itself may be an initial section of some other hierarchy. It follows, as promised, that members of the survivor set at any point in a ranking are equivalently ranked on *every* preceding constraint.

(62) **Corollary**. Let $R=[GH]$ and $x,y \in G(K)$. Then for every C in G, we have $(x \approx y; C)$.
Pf. Apply the Equivalence Lemma to G.

The competitors that survive some initial section of a ranking along with a specific candidate ω are also the only candidates that may still threaten its optimal status by beating it on the not-yet- assessed constraints of the complete ranking — they constitute the potential membership of a bounding set for ω.

We conclude by reconstructing the notion of *relative harmony*: order on the full candidate set with respect to a hierarchy, as projected from the individual orders imposed by constraints. To date, we have dealt only the pursuit of optima and have said nothing at all about how suboptimal forms might relate to each other. But our apparatus allows a natural way to compare any two candidates: we will say that a candidate *x* is *more harmonic* than another candidate *y* relative to a ranking R whenever *x* wins and *y* loses in the set of candidates $\{x,y\}$, in which case we will write $(x \succ y; R)$.

(63) **Def**. **Relative Harmony**. For any candidates x,y, let $K=\{x,y\}$. We say:

$x \succ y$ on a ranking R iff $x \in R(K)$ and $y \notin R(K)$.

Through relative harmony, a constraint hierarchy imposes a stratified strict order on any set of candidates, if each constraint is itself a stratified partial order.[12] A constraint hierarchy is thus itself a constraint. The optimal candidates are simply the maximal elements of the "$\succ$" order . These important properties are captured in the following lemmas.

(64) **Relative Harmony is a Stratified Strict Order**. Let $x,y,z \in K$. Let R be a ranking on some set of constraints Σ, and let "$\succ$" be the relative harmony relation. Then,

(a) "$\succ$" is irreflexive: $\neg\,(x \succ x)$,
(b) "$\succ$" is asymmetric $x \succ y \Rightarrow \neg\,(y \succ x)$
(c) "$\succ$" is transitive: $(x \succ y \;\&\; y \succ z) \Rightarrow (x \succ z)$.
(d) "$\succ$" is stratified: $\forall a,b,x \in K, \neg(a \succ b \vee b \succ a) \Rightarrow (a \succ x \Leftrightarrow b \succ x)$

[12] If a constraint is allowed to be a more general form of partial order, the result of constraint composition may not even be an order at all: see Appendix A.

A strict order is asymmetric and irreflexive, as is reflected in conditions (a) and (b).

(65) **Optimality Maximality Lemma**.
$\alpha \in R(K)$ iff α is a maximal element in the harmonic order $R^\wedge = \langle K; \succ \rangle$ induced by R on K.

For the sake of swiftness, we relegate the proofs to Appendix A, and move directly on to the analysis of favoring.

4.2 Favoring

In this section, we establish the basic properties of favoring constraints and favoring hierarchies. Recall that a *favoring constraint* for a candidate is one for which that candidate is maximal in its candidate set.

Def. Favoring Constraint F is a *favoring constraint* for α over K iff $\alpha \in F(K)$.

As noted above, the *favoring* property generalizes the notion of 'satisfying a constraint' to the OT context: a favored candidate need not literally *satisfy* some criterion, but it must *do best* on it, in the sense that none of its co-candidates do better.

Our major goal is to show that the entire set of favoring constraints can be moved to the top of a ranking, without affecting the success (or failure) of a favored candidate on that ranking. When set to work recursively in §5, this result will allow us to deduce that every potential winner has an exhaustive favoring hierarchy, and every loser a non-exhaustive favoring hierarchy which determines its non-empty maximal bounding set. We will also show that the output of a collection of favoring constraints is determined independent of ranking: it is just the intersection of the tops of each constraint in the collection. We will approach these results through a series of small steps.

First, we note a fact that will become useful later: any ranking on which a candidate wins must begin with a favoring constraint. (If not, some candidate would beat it right away, at the evaluation of the first constraint.)

(66) **Initial Favoring Lemma**. For $\alpha \in K$, and any ranking R=CH of some constraint set Σ, if α wins over R, then C is a favoring constraint for α in K:

$$\alpha \in [CH](K) \Rightarrow \alpha \in C(K).$$

Pf. By the Inclusion Lemma, $[CH](K) = H(C(K)) \subseteq C(K)$. Therefore $\alpha \in [CH](K) \Rightarrow \alpha \in C(K)$. □

Now we set off toward our main goals. The key background observation is that a favoring constraint can never be responsible for the *suboptimal* status of its favored candidate; it cannot displace the favored candidate from any candidate set.

It follows that adding a favoring constraint to a ranking cannot change the prior status of the favored candidate, in the following sense: if α is winner on a ranking sequence R, then α must also win on the extended ranking [RF]; and if α loses on R, it must also lose on [RF]. (The latter point is obvious from the fact that no constraint may put a losing candidate *back* into the winner set.) More generally, if α wins on $R=[GH]$ then α also wins on $[GFH]$, for any G and H.

This result is stated and established in the Favoring Transparency Lemma (70). In the interests of generality, we will assume only the minimum necessary about constraint orders: that they provide maximal elements; we will not require of them that they be stratified orders, such as are produced by violation theory.

In reasoning toward the Favoring Transparency Lemma, we use a simple but useful result about maximal elements in ordered sets: maximality in a set entails maximality in any subset that includes the maximum. If Everest is the tallest mountain on planet Earth, it is the tallest mountain in Asia, and in Nepal. If Buddy is the meanest dog east of the Mississippi, we may legitimately infer that he’s the meanest dog in New Jersey, and in New Brunswick, where he lives.

(67) **Downward Inheritance of Maximality**(DIM). If x is maximal in an ordered set P, it is also maximal in every subset of P to which it belongs.

Let $x\in\max\langle Q;O\rangle$.Then for any $P\subseteq Q$ with $x\in P$, $x\in\max\langle P;O\rangle$.

Pf. $x\in\max\langle Q;O\rangle$ iff $\forall y\in Q, \neg (y>x)$. In particular, then, since $P\subseteq Q$, $\forall z\in P, \neg (z>x)$. □

We can now establish that favoring constraints for α are a kind of identity with respect to α’s fate on a ranking:

(68) **Favoring Extension**. Let $R=[HF]$ be a ranking, with F a favoring constraint for α in K. Then,

$$\alpha\in H(K) \Leftrightarrow \alpha\in[HF](K).$$

Pf. Right-to left: by the Inclusion Lemma, $[HF](K)\subseteq H(K)$. Left-to-right: we argue that since α is in the set $H(K)$ and maximal with respect to F^ over its superset K, we can apply DIM to get the result. That is: by hypothesis $\alpha\in H(K)$, and $H(K)\subseteq K$ by Shrinkage. Since α is favored by F, i.e. maximal in the F^ order over K, it follows by the DIM that α is maximal for F^ over $H(K)\subseteq K$, i.e. that $\alpha\in F(H(K))= [HF](K)$. □

We still need to check that the equivalence of a hierarchy H and its extension HF, as far as α’s winning or losing goes, is preserved when H and HF are themselves initial sections of some larger hierarchy.

To show this, we first note a property that follows directly from the DIM as applied to the harmonic order “$\succ$” imposed by rankings on candidate sets: if a candidate wins on some ranking for a given candidate set, it also wins on a subset of that candidate set, so long as the winner is in the subset. This is pure DIM reasoning: if the winner beats everything in the bigger set, it also beats everything in the subset. (This observation is also useful in its contrapositive form: if α *loses* in a certain candidate set, then α loses on any superset of it.)

(69) **Optimality Inheritance**. Suppose $K'\subseteq K$, and let $\alpha\in K'$. Then: $\alpha\in R(K) \Rightarrow \alpha\in R(K')$.

Pf. Candidate α is maximal in the harmonic order ≻ determined by R over K. By DIM, α is maximal over $\langle K';\succ\rangle$ as well. □

With this in hand, we show that a favoring constraint for α can be inserted in any ranking that α wins on, without disturbing α's primacy.

(70) **Favoring Transparency Lemma**. For any ranking R=[GH], candidate set K, and for any constraint F favoring α in K,

$$\alpha\in[GH](K) \Rightarrow \alpha\in[GFH](K).$$

Pf. In the functional notation, we want $\alpha\in H\circ G(K) \Rightarrow \alpha\in H\circ F\circ G(K)$. We argue by Optimality Inheritance that since α is optimal on H(X) for X=G(K), it must also be optimal for H(Y), for $Y\subseteq X$, where $Y=[GF](K)=F\circ G(K)$. G(K) will play the superset role, [GF](K) the subset role in Optimality Inheritance.

First, note that $[GF](K) \subseteq G(K)$ by Inclusion. We need $\alpha\in[GF](K)$ to be able to invoke Optimality Inheritance. By hypothesis,we have $\alpha\in[GH](K)$, yielding $\alpha\in G(K)$ by Inclusion. By Favoring Extension, $\alpha\in[GF](K)$. So we have, as desired, $\alpha\in[GF](K) \subseteq G(K)$.

Now, by Optimality Inheritance, $\alpha\in H(G(K)) \Rightarrow \alpha\in H(F\circ G(K))$. □

Favoring Transparency leads to a very useful result: favoring constraints can be promoted in a ranking without affecting the optimality/suboptimality of the favored candidate. Schematically, ...HF... can be flipped into ...FH... without affecting the status of F's favorites. This fact will allow us to re-arrange hierarchies into a canonical form that is most favorable to a targeted candidate and least favorable to its potential bounding set— a favoring hierarchy.

The proof turns on the following fairly obvious fact: a constraint cannot be meaningfully repeated lower down in a hierarchy.

(71) **Repetition Futility**. For any constraint hierarchy X, any constraint C, any candidate set K:

$$[CXC](K) =[CX](K)$$

Pf. By Favoring Extension, [X] = [XC] over any candidate set that consists entirely of elements favored by C. But C favors every element of the set C(K). Hence, [X](C(K)) = [XC](C(K)) = C(X(C(K)). □

(72) **Corollary**. For any constraint hierarchies X,Y,Z, any constraint C, and candidate set K:

$$[XCYCZ](K) = [XCYZ](K)$$

Pf. Since [CYC] is merely a function on the set of candidates X(K), by Repetition Futility we can replace it with [CY]. Now, we have

$$[XCYCZ](K) = [CYCZ]\ (X(K)) = Z([CYC](X(K)) = Z([CY](X(K))) = [CYZ](X(K))=[XCYZ](K).$$

The first two steps are justified by Compositionality of Ranking, the third by Repetition Futility, the last two by Compositionality again. □

We can now easily establish the desired result.

(73) **Favoring Promotion Theorem**. Let K be a set of candidates, Σ a set of constraints, Z a constraint hierarchy, and let F be a favoring constraint for α in K. Then:

$$\alpha\in [\ldots \mathrm{ZF}\ldots](K) \Rightarrow \alpha\in [\ldots \mathrm{FZ}\ldots](K)$$

Pf. $\alpha\in [\ldots \mathrm{ZF}\ldots](K) \Rightarrow \alpha\in[\ldots \mathrm{FZF}\ldots](K) \Rightarrow \alpha\in[\ldots \mathrm{FZ}\ldots](K)$.
The first step follows from Favoring Transparency, the second from Repetition Futility. □

Favoring Promotion gives us real power: it ensures that any candidate α winning on a ranking R will also win on any permutation R′ differing from R only in that *all* favoring constraints for α have been promoted to the front. This provides the following necessary condition for optimality, which is the first step toward the Winner/Loser Theorem.

(74) **Favoring/Residue Lemma**. Let R be any ranking of a set of constraints Σ. Consider the set $\mathscr{F} = \mathscr{F}(\alpha, K, \Sigma)$ of the constraints in R favoring α in K, and let $\mathscr{F}^{>>}$ be *any* ranking for them. Let $[\Sigma-\mathscr{F}]^{R}$ be a hierarchy consisting of the residual constraints in Σ ranked according to their order in R. Then the ranking $[\mathscr{F}^{>>}][\Sigma-\mathscr{F}]^{R}$ has α as an optimum whenever α is optimal for R.

$$\alpha\in R(K) \Rightarrow \alpha\in[\mathscr{F}^{>>}][\Sigma-\mathscr{F}]^{R}(K).$$

Pf. Suppose $\alpha\in R(K)$. By the Initial Favoring Lemma, R=FH for some favoring constraint, hence $\mathscr{F}^{>>}\neq\emptyset$. Now, starting from R, build $R'=[\mathscr{F}^{>}][\Sigma-\mathscr{F}]^{R}$stepwise by promoting to the front the highest-ranked favoring constraint F that follows a non-favoring constraint in R. Starting with $R=R_0$ and terminating with $R_n=R'$, each promotion transforms an intermediate ranking $R_i= [\mathrm{YFZ}]$ into $R_{i+1}=[\mathrm{FYZ}]$, with the Favoring Promotion Theorem ensuring that at each step $\alpha\in R_i(K)\Rightarrow \alpha\in R_{i+1}(K)$. Therefore, by transitivity of implication, $\alpha\in R(K) \Rightarrow \alpha\in R'(K)$. □

We conclude by showing that a favoring stratum applied to a candidate set returns the intersection of the tops of all the favoring constraints in it. This provides an upper bound for the set of optima of any ranking with the shape $[\mathscr{F}^{>>}][\Sigma-\mathscr{F}]^{R}$, as well as for each bounding set, since the intersection may include candidates that co-occur with the favored candidate at the top of each favoring constraint but are beaten elsewhere.

As a preliminary, we establish that a favored candidate wins on any ranking of a set of favoring constraints.

(75) **Favoring Stratum Optima.** Let $\mathscr{F}$ be a set of favoring constraints for α in K. Then α wins on any possible ranking $\mathscr{F}^{>>}$ over $\mathscr{F}$.

$$\forall\mathscr{F}^{>>}, \text{ a ranking of } \mathscr{F},\ \alpha\in\mathscr{F}^{>>}(K).$$

Pf. For any ranking $\mathscr{F}^{>>}$ over $\mathscr{F}$, let I^{k+1} and I^{k} be two initial sections of $\mathscr{F}^{>>}$ such that $I^{k+1} = I^{k}F$. Then, by Favoring Extension, $\alpha\in I^{k+1}(K)$ iff $\alpha\in I^{k}(K)$. Hence, by transitivity of implication, on all such statements over initial sections I^{j} for $0\leq j \leq n$, we have $\alpha\in I_n(K)=\mathscr{F}^{>>}(K)$ iff $\alpha\in I_0(K) = K$, and since $\alpha\in K$ it follows that $\alpha\in\mathscr{F}^{>>}(K)$. □

(76) **Favoring Intersection Lemma.** Let $\mathscr{F} = \{F_1,...,F_n\}$ be a set of favoring constraints for α in K. Then the set of optima $\mathscr{F}$(K) for any ranking $\mathscr{F}^{>>}$ on $\mathscr{F}$ is equal to $\cap_i$ F_i(K), $1 \le i \le n$.

$$\mathscr{F}(K) = \cap_i F_i(K).$$

Pf. By Favoring Stratum Optima, we have for any $\mathscr{F}^{>>}$ over $\mathscr{F}$, $\alpha \in \mathscr{F}^{>>}(K)$. By the Equivalence Lemma it follows that $\forall x \in \mathscr{F}^{>>}(K)$, we must have $(x \approx \alpha; F^\wedge)$ for any F in $\mathscr{F}$. Since for all $F \in \mathscr{F}$ we have by hypothesis $\alpha \in F(K)$, it follows that $x \in F(K)$, and therefore that for any ranking $\mathscr{F}^{>>}$ over $\mathscr{F}$, $\mathscr{F}^{>>}(K) \subseteq \cap_i F_i(K)$.

For the reverse inclusion relation, let $x \in \cap_i F_i(K)$. Then for any $F \in \mathscr{F}$, we have $x \in F(K)$. Hence all F in $\mathscr{F}$ are favoring constraints for *x*, from which it follows, by Favoring Stratum Optima, that $x \in \mathscr{F}^{>>}(K)$ for any ranking $\mathscr{F}^{>>}$ over $\mathscr{F}$. Therefore $\cap_i F_i(K) \subseteq \mathscr{F}^{>>}(K)$. □

5. Winners and Losers

In this section we establish the two central related results: the Winner/Loser Theorem, which asserts that a candidate is a winner if and only if it has an exhaustive favoring hierarchy, and the Bounding Theorem, which asserts that a candidate is a loser if and only if it has a non-null bounding set.

5.1 The Winner/Loser Theorem and the Bounding Theorem

We begin by restating the initial theorem on the determination of ranking by winners, which ensures that ω is optimal on any ranking R if and only if ω is at least as good on R as all potential winners, the elements of W(K,Σ). This is equivalent to saying that $\omega \in R(K)$ iff $\omega \in R(W(K,\Sigma))$, where on the right hand side the competition is restricted to potential winners only. This is equivalent in turn to saying that the set of optima with respect to K over Σ is identical to the set of optima with respect to W(K,Σ), i.e. R(K) = R(W(K,Σ)). In this concise form, the theorem transparently shows the dispensability of losers: competing for optimality over W(K,Σ) is equivalent to competing over the potentially much larger loser-inclusive K.

(77) **Determination of Ranking by Winners.**

Let Σ be a set of constraints, and K a set of candidates. Let W(K,Σ) be the set of candidates that are optimal for some ranking R of Σ, and let $\omega \in K$. Then ω is optimal in K on a ranking R iff for every $x \in W(K,\Sigma)$ we have $(\omega \succeq x; R)$. In short,

$$R(K) = R(W(K,\Sigma))$$

Pf. Assume first $\omega \in R(K)$. By the Optimality Maximality Lemma (65), for any $x \in K$, $(\omega \succeq x;R)$. Since $W(K,\Sigma) \subseteq K$, for any $\omega' \in W(K,\Sigma) \subseteq K$ it follows that $(\omega \succeq \omega';R)$, and hence that $\omega \in R(W(K,\Sigma))$. For the reverse inclusion relation, assume $\omega \in R(W(K,\Sigma))$, and let ω' be optimal in K on R, i.e. $\omega' \in R(K)$. From this, we have $\omega' \in W(K,\Sigma)$. But then because $\omega \in R(W(K,\Sigma))$, we have $(\omega \succeq \omega'; R)$. This means that ω must also be a maximal element in K on the $R^\wedge$ order. By the Optimality Maximality Lemma, it follows that $\omega \in R(K)$. □

Let us turn to the Winner/Loser Theorem, which shows that a candidate ω is a winner over constraint set Σ if and only if there exists a favoring hierarchy $\mathcal{H} = \langle \mathscr{F}_1 .. \mathscr{F}_n \rangle$ for ω exhausting Σ.

(78) **Winner/Loser Theorem.** For a set of candidates K, a constraint set Σ, and candidate $\alpha \in K$, α is a winner over Σ iff there is a favoring hierarchy for α , $\mathcal{H}(\alpha) = \langle \mathscr{F}_1 .. \mathscr{F}_n \rangle$, exhausting Σ.

$$\alpha \in W(K,\Sigma) \Leftrightarrow \forall C \in \Sigma,\ C \in \mathscr{F}_i \text{ for some } \mathscr{F}_i \in \mathcal{H}(\alpha).$$

Before advancing to the proof, it is useful to repeat the definition of 'favoring hierarchy' (see (32) in §3.2 above). First we define 'output of a set of favoring constraints', the key building-block of the favoring hierarchy definition.

(79) **Def. Output of Favoring Stratum**. Let $\mathscr{F}(\alpha,K,\Sigma) = \{F_1,..F_n\} \subseteq \Sigma$ be a set of favoring constraints F for α over Σ with respect to K. Then by $\mathscr{F}(K)$ we denote $\mathscr{F}^{>>}(K)$, where $\mathscr{F}^{>>}$ is any ranking of the constraints $\{F_1,..F_n\}$.

Observe that the expression $\mathscr{F}(K)$ is well-defined. From the Favoring Intersection Lemma (76), we know that any ranking $\mathscr{F}^{>>}$ of $\mathscr{F}$ gives the same result when applied to a set K containing α, namely $\cap_j F_j(K)$, and $\mathscr{F}_i(K_i)$ denotes that unique result.

With this settled, the notion 'favoring hierarchy' can be recursively defined.

(80) **Def. Favoring Hierarchy**. Let K be any set of candidates including α, and Σ any set of constraints. Let $\mathscr{F}_i(\alpha,K,\Sigma)$ be a set of favoring constraints for α over Σ with respect to K. Then the *favoring hierarchy* $\mathcal{H}(\alpha)$ is a stratified hierarchy $\langle \mathscr{F}_1 ... \mathscr{F}_n \rangle$ where each favoring stratum $\mathscr{F}_i$ is a non-empty set of favoring constraints recursively defined as follows:

Base step:	Comments:
$K_1 = K$	
$\Sigma_1 = \Sigma$	
$\mathscr{F}_1 = \mathscr{F}(\alpha,K_1,\Sigma_1)$	1st fav. stratum = set of favoring constraints for α over K and Σ
Recursive step:	
$K_{i+1} = \mathscr{F}_i(K_i)$	Next candidate set = co-winners of current favoring stratum.
$\Sigma_{i+1} = \Sigma_i - \mathscr{F}_i$	Next constraint set = current set minus current favoring stratum
$\mathscr{F}_{i+1} = \mathscr{F}(\alpha,K_{i+1},\Sigma_{i+1})$	Next favoring stratum = favoring constraints for α over the new sets of candidates and constraints

Remark. It is instructive to unfold the shape of hierarchy from the recursive definition. Notice first that $K_{m+1} = \mathscr{F}_m(K_m) = \mathscr{F}_m^{>>}(K_m)$, where $\mathscr{F}_m^{>>}$ is any ranking of $\mathscr{F}_m$. From this we have, repeatedly substituting for K_j :

$$\begin{aligned} K_{m+1} = \mathscr{F}_m(K_m) &= \mathscr{F}_m^{>>}(K_m) \\ &= \mathscr{F}_m^{>>}(\mathscr{F}_{m-1}^{>>}(K_{m-1})) \\ &= ... \\ &= [\mathscr{F}_m^{>>} ... \mathscr{F}_1^{>>}](K) \end{aligned}$$

Along the same lines, by repeated substitution we can unfold the definition of Σ_{m+1} :

$$\begin{aligned}\Sigma_{m+1} &= \Sigma_m - \mathscr{F}_m = (\Sigma_{m-1} - \mathscr{F}_{m-1}) - \mathscr{F}_m \\ &= \Sigma - \mathscr{F}_1 - \ldots - \mathscr{F}_m \\ &= \Sigma - \bigcup_{\Theta\leq\varkappa\leq\Upsilon} \mathscr{F}_{\varkappa}\end{aligned}$$

Proof of Theorem. We start with the right to left entailment: if α has an exhaustive favoring hierarchy over K and Σ, then α wins on some ranking. It is straightforward to construct such a ranking from the favoring hierarchy itself.

Let $\mathcal{H}(\alpha) = \langle \mathscr{F}_1 \ldots \mathscr{F}_n \rangle$ be the favoring hierarchy for α over K exhausting Σ. Let R be any refinement of $\mathcal{H}$, such that each $\mathscr{F}_k^{>>}$ is some arbitrary ranking of the constraints in the $\mathscr{F}_k$ stratum. By the remark on unfolding, $K_{n+1} = [\mathscr{F}_1^{>>\cdot} \ldots \mathscr{F}_n^{>>}]$ (K). But by the definition of favoring hierarchy, we have it that $\alpha \in K_i$ for each K_i , the output of the favoring stratum $i-1$. In particular, $\alpha \in K_{n+1}$. So α is optimal on some ranking, and a winner. □

We now prove the reverse entailment: if α is a winner, then it has an exhaustive favoring hierarchy. The proof is a bit trickier, because the result is more contentful: it asserts that a winner over Σ, optimal on some ranking of shape unknown, will always be optimal on one particular kind of ranking.

This conclusion is reached by repeatedly invoking the Favoring/Residue Lemma (74), which asserts that if α wins on a ranking R, then α wins over the hierarchy obtained from R by promoting all favoring constraints for α to the front. The same lemma can then be invoked in reference to the residual hierarchy of nonfavoring constraints left behind by the fronting maneuver, this time fronting the constraints that favor α over the co-winners of the first favoring stratum, which still maintains the optimality of α. And so on, until there are no more constraints left to deal with. (This construction mirrors RCD.) We show that the hierarchy produced in this fashion corresponds in fact to a favoring hierarchy as defined in (80) and is exhaustive of Σ.

For purposes of the proof, it is useful to identify and define the intermediate product of the construction: a 'Quasi-Favoring Hierarchy' (QFH) for α over K and Σ. Let us say that a ranking R of Σ is a QFH of depth *n* for $\alpha \in K$, if R can be partitioned into *n*+1 segments, of which at least the first *n* are non-empty, and where the k^{th} segment, for $k \leq n$, is precisely a ranking of the constraints in the set $\mathscr{F}_k$ as it is defined in (80). Let us call the initial *n* segments the 'favoring section' of the ranking and the *n*+1st segment the 'residue'. The Favoring Hierarchy for α over K and Σ is then the QFH for α with a null residue. We now establish the following claim: if $\alpha \in R(K)$ for an arbitrary ranking R over Σ, then α is optimal for every quasi-favoring hierarchy QFH^R over Σ in which the residue is a set of constraints ranked according their order in R, including the case where the residue is null.

Assume $\alpha \in W(K,\Sigma)$, that is: $\alpha \in R(K)$ for some R over Σ.

[1] Let $\mathscr{F}_1 = \mathscr{F}(\alpha, K, \Sigma) \subseteq \Sigma$ be the set of favoring constraints for α in R relative to K, which we know to be non-empty by the Initial Favoring Lemma (66). Let $R_1 = [\mathscr{F}_1^{>>}][\Sigma_2]^R$ be a hierarchy where $\mathscr{F}_1^{>>}$ is any ranking over $\mathscr{F}_1$, and $[\Sigma_2]^R$ is a ranking of all the residual non-favoring constraints

$\Sigma_2 \subseteq \Sigma$ which respects their order in R. By construction, R_1 is the QFH^R of depth 1 for α over K and Σ in which the constraints in the residue Σ_2 are ranked according to their ranking in R. By the Favoring/Residue Lemma (74), we have $\alpha \in R(K) \Rightarrow \alpha \in R_1(K)$. Therefore, α is optimal for the QFH^R of depth 1.

[2] Now suppose that we have a QFH^R for α over Σ and K of every depth *m* for some $m \geq 1$. We have therefore achieved a ranking of the form $R_m = [\mathscr{F}_1{}^{>>} ... \mathscr{F}_m{}^{>>}]\ [\Sigma_{m+1}]^R$, for an $m \geq 1$, where $\alpha \in R_m(K)$. If $\Sigma_{m+1} = \varnothing$, we are done, because $\langle \mathscr{F}_1 ... \mathscr{F}_m \rangle$ then constitutes an exhaustive favoring hierarchy for α over K and Σ, and there can be none of greater depth.

Let us assume then that $\Sigma_{m+1} \neq \varnothing$. We need to show that R_m can be extended to a QFH^R of depth *m*+1, preserving the optimality of α. We know that $[\mathscr{F}_1{}^{>>} ... \mathscr{F}_m{}^{>>}](K) = K_{m+1}$. Therefore by Compositionality (54), we have

$$\begin{aligned}(*)\qquad R_m(K) &= [\mathscr{F}_1{}^{>>} ... \mathscr{F}_m{}^{>>}][\Sigma_{m+1}]^R (K)\\ &= [\Sigma_{m+1}]^R\ (\ [\mathscr{F}_1{}^{>>} ... \mathscr{F}_m{}^{>>}](K)\ )\\ &= [\Sigma_{m+1}]^R(K_{m+1})\ .\end{aligned}$$

Let us now gather all those constraints in Σ_{m+1} that favor α with respect to K_{m+1}. There must be at least one such, for nonnull Σ_{m+1}, because $\alpha \in R_m(K)$ is equivalent to $\alpha \in [\Sigma_{m+1}]^R\ (K_{m+1})$, and by Initial Favoring (66), $[\Sigma_{m+1}]^R$ must begin with a constraint that favors α over K_{m+1}. We now have a nonnull favoring stratum $\mathscr{F}_{m+1}$, which precisely accords with the definition of $\mathscr{F}_{m+1}$ in (80). Let $\Sigma_{m+2} = \Sigma_{m+1} - \mathscr{F}_{m+1}$. Then by the Favoring/Residue Lemma,

$$(**)\qquad [\Sigma_{m+1}]^R(K_{m+1}) = [\mathscr{F}_{m+1}{}^{>>}][\Sigma_{m+2}]^R(K_{m+1})$$

We now set

$$(***)\qquad R_{m+1}(K) = [\mathscr{F}_{m+1}{}^{>>}][\Sigma_{m+2}]^R(K_{m+1})$$

and unfolding K_{m+1}, we get

$$R_{m+1}(K) = [\mathscr{F}_1{}^{>>} ... \mathscr{F}_{m+1}{}^{>>}][\Sigma_{m+2}]^R\ (K)$$

R_{m+1} is therefore the QFH^R of depth *m*+1. Furthermore we have from (*) that $R_m(K)=[\Sigma_{m+1}]^R(K_{m+1})$ and from (**) and (***) $[\Sigma_{m+1}]^R(K_{m+1})=R_{m+1}(K)$, hence $R_m(K)=R_{m+1}(K)$, and R_{m+1} has precisely the same optima over K as R_m. Since $\alpha \in R_m(K)$, we have $\alpha \in R_{m+1}(K)$, as desired.

[3] From [1] and [2], it follows that there exists a QFH^R of every possible depth (up to exhaustion of Σ) with the same optima as R. In particular, the QFH^R construction terminates in a Favoring Hierarchy for α. This shows that $\alpha \in R(K)$ entails the existence of a favoring hierarchy for α that exhausts Σ. □

We may now proceed with the complementary theorem on Bounding, which associates every loser with a non-empty bounding set. For this purpose, we first prove as a lemma that $B^{Max}(\alpha)$ — i.e. the collection of all residual candidates that strict bound α on a residual constraint (see def. (40)) — is a bounding set for α relative to K and Σ. Let us first recall the relevant definitions.

(81) **Def. Set of Residual Constraints.** For any constraint set Σ, and favoring hierarchy $\mathcal{H}(\alpha)$ over Σ, the set of residual constraints Res(Σ) is formed by all and only the constraints in Σ but not in $\mathcal{H}(\alpha)$:

$$Res(\Sigma) = \Sigma - \{C: C \in \mathcal{H}(\alpha)\}$$

(82) **Def. Set of Residual Candidates.** For any constraint set Σ, candidate set K, and favoring hierarchy $\mathcal{H}(\alpha)=\langle \mathscr{F}_1 \ldots \mathscr{F}_n \rangle$ for α in K, the set of residual candidates Res(K) is formed by all and only the candidates in K that co-win with α on each favoring stratum $\mathscr{F}_i$:

$$\text{Res(K)} = [\mathscr{F}_1 \ldots \mathscr{F}_n](\text{K})$$

(83) **Def. Maximal Bounding Set.** For any constraint set Σ, candidate set K, and α in K, let $B^{Max}(\alpha)$ be formed by all and only those candidates in the set of residual candidates Res(K) that are strictly better than α on some constraint in the set of residual constraints Res(Σ):

$$B^{Max}(\alpha) = \{x : x \in \text{Res(K) and } \exists C \in \text{Res}(\Sigma)\ (x > \alpha; C^\wedge)\}$$

With these in hand, we may show that B^{Max} is indeed a bounding set.

(84) **B^{Max} Bounding Lemma.** For any constraint set Σ, candidate set K, and α in K, $B^{Max}(\alpha)$ constitutes a bounding set for α relative to Σ and K.

Pf. [1] B^{Max} satisfies Strictness, since by definition $\forall \beta \in B^{Max}$, $\exists C \in \text{Res}(\Sigma)$ $(\beta > \alpha; C^\wedge)$. .

[2] B^{Max} satisfies Reciprocity: let β be any member in B^{Max}, then:

(i) for any C in Res(Σ), if $(\alpha > \beta; C^\wedge)$ then by definition of residual constraint, since no residual constraint is favoring for α over Res(K), we have $\exists \gamma \in \text{Res(K)}$ $(\gamma > \alpha; C^\wedge)$. By hypothesis, $\gamma \in B^{Max}$, thus B^{Max} satisfies Reciprocity over Res(Σ). With this and [1] it follows that B^{max} is a bounding set for α over Res(Σ).

(ii) Now, for any C *not* in Res(Σ), it must be the case that $(\beta \approx \alpha; C^\wedge)$ by the Equivalence Lemma (61), because by hypothesis α and β are residual candidates, hence co-winners over $\mathcal{H}(\alpha)$. Reciprocity is thus vacuously satisfied over any $C \in \mathcal{H}(\alpha)$.

[3] Note finally that if B is a bounding set for α over some set of candidates $S \subseteq K$, then B is also a bounding set over K. (This holds because the definition of Bounding Set only mentions the elements of the bounding set and their relation to α; it doesn't matter what other candidates are or are not in the same set with α and the members of the bounding set.) Therefore, since B is bounding over Res(K), it is also bounding over $K \supseteq \text{Res(K)}$.

From [1], [2], and [3] it follows that B^{Max} is a bounding set for α relative to K and Σ. □

The Bounding Theorem now follows straightforwardly: B^{Max} in fact guarantees a non-empty bounding set whenever $\mathcal{H}(\alpha)$ does not exhaust Σ.

(85) **Bounding Theorem**. For any constraint set Σ and candidate set K, a candidate *z* in K is suboptimal on every ranking R over Σ iff there is in K a non-empty bounding set B(z) for *z*.

$$z \notin W(K,\Sigma) \Leftrightarrow B(z) \neq \varnothing$$

Pf. Let us start with the left-to-right entailment. By the Winner/Loser Theorem, the favoring hierarchy $\mathcal{H}(z)$ cannot exhaust Σ, hence $\text{Res}(\Sigma) \neq \varnothing$. By definition, each residual constraint C allows for a strict bound δ in Res(K) such that $(\ \delta > z; C^\wedge)$. By definition, all such δ are in B^{Max}, hence B^{Max} $\neq \varnothing$. Then, by the B^{Max} Bounding Lemma, B^{Max} constitutes a non-empty bounding set for *z* relative to K and Σ.

Let us now prove the reverse implication: $B(z)\neq\emptyset \Rightarrow z\notin W(K,\Sigma)$. For any ranking $R=[C_1...C_n]$ over Σ, let D be the *leftmost* constraint for which *z* loses against a member *b* of B(z), i.e. (b>z;D^). We know that such D exists, because by hypothesis B(z) is non empty, and given strictness, for any $x\in B(z)$ there is at least one constraint C on which (x>z;C^). It follows that R must have the shape R=HDX. Then by Inclusion, $R(K)\subseteq[HD](K)$. We now only need to prove that *z* loses on the initial section R′=HD, because shrinkage guarantees that once eliminated from the set of winners, *z* cannot reenter it.

Assume *z* does not lose on R′. Since (b>z;D^), it must be the case that for some higher ranked constraint C in H it is the case that (z>b;C^). But by hypothesis B(z) satisfies reciprocity, therefore there must exist $\beta\in B(z)$ such that (β>z;C^), contradicting the original hypothesis that D was the leftmost such constraint. Therefore $z\notin R'(K)$, thus by Inclusion $z\notin R(K)$, and since the above reasoning applies to any ranking R over Σ, it follows $z\notin W(K,\Sigma)$. □

5.2 Maximal and Minimal bounding sets

This final section is devoted to demonstrating the results on bounding sets. We begin by proving two lemmas about Reciprocity and Strictness that underlay our informal discussion of favoring hierarchies in §3.3. The first shows that non-residual candidates are inevitably nonmembers of any bounding set ('non-bounds') because they necessarily fail Reciprocity. The second shows that non-bound status also extends via failure of Strictness to any candidate that is weakly bounded by α on all residual constraint, i.e any candidate φ such that ($\alpha\geq\varphi$;C) for all $C\in Res(\Sigma)$.

On the basis of these two lemmas, we proceed to prove the uniqueness and maximality of the bounding set B^{Max}. We then prove the Blocking Set and Minimal Bounding Set theorems, showing how minimal bounding sets never use more than one residual candidate per residual constraint, and conclude by showing that covering sets always include a bounding set.

The Reciprocity- and Strictness-Failure Lemmas are proved *per absurdum*, showing how positing the existence of a bounding set containing the candidates at issue necessarily leads to a contradiction.

(86) **Reciprocity Failure Lemma**. For any candidate set K, any constraint set Σ, and any α, let $Q(\alpha)=[\mathscr{F}_1...\mathscr{F}_n]R$ be a quasi-favoring hierarchy for α exhausting Σ and with $R\neq\emptyset$. Then for any $\beta\in K$, if β is not a co-winner with α on $\mathcal{H}(\alpha)=[\mathscr{F}_1...\mathscr{F}_n]$, it is excluded from any bounding set $B(\alpha)$ relative to K and Σ.

$$\forall\alpha,\beta\in K,\ \ \beta\notin[\mathscr{F}_1...\mathscr{F}_n](K)\Rightarrow\forall B(\alpha)\ \beta\notin B(\alpha)$$

Pf.

[1] Assume there exists B, a non-empty bounding set for α over K and Σ, such that $\beta\in B$ for some non-residual $\beta\in K$, and let us call B′ the set of all non residual candidates like β in B. Obviously B′ is a subset of B and it is not empty, since it contains at least β.

[2] Let $\mathcal{H}^{>>}$ be any ranking refining $\mathcal{H}(\alpha)=[\mathscr{F}_1...\mathscr{F}_n]$. From the Winner/Loser Theorem, we have $\alpha\in\mathcal{H}^{>>}(K)$ and by Inclusion, α also wins over any initial section I of $\mathcal{H}^{>>}$.

[3] By hypothesis, $\beta \notin [\mathcal{F}_1 \ldots \mathcal{F}_n](K)$. Let D be the *highest ranked* constraint in $\mathcal{H}^{>>}$ such that for some β in B′, (α>β;D^). Note that D must exist, because B′ is non-empty, and since its members are not residual candidates they must have been eliminated on some such D.

[4] By hypothesis, B satisfies Reciprocity, hence there must be some β′∈B such that (β′>α;D^). But by [2] above, α∈I(K) for any initial section I=GD of $\mathcal{H}^{>>}$ ending with constraint D. This implies that β′ is strictly bound by α on some constraint D′ higher ranked than D in I, which in turn implies that β′ is a non-residual candidate in B′, contradicting the hypothesis that D is the highest constraint where α strictly bounds a member of B′. It follows that B cannot satisfy Reciprocity on β, contradicting the hypothesis that B is a bounding set. □

(87) **Strictness Failure Lemma**. For any candidate set K and any constraint set Σ, let $Q(\alpha)=[\mathcal{F}_1 \ldots \mathcal{F}_n]R$ be a quasi-favoring hierarchy for α exhausting Σ and with R a ranking over a non-empty set of residual constraints Res(Σ). Then, any β in K that does not strictly bound α on any of the residual constraints in R is excluded from any bounding set B(α) relative to K and Σ.

$$\forall \alpha,\beta \in K,\ \forall C \in R,\ \ (\beta \leq \alpha; C\hat{}) \Rightarrow \forall B(\alpha)\ \beta \notin B(\alpha)$$

Pf. Either β is a residual candidate in $\mathrm{Res}(K)=[\mathcal{F}_1 \ldots \mathcal{F}_n](K)$ or it is not. If it is not, the result follows directly from the Reciprocity Failure Lemma. Let us consider the case where β∈Res(K), and assume there exists a bounding set B including β among its members.

[1] By hypothesis, ∀C ∈ Res(Σ) it holds that (β≤α;C^), hence β does not satisfy Strictness on any constraint in Res(Σ).

[2] By definition of Res(K), β is a co-winner of α in $\mathcal{H}(\alpha)=[\mathcal{F}_1 \ldots \mathcal{F}_n]$. Therefore, by Equivalence, $\forall C \in \mathcal{H}(\alpha)$ (β≈α;C^), hence β cannot satisfy Strictness on any constraint in $\mathcal{H}(\alpha)$.

[3] Since there is no other constraints in Σ other than those in $\mathcal{H}(\alpha)$ and Res(Σ), it follows that β cannot satisfy Strictness, contradicting the hypothesis that B is a bounding set. □

The above two lemmas form the core of the Maximal Bounding Set Theorem, stated below, which asserts the maximality and uniqueness of the bounding set B^{Max}.

(88) **Maximal Bounding Set Theorem.** For any constraint set Σ, candidate set K, and α in K, $B^{Max}(\alpha)$ constitutes the unique maximal bounding set for α relative to Σ and K.

Pf. That $B^{Max}(\alpha)$ is a bounding set for α relative to K and Σ follows from the B^{Max}-Bounding Lemma (84).

[1] $B^{Max}(\alpha)$ is maximal. By definition, all residual candidates in Res(K) that strictly bound α on some constraint C∈Res(Σ) are in $B^{Max}(\alpha)$. It follows that any larger set B^+ necessarily includes either a residual candidate δ such that (α≥δ;C^) for every C in Res(Σ), or a non-residual candidate γ. The Strictness- and Reciprocity-Failure Lemmas ensure that B^+ fails Strictness on δ and Reciprocity on γ, thus not qualifying as a bounding set for α. Hence $B^{Max}(\alpha)$ is the maximal bounding set for α relative to K and Σ.

[2] $B^{Max}(\alpha)$ is unique. In order to avoid being a subset of $B^{Max}(\alpha)$, any maximal bounding set B′ distinct from $B^{Max}(\alpha)$ would have to include either a non-residual candidate γ, or a residual candidate δ such that (α≥δ;C^) for all C in Res(Σ). As shown in [2] above, this entails non-bounding status for B′, showing the uniqueness of $B^{Max}(\alpha)$. □

The next proof demonstrates that blocking sets —defined over a disfavoring constraint system Δ(α) by picking one strict bound for α on each constraint— form a bounding set relative to Δ. This result, independent from favoring hierarchies, is then used in the ensuing Minimal Bounding Theorem to show that minimal bounding sets are always subsets of some blocking set defined over the set of residual constraints Res(Σ). We first repeat the relevant definitions.

(89) **Def. Disfavoring System.** For any set of candidates K, a set of constraints S is a *disfavoring system* Δ(z) for a candidate *z*∈K iff on each constraint of S some element α in K strictly bounds *z*:

$$\Delta(z) = \{C : C\in S \ \&\ \exists\alpha\in K, \alpha>z \text{ in } C\}$$

(90) **Def. Blocking Set.** Let Δ(z) be a disfavoring system defined over some constraint set S and set of candidates K, then B^{Δ} is a blocking set for Δ(z) iff for each constraint C∈S the set B^{Δ} contains exactly one designated element α that strictly bounds *z* on C.

We now demonstrate that disfavoring systems give rise to blocking sets that are bounding sets over the constraints in the system.

(91) **Blocking Set Theorem.** For any candidate set K and disfavoring system Δ(z), every corresponding blocking set B^{Δ} constitutes a bounding set B(z) over Δ(z).

$$\forall\Delta(z),\ B^{\Delta} = B(z)$$

Pf. The blocking set B^{Δ} satisfies Strictness over Δ by definition, since each of its members is a strict bound of α for some C in Δ. Likewise B^{Δ} satisfies Reciprocity, since for any $\beta\in B^{\Delta}$ such that (z>β;C^) there must be by definition a designated member $\gamma\in B^{\Delta}$ such that (γ>z;C^). □

Before examining the theorem on Minimal Bounding, it is worth recalling that the property characterizing a blocking set is that each member is a *designated* bound for some constraint C, and that this property does not exclude the possibility of multiple bounds for each constraint; see the discussion of ex.(46) in §3.3. As for minimal bounding sets, remember that they may lack a strict bound for some of the disfavored constraints, as in ex.(53) in §3.3.

The proof of the theorem examines all a priori logically possible articulations of minimal bounding sets, rejecting some as yielding non-minimal sets, while showing all others to entail a subset relation with some blocking set.

(92) **Minimal Bounding Set Theorem.** For any candidate set K, constraint set Σ, and candidate α∈K, let B be a minimal bounding set for α over K and Σ, then B is a subset of some blocking set B′ over Res(Σ) and drawn from Res(K).

Pf. Let B be a minimal bounding set relative to K and Σ. By the Maximal Bounding Set Theorem, $B\subseteq B^{Max}(\alpha)$, and hence by definition of $B^{Max}(\alpha)$ the set B consists entirely of residual candidates in Res(K) that strictly bound α on some constraint C in Res(Σ). Let us consider all logically possible hypotheses over the structure of B:

[1] If B has a strict bound β for α for each $C\in Res(\Sigma)$, let B' be the set formed by picking for each C one such β from B. By construction, B' is a blocking set over $Res(\Sigma)$ and therefore a bounding set for α, given the Blocking Set Theorem. Either B' is a proper subset of B, in which case B is not minimal against the initial hypothesis, or $B'=B$, and hence $B\subseteq B'$ completing the proof.

[2] Else, for some constraint $D\in Res(\Sigma)$ the set B lacks a strict bound β for α. Only two alternatives are given:

[2.1] If every element of B is a strict bound for α each on a distinct residual constraint $C\neq D$, then let B' be the set formed by adding to B a strict bound β' for each constraint D for which B has no strict bound for α. Note that by definition of $Res(\Sigma)$ there must be such β' for each D. Then the newly formed set B' is a blocking set for $Res(\Sigma)$ over $Res(K)$, and by construction $B\subseteq B'$.

[2.2] Else, B necessarily contains some element γ which cannot be designated as the strict bound of any constraint because each constraint C_i where $(\gamma>\alpha;C_i\hat{})$ is already associated with some designated bound β in B for which $(\beta>\alpha;C_i\hat{})$.

But then B is not minimal, contradicting the initial hypothesis: the proper subset $S=B-\{\gamma\}$ is in fact a bounding set for α over Σ. Since each member of B satisfies Strictness, so does each member of its subset S. Moreover, S satisfies Reciprocity: since B is a bounding set and given the properties of γ examined above, $\forall\beta\in S$ if $\exists C\ (\alpha>\beta;C\hat{})$, then $\exists\beta'\in S$ such that $(\beta'>\alpha;C\hat{})$ and $\beta\neq\gamma$. Hence B cannot be both minimal and have the structure described above.

[3] Any minimal set B must thus have the structure of either point [1] or point [2.1], both of which were shown to entail $B\subseteq B'$, with B' a blocking set over $Res(\Sigma)$ and drawn from $Res(K)$. □

Our last theorem concerns covering sets, i.e. set of candidates each guaranteed to beat some element α on a specific ranking R of the $n!$ possible rankings of a constraint set of cardinality n. The theorem is reformulated in terms of relative harmony over a ranking. We show that any covering set COV(z) always includes a bounding set B(z). The theorem is re-stated from 12.

(93) **Covering Theorem**. Let $COV(z)\subseteq K$ be such that for every ranking R of Σ, there is an element $c\in COV(z)$, $(c\succ z;R\hat{})$. Then there is a non-empty set $B(z)\subseteq COV(z)$, with B(z) a bounding set for z:

$$\forall R\ \langle\Sigma,R\rangle,\ \exists\alpha\in COV(z),\ (\alpha\succ z;R)\Rightarrow\exists B(z)\neq\varnothing,\ B(z)\subseteq COV(z)$$

Pf. Let COV(z) be any set such that $\forall R$ on Σ, $\exists\alpha\in COV(z)$ with $(\alpha\succ z;R\hat{})$. Then COV(z) contains a bounding set for z. Consider $K=\{z\}\cup COV(z)$.

Construct the favoring hierarchy $\mathcal{H}(z)$ over K. This favoring hierarchy cannot exhaust Σ, else $\forall R'$ refining $\mathcal{H}(z)$, and $\forall\alpha\in COV(z)$, we have $(z\succ\alpha;R'\hat{})$, against the hypothesis that COV(z) contains a candidate better than for z on each possible ranking of Σ.

Then the Maximal Bounding Theorem ensures the existence of a maximal bounding set $B^{Max}(z)$ drawn from $Res(K)-\{z\}$, and since $Res(\Sigma)\neq\varnothing$, by definition $B^{Max}(z)\neq\varnothing$, since each residual constraint has at least one strict bound for z.

Since $Res(K)-\{z\}\subseteq COV(z)$, it follows $B^{Max}(z)\subseteq COV(z)$. □

References

Davey, B.A. & H.A. Priestley. 1990. *Introduction to Lattices and Order.* CUP: Cambridge.

Karttunen, Lauri. 1998. The Proper Treatment of Optimality in Computational Phonology. ROA-258. Rutgers Optimality Archive, http://ruccs.rutgers.edu/roa.html.

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

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. See also ROA-149.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. Linguistic Inquiry 27. ROA-118.

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

Tesar, Bruce. 1999. Overcoming Ambiguity in Language Learning. Ms. Rutgers University.

Tesar, Bruce & Paul Smolensky. 1995. The Learnability of Optimality Theory. *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*, 122-137.

Tesar, Bruce & Paul Smolensky. 1996ab. Learnability in Optimality Theory. ROA-155, ROA-156. (Short and long versions).

Tesar, Bruce & Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29.2, 229-268.

Tesar, Bruce & Paul Smolensky. In press. *Learnability in Optimality Theory*. MIT Press: Cambridge.

Appendix A. Relative Harmony as a Strict Order

We show here that relative harmony "$\succ$" is a strict order whose maximal elements are the optimal candidates of the ranking over which "$\succ$" is defined. Recall the definition of relative harmony:

(94) **Def**. **Relative Harmony**. For any candidates x,y, let K={x,y}. We say:

$x \succ y$ on a ranking R iff $x \in R(K)$ and $y \notin R(K)$.

For purposes of the proof we assume a constraint to be a *stratified* partial order. This allows to treat membership in the same stratum of a given constraint as an equivalence relation. If we make only the weaker assumption that a constraint is a partial order in which each subset has a maximal element, we can still define optimality successfully in the way we have done it, but the proposed definition of relative harmony no longer yields an order relation. To see this consider the following hierarchy $R = C_1 >> C_2$:

C_1^	C_2^
a ∥ x	b
|	|
b	x
	|
	a

We intend C_1^ to impose only *a*>*b* so that *x* is unordered with respect to *a* and *b*. Note that *x* is maximal in C_1^ as well as *a*. Consider now the fate of each pairing:

R({a,x})	=	x	'x>a'
R({a,b})	=	a	'a>b'
R({b,x})	=	b	'b>x'

Transitivity fails and we have no order relation. With stratification of the constraint orders, however, we are guaranteed that the definition of Relative Harmony produces an order, and a strict one.

(95) **Relative Harmony is a Stratified Strict Order**. Let $x,y,z \in K$. Let R be a ranking on some set of constraints Σ, and let "$\succ$" be the relative harmony relation. Then,

(a) "$\succ$" is irreflexive: $\neg (x \succ x)$,

(b) "$\succ$" is asymmetric $x \succ y \Rightarrow \neg (y \succ x)$

(c) "$\succ$" is transitive: $(x \succ y \ \&\ y \succ z) \Rightarrow (x \succ z)$.

(d) "$\succ$" is stratified: $\forall a,b,x \in K, \neg(a \succ b \vee b \succ a) \Rightarrow (a \succ x \Leftrightarrow b \succ x)$

Let us start by showing that "$\succ$" is a strict order, i.e. irreflexive and transitive. The proof of transitivity requires working through some details, but the idea is straightforward and can be seen clearly in the relevant comparative tableaux. Consider a tableau for $x \succ y$: it consists of a number, possibly zero, of blank cells, followed by a single box containing *x*, followed by a bunch of cells whose content is irrelevant.

$x \succ y$	.....	x	▨

The tableau for $y \succ z$ has the same form. Clearly these tableaux can match up in three different ways.

(i) The initial blank portions are identical.

$x \succ y$	....	x	▨
$y \succ z$	....	y	▨

Here we get $x > y$ and $y > z$ from the constraint whose column contains x and y. By definition of 'constraint', this is an order, so transitive, so $x > z$, and consequently $x \succ z$.

(ii) The blank portion for $y \succ z$ is longer.

$x \succ y$	....	x	▨		
$y \succ z$	....		....	y	▨

Here we have $x > y$ and $y \approx z$ on the heavily-outlined central constraint. Therefore, since constraints are assumed to be stratified orders, and since members of the same stratum share order properties, we have $x > z$, and so $x \succ z$.

(iii) The blank portion for $x \succ y$ is longer.

$x \succ y$	....		....	x	▨
$y \succ z$	....	y	▨		

Here we have $x \approx y$ and $y > z$ on the central heavily-outlined constraint, therefore x and y share order properties, so $x > z$, so $x \succ z$.

We now go on to spell out the algebraic details of this argument.

(96) **Relative Harmony is a Stratified Strict Order**. Let $x,y,z \in K$. Let R be a ranking on some set of constraints Σ, and let "$\succ$" be the relative harmony relation. Then

(a) "$\succ$" is irreflexive: $\neg (x \succ x)$
(b) "$\succ$" is asymmetric: $\neg ((x \succ y) \wedge (y \succ x))$
(c) "$\succ$" is transitive: $(x \succ y \ \& \ y \succ z) \Rightarrow (x \succ z)$
(d) "$\succ$" is stratified: $\forall a,b,x \in K, \neg(a \succ b \vee b \succ a) \Rightarrow (a \succ x \Leftrightarrow b \succ x)$

Pf. (a) Irreflexivity is immediate, since by the definition of "$\succ$", $x \succ x$ iff $x \in R(\{x\})$ and $\neg(x \in R(\{x\}))$, an obvious contradiction.

(b) Asymmetry. Clear because $x \in R(\{x,y\})$ and $\neg R(\{x,y\})$ are inconsistent.

(c) Transitivity. Suppose $x \succ y$. Then $x \in R(\{x,y\})$ and $y \notin R(\{x,y\})$. From the definition of optimality, there must be at least one initial section H of R, such that R=[HCG] and such that y is strictly bound by *x* in C, i.e. $x \in [HC](\{x,y\})$ and $y \notin [HC](\{x,y\})$. If there is more than one such section, choose H to be the shortest such. By construction, $x \in H(\{x,y\})$ and $y \in H(\{x,y\})$. By the Equivalence Lemma, $(x \approx y; C_i)$ for every C_i in H. We have now divided R into 3 parts: an initial section H, on which $x \approx y$, a constraint C on which they differ, with $(x>y;C)$, and a coda G in which their relations are undetermined (and immaterial to the argument).

Now suppose $y \succ z$ and repeat the analysis, yielding $R=[H'C'G']$, with $y \approx z$ on H', $(y>z;C'^{\wedge})$ and G' the uninformative coda.

Let us now investigate the possible relations of H to H' and C to C'.

First, suppose $H=H'$. In this case $C=C'$ and we have $(x>y;C^{\wedge})$ and $(y>z;C^{\wedge})$. Therefore by transitivity of $>_C$ we have $(x>z;C^{\wedge})$. Furthermore, since $x \approx y$ on all constraints in H and $y \approx z$ on all such constraints, we have $x \approx z$, by transitivity of $\approx$ in stratified hierarchies. Consider now the candidate set $K=\{x,z\}$. We must have $x,z \in H(K)$. Since $(x>z;C^{\wedge})$, we have $x \in [HC](K)$ and $z \notin [HC](K)$. By Inclusion, $x \in R(K)$ and $z \notin R(K)$. That is, $x \succ z$, as desired

Next suppose $H'=[HJ]$, $J \neq \emptyset$, i.e. suppose that H' is longer than H. In this case, we have $(y \approx z;C^{\wedge})$ because $y \approx z$ over all of H' and C is in H'. But since $(x>y;C^{\wedge})$ and $(y \approx z;C^{\wedge})$, we have $(x>z;C^{\wedge})$. Once again, since $x \approx y \approx z$ over H, C will be the first constraint that distinguishes x and z, favoring x. As before, this entails that $x \succ z$, as desired.

Finally, suppose that $H=[H'J]$, i.e. that H is longer than H'. Now we have $(x \approx y;C'^{\wedge})$ since C' falls inside H. But we have $(y>z;C'^{\wedge})$ and therefore because $x \approx y$, $(x>z;C'^{\wedge})$. As above, this leads immediately to the conclusion that $x \succ z$.

(d) Stratification. Let *a*, *b* in K be such that $\neg(a \succ b \vee b \succ a)$, which entails:

(*) $(a \preceq b \wedge b \preceq a)$.

Let *x* in K be such that $a \succ x$. Now consider the relation between *x* and *b*. First note that for any ranking R and $K=\{b,x\}$, R(K) has at least one winner, hence *b* and *x* are ordered with respect to each other in "$\succ$". It cannot be the case that $x \succeq b$, because in this case it would hold $a \succ b$ by transitivity on the strict order "$\succ$", against (*). Therefore, it must be the case that $b \succ x$. The reverse entailment $(b \succ x \Rightarrow a \succ x)$ follows along the same lines once *a* is switched with *b*. □

Now we show that optimal elements are maximal in the harmonic order and vice versa, under the definition of optimality we have given.

(97) **Optimality Maximality Lemma**. $\alpha \in R(K)$ iff α is a maximal element in the harmonic order $\langle K; \succ \rangle$ induced by R on K.

Pf. Optimality ⇒ maximality. Suppose not. I.e. suppose that $\alpha \in R(K)$ and there is an $x \in K$ such that $x \succ \alpha$. Therefore, $x \in R(\{x, \alpha\})$ and $\alpha \notin R(\{x,\alpha\})$. This latter condition means that there is some constraint C on which $(x>\alpha;C^{\wedge})$ so that $\alpha \notin C(\{x,\alpha\})$. Consider the highest ranked such constraint, C^h. In the initial section H of $R=[HC^hG]$, we cannot have any constraint on which $\alpha>x$, else $\alpha \succ x$. So we must have $x \approx \alpha$ on every constraint in H. Now, by assumption $\alpha \in R(K)$, so from the Inclusion Lemma, $\alpha \in H(K)$. Therefore $x \in H(K)$. So *both* x and α are in H(K). By the definition of constraint evaluation, we must have $\alpha \notin [HC^h](K)$, implying $\alpha \notin R(K)$. Contradiction.

Maximality ⇒ optimality. If α is maximal, then for every 2 element set $\{\alpha,x\}\subseteq K$, we have $\alpha\in R(\{\alpha,x\})$. Let ω be an optimal form on R, which by what was just shown must also be maximal. Now consider the candidate set $\{\alpha,\omega\}$. By maximality of α and ω, we have it that $\alpha,\omega\in R(\{\alpha,\omega\})$. By Equivalence, $\alpha\approx\omega$ on all constraints of R. Therefore α must also be optimal. □

Appendix B. The Relation between Simple and Collective Harmonic Bounding

The specific-to general relation between simple and collective harmonic bounding emerges in its full clarity as soon as we define both notions through more purely order-theoretic concepts, such as the following notion of a 'strict bound' for a set, which states that an element *b* 'strictly' bounds a set *S* when it is better than or order-equivalent to all the elements of S, and on top of that strictly better than at least one of them.

(98) **Def. Strict Bound for a Set**. An element $b\in K$ is a *strict bound* for $S\subseteq K$ iff

(1) b is a bound for S ($b\geq x$, for all $x\in S$), and

(2) $b>x$ for *some* $x\in S$.

Simple harmonic bounding may then be restated as shown below. Note that condition (1) below is equivalent to the original Strictness condition, demanding that $\beta>z$ on some constraint. Likewise, condition (2) is equivalent to the original weak-bounding condition, demanding that $\beta\geq z$ on all constraints.

(99) **Def. Harmonic Bounding**. A candidate $z\in K$ is *harmonically bounded* if there exists a candidate $\beta\in K$ meeting two conditions:

(1) **Strictness**. β is a strict bound for $\{z\}$ on at least one constraint in Σ.

(2) **Weak Bounding**. z is *not* a strict bound for $\{\beta\}$ on any constraint in Σ.

The generalization to bounding sets is immediate and requires nothing more than extending to a set B the same two conditions defining harmonic bounding.

(100) **Def. Bounding Set.** A set $B\subseteq K$ is a bounding set B(z) for $z\in K$ relative to a constraint set Σ iff B has these properties:

(1) **Strictness**. Every $\beta\in B$ is a strict bound for $\{z\}$ on at least one constraint in Σ.

(2) **Reciprocity**. z is *not* a strict bound for B on any constraint in Σ.

Condition (2) states that on any constraint $C\in\Sigma$ it should *not* be the case that (i) $\forall b\in B$, $(z\geq b;C)$, *and* (ii) $\exists b\in B$, $(z>b;C)$. This is equivalent to requiring for all C in Σ that *either* (i) $\exists b\in B$, $(b>z;C$^), *or* (ii) $\forall b\in B$, $(b\geq z;C$^), which in turn is equivalent to the original Reciprocity condition.