

Ranking and necessity: the Fusional Reduction Algorithm

Adrian Brasoveanu · Alan Prince

Received: 11 August 2009 / Accepted: 12 November 2009 / Published online: 31 August 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract Understanding a linguistic theory within OT requires an exact characterization of the ranking conditions necessitated by data. These conditions determine the formal shape of the grammar while providing the crucial link between the data and its interpretation. We introduce an algorithm ('Fusional Reduction', FRed) which calculates the necessary and sufficient ranking conditions inherent in any collection of candidates and presents them in a maximally concise and informative way. The algorithm, stemming from the original proposal of Brasoveanu 2003, is set within the fusional ERC theory of Prince 2002a. In this context, the Most Informative Basis and the Skeletal Basis emerge as the two important types of reduced representations of ranking structure. We examine their properties and show how FRed produces them from data. Fine-grained FRed is compared with broad-stroke RCD (Tesar and Smolensky 1993, Tesar 1995 *et seq.*), and RCD is reinterpreted and embraced within FRed as a simplified, information-losing sub-case. Finally, FRed is compared with other related algorithms in structure, worst-case complexity, and relevance to the analytical enterprise. This paper revises Brasoveanu and Prince 2005, 2007; Prince and Brasoveanu 2010 gives a more formal perspective, with proof of the theorems.

Keywords Optimality Theory · Ranking · ERC · Fusional Reduction · RCD

A. Brasoveanu (✉)
UC Santa Cruz, Santa Cruz, USA
e-mail: abrsvn@ucsc.edu

A. Prince
Rutgers University, New Brunswick, USA
e-mail: prince@rucss.rutgers.edu

1 Beyond the sufficient

With a constraint set in hand, the first order of analytical business in OT is to match candidates with violation profiles. One of two problems then arises, depending on what is known (or assumed) about the candidates and the constraint hierarchy.

The Selection Problem. If a ranking has been imposed, the problem is to determine which candidates come out as optimal under that ranking.

The Ranking Problem. If the candidates desired to be optimal have been identified, the problem is to find the rankings, if any exist, that render them optimal.

The Selection Problem arises when prior analysis or prior assumptions have fixed a ranking for the constraint set. Selection of optima can be obtained by a simple repeated process of filtration, conducted under the rubric “take the best, ignore the rest” (as Gigerenzer and Goldstein 1996 sharply phrase it; see also Gigerenzer et al. 1999). The best overall must lie among the best on the top-ranked constraint; and must lie among the best of *those* as evaluated by the next-highest-ranked constraint, and so on down the hierarchy, selecting the best of the best, until the optimal violation profiles emerge.

The Ranking Problem arises when the desired optima are known, as from empirical observation and analysis, but the rankings they require are yet to be discovered. The task is different in character from selection and rather more complex. It is based on comparisons of relative goodness between grammatical and ungrammatical forms—between desired optima and their suboptimal competitors. Any such comparison yields a pattern of successes, failures, and moot contests on individual constraints. Typically, only certain rankings will be consistent with a choice of desired optima. The partial information gleaned from each of many instances of comparing optimum *vs.* competitor must be integrated to produce the definitive set of ranking conditions imposed by the data. Taken together, these conditions delimit the set of rankings—total orders on the constraint set—that yield the observed grammatical forms and predict the shape of the language beyond the data considered.

In brief, selection goes from ranking to optima, and presents little difficulty. The Ranking Problem runs the other way, from (desired) optima to ranking, and has a logic of its own. The analyst faces the Ranking Problem every working day. The tools provided in this paper will, for the first time, resolve it in full generality.

To illustrate these problems, and their distinctness, let’s look at an example based on the Lardil analysis worked out in Prince and Smolensky 2004: 140ff. We have renamed the constraints to accord with more recent usage,¹ and we have omitted the constraints that award all candidates the same number of violations. The special non-phonetic notations in the candidate transcription are ‘A’ for ‘inserted [a]’ and ‘■’ for ‘deleted segment’. They are intended only to make the key input–output disparities obvious to the eye. Syllables are marked off by periods.

¹DEPV: a vowel in the output corresponds to one in the input. MAX: a segment in the input corresponds to one in the output. NOCODA: no syllable ends in a consonant. ANCHOR-R: a segment at the right edge of a morphological word has a correspondent at the right edge of a syllable. These constraints assess one violation for each configuration of which they are false. These rework, respectively, the original FILL^{Nuc}, PARSE, –CODA, and ALIGN.

(1) **The Selection Problem: Ranking given.**

Lardil /kentapal/ → [???'dugong' (*dugong dugon*: aquatic mammal species)

/kentapal/ →	DEPV	ANCHOR-R	MAX	NOCODA
.ken.ta.pal.				**
.ke.nA.ta.pa.■	* !	*	*	

The left-to-right order of the constraints mirrors their actual ranking (as shown in Prince and Smolensky 2004: 132–138). The desired optimum beats its competitor definitively on DEP V, its suboptimality due to epenthesis: a victory (and a defeat) marked by the appearance of the exclamation point.

To see a less trivial filtration, we need to have the optimum face off against more than one competitor. Here’s an example, expanded from (1):

(2) **More elaborate selection.** Given this ranking, Lardil /kentapal/ → [???'dugong'

/kentapal/ →	DEPV	ANCHOR-R	MAX	NOCODA
.ken.ta.pal.				**
.ken.tap.al.				*** !
.ken.ta.pa.■		* !	*	*
.ke.nA.ta.pa.■	* !	*	*	

An exclamation point marks where a losing candidate is ejected from the set of ‘best’ candidates, given this ranking. Shading emphasizes that the candidate is no longer in the filtered set.

Matters look quite different when we start with knowledge that [.ken.ta.pal.] is the output from /kentapal/ and assume nothing about the constraint ranking. Let’s take the same two violation profiles as in (1) and annotate the tableau according to the *relative* performance of the competitors on each constraint. The desired optimum is listed first and candidates desired to be suboptimal are indented below it. The symbol ‘W’ generalizes ‘!’ and indicates that the desired optimum is locally better on the one constraint by virtue of having fewer violations than its suboptimal competitor. The new symbol ‘L’ indicates that the *suboptimum* does better on the constraint than the desired optimum. Only the suboptimum’s row is annotated, for the violation profile of the desired optimum functions as a kind of yardstick against which everything else is measured. We switch to numerals as violation markers to avoid the distractions of visual splatter.

(3) **The Ranking Problem: Optimum given.** Lardil /kentapal/ → [.ken.ta.pal.] ‘dugong’

/kentapal/ →	DEPV	ANCHOR-R	MAX	NOCODA
ken.ta.pal	0	0	0	2
ke.nA.ta.pa.■	1 W	1 W	1 W	0 L

This is an exact map of what the constraint set says about these data. The comparison shows that NOCODA (displaying L) must be subordinated in the ranking, because it

favors the suboptimum, which had better do worse than the optimum over the hierarchy. It follows that NOCODA must be dominated by some constraint of opposite polarity that favors the desired optimum—and we have three different choices (each marked W). Comparing output [.ken.ta.pal.] with * [.ke.na.ta.pa.] gives us no grounds for choosing among the three potentially dominating constraints, and yields, therefore, quite weak information about what the hierarchy must be. Just comparing these competitors, we have no idea whether * [.ke.na.ta.pa.]’s avoidance of [.ken.ta.pal.]’s codas is rejected because (1) you can’t insert to avoid codas, (2) you can’t delete to avoid them, or (3) you must preserve word-final consonants. To decide the matter, further evidence must be taken into account. Other comparisons will impose tighter conditions, which make it clear that the first of these is the correct and only explanation for the failure of * [.ke.na.ta.pa.]’s avoidance of [.ken.ta.pal.]’s codas.

With a collection of such comparisons in hand, we may ask for a characterization of the set of successful rankings, those that give the data in its entirety. The answer we will be satisfied with varies with our goals. It is commonly the case that a number of different rankings will suffice—after all, many constraints simply do not conflict, many others lose the ability to conflict when they are subordinated in certain ways, and ranking serves to resolve conflicts. (Example: those constraints that are fully satisfied in every optimal form can be ranked in any order whatsoever at the top of the hierarchy.) The *learner*, for whom the grammar is a black box, is looking for a ranking sufficient to produce correct output and will be happy with any such.² The data modeler might be similarly satisfied, and need know nothing more than that a grammar exists, happy that the grammar (whatever it is) reproduces the observed facts. But a more ambitious analyst will want to know the details of the constraint relations and their factual basis. Such information determines the explanatory structure of a theory, which relates data to constraints and their interaction. And it is crucial to the dynamics of investigation and theory construction. Modifying constraint definitions, adding new constraints, amalgamating or subdividing constraints—the success of any of these theory-developing maneuvers depends on the details of the interactions. The serious analyst, then, will be interested in precisely delimiting the data-theory relation: the necessary and sufficient conditions imposed by the data. Because of the inherent logic of the Ranking Problem, it need not be entirely straightforward to obtain them.

As Tesar found, we can easily derive a sufficient ranking from a collection of comparative data if we ask not what constraints *must* be ranked at the top but what *can* be ranked there. We gather all those rankable constraints into a ‘stratum’, dismiss the data they solve, and proceed to re-ask the question of the remaining, smaller set of data, obtaining a new, lower stratum. We continue in this fashion until all the data is accounted for, yielding an ordered collection of strata. The stratification procedure is known as ‘Recursive Constraint Demotion’ (RCD) and a grammar derived from it will work, if any grammar works.³ Since the constraints within a stratum do not

²Modulo *restrictiveness*, as reviewer A reminds us: see Prince and Tesar 2004, Hayes 2004 for recent proposals. Observe that even with restrictiveness as a limitation on grammar choice, many alternative rankings may still exist.

³Basic references include Tesar and Smolensky 1993, 2000; Tesar 1995; see also Samek-Lodovici and Prince 1999; Prince 2002a, 2002b. It is used in OTSoft (Hayes et al. 2004), and an Excel™-based

conflict, we can construct a successful grammar (a total ordering of the constraint set) which will select optima by picking any ranking order for the constraints within each stratum and patching these segments together so as to respect the stratal order. The trade-off is that while we have sufficiency, we have abandoned necessity.

A simple formal example brings us to the heart of the matter. Imagine that the entire constraint set has only three members, and suppose that applying RCD to the data of a particular language delivers up the following two-stratum hierarchy. (A double pipe demarcates the strata.)

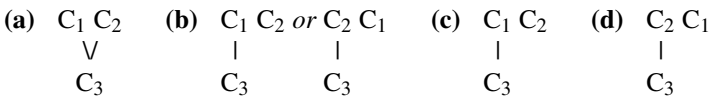
(4) **A stratified hierarchy.** $\parallel C_1, C_2 \parallel C_3 \parallel$

We know that C_1 and C_2 can be ranked at the top, and that C_3 cannot. No fewer than four distinct ranking restrictions, driven by different data, could lie behind this one stratified hierarchy:

- (a) Both C_1 and C_2 must dominate C_3 .
- (b) Either C_1 or C_2 must dominate C_3 .
- (c) C_1 must dominate C_3 , but C_2 can go anywhere.
- (d) C_2 must dominate C_3 , but C_1 can go anywhere.

These different situations—each quite common in the natural world—can be perspicuously summarized as (sets of) partial ordering diagrams. In each case, any linear order that respects the specified ranking restrictions, shown by the lines, will count as legitimate realization.

(5) **Rankings behind the RCD stratification.** $\parallel C_1, C_2 \parallel C_3 \parallel$



In all these mini-grammars, each arising from different data, C_3 must be subordinated or the wrong candidate will be designated as optimal. But they differ in what C_3 must be subordinated to. In (b), for example, the right outcome is achieved if C_1 or C_2 comes ahead of C_3 in a ranking, but in (c) it must absolutely be the case that C_1 precedes C_3 in a successful hierarchy. Of these, the latter three induce greediness artifacts in the RCD output $\parallel C_1, C_2 \parallel C_3 \parallel$. In each, some constraint will land in the top stratum because it can be there, not because it must.

The four situations differ in their explanatory structure and in the mutability of the constraints involved. In case (c), for example, the violability of C_3 is due entirely to the possibility of ‘better satisfying’ C_1 , and it is appropriate to say that C_1 ‘forces’ the violation of C_3 or that only C_1 ‘licenses’ C_3 violation. But this is not true of any of the other patterns. In cases (b) and (c), the constraint C_2 is wide open to reformulation, but not in the other systems.

To see the force of such considerations, imagine that each of the four mini-systems is confronted with further data indicating unambiguously that $C_3 \gg C_2$. Systems (a)

implementation is available in OTWorkplace (Prince 2008c; Prince and Tesar 2008). Prince 2009 offers an accessible walk-through of the algorithm, from a perspective closely allied to the one developed here.

and (d) are immediately kaput—they simply can't accommodate the facts, since each requires that $C_2 \gg C_3$. System (c) is refined by an additional ranking, which it handles uncomplainingly, while system (b) loses a disjunct and takes on the same form as system (c)— $C_1 \gg C_3 \gg C_2$. This shows that neither the learner nor the analyst can settle for the RCD form of available data, at the risk of grossly miscalculating its import.⁴

In this paper, we present and justify an algorithm—'Fusional Reduction', FRed—that pulls out the necessary and sufficient ranking conditions inherent in a set of data and puts them in a canonical form that is maximally concise and informative. The algorithm originates in Brasoveanu 2003; here it is revised and extended, in the setting of the theory of Prince 2002a. 'Maximally concise' means that there is no other presentation that uses fewer comparative expressions. Among such maximally concise ranking summaries, of which there may be several, FRed will natively produce the one that is 'maximally informative' in the sense that each element in the description gives complete information about dominated constraints. This form we will call the 'Most Informative Basis'; the notion 'informative' will be made precise for this usage. From it, another equally concise representation will be derived, one which describes the ranking relation with maximum locality, omitting mention of all dominated constraints whose status can be derived collectively. This is the 'Skeletal Basis', which directly maps ranking diagrams such as those in (5) and provides the direct link between data and the rankings they require. Each will have its uses in the analysis of OT systems. Through FRed, then, we will gain a complete algorithmic solution to the Ranking Problem, understood in its most demanding form. FRed has been implemented, along with a number of other analytically useful calculations, as part of the RUBOT component of OTWorkplace, an open-source software package that runs inside Excel™ (Prince 2008c; Prince and Tesar 2008).

Such matters may be addressed at various levels of technical detail. In the interests of accessibility, we will steer clear of the finest grain of formal argument. Prince and Brasoveanu 2010 gives proofs and deeper analysis of all our claims. Here the focus will be on the qualitative content of the notions involved and the step-by-step procedures involved in implementing them. We believe that it is both possible and useful for workers in OT (and perhaps for workers in other theories as well, who may be interested in reaching similar levels of analysis of their own data-theory relation) to see how things work at the level we pursue here. Beyond usefulness lies logical compulsion: when the facts can be obtained, it is time to relinquish heuristics, guesswork, and the assertion, however tentative, of falsehoods. We will provide explicit definitions for everything we talk about, usually appended to the discursive presentation, for those who wish to examine them. No prior background in the relevant material will be assumed.

The requisite notional underpinnings will be developed in full in the next section (Sect. 2). With these tools in hand, we move to an exposition of the FRed algorithm and its output, the Most Informative Basis (Sect. 3). FRed is then refined so as

⁴Hence Tesar 1997a, 1997b finds that the set of ranking arguments, the *ERCs* in the terminology to be adopted here, and not an artifact-susceptible intermediate ranking, is what must be carried forth in learning. Tesar and Prince 2005 term this fund of irreplaceable knowledge the 'Support' for the ranking system.

to also produce the Skeletal Basis, whose importance we justify (Sect. 4). We then place FRed with respect to RCD, with the aim of developing further insight into both (Sect. 5). We conclude with some notes on computational complexity and on the relationship of FRed to other work in the area, such as the algorithm for finding bounding minima in Samek-Lodovici and Prince 2005 and the ERC set simplification algorithm of Prince 2002a: 31–34.

2 The consequences of comparison

In this section we review the logic of pairwise candidate comparison that underlies the entire calculus of ranking argumentation in OT. We draw on Prince 2002a, *Entailed Ranking Arguments*, henceforth *ERA*.

2.1 The Elementary Ranking Condition

Extracting ranking information from even a single pairwise comparison need not be trivial. Each individual constraint awards one of three evaluations to a competing pair:

- W the desired optimum is better, with fewer violations than its competitor
- L the *sub*optimum is better, with fewer violations; the desired optimum loses to it
- e neither is better, because the two are equal in violations

We will say that a constraint ‘distinguishes’ two candidates if it awards one of the ‘polar’ values W or L, and not the ‘neutral’ value e, to their comparison. A typical result, seen in another Lardil example (Prince and Smolensky 2004: 140), looks like this (as above, the suboptimal competitor is indented below the desired optimum):

(6) **Violation tableau with comparative annotations:** /ŋaluk/ → [.ŋa.lu.]

/ŋaluk/ →	CODACOND	DEPV	ANCHOR-R	DEPC	MAX	NOCODA
.ŋa.lu.■	0	0	1	0	1	0
.ŋa.luk.	1 W	0 e	0 L	0 e	0 L	1 W

The constraint CODACOND,⁵ unmentioned in (3) because neutral and therefore irrelevant, is polar here; we have also brought in a couple of neutral constraints for illustrative purposes. Let us emphasize a key notational fact: in this paper, constraint order in a tableau header does not represent a ranking order.

(7) **Notational convention: constraint listing in tableaux.**

Since we are dealing with the Ranking Problem, we will *never* presuppose, henceforth, that the order of cited constraints in a tableau reflects a ranking order, nor will we distinguish in vertical line style between ranked and unranked constraints.

⁵CODACOND: A coda consonant can have only Coronal place or place shared with another consonant.

A tableau like (6) can be reduced to its core by suppression of the violation profiles, which have done their work when the comparative values are calculated from them. The result is a pure ‘comparative tableau’ (CT).⁶

(8) **Comparative tableau (CT).**

/ŋaluk/ →	CODACOND	DEPV	ANCHOR-R	DEPC	MAX	NOCODA
.ŋa.lu.■ ~ .ŋa.luk.	W	<i>e</i>	L	<i>e</i>	L	W

The match of a desired optimum with its competitor is concisely indicated here.

(9) **Notational convention: optimum~suboptimum comparisons.**

We write ‘ $q \sim z$ ’ to denote the comparison of a desired optimum q with its competitor z . We will suppress the repeated input term whenever convenient.

For example, in (8) above, we are actually comparing q : (ŋaluk→.ŋa.lu.■) with z : (ŋaluk→.ŋa.luk.). These are two competing input–output relations based on the same input.

Each row detailing the evaluations of some $q \sim z$ comparison over the entire constraint set is associated with a ranking restriction, an ‘**Elementary Ranking Condition**’ or ERC, that always has the same logical form. In CT (8), for example, the requirement is this:

either CODACOND or NOCODA dominates both ANCHOR-R and MAX.

Every ranking in which the desired optimum (ŋaluk→.ŋa.lu.) is judged better than the competitor (ŋaluk→.ŋa.luk.) will satisfy this condition. An ERC proclaims that *some* constraint assessing **W** must dominate *all* constraints assessing **L**.

The ERC is a direct consequence of the way that optimality is defined. Recall that this requires two *distinct* ‘better than’ relations (Prince and Smolensky 2004: Chap. 5; Prince 2009; see exs. (16), (17), (18) below for explicit formulations). A form is optimal when it is *better than* all of its violation-distinct competitors *over the whole hierarchy*. This global ‘better than’ in turn derives from the local, violation-sensitive judgments of the individual constraints, as represented in the **W-L-*e*** values. In terms of a single constraint, *better than* means ‘incurs fewer violations’. Globally, one candidate is better than another *over the whole hierarchy* when it is locally better on the highest-ranked constraint that distinguishes them (to use the concise formulation of Grimshaw 1997). This leads to the ERC: the comparative profile, in **W-L-*e*** values, determines which constraints could possibly serve in the decisive role as the highest-ranked distinguishing constraint (any of those assessing **W**), and which constraints must absolutely be banned from that position (all of those assessing **L**). The comparative profile also identifies those constraints that are completely irrelevant to the status of the competitors, those neutrals assessing *e*.

⁶For the CT per se, see Prince 1998, 2000, 2002b. The relevant logic, presented here, is developed in ERA. Demonstrations of its utility in analyzing OT systems are found in Prince 2006b.

The essence of the comparative tableau lies in the sequences of values it assigns to comparisons. Let us recognize the relevant part of the tableau row as a distinct entity, a ‘comparative vector’. Such a vector is a list of W-L-*e* values. Position in the list corresponds to an enumeration of the constraint set. The enumeration is arbitrary and serves only to keep track of the constraints, in much the same sense as the sequence of entries in a vector of Cartesian coordinates keeps track of the spatial dimensions, which have no intrinsic ordering. Following the standard Cartesian lingo, the entries in a vector will be called its ‘coordinates’. For example, the comparative tableau (8) displays a single vector, which we can write in the following compressed form:

(10) **Comparative vector:** (W, e, L, e, L, W)

Each such vector is associated with an ERC, a logical expression. To represent the ranking relations imposed by a set of data, we take the ERC as the basic unit of description. We will see how certain operations on the coordinates of comparative vectors (i.e. the columns of comparative tableaux) compute the logical properties of ERCs and ERC sets. These will enable us to define the algorithm which takes a collection of vectors and renders it into its most compact, informative equivalent.

The ERC intuitively says ‘some W dominates all L’s’. In the interests of explicitness, let us formally define it here using the notion of the W-set of a vector **a**, written $W(\mathbf{a})$ —those constraints contributing W to **a**—and the parallel notion of its L-set, $L(\mathbf{a})$.⁷

(11) **Elementary Ranking Condition** ($\forall\exists$ form).

Let $\mathbf{a} = [x \sim y] \in \{W, L, e\}^n$ be a comparative vector associated with Σ_n , an arbitrary enumeration of the constraint set Σ . Writing

$$W(\mathbf{a}) = \{C_k \in \Sigma \mid \mathbf{a}[k] = W\} \text{ ‘those constraints assessing W of } \mathbf{a}\text{’}$$

$$L(\mathbf{a}) = \{C_k \in \Sigma \mid \mathbf{a}[k] = L\} \text{ ‘those constraints assessing L of } \mathbf{a}\text{’}$$

The ERC associated with **a** is this:

$$\forall C \in \Sigma \exists D \in \Sigma (C \in L(\mathbf{a}) \supset (D \in W(\mathbf{a}) \ \& \ D \gg C))$$

‘Every constraint assessing L is dominated by some constraint assessing W.’

(12) **Elementary Ranking Condition** ($\exists\forall$ form). Using the same notation,

$$\exists D \in \Sigma \forall C \in \Sigma (C \in L(\mathbf{a}) \supset (D \in W(\mathbf{a}) \ \& \ D \gg C))$$

‘Some constraint assessing W dominates all constraints assessing L.’

The ERC provides a new perspective on the Cancellation-Domination Lemma of Prince and Smolensky 2004: 153–154, which states that a candidate A is globally better than an alternative B iff “every mark incurred by A is either (i) cancelled by an identical mark incurred by B, or (ii) dominated by a higher-ranking mark incurred

⁷For material implication (ordinary Boolean ‘if-then’), we use the Russellian symbol \supset ‘horseshoe’, currently waning in popularity, because we will use \Rightarrow and \rightarrow for other notions below.

by B.” Shifting focus from marks to constraints: A is globally better than B iff each constraint evaluating $[A \sim B]$ either (i) assesses W or e , or, if not, (ii) is dominated by another constraint, which assesses W. This will be the case iff ‘some W dominates all L’s’.

The syntactically distinct forms of the ERC given in (11) and (12) are equivalent because the linear ordering of the constraint hierarchy means that one of the W-assessors must dominate all the others. If every constraint assessing L is dominated by some W-assessor ($\forall \exists$) then it follows, by linearity of the ranking, that one of the W-assessors dominates all the L’s ($\exists \forall$). We use the some–all form ($\exists \forall$) in (12) as our standard.

The truth or falsity of an ERC is determined with respect to a ranking on the constraint set. An ERC is about rankings. A constraint hierarchy is then a *model* in which an ERC is judged true or false. When an ERC is true of some hierarchy H, we will say, following the usual usage, that the hierarchy H *satisfies* the ERC. If all ERCs in a collection are true of a hierarchy, we will say that the hierarchy satisfies the entire ERC set.

The force of the ERC associated with the comparison $x \sim y$, then, is that x is *better than* y on H if and only if H satisfies the associated ERC. From the selectional point of view, a totally-ordered hierarchy that satisfies an ERC is one that successfully dismisses the competitor desired suboptimal. Similarly, if every member of a set of ERCs is simultaneously satisfied by a hierarchy, then that hierarchy guarantees that every comparison behind the ERC set comes out in favor of its desired winner.

Since ERCs and ERC vectors stand in a one-to-one relation, let us simplify the discussion by extending the usage of ‘satisfies’ to encompass both entities. We will speak of a hierarchy *satisfying* an ERC *vector*, where this means that the associated ERC, a logical formula, is satisfied. It is possible to make this usage central, rather than extended, through a direct model-theoretic account that gives the ERC vector the status of an expression in a formal language, directly interpreted. Since it is useful to have the ERC available as an expression in logical syntax, we will defer this option.

To show the force and utility of ERC representation, we return to the four-ways ambiguous RCD hierarchy cited in (4) and show how the various cases come out as expressed in ERC vectors:

- (13) **Rankings behind the RCD stratification.** $\parallel C_1, C_2 \parallel C_3 \parallel$
- | | |
|--|----------------------------|
| (a) Both C_1 and C_2 must dominate C_3 . | $\{(W, e, L), (e, W, L)\}$ |
| (b) Either C_1 or C_2 must dominate C_3 . | (W, W, L) |
| (c) C_1 must dominate C_3 , but C_2 goes anywhere. | (W, e, L) |
| (d) C_2 must dominate C_3 , but C_1 goes anywhere. | (e, W, L) |

Expressions like these are ubiquitous in the analysis of language data.

2.1.1 Appendix to 2.1

We conclude with a concise record of the notions referred to in deriving the ERC.

- (14) **Constraint.** A constraint assigns violations to candidates; thus, a constraint is a function from the universal set of candidates U to the nonnegative integers, i.e. $C:U \rightarrow \mathbb{N}$.
- (15) **Constraint hierarchy.** Given a set $\Sigma = \{C_1, \dots, C_n\}$ of constraints, a grammar or ‘constraint hierarchy’ is a total order, written \gg , on the constraint set.
- (16) **‘Better than on a constraint’.** For $x, y \in U$, and for a constraint $C \in \Sigma$, we say ‘ x is better than y on C ’, writing $x \succ_C y$, iff $C(x) < C(y)$. In this case, we also say ‘ x is locally better than y ’.
- (17) **‘Better than on a constraint hierarchy’.** For H , a total ordering of Σ , and $x, y \in U$, we say ‘ x is better than y on H ’, writing $x \sqsupset_H y$, iff (a) $\exists C \in \Sigma$ such that $x \succ_C y$; and (b) $\forall D \in \Sigma$ if $D \gg C$, then $D(x) = D(y)$, or equivalently, (b’) $\forall D \in \Sigma$ if $y \succ_D x$, then $C \gg D$. That is: there is a constraint distinguishing x and y , and the highest-ranking such constraint favors x . In this case, we also say ‘ x is globally better than y ’. We use a different symbol for the two different ‘better than’ relations.
- (18) **Optimal.** For $x \in K$, a candidate set, and a constraint hierarchy H , x is ‘optimal’ on H (and in K) iff $\neg \exists y \in K$ such that $y \sqsupset_H x$. ‘No candidate is globally better’.
- (19) **Comparative values.** A function \hat{C} is associated with each constraint C , assigning to an ordered pair of candidates $(x, y) \in U \times U$ the following values: $\hat{C}[x \sim y] = \text{cmp}(C(y) - C(x))$, where $\text{cmp}(n) = W$ if $n > 0$, e if $n = 0$, L if $n < 0$. This transforms $C:U \rightarrow \mathbb{N}$ into $\hat{C}:U \times U \rightarrow \{W, L, e\}$.⁸
- (20) **Comparative vector.** Let Σ be an arbitrary enumeration of a constraint set, so that $\Sigma = \{C_1, \dots, C_n\}$. A comparative vector (or ERC vector) \mathbf{v} over Σ with respect to an ordered pair of candidates $x, y \in U$ has coordinates given by $\mathbf{v}[k] = \hat{C}_k([x \sim y])$.
- (21) **W-set. L-set.** The W-set of a comparative vector \mathbf{v} , written $W(\mathbf{v})$, is the set of constraints that give rise to W coordinates in \mathbf{v} . Similarly for the L-set.

$$W(\mathbf{v}) = \{\hat{C}_k \mid \mathbf{v}_k = W\}$$

$$L(\mathbf{v}) = \{\hat{C}_k \mid \mathbf{v}_k = L\}$$

- (22) **Elementary Ranking Condition (ERC).** Each comparative vector gives rise to an ERC defined as in (11) and (12).
- (23) **Satisfying an ERC.** An ERC φ over a constraint set Σ is satisfied by linear ranking order H on Σ iff the ERC is true of H . In that case, we write $H \models \varphi$.

⁸It is natural to identify ‘cmp’ with ‘sgn’, the function that returns +1 for positive arguments, 0 for 0, and -1 for negative arguments (*ERA*: 55ff., 100ff.) This portrays a comparative vector as a vector over $GF(3)$, the Galois (i.e. finite) Field with 3 elements, where in fact the relations and operations defined in the next section can be stated algebraically. See fn.15. In the interest of retaining a palpable connection with the winner–loser structure of OT, we will use the standard symbols W, L, e .

- (24) **Satisfying a vector.** Given a hierarchy H and an ERC vector \mathbf{v} , we will also say that H satisfies \mathbf{v} , writing $H \models \mathbf{v}$, iff H satisfies the ERC associated with \mathbf{v} ; that is, iff H is such that some constraint in $W(\mathbf{v})$ dominates every constraint in $L(\mathbf{v})$.

2.2 Informativeness and the basis

Identical rankings may be demanded by different ERC sets. The simplest nontrivial case involves only three constraints, linearly ordered, and shows quite clearly how this can happen. We use bolded numbers **1, 2, 3**, etc. to index constraints, and bolded alphabetic characters **a, a', b**, etc. to name ERC vectors. The same character but bracketed [**a**], [**a'**], etc., names the associated ERC.

- (25) **Equivalent ERC sets:** $1 \gg 2 \gg 3$

CT ₁	1	2	3	CT ₂	1	2	3	CT ₃	1	2	3
a	W	L	L	a'	W	L	<i>e</i>	a''	W	L	W
b	<i>e</i>	W	L	b	<i>e</i>	W	L	b	<i>e</i>	W	L

Taken individually, the first-row vectors **a**, **a'**, and **a''** are *not* equivalent, as is evident from their associated ERCs.

$$\begin{aligned}
 \mathbf{a}: & \quad (W,L,L) & [\mathbf{a}]: & \quad 1 \gg 2 \ \& \ 1 \gg 3 \\
 \mathbf{a}': & \quad (W,L,e) & [\mathbf{a}']: & \quad 1 \gg 2 \\
 \mathbf{a}'': & \quad (W,L,W) & [\mathbf{a}'']: & \quad 1 \gg 2 \ \vee \ 3 \gg 2
 \end{aligned}$$

Nonetheless, in the context of vector **b**, which requires $2 \gg 3$, these three all yield exactly the same conclusion: sets CT₁, CT₂, and CT₃ are satisfied by one and the same linear order $1 \gg 2 \gg 3$ on the constraint set, and by no other.

Further collections of ERCs can yield the same result: for example, the union of all three cited sets; the union of any two of them; numerous other admixtures and augmentations. We may freely add in vectors which consist solely of W's and *e*'s: in such cases, no constraints disfavor the desired winners, and no ranking is required to make them win; such vectors place no restrictions and may be added or taken away from any ERC set without changing its content.

All of these variants are commonly encountered. Here is a four-constraint case from Lombardi's important voicing typology paper (Lombardi 1999: 286, ex. (31)), which includes all of them. For clarity, we suppress *e* in favor of blankness. Bold font style draws attention to relevant consonants in the input and in the output to unfaithful mappings.⁹

⁹Constraint definitions: AGREE demands that adjacent obstruents have identical voicing values. IDLAR demands that the input voicing value for an obstruent match that of its output correspondent. IDONSLAR demands the same, but is restricted to those output obstruents occurring in an onset and released into a sonorant. *LAR demands that an output obstruent be voiceless. All constraints assess one violation for each configuration that fails to meet their demands.

(26) **Lombardi's analysis of Swedish.**

	AGREE	IDLAR	*LAR	IDONSLAR
a. <i>stekde</i> → <i>stek.te</i> ~ <i>steg.de</i>			W	L
b. <i>sku:g</i> → <i>sku:g</i> ~ <i>sku:k</i>		W	L	
c. <i>vigsəl</i> → <i>vik.səl</i> ~ <i>vig.səl</i>	W	L	W	
d. <i>vigsəl</i> → <i>vik.səl</i> ~ <i>vig.zəl</i>			W	W
e. <i>ægde</i> → <i>æg.de</i> ~ <i>æk.te</i>		W	L	W
f. <i>stekde</i> → <i>stek.te</i> ~ <i>stek.de</i>	W	L	W	L
g. <i>ægde</i> → <i>æg.de</i> ~ <i>æk.de</i>	W	W	L	

Lombardi is considering the voicing patterns in obstruent clusters and aims to show that her constraint system provides a grammar for the kind of pattern seen in Swedish.¹⁰ If we are looking for the ranking, only a proper subset of the cases need be considered. (When, like Lombardi, we turn to argue that the established ranking selects the desired optima, we must deal with entire candidate sets, since to be optimal is to be better than *all* violation-distinct competitors.)

The required ranking, which is total and follows (in this particular case) the listed order of constraints in the tableau, is determined by any of the following ERC sets (as well as by any sets containing them):

a, b, c

a, b, f

a, e, c

a, e, f

For illustrative purposes, tableau (26) distinguishes the first of these. Note that the entire grammar of voicing is fixed by just three pieces of comparative data.

A couple of other remarks: vector **d** (*e,e,W,W*) is completely uninformative, since the desired optimum harmonically bounds its competitor: it is always equal to or better than the competitor, as is clearly displayed, and will win on any ranking.¹¹ (Harmonic bounding of this sort is an important fact about the way the constraint set operates to exclude certain items universally, but it does not help with the Ranking Problem.) Row **g**, even though it does not involve harmonic bounding, is also uninformative in context, because any ranking that satisfies **b** (*e,W,L,e*) must also satisfy the weaker requirements of **g** (*W,W,L,e*), though not vice versa.

¹⁰Although our focus is on the structure of the analysis, and not on voicing per se, it might be useful to have a sense of how this works. The action of the given hierarchy on the universal set of candidates can be understood as follows (Prince 2006a). All obstruent clusters agree in voicing in the output (AGREE). If an input cluster is heterogeneous, the most frequent input value (if there is one) takes over, as it requires the fewest breaches of faithfulness (IDLAR). If an input cluster is evenly split between voicing values, it emerges voiceless (*LAR). (As for IDONSLAR, all cases are decided before it is reached, so it plays no role.) In the cases of unevenly-apportioned clusters, this doesn't match the generalization cited in the text: "Swedish has been described as having bidirectional spread of [-voice]; in other words, a voiced obstruent devoices next to a voiceless obstruent. . ." (Lombardi 1999: 285), but it's what the hierarchy says. On the relation of the predicted pattern to the facts, see (60) below.

¹¹Harmonic bounding: Prince and Smolensky 2004; Samek-Lodovici 1992; Samek-Lodovici and Prince 1999, 2005.

Since our target is the most concise and informative representation of the ranking conditions, we will be interested only in those ERC vector sets of *smallest size*. Tableau (26) has 7 vectors; all are distinct, but only 3 are needed to encapsulate the tableau’s ranking restrictions. Observe that the ERC vector is a free-standing formal object with a fully determinate meaning; we can (and will) use vectors to encode ranking relations that arise by implication from data.

Let us introduce the term ‘basis’ to refer to any minimal-cardinality set of ERC vectors, of whatever origin, that fully determines the rankings required by some designated set, which we will call the ‘reference set’.

(27) **Basis.** A basis is a minimal cardinality collection of ERCs equivalent to a reference set.

There are quite a few three-member ERC vector sets—bases—for Lombardi’s Swedish hierarchy: 27 in total. Tableau (26) contains 4 of them, each directly produced by analysis of comparative data. Among these yet further distinctions may be made. Since the distinctions are fully exemplified in the simpler abstract case we began with, which has 3 rather than 4 constraints in linear order, let us return to that.

(28) **Equivalent ERC sets:** $1 \gg 2 \gg 3$

CT ₁	1	2	3	CT ₂	1	2	3	CT ₃	1	2	3
a	W	L	L	a'	W	L	e	a''	W	L	W
b	e	W	L	b	e	W	L	b	e	W	L

Two basic observations may be made about the three-constraint situation in (28):

- The ranking relation $2 \gg 3$ comes from just one vector, **b**: (e,W,L).
- The ranking relation $1 \gg 2$, in the context of **b**, has *three* distinct possible sources:

$$\begin{array}{ll}
 \mathbf{a}: & (W,L,L) \quad [\mathbf{a}]: \quad 1 \gg 2 \ \& \ 1 \gg 3 \\
 \mathbf{a}': & (W,L,e) \quad [\mathbf{a}']: \quad 1 \gg 2 \\
 \mathbf{a}'': & (W,L,W) \quad [\mathbf{a}'']: \quad 1 \gg 2 \ \vee \ 3 \gg 2
 \end{array}$$

Vector **a** is the most informative of those concerned with constraint **1**, in the sense that it spells out every condition on the ranking of that constraint. The others either omit a requirement (**a'**) or hide the truth among a disjunction that turns out to be irrelevantly weak (**a''**).

Here we see the three extremes in basis structure:

- (1) CT₁ exemplifies what we will call the ‘Most Informative Basis’ (**MIB**): for each ERC, it gives the fullest possible listing of constraints dominated by (some element of) the W-set.
- (2) CT₃ is an instance of its direct opposite, the ‘Least Informative Basis’ (**LIB**), because it is filled up with as many local disjunctions as possible, giving the locally weakest (and most opaque) picture of the ranking structure.
- (3) CT₂ is an instance of what we will call the ‘Skeletal Basis’ (**SKB**): it suppresses all relations collectively derivable (here, from transitivity), omitting from the list of dominated constraints those that cannot be *immediately* dominated by the

highest-ranked member of the W-set. For this reason, the Skeletal Basis is quite useful, for example in constructing the diagrams that represent ranking relations and in isolating the particular data that contributes each local domination relation (here $1 \gg 2$). The virtues of the Skeletal Basis are taken up in Sect. 4.

To characterize these distinctions in terms of overall W-L-*e* content: the **LIB** has the most W's possible, the **MIB** the most L's, and the **SKB** the most *e*'s (therefore the least total number of W's and L's).

Both the **MIB** and Skeletal Basis have the extremely desirable property that any disjunction in one of their ERCs corresponds to a real disjunction in the overall system. Any W in a **MIB** or Skeletal Basis ERC vector appears as an actual dominator in some linear ranking countenanced by that basis (proved in Prince 2006c: 13). These bases are locally trustworthy. Since the way to the Skeletal Basis lies through the **MIB**, we now turn to sharpening the logical tools that will lead to the **MIB**-producing Fusional Reduction Algorithm (FRed).

Informativeness has various intuitive meanings, depending on what you are interested in being informed about. For the present initiative, the informativeness relation among ERCs is understood to be that of *logical entailment*.

So: what do we mean by *entailment*? The vector $\mathbf{a} (W,L,L)$ entails $\mathbf{a}' (W,L,e)$ in exactly the same way that 'p&q' entails 'p'. Similarly, $\mathbf{a}' (W,L,e)$ entails $\mathbf{a}'' (W,L,W)$ in the same way that 'p' entails 'p∨q'. Every circumstance in which the first of these is true is one in which the second is true. In model-theoretic talk: every model that satisfies the first also satisfies the second. Extensionally, the set of rankings admitted by \mathbf{a} is a (proper) subset of those admitted by \mathbf{a}' , and the total rankings admitted by \mathbf{a}' are a (proper) subset of those admitted by \mathbf{a}'' . These relations may be readily seen if we simply list the rankings (total orders) allowed by each ERC.

(29) **Linear rankings admitted by each ERC.**

Vector	$\mathbf{a} = (W,L,L)$	$\mathbf{a}' = (W,L,e)$	$\mathbf{a}'' = (W,L,W)$
ERC	$1 \gg 2 \ \& \ 1 \gg 3$	$1 \gg 2$	$1 \gg 2 \ \vee \ 3 \gg 2$
Rankings:	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$
	$1 \gg 3 \gg 2$	$1 \gg 3 \gg 2$	$1 \gg 3 \gg 2$
		$3 \gg 1 \gg 2$	$3 \gg 1 \gg 2$
			$3 \gg 2 \gg 1$

Any ranking good for \mathbf{a} also works for \mathbf{a}' and \mathbf{a}'' ; any ranking good for \mathbf{a}' works for \mathbf{a}'' as well. If we are looking at an overall ranking situation where \mathbf{a} is satisfied, then \mathbf{a} is the most informative of the trio about that situation. Our $1 \gg 2 \gg 3$ in example (28) is exactly such a case: ERC [\mathbf{a}] (hence vector \mathbf{a}) provides the greatest possible amount of information about $\mathbf{1}$ and its necessary dominees, namely that $\mathbf{1}$ dominates both $\mathbf{2}$ and $\mathbf{3}$, a fact unmentioned in either \mathbf{a}' or \mathbf{a}'' .

'Informativeness', for us, then, *is* entailment. A stronger, asymmetrically entailing ERC is more informative, more restrictive of rankings, than the ERCs that it asymmetrically entails. Since entailment is based on satisfaction, it extends just as directly from ERCs (as logical expressions) to their corresponding vectors. So we will speak of entailment, thus extended, as a relation among vectors as well, exactly tracking the relation among their associated ERCs.

The informativeness relation holds meaningfully among vectors arising from a given reference set, which is what they are being informative about. If you know ‘p&q’ you know more than if you know ‘p’ alone—so long as ‘p’ and ‘q’ are both informative in the first place.¹² In particular, we are only interested in vectors that are entailed by the reference set, because we seek to characterize its essential content.

There is then a hierarchy of informativeness among **a**, **a'**, and **a''**, running from more to less, based on asymmetric entailment. This local hierarchy propagates to the bases containing them. The three different bases for **1**≫**2**≫**3** can be rated according to their relative informativeness. All must contain exactly two vectors and one of those must be **b** (*e,W,L*), but the second can be chosen from among these:

$$\mathbf{a}: (W,L,L) \quad \mathbf{a}': (W,L,e) \quad \mathbf{a}'': (W,L,W)$$

If we seek maximal informativeness, we choose **a**, since it asymmetrically entails the others. Our choice, taken with **b** (*e,W,L*), yields the **MIB**. If we seek to focus on local relations and avoid consequences collectively derivable through transitivity, we choose **a'**, arriving at the Skeletal Basis. If we are for some reason interested in keeping up the appearance of openness in our local options, we choose **a''**, yielding the Least Informative Basis.

The minimality restriction of the size of the *basis* also determines the extent of entailment relations within it: none can exist. If one basis element were entailed by others, then it could be removed without loss of information, which means that the original putative ‘basis’ didn’t have minimal cardinality and couldn’t have been a basis at all.

Our overall goal can now be made quite concrete: given a reference set of ERC vectors, we wish to operate upon it to obtain a minimal vector set giving back the entirety of required rankings. In addition to conciseness, we want informativeness: maximal local informativeness gives the **MIB**. From the elements of the **MIB**, the Skeletal Basis may be constructed. Although there will in general be many bases for a typical reference set, both the **MIB** and the Skeletal Basis are unique, a not-quite-trivial fact that is proved in Prince and Brasoveanu 2010.

For Lombardi’s Swedish as evidenced in (26), the **MIB** will contain exactly these three elements:

$$\begin{aligned} \mathbf{b}_1: (W,L,L,L) & \quad \mathbf{1} \gg \mathbf{2} \ \& \ \mathbf{1} \gg \mathbf{3} \ \& \ \mathbf{1} \gg \mathbf{4} \\ \mathbf{b}_2: (e,W,L,L) & \quad \mathbf{2} \gg \mathbf{3} \ \& \ \mathbf{2} \gg \mathbf{4} \\ \mathbf{b}_3: (e,e,W,L) & \quad \mathbf{3} \gg \mathbf{4} \end{aligned}$$

This gives the following CT (*e* omitted for clarity):

(30) **MIB for Lombardi’s Swedish.**

MIB of (26)	1	2	3	4
b ₁	W	L	L	L
b ₂		W	L	L
b ₃			W	L

¹²For example, ‘2+2 = 4’ is entailed by everything and ‘2+2 = 5’ entails everything, but neither ‘p&(2+2 = 4)’ nor ‘p&(2+2 = 5)’ is notably more informative than ‘p’ by itself.

This is skeletalized by reducing all collectively-derivable L to *e* (which is equivalent to blank):

(31) **SKB for Lombardi’s Swedish.**

SKB of (26)	1	2	3	4
\mathbf{b}'_1	W	L	<i>e</i>	<i>e</i>
\mathbf{b}'_2		W	L	<i>e</i>
\mathbf{b}_3			W	L

Other bases exist, but they all contain weaker or more confounded elements. For example, \mathbf{b}_1 (W,L,L,L) can be replaced by (W,L,L,W) or (W,L,W,W) or (W,L,*e*,W) or various others, but any such are asymmetrically entailed by \mathbf{b}_1 , since they omit requirements present in \mathbf{b}_1 and introduce disjunctions that are cancelled out in the broader scheme of things.

Importantly, there is no restriction that the **MIB**, or any other basis, be a subset of the original set. In the case at hand, only vectors \mathbf{b}_3 (**MIB**) and \mathbf{b}'_2 (**SKB**) appear in the raw data, as (26)a and (26)b. The others must be constructed by *combining information* that is directly available. We now leave behind the notion of the ERC vector as the creature of data observation and we advance to regarding it as formal expression, with a certain interpretation, in which guise it serves as the essential element for representing and calculating ranking requirements.

2.2.1 Appendix to 2.2

For purposes of explicitness, we record the notions involved in the discussion.

- (32) **Entailment among vectors.** Let **A** be a set of ERC vectors, and **b** be an ERC vector. Then $\mathbf{A} \models \mathbf{b}$ iff every ranking that satisfies **A** also satisfies **b**. For vectors **b,c**, if $\mathbf{b} \models \mathbf{c}$ but not vice versa, then **b** ‘asymmetrically entails’ **c**. Entailment between sets of ERC vectors is as follows: for **A** and **B** sets of ERC vectors, $\mathbf{A} \models \mathbf{B}$ iff $\mathbf{A} \models \mathbf{b}$ for every $\mathbf{b} \in \mathbf{B}$.
- (33) **Informative.** Let **A** be an ERC vector set, and **b,c** be ERC vectors. Suppose also that $\mathbf{A} \models \{\mathbf{b}, \mathbf{c}\}$. Then ‘**b** is more informative than **c** about **A**’ iff **b** asymmetrically entails **c**.
- (34) **Basis.** Let **A** be an ERC vector set and suppose **B** is such that $\mathbf{B} \models \mathbf{A}$ and $\mathbf{A} \models \mathbf{B}$, i.e. **A** and **B** are equivalent ERC vector sets. Further suppose there is no set **C** with the same property such that $|\mathbf{C}| < |\mathbf{B}|$ (i.e. the cardinality of **C** is strictly less than the cardinality of **B**). Then we say that **B** is a *basis* for **A**.¹³
- (35) **Most Informative Basis (MIB).** Let **B** be a basis for **A**. **B** is the **MIB** for **A** iff **B** has the most L’s of any basis. Equivalently, iff for every $\mathbf{b} \in \mathbf{B}$ and every **x** such that $\mathbf{A} \models \mathbf{x}$, if $\mathbf{x} \models \mathbf{b}$, then $\mathbf{x} = \mathbf{b}$. Equivalently, if $\mathbf{x} \models \mathbf{b}$ and $\mathbf{x} \neq \mathbf{b}$, then $\mathbf{B} \setminus \cup \{\mathbf{x}\}$ is not a basis for **A**.

¹³It is also possible to define *basis* in terms of logical properties and derive its minimal cardinality; noted below, Sect. 3.3, and developed in Prince and Brasoveanu 2010. For present purposes, we settle for the direct approach.

- (36) **Least Informative Basis (LIB).** Let \mathbf{B} be a basis for \mathbf{A} . \mathbf{B} is the **LIB** for \mathbf{A} iff \mathbf{B} has the most W 's of any basis. Equivalently, iff for every $\mathbf{b} \in \mathbf{B}$ and every \mathbf{x} such that $\mathbf{A} \models \mathbf{x}$, if $\mathbf{b} \models \mathbf{x}$ and $\mathbf{x} \neq \mathbf{b}$, then $\mathbf{B} \setminus \mathbf{b} \cup \{\mathbf{x}\}$ is not a basis for \mathbf{A} .
- (37) **Skeletal Basis (SKB).** Let \mathbf{B} be a basis for \mathbf{A} . \mathbf{B} is the **SKB** for \mathbf{A} iff \mathbf{B} has the most e 's of any basis. Equivalently, using the notion of fusion introduced in Sect. 2.3 below, notated with the ' \circ ' symbol, \mathbf{B} is the **SKB** for \mathbf{A} iff for every $\mathbf{b} \in \mathbf{B}$ and for every \mathbf{x} such that $\mathbf{A} \models \mathbf{x}$, if $\mathbf{x} \circ \mathbf{b} = \mathbf{b}$ and $\mathbf{x} \neq \mathbf{b}$ then $\mathbf{B} \setminus \mathbf{b} \cup \{\mathbf{x}\}$ is not a basis for \mathbf{A} .

2.3 ERCs and consequences

Defining and finding the Most Informative Basis turns on the notion of *entailment* between ERCs. Given an ERC set \mathbf{A} , the elements of a basis are drawn not just from \mathbf{A} , but from the entire set of ERCs entailed by \mathbf{A} . The relation of informativeness between ERCs is based on entailment. The elements of a basis are logically independent, in the sense that no one of them is entailed by any collection of others.

The relation of entailment between ERCs has been studied in *ERA*. Here we present the basic findings that underlie the present investigation and move forward to develop the key ideas that lead to the Fusional Reduction Algorithm.

2.3.1 Single ERC entailment

Entailment relations fall naturally into two types. The simplest involves entailments following from a single ERC. More complex is the relation between a multi-member set of ERCs and its consequences. We will see that the second reduces to the first, when an appropriate method of combining ERCs, 'fusion', is introduced. The algorithm FRed depends on this reduction.

Let us first examine single ERC entailment. Given an ERC vector that contains at least one L coordinate, we can derive an entailment from it by removing an L and replacing it with an e . We write ' \rightarrow ' between vectors connected by this operation.

(38) **L-retraction.**

$$\begin{array}{lcl} (W,L,L) & \rightarrow & (W,L,e) \\ \mathbf{1} \gg \mathbf{2} \ \& \ \mathbf{1} \gg \mathbf{3} & \models & \mathbf{1} \gg \mathbf{2} \end{array}$$

Similarly, if we replace an e with a W , we also obtain a legitimate entailment. Again, we signify the relation between vectors with ' \rightarrow '.

(39) **W-extension.**

$$\begin{array}{lcl} (W,L,e) & \rightarrow & (W,L,W) \\ \mathbf{1} \gg \mathbf{2} & \models & \mathbf{1} \gg \mathbf{2} \ \vee \ \mathbf{3} \gg \mathbf{2} \end{array}$$

Putting aside those 'trivial' vectors that are satisfied in all models or in none, it turns out that *every* entailment from a single ERC arises from a sequence of L-retractions and W-extensions (*ERA*: 6, Prop 1.1).¹⁴

¹⁴Recall that a model is a grammar or total ranking of the constraints (12, 13), yielding the semantics of satisfaction (see ex. (24)). We are thus taking a model-theoretic perspective on the logic of natural language

The trivial vectors deserve a moment’s notice. These lack the familiar configuration in which both W’s and L’s are present, and they therefore do not express a contentful ranking relation. They come in two species, valid and invalid:

- (1) **Valid.** Those lacking L impose no ranking requirements, because there is no constraint that must be subordinated; they are therefore satisfied by any hierarchy and universally valid.
- (2) **Invalid.** Those lacking W but containing L no ranking can satisfy, because they contain no legitimate dominator to satisfy the requirement that L be dominated.

To provide for convenient reference to these entities, let us designate the set of all valid vectors of any length by the name W^* , mnemonically marking their composition from W’s and e’s. The set of invalid vectors we will designate L^+ , indicating that its members contain at least one L (but no W’s).

Trivial vectors, because of their indiscriminate logic, misbehave with respect to L-retraction and W-extension. Boolean logic tells us that invalid statements entail anything, and anything entails a valid statement. But L-retraction and W-extension are more particular. For example,

$$(40) \quad (e,L) \not\rightarrow (L,W) \quad \text{but} \quad (e,L) \models (L,W): \textit{ex falso quodlibet}.$$

The relationship $e \rightarrow L$ in the first coordinate is not sanctioned by L-retraction. Similarly,

$$(41) \quad (W,L) \not\rightarrow (e,W) \quad \text{but} \quad (W,L) \models (e,W): \textit{verum ex quodlibet}.$$

Here the first-coordinate relationship $W \rightarrow e$ runs in the wrong direction, but has no effect on (Boolean) entailment.

L-retraction and W-extension can be brought under one roof if we see the entailment relation as being based on an order. Assume the scale $L < e < W$.¹⁵ Then we can say that **a** ‘arrows’ **b** by definition iff every coordinate of **a** is *less than or equal to* the corresponding coordinate of **b** in this order.¹⁶ Our fundamental result identifies the ‘arrowing’ relation as the perfect mirror of semantic entailment, as long as trivial ERCs are not involved.

- (42) **Entailment over individual nontrivial vectors \Leftrightarrow Arrow.**
 For **a, b** nontrivial, $\mathbf{a} \rightarrow \mathbf{b}$ iff $\mathbf{a} \models \mathbf{b}$.
 Proof. ERA: 6, Prop 1.1.

grammar. The centrality of entailment in OT metatheory attests to the utility of the approach, here and in linguistics generally.

¹⁵The arithmetization of fn. 8 (with $L = -1, e = 0, W = 1$) would serve handily. See Meyer 1975: 400 for the original introduction of the scalar idea for interpreting arrow and related connectives; see ERA: 55–57 for discussion.

¹⁶This extends the familiar Boolean relation of material implication. There the scale is $F < T$, often mirrored numerically by taking $F = 0$ and $T = 1$. Then $p \supset q$ is true whenever the truth value of p is less than or equal to that of q .

Observe that moving from ‘arrow’ to entailment is always legitimate, regardless of triviality. The arrow relation *refines* entailment, adding further structure but never contradicting it.

2.3.2 Multiple ERC entailment

Entailments may follow from a set of ERCs that do not follow from any individual member of the set. Consider this pair:

(43) **1>>2 and 1>>3.**

	1	2	3	ERC
a	W	L	e	1>>2
b	W	e	L	1>>3

Neither **a** nor **b** independently yields the conclusion that **1** must dominate both **2** and **3**. In this kind of case, there is a vector that exactly encapsulates the set, namely (W,L,L). But when transitivity of ranking order is involved, such a handy reduction will not be available. Consider the simplest case where transitivity asserts itself.

(44) **1>>2>>3.**

	1	2	3	ERC
a	W	L	e	1>>2
b	e	W	L	2>>3

From these two vectors, it follows that **1>>3**. But the vector (W,e,L) expressing this requirement is *not* individually arrowed (or entailed) by either **a** or **b**:

(45) **Failures of simple arrowing.**

	Non-arrowed	Non-entailed	Cause
a:	(W,L,e) $\not\rightarrow$ (W,e,L)	1>>2 \models 1>>3	e \rightarrow L in 3
b:	(e,W,L) $\not\rightarrow$ (W,e,L)	2>>3 \models 1>>3	W \rightarrow e in 2

Furthermore, there is no way to combine **a** and **b** into a third vector that expresses the full content of both. The reason is straightforward: the totality of information in **a** and **b** is represented in the logical conjunction of their associated ERCs, and this expression is simply not of the ERC form.

(46) **Conjunction.** [a]&[b] = 1>>2 & 2>>3

In this conjunction, constraint **2** is identified both as a dominator (W) and a dominee (L). But any ERC must treat the W and L classes as disjoint.¹⁷ An ERC relates two disjoint classes of constraints; transitivity mentions three constraints falling into two overlapping classes.

¹⁷In an equivalent to the vectorial representation, an ERC may be expressed as an ordered pair of sets of constraints, the W-set and the L-set of the candidate pair: (W,L). See ERA: 2.

What’s needed is a different kind of logical combination, one which may sacrifice some information, but preserves enough to support the calculation of entailments. The appropriate method is ‘fusion’, which combines values according to the following scheme (ERA: 8):

- (47) **Fusion of values.**
 $L \circ X = X \circ L = L$ ‘L is dominant’
 $e \circ X = X \circ e = X$ ‘e is identity’
 $W \circ W = W$ ‘fusion is idempotent: more generally, $X \circ X = X$ ’

We write $X \circ Y$ for ‘X fused with Y’ where X,Y are variables over $\{W,L,e\}$. We will speak of the components of a fusion as ‘fusands’. To get the fusion of multi-coordinate vectors, we apply the operation to corresponding coordinates, paralleling the way vectors are added. As is customary, we overload the fusion symbol, using the same notation for the derived and basic operations.

- (48) **Fusion of vectors.** Fuse coordinate-wise. Writing $v[k]$ for the k^{th} coordinate of v , we define the coordinates of the vector $a \circ b$ in terms of the fusion of its coordinates.

$$a \circ b[k] =_{\text{def}} a[k] \circ b[k]$$

That is: the k^{th} coordinate of $a \circ b$ is the fusion of the k^{th} coordinates of a and b .

Like more familiar operations such as *and*, *or*, *plus*, and *times*, the order of fusion of multiple entities is immaterial; it therefore extends smoothly from pairs to multi-element sets.¹⁸ We therefore adopt a general notation for fusion of multiple vectors.

- (49) **Notational convention: Fusion of sets of ERC vectors.**
 We write fA for the fusion of a nonempty set of ERC vectors A , with the convenient proviso that $f\{v\} = v$ for singleton sets.

Returning to case (44) with fusion in hand, we can now derive the desired collective consequence. First, we fuse the two vectors:

- (50) **Fusion and transitivity.**

	1	2	3
a	W	L	e
b	e	W	L
a o b	W	L	L

The vector representing $1 \gg 3$ is arrowed, and therefore entailed, by the fusion $a \circ b$:

$$(W,L,L) \rightarrow (W,e,L) \quad \text{nb: } L \rightarrow e \text{ in } \mathbf{2}$$

¹⁸Fusion is commutative by definition and associativity follows from a straightforward calculation. The term itself, and the \circ notation come from relevance logic (Anderson and Belnap 1975). See Sect. 2.3.4 below for discussion.

Fusion has delivered a vector that allows the collective consequence $1 \gg 3$ to be derived from the rules for individual entailments. This pattern turns out to be entirely general.

A fusion-derived vector is telling us something about ranking requirements that the entire fusing set also tells us. Fusion respects the content of a set of vectors. More important: it suffices, properly used, to recover *any* ranking requirement imposed collectively by members of that set.

To get a glimpse of the utility of fusion, let's return to the Lombardi analysis, gathering some highly disjunctive ERCs that nonetheless jointly and completely determine the ranking.

(51) **A basis for Lombardi's Swedish.**

	1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a. <i>stekde</i> → <i>stek.te</i> ~ <i>steg.de</i>			W	L
e. <i>ægde</i> → <i>æg.de</i> ~ <i>æk.te</i>		W	L	W
f. <i>stekde</i> → <i>stek.te</i> ~ <i>stek.de</i>	W	L	W	L

Each ERC-internal disjunct marked **W** suggests an illusory explanation for the optimality of a correct form. For example, comparison **e** taken by itself allows for **4:IDONSLAR** to be regarded as explaining the choice of [*æg.de*] over **[æk.te]*. But comparison **a** shows that this cannot be right; it is only **2:IDLAR** that can be doing the job. Fusion allows us to integrate the key information from other ERCs, suppressing the nonviable disjunctions (bolded and italicized) that come in with the raw data. Suppressing them all via fusion leads in this case directly to the **MIB**.

(52) **Relevant fusions.**

MIB	1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a			W	L
a◦e		W	L	L
a◦e◦f	W	L	L	L

Fusion always expresses a ranking restriction inherent in the original data. Those displayed here are particularly useful, since the last two rows are individually stronger than the corresponding rows in the original basis (51), deriving directly from the data, in which several factors are confounded. Selective fusion removes the confounds, and the resulting basis in (52) resolves the Ranking Problem with maximal informativeness in each ERC, the hallmark of the **MIB**.

- (53) Fusion Associated ERC
a◦e **2:IDLAR** \gg {**3:*LAR, 4:IDONSLAR**}
a◦e◦f **1:AGREE** \gg {**2:IDLAR, 3:*LAR, 4:IDONSLAR**}

It is immediate from inspection of the tableau that a total order on the constraint set is mandated.

To put the content-respecting property of fusion more exactly: the fusion of any set of vectors is entailed by that set, so that fusion never leads us outside the realm of consequences of the original set. This is a ‘closure’ property, since the set of entailed vectors is closed under fusion.

(54) **Closure: Fusion respects entailment.**

$$\mathbf{A} \models f \mathbf{A}.$$

Proof. ERA: 10, Prop. 2.1.

The closure property is true of a variety of possible combining operations, some uselessly weak—for example, if we replace X-combine-L with *e*, yielding a valid ERC, the entailment would hold vacuously (*verum ex quodlibet*). But fusion has a great strength: it reduces the collective entailment problem to single ERC entailment.

For any ERC vector *v* entailed by a set *A*, we are guaranteed that there is a subset of *A* that fuses to a single ERC vector that entails *v*.

(55) **Entailment by fusion.** Let *v* be an ERC vector and *A* a set of ERC vectors.

$$\text{If } \mathbf{A} \models \mathbf{v}, \text{ then for some } \mathbf{X} \subseteq \mathbf{A}, f \mathbf{X} \models \mathbf{v}.$$

Proof. ERA: 14, Prop. 2.5.

Even better, for all nontrivial entailments, we have a perfect match between entailment and *W*-extension/*L*-retraction—equivalently, the ‘arrow’ ordering relation on vectors.

(56) **General nontrivial entailment** \Leftrightarrow **Arrow.** Let *v* be a nontrivial ERC vector and *A* a satisfiable set of ERC vectors.

$$\mathbf{A} \models \mathbf{v} \text{ iff for some } \mathbf{X} \subseteq \mathbf{A}, f \mathbf{X} \rightarrow \mathbf{v}.$$

Proof. RL from (42) and (54).

LR from (55) and (42). See ERA: 6, 10, 14.

Only the left-to-right (LR) direction—entailment to arrow—actually needs the caveats about nontriviality and satisfiability. This reflects the mismatch, noted above, between the hyperparticularity of arrow and the nonparticularity of entailment with respect to relations involving valid and invalid vectors. Compare the observations in (40) and (41), which identify specific instances of divergence.

The arrow relation, however, is a sturdily reliable guide to entailment, without qualification:

(57) **Arrow** \Rightarrow **Entailment** (all vectors).

$$\text{If for some } \mathbf{X} \subseteq \mathbf{A} \text{ we have } f \mathbf{X} \rightarrow \mathbf{v}, \text{ then } \mathbf{A} \models \mathbf{v}.$$

Proof. Follows directly from ERA: 10, Prop. 2.1.

Fusion, then, is pretty much as strong as possible. The key fact is that an entailed ERC, whatever the complexity of the logic, is always entailed by a single ERC that is the fusion of some subset of the reference set. With this result, we have completely transported the entailment problem from the semantics of ranking arguments into the domain of vectorial operations and relations.

2.3.3 Fusion and failure

The flipside of entailment is inconsistency/unsatisfiability. If p entails q , then p and $\neg q$ are inconsistent—and vice versa. Example: being square (p) entails having corners (q), so you can't have a *cornerless* square ($\neg q$ and p). An ERC set may be internally inconsistent, with no hierarchy satisfying it. This is no exotic mischance: in the course of research, assumptions about the constraint set or linguistic structure frequently reveal themselves as erroneous or incomplete because they give rise to unsatisfiable ERC sets. Similarly, much of Tesar's learnability work depends on inconsistency detection to reject mistaken hypotheses (Tesar 1995 *et seq.*).

Fusion detects inconsistency unerringly. The simplest example takes the form of a pair of constraints which are ranked $1 \gg 2$ by one ERC and $2 \gg 1$ by another. The ERC vectors would be (W,L) and (L,W) . Their fusion to (L,L) marks the case as hopeless, since it demands domination of both and provides no dominator for either. This result hints at the underlying generalization: a set of ERC vectors is unsatisfiable iff it contains a subset that fuses to a vector with no W 's and at least one L , a member of L^+ .

(58) **Unsatisfiability via fusion.**

No hierarchy satisfies A iff for some $X \subseteq A$, $fX \in L^+$.

Proof. ERA Prop. 2.4:11.

To see how this works in general, consider the following more intricate example:

(59) $A = \{a, b\}$.

	1	2	3	4	ERC
a	W	L	W	L	$(1 \gg 2 \ \& \ 1 \gg 4) \vee (3 \gg 2 \ \& \ 3 \gg 4)$
b	L	W	L	W	$(2 \gg 1 \ \& \ 2 \gg 3) \vee (4 \gg 1 \ \& \ 4 \gg 3)$
aob	L	L	L	L	$\exists x \in \emptyset$ s.t. $x \gg \{1,2,3,4\}$

ERC-wise, it will take some tangled work with the propositional calculus, especially the distributive law, accompanied by references to the transitivity and asymmetry of ' \gg ', to crank out the conclusion that no ranking can satisfy both vectors. But fusion delivers the fact directly. The ERC associated with the fusion is invalid, unsatisfiable by any hierarchy; therefore the original set must be unsatisfiable as well. Whatever entails a falsehood must itself be false.

To put this fact to work, let us return again to Lombardi's analysis of Swedish, adding in a piece of data mentioned but untableau'd in the original article: /sku : **g**+**s**+**brand**/ 'forest fire', which comes out as [sku : **ksprand**] not *[sku : **gzbrand**].¹⁹ We attach this fact to the basis (51) in order to assess its status. We know that the cited basis is equivalent to the full set of data considered by Lombardi. If the new datum is already encompassed, it should be entailed by what we already have. If it raises problems, they will be visible in this context.

¹⁹Hat-tip to Sarah Murray for pointing us to the key datum.

(60) **Basis {a,e,f} for Lombardi’s Swedish + additional datum.**

	1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a. <i>stekde</i> → <i>stek.te</i> ~ <i>steg.de</i>			W	L
e. <i>ægde</i> → <i>æg.de</i> ~ <i>æk.te</i>		W	L	W
f. <i>stekde</i> → <i>stek.te</i> ~ <i>stek.de</i>	W	L	W	L
h. <i>sku : gsbrand</i> → <i>sku : ks.prand</i> ~ <i>sku : gz.brand</i>		L	W	L
e◦h		<i>L</i>	<i>L</i>	<i>L</i>

A quick calculation shows that $e◦h = (e,L,L,L)$. Bad news! Observe, too, that the problem in the data does not emerge from fusion of the whole set: $a◦e◦f◦h = (W,L,L,L)$. The theorem is that failure occurs when any subset fails, here just {e,h}.

The underlying problem, of course, is that IDLAR favors the candidate making fewest changes in the voicing value of obstruents (see fn. 10). The desired winner in **h** makes two such changes, its competitor merely one. This is ‘majority rules’: see Lombardi 1999: 295 and Baković 2000: 25ff. for relevant discussion. In the case at hand, its force is mooted by another fact of the language, albeit one that has not been prominent in discussion of the issue: Swedish admits no **z**. When we acknowledge this fact by noting the force of a **z**-barring constraint, call it ***z**, the inconsistency disappears. Now no subset fuses to L^+ .

(61) **More complete view of relevant constraints.**

	0: *z	1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a. <i>stekde</i> → <i>stek.te</i> ~ <i>steg.de</i>				W	L
e. <i>ægde</i> → <i>æg.de</i> ~ <i>æk.te</i>			W	L	W
f. <i>stekde</i> → <i>stek.te</i> ~ <i>stek.de</i>		W	L	W	L
h. <i>sku : gsbrand</i> → <i>sku : ks.prand</i> ~ <i>sku : gz.brand</i>	W		L	W	L
e◦h	W		<i>L</i>	<i>L</i>	<i>L</i>

The methodological moral is two-fold and makes up in widespread applicability what it lacks in subtlety: sound analysis requires inclusion of all relevant constraints and all relevant data. A *relevant* constraint is one assessing a polar value, W or L. *Relevant* data produces a new untailed ERC, new information about the ranking conditions required by the language.

2.3.4 *Fusion and the loss of information*

Why does fusion work? Formally, the response might be: because it does, and provably. Qualitatively, though, we can see that it has the virtue of preserving *necessary* subordination (through the dominance of L) and *possible* domination (through the dominance of W when L is not around). This L-preserving strategy captures the transitivity and asymmetry of domination. In the simplest case, this is straightforward: if **a** says $1 \gg 2$ and **b** says $2 \gg 3$, then both **2** and **3** are necessarily dominated; the

fusion $\mathbf{a} \circ \mathbf{b}$ records this fact and thereby obtains the transitivity-derived relation $\mathbf{1} \gg \mathbf{3}$. Similarly for *asymmetry*: ERCs requiring $\mathbf{1} \gg \mathbf{2}$ and $\mathbf{2} \gg \mathbf{1}$, for example, will fuse to (L,L), because both $\mathbf{1}$ and $\mathbf{2}$ are necessarily dominated. In this way, the logic of the ERC calculus inherently contains the logic of strict order, and fusion brings it out.

From this point of view, it is natural to expect that every unsatisfiable set should contain a subset that fuses to L^+ . Transitivity leads to contradiction between inconsistent ranking requirements, yielding a cycle of required constraint dominations, which is disallowed by the asymmetry of the relation. In such a cycle, every constraint is *necessarily dominated*, earning L somewhere, and thereby eliminating W and e from its fusion.²⁰

But fusion is not a panacea that automatically and indiscriminately resolves all problems. In order to retrieve the consequences of transitivity, the value e must serve as an identity, to preserve *possible domination* in $W \circ e$. This leads to cases where fusion *weakens* informativeness. Consider the following, in which fusion produces an ERC that is entailed by both fusands, and entails neither:

(62) **When fusion weakens.**

	1	2	3	ERC
a	W	e	L	$\mathbf{1} \gg \mathbf{3}$
b	e	W	L	$\mathbf{2} \gg \mathbf{3}$
a \circ b	W	W	L	$\mathbf{1} \gg \mathbf{3} \vee \mathbf{2} \gg \mathbf{3}$

Two complementary questions then arise: when does fusion preserve information exactly? And when does fusion lose information? (We consider only nontrivial ERCs.) These are equivalent to asking when fusion is identical to conjunction (*ERA*: 15–20). Conjunction is entirely information-preserving, since a conjunction entails each of its conjuncts, and the conjuncts jointly entail the conjunction.

We already have $\{\mathbf{p}, \mathbf{q}\} \models \mathbf{p} \circ \mathbf{q}$, the fusands jointly entailing the fusion, from (54). To answer the question of equivalence, then, we need only ascertain the circumstances in which the fusion is guaranteed to entail both fusands (or, complementarily, when it fails to entail one of them). This is the crucial pair of relations:

(63) **Requirements for fusion–conjunction equivalence.**

$$\begin{aligned} \mathbf{p} \circ \mathbf{q} &\rightarrow \mathbf{p} \\ \mathbf{p} \circ \mathbf{q} &\rightarrow \mathbf{q} \end{aligned}$$

A quick review of the rules for fusion reveals these considerations:

- in any coordinate where at least one of \mathbf{p} or \mathbf{q} has L, both entailments are secure. For any $X, Y \in \{W, L, e\}$, we have:

$$L \circ X \rightarrow Y \quad \textit{because 'L arrows anything'}$$

²⁰Observe that if the relation were not asymmetric, such a cycle would be harmless. We can have $a \geq b$ and $b \geq a$.

- in any coordinate where both **p** and **q** have **W**, the two entailments are secure:

$$W \circ W \rightarrow W \quad \textit{because 'W arrows W'}$$

- similarly where both are *e*:

$$e \circ e \rightarrow e \quad \textit{because 'e arrows e'}$$

This leaves only the situation where one vector has *e* and the other has **W**. Here arrow fails from the fusion to the *e*-bearing vector:

$$e \circ W = W \quad \textit{so} \quad e \circ W \not\rightarrow e \quad (\textit{here, } p \circ q \rightarrow p)$$

Fusion becomes equivalent to conjunction, and fully information preserving, precisely when every **W** in the fusion comes only from the fusion of **W**'s in the fusands. This situation is called '**W-compliance**' (*ERA*: 16).²¹ This is true not just of pairs, but also of arbitrary sets. When every **W** in the fusion of an entire set of ERCs comes only from **W**'s, then the fusion of the whole—a single vector—encapsulates everything that the set has to say about ranking. To see the power of fusion in a **W-compliant** environment, consider the following:

(64) **W-compliance.**

	1	2	3	ERC
a	W	W	L	1 » 3 ∨ 2 » 3
b	W	L	W	1 » 2 ∨ 3 » 2
a ∘ b	W	L	L	1 »{ 2,3 }

Here again we face a struggle with the distributive law (supported by the defining order properties of domination and an additional dollop of propositional calculus) if we wish to assault the consequences of the set {**a,b**} within standard logic. But the sole **W** in the fusion derives entirely from fusand **W**'s, signaling **W-compliance**, so that we can be sure that fusion delivers not just a consequence, but a complete equivalent. We may discard both fusands in favor of their fusion and lose nothing.

When **W-compliance** fails in at least one coordinate, due to a collocation of **W**'s and *e*'s there, the fusion is doomed to lose information.²² A dead loss, as it were, occurs when the fusion entails none of its component fusands. A simple example illustrates this outcome:

²¹The converse ('fusion ≡ conjunction implies **W-compliance**') is challenged only when both of the fusands are valid. Consider the simplest possible case, with just one coordinate: $e \circ W \rightarrow e$, yet $e \neq W$. From the logical point of view, **W** and **L** are acting like truth values **T** and **F**; *e* is a third value. **F** behaves in fusion just as it does in conjunction, dominating. Conjunction and fusion part company in their treatment of relations between **T** and the new value *e*. Conjunction must take $T \wedge e$ to be *e*, in order to obtain the eminently desirable ' $p \wedge q \rightarrow q$ ' (consider $p = T$, $q = e$). Thus, when combinations of **T** and *e* are not at issue, fusion collapses with conjunction. See *ERA*: 47ff. for discussion.

²²If there is information to be lost. Fusion of valid vectors, like e.g. (*e*,**W**) and (**W**,*e*), loses nothing because there is nothing to lose.

(65) **Failure of W-compliance I: Increase in disjunctivity.**

	1	2	3	ERC
a	W	e	L	$1 \gg 3$
b	e	W	L	$2 \gg 3$
$a \circ b$	W	W	L	$1 \gg 3 \vee 2 \gg 3$

To recognize the crucial points of departure from W-compliance, let's use the term 'info loss configuration' to describe any *constraint column* that contains both W and e but no L's—equivalently, any coordinate in the vector set that fails to be W-compliant under fusion. In example (65), every W-coordinate in $a \circ b$ belongs to an info loss configuration.

Since it is the vectors that carry the information, we have a related notion that describes their role: for a given constraint fusing to W, its 'info loss residue' is the set of vectors that contribute e to its fusion. Crucially, these will *not be entailed* by the fusion, since $W \rightarrow e$. That's why their domination information is lost. In the case at hand (65), vector **b** is the *info loss residue* of constraint **1**: it supplies the e which is obliterated in the fusion to W. Similarly, vector **a** is the info loss residue of constraint **2**. Neither is entailed by the fusion, so it fails to capture their content. The *total* info loss residue, the union of all residues, embraces the entirety of the data set here, indicating that fusion loses info across the board. In such cases, it is pointless to form the fusion; information is lost from every vector.

In other circumstances, as we've seen, loss of information is compensated for by a local increase in informativeness. Consider the following, which contains a single info loss configuration (constraint **1**) and a single correlated info loss residue (ERC **b**):

(66) **Failure of W-compliance II: Compensatory increase in conjunctivity.**

	1	2	3	ERC
a	W	L	e	$1 \gg 2$
b	e	W	L	$2 \gg 3$
$a \circ b$	W	L	L	$1 \gg 2 \ \& \ 1 \gg 3$

Focus on ERC vector **b** and the information that it delivers. The fusion $a \circ b$ loses all of **b**'s information about constraint **2**, retaining from **b** only the information that constraint **3** must be dominated. But this is not *nothing*—the implied relation between **1** and **3** is nowhere locally expressed in the original set $\{a, b\}$. The fusion, though lossy, provides a more complete account of the ranking restrictions on constraint **1** than any ERC in the set we started out with. The fusion asymmetrically entails one of its fusands, a desirable increase in informativeness. This means that we may *replace* the entailed fusand **a** with the more informative fusion $a \circ b$. We must still retain the unentailed residue **b**, because fusion obliterates its crucial proffer of constraint **2** as the necessary dominator of **3**.

Let us designate as 'fusionally reducible' any collection of vectors that can be replaced by the fusion of a set that includes it. It's clear that W-compliant sets are fusionally reducible in themselves. But the phenomenon is more general. Strikingly, a subset that is not internally W-compliant can still be reducible in the context of a

containing superset, if every info loss configuration column within the subset fuses to L in the fusion of its container. Trivial vectors aside, a subset of vectors is fusionally reducible in the context of a superset if and only if every member of the subset is pairwise W-compliant with the fusion of the superset. In the case just reviewed, the singleton set {a} is fusionally reducible to a◦b. This is more than a trivial swap of one for one, because the fusion is more informative than what it replaces.

Here’s a richer example: the focus is on the behavior of a multi-element subset within a yet larger set. To start off, let’s consider a non-W-compliant pair:

(67) **Fusion, with info loss.**

	1	2	3	4	ERC
a	W	L	e	W	1≫2 ∨ 4≫2
b	W	e	L	e	1≫3
a◦b	W	L	L	W	1≫{2,3} ∨ 4≫{2,3}

Constraint 4 is an info loss configuration; its residue is b. Vectors a and b are not W-compliant, so the pair is not fusionally reducible in the context of itself. Vector a alone is reduced by a◦b. But suppose they in fact sit in a larger world, where further conditions also obtain:

(68) **Fusionally reducible in context, but not internally W-compliant.**

A	1	2	3	4	ERC
a	W	L	e	W	1≫2 ∨ 4≫2
b	W	e	L	e	1≫3
c	e	W	e	L	2≫4
a◦b◦c	W	L	L	L	1≫{2,3,4}

Here the beneficent influence of ERC c dispels the info loss configuration 4. The addition of c introduces a new info loss configuration of its own in 1, of which c is the residue, but that merely means that c is indispensable in characterizing the entire new set.

The fusion a◦b◦c can replace the subset {a,b} without loss of entailments. The fusion of the whole is not merely equivalent to the subset {a,b}. It is stronger. It imposes a relation between 1 and 4 of which the subset is ignorant. The additional requirement 1≫4 is collectively mandated via transitivity involving constraint 2 as expressed in vector c, which declares that 2≫4. It is only vector c that provides the rationale for dismissing the illusory hope, asserted disjunctively in a and not contradicted in b, that constraint 4 is fit to serve as a possible dominator for constraint 2.

We may therefore replace {a,b} with the fusion of the whole. The resulting vector set is equivalent to the original set, with an increase in both conciseness and local informativeness.

(69) **Fusionally reduced set.**

A'	1	2	3	4	ERC
a◦b◦c	W	L	L	L	1≫{2,3,4}
c	e	W	e	L	2≫4

The set of ERC vectors presented in (69) is the smallest that can fully recover the content of the original set, and its component vectors are individually as informative as possible. We have therefore calculated the Most Informative Basis (**MIB**) for $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$.

In this section, we have developed the techniques that yield a syntactic or algebraic counterpart of the essentially semantic—model-theoretic—notion of entailment. Fusion is the key operation on sets of vectors. Its result contains the maximum possible number of L's, picking up every L in the set, identifying every necessarily subordinated constraint, and thereby making the fusion potentially more informative than any single vector in the set. Among the remaining coordinates, the fusion also collects every W, identifying all potential dominator constraints among those that needn't be subordinated. Fusion respects entailment and allows us to reduce nontrivial entailment from any *set* of premise vectors to the arrowing relation between single vectors. The logic of fusion identifies among the constraints those where fusion degrades the information content of the set of vectors that is fused: 'info loss configurations', constraints fusing to W with e in the fusion. Complementarily, we have identified among the vectors 'fusionally reducible sets', those whose informational content is preserved, or even (in context) enhanced, by fusion.

We are now equipped to take on the full Fusional Reduction Algorithm, which forms the techniques used selectively here into a procedure that is guaranteed to produce the **MIB** of any satisfiable set whatever.

2.3.5 Remark on the logical background

Fusion began its life in linguistics under the name 'summation' (Prince 2000). With the introduction of the 'negative', which transforms $[x \sim y]$ to $[y \sim x]$ (*ERA*: 12), a full-fledged logic emerged, identified in *ERA* as the implication-negation fragment of the relevance logic RM3 (Anderson and Belnap 1975; Meyer 1975; Parks 1972), which itself first appears, on its own, in Sobociński 1952. With this identification, notation and terminology was normalized to the relevance logic standard. RM3 is a three-valued propositional logic, in which our W corresponds to T(rue), L to F(alse), and e marks the third value. In this logic, fusion is a kind of weakened analog of conjunction; arrow, a weakened analog of material implication; and negative, of standard negation. The relation between OT and RM3 (as well as its big sibling RM), is explored in some detail in *ERA*: 47–80.

2.3.6 Appendix to 2.3

For the explicit record, we conclude with a concise listing of the key notions and results.

- (70) **Trivial vector.** A vector \mathbf{v} is 'trivial' over a constraint set Σ , iff either of two conditions holds:
- a. For every hierarchy H, a linear ordering of Σ , H satisfies \mathbf{v} .
 - b. For every hierarchy H, a linear ordering of Σ , H does not satisfy \mathbf{v} .

In the first case, we say that \mathbf{v} is *valid*; in the second case, *invalid*. In the first case, $\mathbf{v} \in W^*$, the set of all vectors lacking L; in the second, $\mathbf{v} \in L^+$, the set of vectors with at least one L but no W.

- (71) **Arrow.** Let \mathbf{a}, \mathbf{b} be vectors over a set of n constraints. Vector \mathbf{a} ‘arrows’ a vector \mathbf{b} , written $\mathbf{a} \rightarrow \mathbf{b}$, iff for every coordinate k , $1 \leq k \leq n$, we have $\mathbf{a}[k] \leq \mathbf{b}[k]$, where the order relation is determined by the scale $L < e < W$, of which ‘ \leq ’ is the reflexive closure, ‘less than or equal to’.
- (72) **Individual nontrivial entailment** \Leftrightarrow **Arrow.** For \mathbf{a}, \mathbf{b} nontrivial, $\mathbf{a} \rightarrow \mathbf{b}$ iff $\mathbf{a} \models \mathbf{b}$. (See (42).)
- (73) **Fusion.** Let \mathbf{a}, \mathbf{b} be vectors over a set of n constraints. The ‘fusion’ of \mathbf{a} and \mathbf{b} , written $\mathbf{a} \circ \mathbf{b}$ or $f\{\mathbf{a}, \mathbf{b}\}$, is the vector \mathbf{c} whose coordinates $\mathbf{c}[k]$ are given by the following:

$\mathbf{c}[k] = (\mathbf{a} \circ \mathbf{b})[k] = (\mathbf{b} \circ \mathbf{a})[k]$	‘fusion is commutative’
$\mathbf{c}[k] = X$ if $\mathbf{a}[k] = X$ and $\mathbf{b}[k] = X$	‘fusion is idempotent’
$\mathbf{c}[k] = L$ if $\mathbf{a}[k] = L$	‘L is dominant’
$\mathbf{c}[k] = \mathbf{b}[k]$ if $\mathbf{a}[k] = e$	‘ e is identity’

By convention, $f\{\mathbf{v}\} = \mathbf{v}$.

In the following, \mathbf{A} is a set of ERC vectors, \mathbf{v} an ERC vector, Σ the constraint set.

- (74) **Closure: Fusion respects entailment.** $\mathbf{A} \models f\mathbf{A}$. (cf. (54))
- (75) **Entailment by fusion.** If $\mathbf{A} \models \mathbf{v}$, then for some $\mathbf{X} \subseteq \mathbf{A}$, $f\mathbf{X} \models \mathbf{v}$. (cf. (55))
- (76) **Arrow \Rightarrow Entailment.** If for some $\mathbf{X} \subseteq \mathbf{A}$ we have $f\mathbf{X} \rightarrow \mathbf{v}$, then $\mathbf{A} \models \mathbf{v}$. (cf. (57))
- (77) **General nontrivial entailment \Leftrightarrow Arrow.** For \mathbf{v} nontrivial, and \mathbf{A} satisfiable, $\mathbf{A} \models \mathbf{v}$ iff for some $\mathbf{X} \subseteq \mathbf{A}$, $f\mathbf{X} \rightarrow \mathbf{v}$. (cf. (56))
- (78) **W-compliant.** \mathbf{A} is ‘W-compliant’ iff $\forall k f\mathbf{A}[k] = W \Rightarrow \mathbf{a}[k] = W$ for all $\mathbf{a} \in \mathbf{A}$.
- (79) **Fusion/Conjunction.** For $\mathbf{a}, \mathbf{b} \notin W^*$, $\mathbf{a} \circ \mathbf{b} \models \{\mathbf{a}, \mathbf{b}\}$ iff $\{\mathbf{a}, \mathbf{b}\}$ is W-compliant. More generally, if \mathbf{A} contains no valid vectors, then $f\mathbf{A} \models \mathbf{v}$ for every $\mathbf{v} \in \mathbf{A}$ iff \mathbf{A} is W-compliant.
Proof. ERA: 16, Props. 3.1, 3.2.
- (80) **Info loss configuration (ILC).** Let $C_k \in \Sigma$ provide the k^{th} coordinate for the vectors in \mathbf{A} . We say C_k is an ‘info loss configuration’ over \mathbf{A} iff $f\mathbf{A}[k] = W$ and there is a $\mathbf{v} \in \mathbf{A}$ such that $\mathbf{v}[k] = e$.
A constraint column containing both W and e but no L is an ILC; equivalently, any coordinate in the vector set where the criterion for W-compliance fails to hold.
- (81) **Info loss residue (ILR).** Let $C_k \in \Sigma$ be an info loss configuration over the set of vectors \mathbf{A} . The ‘info loss residue’ of C_k over \mathbf{A} is the set of vectors $\mathbf{v} \in \mathbf{A}$ such that $\mathbf{v}[k] = e$.
An ILR is a set of vectors that have an e in a column that constitutes an info loss configuration.

- (82) **Fusionally reducible.** $X \subseteq A$ is ‘fusionally reducible in A ’, where $|A| > 1$, iff for every $v \in X$, $fA \models v$.
 ‘A set of vectors whose content is encodable via fusion of a containing superset.’

Remark we may have $X=A$, in which case a (W-compliant) set is reducible to its own fusion. We may also have $X \subsetneq A$, in which case only a proper subset of A is reducible via the fusion of all of A .

The requirement $|A| > 1$ means that a singleton set is not taken to be fusionally reducible to itself. A singleton subset *may* be reducible in context: for example, with $A = \{(W,L,W), (e,W,L)\}$, $X = \{(W,L,W)\}$, we have that X is reducible via $fA = (W,L,L)$.

3 The Fusional Reduction Algorithm: FRed

FRed and the objects it creates are developed and explored.

3.1 Fusional Reduction and the MIB

Our primary goal is to calculate the maximally concise, maximally informative representation of the content of any set of ERCs: its Most Informative Basis (MIB). ‘Conciseness’ means precisely that a basis contains the fewest possible number of ERCs. The demand for conciseness has two distinct consequences for the general character of any basis, MIB or not. First, the elements of a basis must be logically independent, in the sense that none of them can be entailed by the rest. Any entailed vector is superfluous, adding nothing to the content, and can be removed—producing a smaller, equivalent set. But no set of ERCs equivalent to the reference set can be smaller than a basis. Second, it follows from conciseness that no basis can contain fusionally-reducible subsets of cardinality greater than one. If a multi-element subset of a certain vector set is fusionally reducible, we can replace it with a fusion, a single vector, thereby reducing the cardinality, and showing that what we had originally couldn’t have been a basis. These two properties, logical independence and fusional irreducibility, cannot be unified.

To see this, observe first that a set can be fusionally irreducible yet contain logically dependent vectors. Consider the following two-element set, taken in isolation:

- (83) **Fusionally irreducible and logically dependent.**

	1	2	3	ERC
a	W	W	L	$1 \gg 3 \vee 2 \gg 3$
b	W	e	L	$1 \gg 3$
a∘b	W	W	L	$1 \gg 3 \vee 2 \gg 3$

The set $\{a, b\}$ is logically dependent, i.e. redundant: **b** entails **a** by W-extension. However, its fusion is identical to the weaker, entailed vector **a**. Therefore, the set $\{a, b\}$ cannot be replaced by its fusion **a∘b** without loss of information.

Conversely, a fusionally reducible set can perfectly well be logically independent, as in the following example:

(84) **Fusionally reducible and logically independent.**

	1	2	3	ERC
a	W	L	e	1 » 2
b	W	e	L	1 » 3
a ◦ b	W	L	L	1 »{ 2,3 }

This is the simplest case of fusional reducibility, a W-compliant set. Here, neither element entails the other, yet their collective content is entirely present in their fusion. A more intricate case of the same phenomenon is seen in the collection of independent disjunctive ERCs, {(W,L,W),(W,W,L)}. Neither entails the other, but their fusion, which has the same form as in (84), entails them both. In the more general case of fusional reducibility illustrated in ex. (68) above, where a proper subset reduces to the fusion of a containing superset, the elements of the reducible set are also logically independent.

The **MIB** is distinguished among bases in that its elements are *maximally informative*. Informativeness is determined by entailment: an entailing ERC vector is strictly more informative than any entailee distinct from it (choosing all of these from among the consequences of a reference set). The maximal informativeness requirement on the **MIB** means that each ERC in the **MIB** is entailed only by itself. Maximal informativeness induces further structure: the **MIB** doesn't even tolerate subsets of cardinality *one* that are fusionally reducible to something else: any such would consist of an individual ERC than can be replaced, in context, with another ERC that asymmetrically entails it.

From such global requirements on basis and **MIB** structure, we can assemble a concrete picture of what individual **MIB** elements look like. Each vector in the **MIB** describes the complete ranking requirements of a unique, minimal W-set.

- (I) **Uniqueness of the W-set.** The W-set of any basis vector must be different from all the other W-sets of any of its fellow basis elements. Two vectors with the same W-set will be W-compliant, and therefore reducible to one, contradicting the minimal cardinality requirement. This holds of all kinds of bases.
- (II) **Completeness of the ranking requirements.** In the **MIB**, the L-set attached to a W-set must be maximal: broadly put, you can't add an L to any basis element in the **MIB**, preserving equivalence with the reference set. Specifically, for any vector **b** in the **MIB**, there can be no other vector **x**, among all of the **MIB**'s entailments, with the same W-set and with an L-set that properly includes that of **b**. In this case, **b** would be asymmetrically entailed by **x**, via L-retraction (38) and thus not maximally informative. The **MIB**, then, consists of a set of vectors with distinct W-sets, each one with the maximal L-set allowed for by its proprietary W-set.
- (III) **Minimality of the W-set.** Accompanying the maximality property of the L-set of a **MIB** vector is the minimality displayed of its W-set. Given the domination requirements on the W-set of any **MIB** vector, as determined by the reference set, that W-set must be minimal, in the sense that it cannot be replaced by any

proper subset of itself. Were it to be so replaced, the result would asymmetrically entail the original, not a possibility if we are looking at the **MIB** in the first place. The same property shows up in the Skeletal Basis: inserting a new W either diminishes the number of e 's, de-skeletalizing, or diminishes the number of L 's, losing an irreplaceable ranking requirement.

There is inevitably a gap, which may even be a chasm, between knowing the structure of an object, knowing what it looks like, and being able to produce it on demand. Up to this point, we have only interpreted the *basis* in various ways; now we want to change the way linguistic analysis can be done by providing an effective, infallible method of creating the **MIB** from data.

The tools we have in hand are fusion and the single-vector entailment scale $L \rightarrow e \rightarrow W$. They will suffice. The **MIB**-producing Fusional Reduction Algorithm (FRed) works, in broad outline, like this:

- a. Fuse everything.
- b. Keep the fusion if it is informative.
- c. Recurse on all vector subsets from which information has been lost.

More specifically, it runs like this:

1. Fuse everything.
2. Use the fusion to identify:
 - each info loss configuration (a constraint assessing W and e)
 - the related info loss residue (the set of vectors containing the fusion's info loss)
3. Check for entailment of the fusion by the total residue; keep the fusion if unentailed.
4. Recurse on *each* info loss residue.

The first step hunts for sets that are fusionally reducible in the context of the whole (including single less-than-maximally informative vectors). The second step finds the info loss configurations and their residues (the vectors from which information is lost in the fusion of the whole). The third step determines whether the fusion—concomitantly, the vector subset whose content it encodes—is itself entailed by other vectors, in particular by those whose info is lost in the fusion. If unentailed, it is kept as part of the output of FRed; it has fusionally reduced some vectors whose content is not otherwise represented in the reference set. The recursion step repeats the same process on all subsets of vectors from which information has been lost: those that create info loss configurations. In this way, FRed traverses the entire set, digging out, condensing, and locally strengthening all of its ranking requirements. The collection of retained fusions turns out to be exactly the **MIB**.

In the general case, the process applies repeatedly, fusing smaller and smaller subsets until none are left. Nowhere in the algorithm are the defining features of the **MIB** explicitly checked; and even the property of logical independence—a global property of bases—is examined only locally (at step 3). Observe that the object computed by FRed, which we will call the 'Fusional Normal Form' (**FNF**) of the original set, is conceptually distinct from the **MIB**, which is defined purely in terms of its relation

to the reference set, without reference to fusion. We must therefore show that these two objects are identical. This is demonstrated in Prince and Brasoveanu 2010, along with a number of other properties of the **MIB**, including existence and uniqueness. Here we will focus on presenting the algorithm itself.

To see how maximal informativeness emerges, it is instructive to examine the touchstone $1 \gg 2 \gg 3$ case. Suppose—only for simplicity of illustration; there are no requirements on the reference set other than that it be consistent—that we start out with the Least Informative Basis, which we will rename ‘**A**’. Let us step through FRed with it.

(85) **LIB for $1 \gg 2 \gg 3$.**

A	1	2	3
a'	W	L	W
b	<i>e</i>	W	L

Step 1: Fuse

FRed’s first step is to form the fusion of the whole. This step has the dual role of condensing fusionally reducible sets and obtaining maximal informativeness.

(86) **Fusion of the whole of **A**.**

A	1	2	3
a'	W	L	W
b	<i>e</i>	W	L
a'◦b	W	L	L

Step 2: Identify lost information

Constraint **1** is an info loss configuration, as indicated in (87) below. This configuration implicates vector **b** as the info loser—all information provided by **b** about its W-set is indeed lost in the fusion. Let us notate the info loss residue of constraint *n* in a set X as ‘ X_n ’. In the case at hand, there is only one such residue, $A_1 = \{\mathbf{b}\}$.

(87) **Info loss configuration in **A**: $A_1 = \{\mathbf{b}\}$.**

A	1	2	3	Remarks
a'	W	L	W	{ a' } is fusionally reducible
b	<i>e</i>	W	L	{ b } is the info loss residue A_1
a'◦b	W	L	L	

Step 3: Check for entailment

Is the fusion entailed by the collectivity of lost information, here just A_1 , namely $\{\mathbf{b}\}$, whose fusion is identical to its sole member?

(88) **Info loss residue A_1 .**

A_1	1	2	3
b	<i>e</i>	W	L

Entailment fails: $\{b\} \neq fA$. Specifically:

$$(e, W, L) \not\rightarrow (W, L, L) \quad \text{nb: } 2: W \rightarrow L$$

The untailed fusion fA is therefore retained as an element of the **FNF** of our original set **A**.

Step 4: Recursion

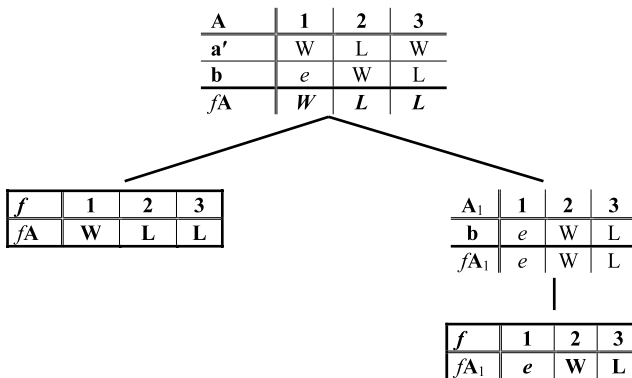
We now continue recursively with the info loss residue A_1 . In this case, the process is trivial, since A_1 contains a single vector. The fusion of A_1 is safely computable by inspection. There is no info loss residue, and no entailment to check. We add this fusion to the **FNF**, and we are done, since there is nothing left for FRed to operate on. The **FNF** of **A**, and therefore its **MIB**, which we have just calculated, is as follows:

(89) $MIB(A) = \{fA, fA_1\} = \{a' \circ b, b\}$.

MIB(A)	1	2	3
$a' \circ b$	W	L	L
b	<i>e</i>	W	L

Application of FRed can be represented perspicuously as the construction of a tree which tracks the stages by which the algorithm generates the **FNF**. We begin with the original set as the root. We place its fusion as the leftmost branch, and any info loss residues as sibling nodes to it. These residue nodes may then be the site of further recursive expansion of the tree. The processing of **A** by FRed, which we have just reviewed, would look like this:

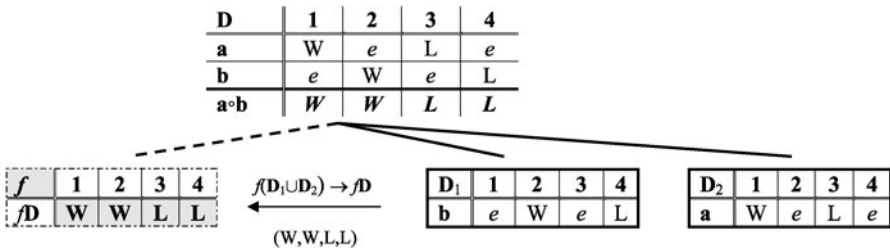
(90) **FRed(A) as a tree.**



The terminal nodes of the tree (boxed) give the **MIB**. For ease of checking the calculation, we also show the fusion of the whole as the bottom row inside the node from which it arises. In the interests of painstaking exactitude, we have drawn out the final step (right branch). We will suppress it in future, being content to box any node that contains a single vector.

To see how the entailment check step can be crucial to the success of the algorithm, let us consider a typical case in which fusion of the whole proves *uninformative*:

(91) **Crucial entailment** in the course of FRed.



Step 1 produces the fusion of the whole, fD . Step 2 assesses its independence from the totality of the info loss residues. We know that the residues must be retained and examined, because their content is surely unentailed by the fusion of the whole. The hanging question is whether they contain in themselves, jointly or severally, all the info that fD encapsulates. Since their fusion entails (indeed, is identical to) the fusion of the whole, we know that it can be abandoned. It is therefore ejected from the **FNF**, a fate we mark by shading and line style.

How is the entailment relation to be determined? The fusion of the whole divides the original set into two disjoint parts: the fusionally-reducible subset—possibly null—which it entails and replaces, and the collection of info loss residues (one for each constraint that is an info loss configuration), which we can call ‘the total residue’ (TR). What matters for the **MIB** is that a retained fusion not be entailed by the total residue. If it were so entailed, it would be redundant with respect to other fusions yet to be calculated. Retaining an entailed fusion would destroy the logical independence of the output of FRed.

The total residue can be a set of any size, given the right starting point. From the fundamental entailment/arrow relation (56), we know that for nontrivial ERCs, entailment from the total residue amounts to arrowing from the fusion of *some subset* of the residue. The appearance of the term ‘some subset’ might suggest the risk of a lengthy calculation. But in this situation, it turns out that we can simply fuse the *entire* residue and use that fusion to check for entailment. Valid and invalid vectors aside, the entailment check runs like this, writing ‘**TR(A)**’ for the total residue of **A**, the union of all its info loss residues.

(92) **Entailment check** (nontrivials). If $fTR(A) \rightarrow fA$, omit fA from the **FNF**. Else, keep it.

As for the practicalities of implementation: we need merely count the raw number of L’s in fA and $fTR(A)$. If they are exactly the same then the fusion of the whole is entailed. If not, then not.

Broadly put: when L-numbers are equal, all the subordination information is present in the residues, and it is only the residues that need to be consulted to determine ranking. To see why this qualitative sense of things is correct, consider the

formal relation between the two fusions, which is tightly restricted because the total residue is *part* of the whole.

Let’s map out the generic possibilities. We make no prior assumption about the existence or nonexistence of an entailment relation between the fusion of the whole and the fusion of the total residue. ‘**FR**’ names the fusionally reducible subset of **A**, ‘**TR**’ its total residue.

(93) **Possible coordinate-wise relations.**

$f\mathbf{A} = f(\mathbf{FR} \cup \mathbf{TR})$	e	W	L
$f\mathbf{TR}$	e	e, W	e, W, L

Each column represents the possible distribution of values in corresponding coordinates; we survey the values in $f\mathbf{A}$ and ask what can coexist in $f\mathbf{TR}$. The presence of e in $f\mathbf{A}$ poses no problems for entailment (first column). W is entailed by anything anyway (second column).

Everything rests on the third column, where the values of $f\mathbf{TR}$ are unconstrained by the value observed in the fusion of its superset. The key fact is this: *L can only be entailed by L*. The only way entailment can fail is when $f\mathbf{A}$ has L and $f\mathbf{TR}$ has e or W . (That is: when the L of $f\mathbf{A}$ comes from some ERC that is not in the total residue.) In this case, $f\mathbf{TR}$ has *fewer* L ’s than $f\mathbf{A}$. If entailment succeeds, the L ’s match up one for one, and their total is perforce the same. Entailment from $f\mathbf{TR}$ to $f\mathbf{A}$ can only be by W -extension (possibly trivial, when the two are identical),²³ when the two expressions have identical L content.

What of the trivial vectors, valid and invalid? These arise frequently enough in the course of empirical investigation, but do not belong in bases. FRed detects them naturally when they threaten to enter the **FNF**, and the algorithm can easily be outfitted to handle them with an extra clause or two. A set consisting entirely of valid vectors may show up as an info loss residue (or even, in principle, as the starting point of the whole process, though only the most insouciant investigator could fail to notice the uniform W^* glare of harmonic bounding that renders every ranking argument vacuous). When a *valid* set is encountered, its fusion should be excluded: it is entailed by anything, including nothing.

An unsatisfiable set is a more alarming discovery, since it indicates failure of the constraint set to analyze the data. When FRed encounters an unsatisfiable set, it should issue an announcement to that effect, and quit, producing no output: an unsatisfiable set does not have a basis. Every unsatisfiable set contains a subset fusing to L^+ , as noted in (58). Any such subset will be isolated and fused in the course of FRed, revealing its identity.

With these considerations in mind, the entailment check step can be rendered as follows:

²³The reader might wish to confirm, by means of a similar calculation, that when $f\mathbf{A}$ entails a vector in the subset of **A** that it fusionally reduces, this must involve L -retraction (perhaps then followed by W -extension from the result).

(94) **Entailment check on a set \mathbf{A} .**

- a. Trivial-True. If $f\mathbf{A} \in W^*$, then omit $f\mathbf{A}$.
- b. Trivial-False. If $f\mathbf{A} \in L^+$, EXIT and announce: ‘ \mathbf{A} is unsatisfiable.’
- c. Else: If $f\mathbf{TR} \rightarrow f\mathbf{A}$, then omit $f\mathbf{A}$ from the **FNF**. If not, retain it.

Here is a complete statement of FRed. The discussion of examples below will clarify the formal steps and the concise comments beside them.

(95) **FRed**

- 0. **Base Step.** #Terminate when \mathbf{A} is empty.
If $\mathbf{A} = \emptyset$ then $\mathbf{FNF}(\mathbf{A}) := \emptyset$
- 1. **Fuse All.** #Collect the fusion of the whole.
 $\text{HoldFus} := \{f\mathbf{A}\}$
- 2. **Identify Lost Information.** #Collect the info loss residues.
 $\mathbf{ILC}(\mathbf{A}) := \{i \mid C_i \in \Sigma \ \& \ f\mathbf{A}[i] = W \ \& \ \exists v \in \mathbf{A} \text{ such that } v[i] = e\}$
 $\mathbf{Res}(\mathbf{A}, i) := \{v \mid v \in \mathbf{A} \ \& \ i \in \mathbf{ILC}(\mathbf{A}) \ \& \ v[i] = e\}$
 $\mathbf{TR}(\mathbf{A}) := \cup_i \mathbf{Res}(\mathbf{A}, i)$
- 3. **Check Entailment.** #Omit trivial and entailed fusions.
If $f\mathbf{A} \in W^*$ then $\text{HoldFus} := \emptyset$
If $f\mathbf{A} \in L^+$, EXIT and announce: ‘ \mathbf{A} is unsatisfiable.’
If $f\mathbf{TR}(\mathbf{A}) \rightarrow f\mathbf{A}$ then $\text{HoldFus} := \emptyset$
Else, $\mathbf{FNF}(\mathbf{A}) := \text{HoldFus}$
- 4. **Recurse.** #Do FRed on each residue.
 $\mathbf{FNF}(\mathbf{A}) := \text{HoldFus} \cup \cup_k \mathbf{FNF}(\mathbf{Res}(\mathbf{A}, k))$

With FRed written out, a formal infelicity in the statement of step 2 becomes visible. There is no need to be so particular about collecting info loss configurations. We could just as well drop reference to the ILC entirely and define a residue for every coordinate that fuses to W, tolerating the empty residues that arise when a W in the fusion comes only from Ws in its component fusands. The Base Step of FRed as stated will process a null residue without complaint. Refining the formulation in this way would cause the FRed tree to become bushier, but it will not have material effects either on the outcome or on the essential features of the process. We therefore put it aside, revisiting it in Sect. 4, where it will become quite useful in clarifying FRed’s relation to RCD and constraint ranking.

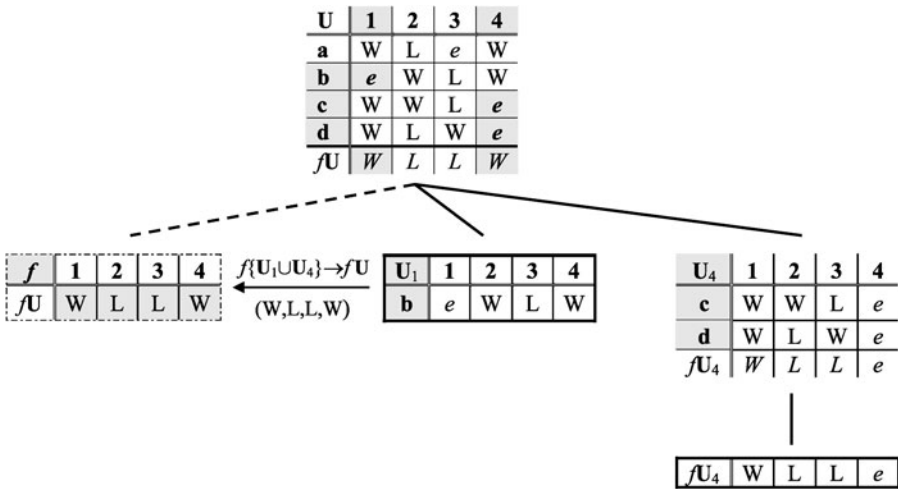
Let’s conclude the illustration of FRed with a couple of examples where recursion does more than report the obvious. Consider the following set of ERC vectors:

(96) **A set \mathbf{U} of vectors.**

U	1	2	3	4
a	W	L	e	W
b	e	W	L	W
c	W	W	L	e
d	W	L	W	e

FRed’s run is pictured in the following tree:

(97) **FRed(U) as a tree.**



Of note in the first round of daughterly processing is the discovery that the whole fusion is redundant (step 3: entailment check). This encapsulates the fact that the fusionally reducible subset $\{a\} = \{(W, L, e, W)\}$ is entailed by its complement, the total info loss residue. Although no single vector in the residue is an entailer, FRed finds the collective entailment without difficulty.

In the second, recursive round of processing, FRed ascertains that the W-compliant subset U_4 can be completely reduced to its fusion. There are no info loss residues, and the procedure terminates, having discovered that the original reference set of four vectors describes a ranking system that requires only two ERCs.

As a final example, let's look closely at a case where FRed has to work recursively and nontrivially on residues—Lombardi's Swedish analysis.

(98) **Round 1: FRed(S), step 1: Fuse All.**

S		1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a.	stekde → stek.te ~ steg.de	e		W	L
b.	sku : g → sku : g ~ sku : k	e	W	L	
c.	vigsəl → vik.səl ~ vig.səl	W	L	W	
d.	vigsəl → vik.səl ~ vig.zəl	e		W	W
e.	ægde → æg.de ~ æk.te	e	W	L	W
f.	stekde → stek.te ~ stek.de	W	L	W	L
g.	ægde → æg.de ~ æk.de	W	W	L	
	fS	W	L	L	L

To focus the info loss, we have omitted most of the e's in the tableau. The crucial cells inducing info loss are highlighted. Only constraint 1 has an info loss residue, which we collect:

(99) **Round 1: FRed(S), step 2: Collect Info Loss Residue of S:** $S_1 = \{a,b,d,e\}$.

S_1		1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a.	stekde \rightarrow stek.te \sim steg.de	e		W	L
b.	sku : g \rightarrow sku : g \sim sku : k	e	W	L	
d.	vigsæl \rightarrow vik.sæl \sim vig.zæl	e		W	W
e.	ægde \rightarrow æg.de \sim æk.te	e	W	L	W

For step 3, we fuse the total residue—here, just S_1 —to enable entailment checking.

$$fTR(S) = fS_1 = (e,W,L,L)$$

It is evident that fS , the fusion of the original set, namely (W,L,L,L), is unentailed.

(100) **Round 1: FRed(S), step 3: Entailment Check.** $fTR(S) \nrightarrow fS$. No action taken.

We now execute the recursion step:

(101) **Round 1: FRed(S), step 4: Recurse:** $FNF(S) := \{fS\} \cup FNF(S_1)$

We must now start all over again with S_1 . As always, we begin with fusion:

(102) **Round 2: FRed(S_1), step 1: Fuse All.**

S_1		1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a.	stekde \rightarrow stek.te \sim steg.de		e	W	L
b.	sku : g \rightarrow sku : g \sim sku : k		W	L	
d.	vigsæl \rightarrow vik.sæl \sim vig.zæl		e	W	W
e.	ægde \rightarrow æg.de \sim æk.te		W	L	W
	fS_1		W	L	L

Only constraint 2 shows info loss. This gives rise to the info loss residue of S_1 , namely S_{12} .

(103) **Round 2: FRed(S_1), step 2: Collect Info Loss Residue of S_1 :** S_{12} .

S_{12}		1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a.	stekde \rightarrow stek.te \sim steg.de		e	W	L
d.	vigsæl \rightarrow vik.sæl \sim vig.zæl		e	W	W

To perform the entailment check, we fuse the total residue of S_1 , which is mercifully just S_{12} .

$$fTR(S_1) = fS_{12} = (e,e,W,L) \nrightarrow (e,W,L,L) = fS_1$$

(104) **Round 2: FRed(S_1), step 3: Entailment Check.** $fTR(S_1) \nrightarrow fS_1$. No action taken.

We advance to the recursion step:

(105) **Round 2: FRed(S_1), step 4: Recurse.** $FNF(S_1) := \{fS_1\} \cup FNF(S_{12})$.

Recursive application of FRed to S_{12} begins with its fusion, which is clearly unentailed, and furthermore indicates that there is no residue.

(106) **Round 3: FRed(S_{12}), step 1: Fuse All.**

S_{12}	1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a. stekde \rightarrow stek.te \sim stegde			W	L
d. vigsæl \rightarrow vik.sæl \sim vig.zæl			W	W
fS_{12}			W	L

The final step is taken:

(107) **Round 3: FRed(S_{12}), step 4: Recurse. $FNF(S_{12}) = \{fS_{12}\} \cup FNF(\emptyset)$.**

The whole process ends abruptly with the 0th step of the next recursion, which informs us that the FNF of the null set is itself. Putting all the pieces together, we are licensed to conclude that for S , the FNF (and therefore the MIB) consists of the fusions we have recursively calculated:

(108) $FNF(S) = \{fS, fS_1, fS_{12}\} = \{a \circ b \circ c \circ d \circ e \circ f \circ g, a \circ b \circ d \circ e, a \circ d\}$

Discussion. Of note in the derivation is the deft elimination of redundant elements. On the first round, three vectors are compressed into one. Here they are:

(109) **Round 1. Fusionally reducible subset.**

c. vigsæl \rightarrow vik.sæl \sim vig.sæl	W	L	W	
f. stekde \rightarrow stek.te \sim stek.de	W	L	W	L
g. ægde \rightarrow æg.de \sim æk.te	W		L	

This is a W-compliant set, and therefore exactly equivalent to its fusion (W,L,L,L). Observe that entailment from the fusion proceeds via L-retraction, as noted in fn. 23.

On the second round, as shown in ex. (102), another two vectors are compressed:

(110) **Round 2. Fusionally reducible subset.**

b. sku : g \rightarrow sku : g \sim sku : k		W	L	
e. ægde \rightarrow æg.de \sim æk.te		W	L	W

These are not W-compliant internally, but they are fusionally reducible in the context of S_1 . Notice also the entailment relation between **b** and **e**. FRed eliminates a redundancy involving entailment, even though it is not specifically looking for it here.

The third round compresses two more vectors, once again W-compliant:

(111) **Round 3. Fusionally reducible subset.**

S_{12}	1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a. stekde \rightarrow stek.te \sim steg.de			W	L
d. vigsæl \rightarrow vik.sæl \sim vig.zæl			W	W

The trivial, valid vector **d** shows up in this residue and is forthwith reduced, even though the algorithm never takes explicit note of it.

There is a final noteworthy property of this example, which is utterly characteristic of FRed: the *disclosure* of info loss residues as recursion progresses. When we advance to the info loss residue of the first round, the fusionally reducible subset $\{\mathbf{c}, \mathbf{f}, \mathbf{g}\}$, presented in (109), is gone, taking its L coordinates with it. Crucially, all L values now disappear from constraint $\mathbf{2}$, rendering it an info loss configuration, due to the e coordinates $\mathbf{a}[2]$ and $\mathbf{d}[2]$. Thus, the set $\{\mathbf{a}, \mathbf{d}\}$ pops up as an info loss residue which must be inspected for ranking information in Round 3. Such newly disclosed residues provide the fodder for further recursion.

We conclude with a few remarks on features of FRed.

Entailment Check. Entailment check asks whether the total residue (TR)—the union of all info loss residues—entails the fusion of the whole. In the general case, we are only guaranteed that any entailment from a set follows from the fusion of some subset. In FRed, it suffices to fuse the *entire* total residue and compare that with the fusion of the whole; no laborious prospecting for subsets of the TR is needed. This special situation arises because the vectors in the total residue are included in the whole, so that the following strong result applies:

(112) **Fusion entailed by fusion.** Let \mathbf{A} be an ERC vector set, such that $f\mathbf{A} \notin W^*$. Let $\mathbf{X} \subseteq \mathbf{A}$ be such that $f\mathbf{X} \notin L^+$. Then:

$$\mathbf{X} \models f\mathbf{A} \text{ iff } f\mathbf{X} \rightarrow f\mathbf{A}.$$

Proof. See Prince and Brasoveanu 2010.

A further point of interest: conducting the check very locally suffices to determine whether entailment of the whole fusion occurs *anywhere*. This is not too surprising at the very first step of the algorithm, where the entire set under consideration is split into two parts. But after that, in the full-blown general case, there can be many, many info loss residues, each sprouting many further residues, generating a massively branching derivation tree, in which entailment could, one is tempted to imagine, lurk in some far-distant node or, worse, be spread in fragments throughout a number of them. But any entailing set must always end up entirely represented in the total info loss residue that is sister to the fusion, and no global traverse is necessary to uncover entailment.

This follows from the way residues separate off during the descent through the telescoping sequence of proper subsets as we move down the branch from the root to the residue \mathbf{R} that yields the target entailed fusion $f\mathbf{R}$. Every vector that is in the complement of \mathbf{R} must have a W where \mathbf{R} has an e . Furthermore, since a residue comes from a column consisting of only W and e , these entailment-denying Ws cannot be rescued through fusion with another ERC with L at the same coordinate: such a vector simply doesn't exist. Further perspective on this property, relating it to ranking structure, is given in Sect. 4 below.

Residues. FRed can handle all kinds of relations between info loss residues. Residues may overlap, stand in subset–superset relations, or even be identical. Here's an example which shows the variety of configurations available:

(113) **Relations between info loss configurations.**

A	1	2	3	4	5
a	W	W	<i>e</i>	L	
b	W	<i>e</i>	<i>e</i>		
c	<i>e</i>	<i>e</i>	W	L	L
<i>fA</i>	W	W	W	L	L

Residues Relations
 $A_1 = \{c\}$ $A_1 \subseteq A_2$
 $A_2 = \{b, c\}$ $A_2 \cap A_3 = \{b\}$
 $A_3 = \{a, b\}$ $A_1 \cap A_3 = \emptyset$

Because we must pursue every info loss configuration, we are necessarily going to encounter repetition in the branches of the process. For example, a little calculation shows these residue duplications implicit in (113):

$$A_{21} = \{c\} = A_1$$

$$A_{23} = \{b\} = A_{32}$$

FRed sails through all such relations smoothly, repeating itself exactly over identical residues, no matter how they are derived. Since the **FNF** is the *set* of collected fusions, any duplications will have no effect on the outcome.

Let’s glance back now, with FRed well in hand, at the properties of the **MIB** as an object (Sect. 3.1), to see qualitatively how they emerge. (I) The *Uniqueness of the W-set* in each **MIB** vector follows because FRed collapses all fusionally reducible sets: should two vectors share a common W-set, they will be W-compliant, and hence fusionally reducible, and hence fusionally reduced. (II) The *Completeness of the Ranking Requirements* is due to the maximization of the number of L’s in the **MIB** vectors, which comes about because each such vector is derived by fusion, which by definition preserves all L’s in its fusands. (III) The *Minimality of the W-set* follows jointly from the fusional maximization of L-content, on the one hand, and the maximization of *e*’s on the other. This latter is achieved in the course of the relentless pursuit of every info loss configuration, using their info loss residues to eject *via* the entailment check step precisely those redundant fusions with excessive numbers of W’s.

3.2 The Skeletal Basis

The virtues of the Skeletal Basis are both theoretical and empirical. We will find that FRed can produce it with a simple modification in one clause.

Each member of the **MIB** describes the entirety of ranking relations of a unique W-set. Recall the **MIB** for $1 \gg 2 \gg 3 \gg 4$, as in Lombardi’s Swedish:

(114) **MIB for $1 \gg 2 \gg 3 \gg 4$.**

	1	2	3	4
a	W	L	L	L
b		W	L	L
c			W	L

From this, we can directly read off everything that constraint **1** must dominate, and so on for all the others. But such maximal local informativeness also carries a cost: from vector **a** alone we cannot tell which of constraint **1**'s necessary subordinates it must *immediately* dominate. Because a **MIB** vector contains all the information derivable from collective interactions, in this case from transitivity, it lumps the dominated constraints together into one class.

Among the potentially many bases for a reference set, there is another extremal type, the one that contains *no* information derivable from transitivity or other L-introducing interactions. This is the Skeletal Basis, which contains the sparsest possible deployment of W's and L's. For $1 \gg 2 \gg 3 \gg 4$, it will look like this:

(115) **Skeletal Basis for $1 \gg 2 \gg 3 \gg 4$.**

	1	2	3	4
a	W	L		
b		W	L	
c			W	L

Only the immediate domination requirements on the W-set of each component vector are represented. The Skeletal Basis is characterized by having as many as *e*'s as possible.

In certain respects, the Skeletal Basis gives a clearer view of the ranking situation than the **MIB**. Most obviously, when it's possible to construct a Hasse diagram of ranking relations, the Skeletal Basis provides what's needed to connect adjacent nodes, and no more.²⁴ To work from the **MIB** would require erasing from the diagram all connections due to transitivity. The **MIB** provides the transitive closure of the ranking relations; the Skeletal Basis provides the 'transitive reduction'. But more is at stake than transitivity. Consider the subtle effect seen in the following case:

(116) **A twist on the 'informativeness' of the MIB.**

MIB	1	2	3	4
a	W	W	L	L
b	W	<i>e</i>	<i>e</i>	L

This is a **MIB**: neither vector entails the other, nor can they be fusionally reduced. Nevertheless, there is an excessively weak local statement lurking here. ERC [**a**] holds that **1** *or* **2** must dominate **4**. So: ERC [**a**] tells us that **1** *may* dominate **4**. Yet [**b**] asserts that **1** *must* dominate **4**.

Let us construct the Skeletal Basis for the same ranking system:

²⁴The Skeletal Basis, when no disjunctions are present, provides the incidence matrix of the associated Hasse diagram, which is just a directed graph. Whenever an irreducible disjunction is present, we don't have a single partial order, but a set of them, requiring multiple Hasse diagrams. In general, an ERC set determines a hypergraph rather than a graph, of which the **SKB** is the incidence matrix. See Prince 2006b: 4; 2008a, for discussion.

(117) **The Skeletal Basis.**

SKB	1	2	3	4
a'	W	W	L	<i>e</i>
b	W	<i>e</i>	<i>e</i>	L

The **MIB** can be recovered by replacing **a'** with $\mathbf{a}' \circ \mathbf{b} = \mathbf{a}$, as FRed would. The Skeletal Basis has eliminated the excessively weak local assertion from **a** in favor of the stronger requirement imposed by **b**. Transitivity is involved rather subtly in this argument, which also rests on the linearity of ranking order in a grammar.

Perhaps the easiest way to see what's going on is to blow the ERC vector **a** up into two *W*-compliant components, two 'Primitive Ranking Conditions', in the terms of Prince 2006c.

(118) **Conjunctive blow-up: $\mathbf{a} = \mathbf{a}_1 \circ \mathbf{a}_2 = \mathbf{a}_1 \ \& \ \mathbf{a}_2$.**

	1	2	3	4
a₁	W	W	L	<i>e</i>
a₂	W	W	<i>e</i>	L
b	W	<i>e</i>	<i>e</i>	L

From this it is clear that $\mathbf{b} \rightarrow \mathbf{a}_2$ by *W*-extension. In short: although **b** does not entail **a**, it entails an implicit conjunctive component within **a**. The Skeletal Basis rids us of this enfeebled conjunct. Thus, if we want to grasp all the ranking relations in their detail, we have good reason to examine the Skeletal Basis.

How then to find the Skeletal Basis? No operation analogous to fusion will combine pieces of data to produce it. Happily, FRed produces the relevant information in the course of its run, exactly where the fusion of the whole is compared with the fusion of the total residue. As found above in examining the entailment check step, the fusion of the whole will always contain a superset of the *L* coordinates that lie in the fusion of the total residue. Every *L* in the total residue must appear as *L* in the fusion of the whole, and non-residue ERCs may provide a few more, as illustrated in table (93). If we simply strike out all those *L*'s that are shared between the two, we will have 'skeletalized' the fusion of the whole. Logically speaking, we are simply removing entailed conjuncts from the fusion of the whole.

To see how this works, let's continue with $\mathbf{1} \gg \mathbf{2} \gg \mathbf{3} \gg \mathbf{4}$, using Lombardi's Swedish as a source.

(119) **Fusion of the whole.**

a.	<i>e</i>		W	L
b.	<i>e</i>	W	L	
c.	W	L	W	
d.	<i>e</i>		W	W
e.	<i>e</i>	W	L	W
f.	W	L	W	L
g.	W	W	L	
<i>fS</i>	W	L	L	L

The total residue is merely the residue of constraint **1**, namely $S_1 = \{a,b,d,e\}$. Its fusion is computed here:

(120) **The total residue of Round 1.**

a.			W	L
b.		W	L	
d.			W	W
e.		W	L	W
fS_1		W	L	L

Striking out the L's shared by the two fusions, which occur in constraints **3** and **4**, will yield a member of the Skeletal Basis. We write $x \setminus y$ for 'x skeletalized by y', that is: x with all of its L's shared with y reduced to e.

(121) **Skeletalization.**

fS	W	L	L	L
fS_1		W	L	L
$fS \setminus fS_1$	W	L	e	e

Skeletalization may appear to be a wildly abstract gesture, since it takes place high atop a tower of formal constructions. But it leads directly back to the data. The skeletalized vector indexes the contrasts that give rise to the ranking relation it expresses. In the case at hand, it is composed as follows:

$$fS \setminus fS_1 = a \circ b \circ c \circ d \circ e \circ f \circ g \setminus a \circ b \circ d \circ e$$

This identifies the set $\{c,f,g\}$ as containing the responsible parties, and $\{a,b,d,e\}$ as irrelevant. Going back to the data itself, we mark the W's and L's shared with the Skeletal Basis:

(122) **Content providers for 1 ≫ 2.**

S		1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
c.	vigsæl → vik.sæl ~ vig.sæl	W	L	W	
f.	stekde → stek.te ~ stek.de	W	L	W	L
g.	ægde → æg.de ~ æk.de	W	W	L	

A glance at this collection reveals that vector **g** tells us nothing about **1 ≫ 2**; therefore, **c** and **f** are the ones that compel the ranking; in this case, either will suffice to give it. This is not a conclusion that one could confidently assert based on these data in isolation. Both **c** and **f** contain internal disjunctions that locally suggest that **1** is merely a possible dominator of **2**. The fusion of the whole dispels this illusion, but throws the entire data set into the mix, completely obscuring the source of the ranking effect even while establishing its existence.

Pursuing this observation a bit more systematically, let's reconsider the following data from Lombardi's Swedish, repeated from (51), again marking the Skeletal Basis entries.

(123) **Data from Lombardi's Swedish.**

		1: AGREE	2: IDLAR	3: *LAR	4: IDONSLAR
a.	stekde → stek.te ~ steg.de			<i>W</i>	<i>L</i>
e.	ægde → æg.de ~ æk.te		<i>W</i>	<i>L</i>	W
f.	stekde → stek.te ~ stek.de	<i>W</i>	<i>L</i>	W	L

Each vector other than the first contains nonskeletal material, reflecting the intermingling of various properties in the data. The Skeletal Basis allows us to precisely identify the contribution of each individual data contrast, no matter how confounded it may be. At the same time, it identifies the key constraint conflict inherent in the data resolved by the ranking restriction.

Of the vector **f**, for example, we may discern the following properties: (1) The constraint *LAR, which provides a nonskeletal *W* in **f**, is *not* the explanation for the choice of the voiceless cluster in outputs like [stek.te] in this language. (2) The nonskeletal *L* under IDONSLAR is *not* proper to the derivation of this form, but follows from another, broader generalization about the language. This fact is only apparent in the Skeletal Basis, as the nonskeletal *L* in question will appear in the **MIB**. (3) The data contrast in **f** is uniquely involved with the conflict between AGREE and IDLAR.

Globally, comparing the 3 ERC vectors of (123) with the Skeletal Basis establishes that the cited forms, though but a tiny subset of the relevant data, suffice to entirely determine the grammar of voicing in the language. These data, in context, contribute the essential material—*W*'s and *L*'s in the right places—to the Skeletal Basis. Because of this, they demonstrably yield a Support (see fn. 4 for this notion) for the Skeletal Basis, and therefore for the grammar as a whole. The Support (123) is minimal, in that nothing may be removed from it.

Finding data that constitutes a minimal Support for the ranking can be done algorithmically via FRed. For each member of the Skeletal Basis, the data supporting it must lie in a residue whose fusion leads exactly to that basis vector. Compare the ERC vectors in that residue to the Skeletal Basis element that derives from the fusion of the residue, just as in (121) above. Choose enough of them to give the *W*'s and *L*'s in the basis element, and you're set. (In the case at hand, only one data-based ERC vector is required to support each basis vector.) A minimal Support for the grammar need not be unique, but it is invaluable: it identifies a core set of data which controls the analysis of every datum, attested or possible, in the entire language. All the ranking arguments needed for an analysis are contained in such a minimal Support, as is the essential data for understanding how the grammar parses the language.²⁵

The natural place to site skeletalization within FRed is at the entailment check step. As noted above (discussing ex. (93)), the *L*-set of the fusion of the whole is a superset, not necessarily proper, of the *L*-set of fusion of the total residue.

Entailment may therefore be checked in the following way: go through every *L* coordinate of $f^{\mathbf{TR}}$ and replace the corresponding coordinate in $f^{\mathbf{A}}$ with *e*. The end

²⁵The Support-seeking algorithm sketched here has been implemented in OTWorkplace (Prince 2008c; Prince and Tesar 2008).

result is the skeletalization of $f\mathbf{A}$ by $f\mathbf{TR}$. If this is a vector with *no* L's, a member of W^* , a valid vector, then we have $f\mathbf{TR} \rightarrow f\mathbf{A}$.

If not, then the result of this calculation is a member of the Skeletal Basis. We have taken away every L from $f\mathbf{A}$ that can be recovered from the data in the \mathbf{TR} . What's left must be due to the fusionally reducible set that $f\mathbf{A}$ replaces, and indeed only as much of that as is not attributable to anything in the \mathbf{TR} . In short: entailment check itself is no more than a test of $f\mathbf{A} \setminus f\mathbf{TR}$ to see whether it's in W^* .

To incorporate these remarks, we rewrite the entailment check step of FRed (95) as follows:

(124) **FRed: Step 3 Revised. Entailment Check and the Skeletal Basis.**

- If $f\mathbf{A} \in L^+$, EXIT and announce: #Detect inconsistency.
- ‘ \mathbf{A} is unsatisfiable.’
- $\mathbf{s} := f\mathbf{A}$ #Potential Sk. Basis element.
- For every coordinate k , #Eliminate \mathbf{TR} L's from \mathbf{s} .
- if $f\mathbf{TR}[k] = L$ then $\mathbf{s}[k] := e$
- If $\mathbf{s} \in W^*$ then $\text{HoldFus} := \emptyset$ #Omit entailed fusion from \mathbf{MIB} .
- else $\mathbf{SKB} = \mathbf{SKB} \cup \{\mathbf{s}\}$ #Save Sk. Basis element.

Here we construct a potential member \mathbf{s} of the Skeletal Basis from a corresponding potential member of the \mathbf{MIB} . If \mathbf{s} is rendered valid by the L-replacement step, then we forget it and erase the fusion of the whole from the location where it is (temporarily) stored. If \mathbf{s} is *not* trivially valid, then we retain the fusion of the whole for the \mathbf{MIB} and add \mathbf{s} to the Skeletal Basis.

We conclude with a note on the relation of the Skeletal Basis to the system of comparative values. Just as the \mathbf{MIB} displays a maximum number of L's and the Least Informative Basis the maximum number of W's, so the Skeletal Basis has the maximum number of e 's. In the case of the \mathbf{MIB} and the \mathbf{LIB} , these properties connect with the entailment order $L < e < W$. \mathbf{MIB} elements occupy the lowest possible position in the entailment hierarchy given by this order. The elements of the \mathbf{LIB} occupy the highest possible position. Skeletality can be defined by the same strategy, if we turn away from the entailment order and recognize a distinct ‘fusional order’ $<_f$ on the comparative values, according to which $L <_f W <_f e$.

The fusional order earns its name because fusion takes the minimal value in this order (see Meyer 1975: 400, ERA: 55–6; the matter is further explored in Prince and Brasoveanu 2010 when we dig deeper into the formal details of FRed.) Extend the order on values to an order on vectors in the usual coordinate-wise fashion, parallel to the treatment of arrow in (71). Each vector in the Skeletal Basis must then be *maximal in the fusional order* in the sense that no greater vector can replace it, preserving basis status. Compare this with definition given in ex. (37).

- (125) **Skeletal Basis.** \mathbf{B} is the skeletal basis for \mathbf{A} iff \mathbf{B} is a basis for \mathbf{A} , and for every $\mathbf{b} \in \mathbf{B}$ and for every \mathbf{x} such that $\mathbf{A} \models \mathbf{x}$, if $\mathbf{x} >_f \mathbf{b}$ then replacing \mathbf{b} in \mathbf{B} with \mathbf{x} fails to yield a basis for \mathbf{A} .

3.3 Atomic generation

A basis is a minimal cardinality collection of ERC vectors that entails every vector in a reference set. In some cases, entailment may involve a simple relationship between

one ERC vector in the basis and some other drawn from the reference set (or from its consequences). But entailment may on occasion require the participation of more than one vector from the basis. Here's a typical case, repeated from ex. (91):

(126) **Multi-ERC entailment.**

D	1	2	3	4	ERC
a	W	e	L	e	1>>3
b	e	W	e	L	2>>4
a◦b	W	W	L	L	1>>{3,4} ∨ 2>>{3,4}

Take the set **D** as the reference set. It's a **MIB**, and therefore its own **MIB**. Among its consequences is the one encoded in the vector (W,W,L,L). Yet this vector is entailed neither by **a** nor by **b** in isolation. It takes the two of them together to get it.

For certain purposes it may be convenient or even necessary to have at hand a list of ERCs from which everything in the reference set follows via *single* ERC entailment. For example, this is required by the procedure in Merchant 2008 for the extraction of ranking information shared by two grammars, which combines ERCs pairwise, one from each grammar, by the 3-valued 'or' of RM3 (ERA: 51, ex. (99)).

We know that such a set must exist—trivially, the entire set of ERC entailments has this property; it is finite and can be exhaustively listed. More interestingly, we know from the basic theorem of the ERC calculus, stated in ex. (55), that anything entailed by a reference set must follow from some single ERC which is the fusion of one of its subsets. If we therefore collect the 'fusional closure' of the reference set—the entire set of fusions of *all* of its subsets—we surely have all the equipment we need to guarantee single ERC entailment.

Let us call any set **G** that is logically equivalent to a reference set **A** a 'generator' or 'generating set' for **A**. If all entailments can be obtained from **G** via single ERCs, let us call **G** an 'atomic generator'. The fusional closure of **A** is, in these terms, an 'atomic generator' for **A**. Within every atomic generator lies a *minimal* atomic generator (**MAG**), a subset of smallest size that atomically generates **A**. The **MAG** must be unique—it is simply the set of elements of the fusional closure that are minimal in the coordinate-wise entailment order based on $L \rightarrow e \rightarrow W$, and the finiteness of the set gives us uniqueness unproblematically. Furthermore, the **MAG** is a subset of *any* atomic generator, as shown in Prince and Brasoveanu 2010 along with the other claims of this section.

We have seen how FRed produces the **MIB** and the Skeletal Basis: FRed also produces the **MAG**, in passing, as it were. If we suspend the entailment check step and simply collect all fusions produced by FRed, we will have accumulated an atomic generator. Since the fusional products of FRed generate the reference set atomically, and since the **MAG** lies inside any atomic generator, we may obtain it by processing the output of unfettered FRed.

Filtering the **MAG** from the total fusional production of FRed can't be done quite as tidily as the general entailment check step, which settles the global entailment question by looking at just one additional fusion (that of the total residue **TR**). It's tempting to imagine that the task could be accomplished without fusion by looking through all the **TR** vectors individually. But further processing among the residues may crucially collapse some vectors. Here's a simple example:

(127) **Atomic entailment check cannot be entirely local.**

A	1	2	3	4
a	W	W	<i>e</i>	L
b	<i>e</i>	W	L	<i>e</i>
c	<i>e</i>	W	<i>e</i>	L
fA	W	W	L	L

There’s one and only one residue, $A_1 = \{b,c\}$. Neither of its elements atomically entails fA . Nevertheless, in its next round, FRed will find fA_1 , collapsing the W-compliant **b** and **c**, which does entail fA . In fact, the **MAG** for **A** consists of the one ERC (*e,W,L,L*), which is exactly **b•c**.

This example points to the correct way to filter for atomic entailment: every time a new fusion is obtained, you must look back up the branch of FRed it lies on, and check to see whether previous fusions on that branch are entailed by it. But you don’t have to look very far: only up to the fusion of its immediate mother in the FRed tree. Thus, while general entailment check need only examine the fusion of the total residue **TR**, the check for atomic entailment will look at the fusion of each residue separately, unamalgamated with the fusions of its sisterly residues, but need look at nothing more.

To see why this is so, recall that any single branch of FRed involves a superset–subset chain of ERC sets: a residue contains a proper subset of the ERCs in its parent node; that parent a proper subset of its parent, and so on up to the top. This means that a higher node has the L-content of any and all nodes lower than it on the branch, and possibly more. As you descend a branch, L-content can at best stay the same, and may decrease. For the fusion of a lower node to entail the fusion of a higher node on a branch, then, they must exactly have the *same* number and distribution of L-coordinates (*L is entailed only by L*). But as we descend a branch, the *e*-content is strictly *increasing*. A residue A_I , for I some sequence of coordinates, must have *e*’s at every position named in I. A residue A_{IJ} will have all those *e*’s and, in addition, those named in the sequence J. Similarly for A_{IJK} , for a sequence K.

Suppose now that fA_{IJK} entails fA_I . They must have *identical* L-sets. Furthermore, A_{IJ} must share that L-content, since it contains A_{IJK} . Therefore, the L-set of the intermediate node A_{IJ} is identical to that of both A_I and A_{IJK} . Since *e*-content increases strictly from A_I to A_{IJ} to A_{IJK} , and in a proper subsetting fashion, it follows (given constancy of L-content) that W-content *decreases* in the same way. The W-set of fA_I is a strict superset of that of fA_{IJ} . It follows immediately that fA_{IJ} entails fA_I by W-extension. Since we only assumed that fA_{IJK} entails fA_I , this shows that any distal entailment relation along a branch enforces stepwise local entailment relations (by W-extension) all along the branch, running upward.

To close the argument, we note that every atomic entailer for a fusion fX , created by FRed in the processing of reference set **A**, will lie in residues of **X**. That is, no distinct atomic entailer can be lurking in other distinct branches of FRed, or scattered among them. This follows for reasons paralleling those that lead to the similar property shown by general entailment, enabling the very local entailment-check step of FRed (further discussed below).

Putting these considerations together, we arrive at a statement of the type of entailment checking needed if we wish to add the Minimal Atomic Generator (**MAG**) to FRed's output.

(128) **Atomic entailment check.**

Let \mathbf{A} be an ERC set or any of its info loss residues. Add $f\mathbf{A}$ to the **MAG** iff $f\mathbf{A}$ is unentailed by the fusion of any of its info loss residues.

The deeper interest of atomic generation lies in the observation that FRed natively produces the **MAG** among its fusional products. Atomic entailment check eliminates from these products those that are redundant with respect to single-ERC entailment. The general entailment-check step of FRed proper eliminates *all* redundancies, and thereby finds the **MIB** inside the **MAG**. This identifies the **MIB** as a fully nonredundant set of vectors, none of which is atomically entailed by anything other than itself. The **MIB** is therefore definable in logical predicates only, and the minimal cardinality requirement, which we have taken as basic here, can be shown to follow from the logical characterization. This perspective underlies the approach to **MIB** and basis structure developed in Prince and Brasoveanu 2010.

3.3.1 Appendix to 3.3

We conclude with definitions of the new notions introduced in Sect. 3.3.

(129) **Fusional closure.** For an ERC (vector) set \mathbf{A} , the fusional closure of \mathbf{A} , written $\mathbf{fuscl}(\mathbf{A})$, is the set of fusions of every nonempty subset \mathbf{A} .

$$\mathbf{fuscl}(\mathbf{A}) = \{f\mathbf{X} \mid \mathbf{X} \subseteq \mathbf{A}, \mathbf{X} \neq \emptyset\}$$

(130) **Generator/Generating set.** For \mathbf{A} a ERC (vector) set, an ERC (vector) set \mathbf{G} is a generator (generating set) for \mathbf{A} iff \mathbf{G} and \mathbf{A} have exactly the same entailments.

For every ERC (vector) α , $\mathbf{G} \models \alpha$ iff $\mathbf{A} \models \alpha$; that is, $\mathbf{G} \models \mathbf{A}$ and $\mathbf{A} \models \mathbf{G}$.

(131) **Atomic entailment.** A set \mathbf{G} entails an ERC (vector) α *atomically* iff $\exists \beta \in \mathbf{G}$ such that $\beta \models \alpha$.

(132) **Atomic generating set.** \mathbf{G} is an atomic generating set for \mathbf{A} iff \mathbf{G} is a generating set for \mathbf{A} and, for every ERC (vector) α , if $\mathbf{A} \models \alpha$, then there is a $\beta \in \mathbf{G}$ such that $\beta \models \alpha$.

(133) **Minimal Atomic Generator (MAG).** \mathbf{G} is a minimal atomic generator (**MAG**) for \mathbf{A} iff \mathbf{G} is an atomic generator for \mathbf{A} and no proper subset of \mathbf{G} is an atomic generator for \mathbf{A} .

4 FRed and RCD

Here we examine the relation of FRed to RCD. Reversing perspective, RCD is viewed as a gatherer of information, and FRed as an explorer of rankings.

The Ranking Problem can be addressed from different angles, with different goals. Recursive Constraint Demotion (RCD) efficiently delivers a ranking, if one exists, sufficient to satisfy all the ERCs on hand, and also discovers when there is no such ranking. With less efficiency but greater detail, FRed concisely presents the entirety of ranking conditions inherent in an ERC set. It is instructive in both directions to compare the two.

RCD is usually cast as a ranking algorithm, producing a concrete result—a stratified hierarchy, from which a linear ranking may be chosen. FRed, in apparent contrast, collects information *about* rankings, presenting not a hierarchy of any sort but, more abstractly, a set of ERCs. Much of the apparent conceptual distance between the two algorithms is a matter of analytic perspective that can be eliminated by re-considering RCD as an information producer.

The first action in RCD, as in FRed, is to compute the fusion of the whole (ERA: 21–26; Prince 2009). Hierarchy-wise, this tells us which constraints are immediately rankable: those that fuse to *e* or *W*. We may also think of this as identifying those constraints that must be demoted: those that fuse to *L*. Informationally, the fusion of the whole gives us an ERC which summarizes a set of ranking conditions. To normalize RCD to the FRed model, let us put aside the hierarchizing interpretation and proceed informationally: we want to *retain* the fusion of the whole as the output of the first step of RCD, postponing any consideration of ranking per se. In this new conception, the fusion is not a mere tool to determine rankability, but the principal goal of the calculation, as it is in FRed.

RCD in its earliest formulations advances to its recursive step by a kind of double elimination. Those ERCs satisfied by the newly stratified constraints are omitted from further consideration; similarly omitted are those *constraints* that have been ranked. We are then left with a smaller set of ERCs *and* a smaller set of constraints to deal with. FRed, in contrast, works by eliminating only the satisfied ERCs and always keeps the entire set of constraints intact. But this contrast is illusory. The double-elimination conception is rooted in the ‘mark-data pair’ data structure (Tesar and Smolensky 1993; Tesar 1995 *et seq.*), which couples a list of the constraints crucially violated by the desired optimum with another list of those crucially violated by its competitor; that is: the *L*- and *W*-sets of the comparison. When we move to the ERC vector representation, with its explicit identity (*e*), the constraints fall into three sets, not two, with inertness formally recognized as presence of *e*, not absence from the accounting. We need no longer remove the stratified constraints, because they award *e* to the retained ERC vectors and are therefore inert. To see this, consider the following simple example:

(134) **RCD by double elimination (classic).**

I	1	2	3	→	II	1	2	3	→	III	1	2	3
a	W	L	e										
b	e	W	L		b		W	L					

Here each stage removes a constraint and the ERC it explains. That we need only remove ERCs is apparent in this reconceptualization:

(135) **RCD by single elimination.**

I	1	2	3
a	W	L	e
b	e	W	L

→

II	1	2	3
b	e	W	L

→

III	1	2	3

In stage I we find that constraint **1** is rankable. This eliminates vector **a**, leaving **b** to be reckoned with. At stage II, the ranking of constraint **2** satisfies **b**, leading to stage III, a kind of optimality-theoretic nirvana. Free of ERCs, we step off the wheel of recursion. In the process, there is no need to literally remove *constraints* from the scene. At the second stage, as shown in (134), constraint **1** would have been classically removed. Here we simply keep it, as its presence does no harm. We may now conceive of RCD as compiling a set of vectors via fusion, paralleling FRed. Call this the ‘Constraint Demotion Normal Form’ or **CDNF** of the original set.

(136) **Constraint Demotion Normal Form (CDNF).**

CDNF	1	2	3
a b	W	L	L
b	e	W	L

The **CDNF** is exactly the same as the **FNF** in this case. In other cases, of course, they will diverge; for example, the set consisting of (W,e,L) and (e,W,L) has the Demotion Normal Form (W,W,L), but its Fusional Normal Form, and **MIB**, is identical to the original.

These differences between FRed and RCD arise because of the differing ways that they move to recursion. To clarify these differences, we must first surmount an obstacle of perspective. In familiar RCD, we talk of *removing* ERCs (those satisfied by some rankable constraint assessing W) and in FRed we talk of *retaining* ERCs (those involved in info loss configurations), but these are complementary descriptions of the same principle of selection. In example (135), the constraint **1** is an info loss configuration. RCD proceeds by examining the info loss residue of this constraint—vector **b**—exactly as FRed does.

RCD and FRed both continue recursively with info loss residues. But where FRed explores each one separately, RCD makes a grosser calculation. RCD retains precisely those ERCs which earn *e* in every rankable constraint. RCD, then, continues with the *intersection* of the info loss residues of the constraints that fuse to W in the fusion of the whole. This is equivalent to removing those ERCs satisfied by some W and continuing with what’s left. An example will bring out the pattern:

(137) **RCD intersects info loss residues of rankable C:**

$$\{\mathbf{d}, \mathbf{e}\} = \mathbf{A}_1 \cap \mathbf{A}_2 \cap \mathbf{A}_3 \cap \mathbf{A}_4.$$

A	1	2	3	4	5	6	7	8
a	W	e	W	W	L	e	e	L
[-13.9pt] b	e	W	e	W	e	L	e	e
c	e	e	W	e	e	L	e	e
d	e	e	e	e	e	W	L	e
e	e	e	e	e	W	e	L	e
fA	W	W	W	W	L	L	L	L

The info loss residues of the rankable constraints, \mathbf{A}_1 through \mathbf{A}_4 , are signaled by shading: their intersection is boxed in heavy lines. Every ERC belongs to some residue, in this case.

$$\begin{aligned} \mathbf{A}_1 &= \{\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}\} \\ \mathbf{A}_2 &= \{\mathbf{a}, \mathbf{c}, \mathbf{d}, \mathbf{e}\} \\ \mathbf{A}_3 &= \{\mathbf{b}, \mathbf{d}, \mathbf{e}\} \\ \mathbf{A}_4 &= \{\mathbf{c}, \mathbf{d}, \mathbf{e}\} \\ \mathbf{A}_1 \cap \mathbf{A}_2 \cap \mathbf{A}_3 \cap \mathbf{A}_4 &= \{\mathbf{d}, \mathbf{e}\} \end{aligned}$$

RCD removes from \mathbf{A} the vectors $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ because they are satisfied by the constraints fusing to W . Complementarily, RCD *continues* with the intersection of all residues: $\{\mathbf{d}, \mathbf{e}\}$. Observe that the intersection is not among the four residues, though it will eventually appear in various branches of FRed devolving from each.

Let us proceed:

(138) **Next Round of RCD.**

$\cap_i \mathbf{A}_i$	1	2	3	4	5	6	7	8
d					<i>e</i>	W	L	<i>e</i>
e					W	<i>e</i>	L	<i>e</i>
$f \cap_i \mathbf{A}_i$					W	W	L	<i>e</i>

Need we go on to a yet further round? The intersection of the residues here is *null*.

Spelling it out:

$$\begin{aligned} \mathbf{A}_5 &= \{\mathbf{d}\} \\ \mathbf{A}_6 &= \{\mathbf{e}\} \\ \mathbf{A}_5 \cap \mathbf{A}_6 &= \emptyset \end{aligned}$$

This signals the successful conclusion of RCD. We have collected all the fusions we need, when our goal is just to find a ranking *sufficient* to satisfy the original reference set.

The output is a reduced set of ERCs, which carries the information we need. Our example yields the following Constraint Demotion Normal Form.

(139) **CDNF(A)**

CDNF(A)	1	2	3	4	5	6	7	8
$f\mathbf{A}$	W	W	W	W	L	L	L	L
$f \cap_i \mathbf{A}_i$	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	W	W	L	<i>e</i>

We can now formulate RCD in the manner of FRed, as a collector of fusions. Observe that we will also want a null intersection in the case where any single constraint fusing to W contains only W 's (i.e., is W -compliant). Such a constraint is rankable, and its ranking satisfies all remaining ERCs, bringing the RCD process to a successful conclusion. We must say not that it lacks a residue, but that it has a *null* residue, which when intersected with any other residues ensures the desired null intersection. We therefore modify step 2 in the manner noted in Sect. 3.1: we define a residue for every coordinate fusing to W . In this understanding, an info loss configuration occurs when a constraint has a non-null residue.

- (140) **RCD as information gatherer.**
- 0. **Base Step.** #Terminate when \mathbf{A} is empty.
 If $\mathbf{A} = \emptyset$ then $\text{CDNF}(\mathbf{A}) := \emptyset$
 - 1. **Fuse All.** #Collect the fusion of the whole.
 $\text{HoldFus} := \{f\mathbf{A}\}$
 - 2. **Identify Info Loss Residues and Their Intersection.**
 $\text{Res}(\mathbf{A}, i) := \{v \mid v \in \mathbf{A} \ \& \ i \in W(f\mathbf{A}) \ \& \ v[i] = e\}$ #Each W in $f\mathbf{A}$ has a residue
 $\text{IR}(\mathbf{A}) := \cap_i \text{Res}(\mathbf{A}, i)$
 - 3. **Check Consistency.**
 If $f\mathbf{A} \in L^+$, EXIT and announce: #If L^+ fusion, announce failure.
 ‘ \mathbf{A} is unsatisfiable.’
 Else, $\text{CDNF}(\mathbf{A}) := \text{HoldFus}$ #Else, save the fusion in CDNF.
 - 4. **Recurse.** #Recurse on the intersection IR .
 $\text{CDNF}(\mathbf{A}) := \text{HoldFus} \cup \text{CDNF}(\text{IR}(\mathbf{A}))$

We may dissect the current result (139) into three strata if we wish. With each constraint placed as high as possible, our example yields the following tripartite structure:

$$\{1,2,3,4\} \gg \{5,6,8\} \gg \{7\}.$$

- (141) **CDNF(A).**

$\text{CDNF}(\mathbf{A})$	1	2	3	4	8	5	6	7
$f\mathbf{A}$	W	W	W	W	L	L	L	L
$f \cap_i \mathbf{A}_i$	e	e	e	e	e	W	W	L

Stratification is shown by heavy lines and rearrangement of columns.

The logic of fusion allows us to bridge between the fusional representation of RCD and the stratifying conception. Following *ERA*: 22ff., 83ff., let’s extend the fusion operation to constraints, so that it can be applied to *columns* in a tableau as well as to rows. With a constraint taken to be a column vector over $\{W, L, e\}$, the familiar rules for fusion apply without change. A ‘stratum’ is then understood to be the fusion of a collection of constraints—namely, those that are ‘rankable’: those giving rise at some stage of RCD to e or W in the fusion of the whole remaining ERC set.

On this view, our eight-constraint example turns out like this:

- (142) **RCD with constraint fusion.**

A	1◦2◦3◦4	5◦6◦8	7
$c(f\mathbf{A})$	W	L	L
$c(f \cap_i \mathbf{A}_i)$	e	W	L

RCD, then, emerges as FRed with one essential modification: after the fusion of the whole ERC vector set, we interpolate a step where RCD contracts the constraint system by fusing together all the rankable *constraints*. The FRed algorithm then proceeds with the reduced system. In the tableau, we write $c(X)$ for the ‘contraction’ of X that eventually emerges from the fusional lumping of constraint columns.

In this representation, there is at most one info loss configuration—the fused mega-constraint, which gathers all those constraints fusing to e or W ; every other

constraint fuses to L. The info loss residue corresponding to the fused constraint is precisely the intersection of the info loss residues of its fusands.²⁶ If we add one final clean-up operation that fuses all constraints in the lowest stratum, the characteristic result would take on the following form, behind which we imagine a system of 11 constraints, and a set **U** containing various ERCs whose number we need not fix.

(143) **Typical result of RCD.**

CDNF(U)	1◦7◦11	4◦6	2◦5◦9	3◦8◦10
c(fU)	W	L	L	L
c(fU')	<i>e</i>	W	L	L
c(fU'')	<i>e</i>	<i>e</i>	W	L

This pattern—a diagonal of **W**'s fencing off a solid upper triangle of **L**'s—illustrates the form that the outcome will always take.

For RCD on a satisfiable ERC set, an entailment check step will always come up negative; there can be no entailment from the residue to the fusion of the whole. The {*e*, **W**}-set of the fusion of the whole matches to *e*'s in the intersected RCD residue; but its **L**-set corresponds to residual *e*, **W**, or **L**: among these, there must be at least one **W**, rendering entailment impossible. The lower the stratum, the fewer the **L**'s. This behavior is visible in all the exs. (139)–(143).

Under this conception, RCD produces the **MIB** for the fusionally-contracted constraint system, which in the schematic example (143) consists of four derived mega-constraints: {**1◦7◦11**, **4◦6**, **2◦5◦9**, **3◦8◦10**}. When a full linear order is what's required by the ERC set, each stratum contains exactly one constraint. Constraint fusion is in this case trivial, and RCD and FRed agree completely. Whenever fusional contraction involves constraints that are distinct in their values, RCD and FRed diverge. RCD loses ranking information via nontrivial constraint fusion, as may be seen by contemplating the many configurations that give rise to **1◦2**»**3◦4**, while FRed losslessly pursues all strands of information.

Developing this approach in detail requires that the effects of the contraction operation be propagated to all stored references to the constraint system. Because contraction can change the composition of the constraint set at any recursive stage, the character of all preserved fusions will also change. After the first stage of applying RCD to some set **A**, for example, we will have *fA*, which we save and contract as $c_1(fA)$. When the **L**-set of the $c_1(fA)$ is further processed, it too will typically undergo some contraction, and we will also want to be sure that the saved version of $c_1(fA)$ is contracted as well, so that it accords with the developing picture of the constraint set's composition.²⁷ We must similarly revise the saved fusions at every stage.

The core of contractional RCD will proceed according to the following outline, which suppresses the inconsistency check and the termination condition (ERC set emptiness).

²⁶The standard fusional rule $W◦e = e◦W = W$, applying horizontally in the tableau as rankable constraint columns fuse, will eliminate from the residue of the mega-constraint any ERC vector with **W** in a rankable constraint.

²⁷Thus, while at stage 1 we save $c_1(fA)$, at stage 2, we're saving $c_2(c_1(fA))$ and $c_2(f(Res(c_1(fA))))$.

(144) **Contractional RCD (outline).**

- [1] Fuse all ERC vectors and save the result in **CDNF**.
- [2a] Contract the constraint set, fusing all constraints projecting *W* or *e* in [1].
- [2b] Contract the members of **CDNF** according to the same scheme.
- [3] Recurse on the single info loss residue of [2a].

The result is a reduction of the original ERC set to a canonical form, a kind of triangular matrix, echoing the reduction of the ranking requirements to a single linear order on the fusionally-contracted constraint set.

Just as it is profitable to view RCD as a producer of information, so is it worthwhile to examine FRed as a surveyor of rankings. Consider any path in the tree of FRed recursions, running from the reference set at the root to the empty residue that the modified definition of step 2 places at the end of every branch. FRed begins with the entire set under consideration, call it **A**, and proceeds to some info loss residue of **A**. Since the numbering of the constraint set is arbitrary, let us imagine (without loss of generalization) that it is perfectly suited to our expositional purposes, so that the residues along the branch we're discussing are **A**₁, **A**₁₂, **A**₁₂₃, . . . , **A**_{123...k}.

The residue **A**₁ is comprised of those ERCs to which *C*₁ assigns *e*. Complementarily put, **A**₁ is precisely what's left of **A** when all the ERCs earning *W* in *C*₁ are removed from **A**. The effect of removal is precisely the same as that obtained by filtration when *C*₁ is ranked at the top of a hierarchy. The residue **A**₁ is made up of those ERCs that are not yet satisfied by this ranking maneuver; **A**₁ consists of the data about which *C*₁ is silent. RCD continues with those ERCs that are as-yet unsatisfied when the *entire* set of rankable constraints is placed at the top. We may think of FRed as dealing with what happens when each rankable constraint occupies the topmost position in a separate ranking. FRed's search is not mindlessly exhaustive, however. Only those rankings are pursued that need further articulation to resolve the entire original reference set.

To see how this works, consider first a version of the **1**»**2**»**3** case:

(145) **FRed hunts rankings.**

S	1	2	3
a	W	L	W
b	<i>e</i>	W	L

Here FRed produces a single branch $\langle \mathbf{S}, \mathbf{S}_1, \mathbf{S}_{12} \rangle$. To obtain the related ranking, we extend the sequence to $\langle 1, 2, 3 \rangle$, continuing it with any constraints that the last fusion declares to be necessarily dominated. Here the branch of residues is telling us that **1**»**2**»**3** will work: this is the ranking that it unfolds. From $f\mathbf{S} = \langle \mathbf{W}, \mathbf{L}, \mathbf{L} \rangle$, we know that in fact **1** *must* dominate **2**. In this simple case, FRed has just one branch, and, appropriately extended, its indices exhaust the constraint set. The **MIB** is $\{f\mathbf{S}, f\mathbf{S}_1\} = \{\mathbf{a} \circ \mathbf{b}, \mathbf{b}\}$.

What happens when a fusion of the whole contains a *W*-compliant coordinate? This leads directly to a null residue, which therefore marks the end of a branch. If it is ranked, everything is resolved. Consider this expansion of example (145):

(146) **A W-compliant column.**

A	1	2	3	4
a'	W	W	L	W
b'	W	<i>e</i>	W	L

The residue of **1** is empty: $A_1 = \emptyset$. The fusion $fA = (W, W, L, L)$ tells us that **3** and **4** must be dominated. Therefore, the branch $\langle A, A_1 \rangle$ corresponds to the ranking sequences $\langle 1, 3, 4 \rangle$ and $\langle 1, 4, 3 \rangle$. Constraint **2** makes no appearance in these sequences. This means that any ranking of the whole constraint set which respects the orders $\langle 1, 3, 4 \rangle$ or $\langle 1, 4, 3 \rangle$ will also work no matter where constraint **2** is interpolated.

The restrictions on constraint **2** are explored in the branch $\langle A, A_2, A_{23} \rangle$. The last fusion on this branch, which is identical with **b'**, declares that constraint **4** must be subordinated. This yields the ranking sequence $\langle 2, 3, 4 \rangle$, into which the omitted constraint **1** may be freely inserted. A calculation shows that some 10 total orders are admitted by the ERCs of this example. FRed explores them quite efficiently in just two branches.

Pursuing this logic, we find that the branch terminating in $A_{123\dots k} = \emptyset$ corresponds to a linearly-ordered ranking of constraints $C_1 \gg C_2 \gg C_3 \gg \dots \gg C_k$. The last fusion in the branch determines how it must be extended: if m additional constraints must be subordinated, then the ranking orders associated with this branch relate $k + m$ constraints. Any constraints that remain may be freely interspersed in the rankings thus fixed. This enumerates the set of rankings associated with the branch.

Starting back at the root **A**, we find that the fusion fA enumerates all possible top-ranked constraints: these provide the non-L-coordinates of fA . Ignoring the constraints that yield *e* in fA —they do no work, where the job is to outrank all L's with a W—FRed next considers what would happen if each W-supplying constraint were indeed first in a ranking order. The info loss residue of each such constraint contains all the data that is not resolved by ranking it at the top. Moving to the next stage, the fusion of any first-order residue A_j identifies the constraints that may be productively ranked just below C_j , as second in the unfolding linear rankings. And so on, through the residues of residues, until a null residue is reached, and only the constraints supplying L in the last fusion must be placed at the bottom.

FRed does not remember the rankings it unfolds, retaining instead the fusions it creates along the way, an exact record of the conditions that it is exploring. (The entailed, redundant fusions are eliminated in the interest of conciseness.) Each fusion determines the possible informative continuations. Each residue $A_{k\dots m}$ says: if you have continued in *this* way, adjoining C_m to the end of the initial sequence $C_k \gg \dots \gg C_{m-1}$, then here is the data unresolved by that partial ranking, giving further conditions that must be met. The fusion $fA_{k\dots m}$ is the key to what constraints may be ranked next, and identifies their next-order residues. Viewed as ranking algorithms, both RCD and FRed operate locally with respect to developing hierarchies, seeking out the set of the constraints that can be ranked *next*. RCD amalgamates them; FRed explores, for each possibility, the further as-yet-incomplete possibilities that lead to eventual success, if success is available. It is remarkable that if any of the branches so enumerated lead to success, then all of them do; and if any lead to failure, then all fail. The fact that such localism is viable—that, for example, at the very first

step you can determine which constraints can stand in topmost position—reflects the structure of lexicographic optimization that OT is based on.

The ranking perspective on FRed is of direct use in understanding its properties. A valuable relation between constraint ranking order and ERC entailment is established in *ERA*: 29, Prop. 4.3. Given a hierarchy, possibly stratified, let the ‘rank’ of an ERC vector be determined by the highest-ranking constraint that assigns its W,L-pair a polar value. (In a successful hierarchy, this value will be W.) The higher the ‘rank’ of the ERC vector, the higher in the given hierarchy is the constraint that dismisses its suboptimal data.

The relevant result is that, in any successful hierarchy, a nontrivial ERC vector can be entailed only by vectors that are at the same or lower position in the ERC rank order. The rank relation in a successful hierarchy *refines* the arrow/entailment relation, in the sense that, if $\mathbf{a} \rightarrow \mathbf{b}$ for two distinct ERC vectors, then the rank of \mathbf{a} is lower than or equal to that of \mathbf{b} . Turning this around contrapositively: if \mathbf{b} is of lower rank than \mathbf{a} , then we cannot have $\mathbf{a} \rightarrow \mathbf{b}$. Even more strongly, Prop. 4.4 of *ERA*: 30 shows that any *set* of entailing ERCs is similarly restricted. For \mathbf{S} a consistent set of nontrivial ERCs, if $\mathbf{A} \subseteq \mathbf{S}$ entails $\mathbf{b} \in \mathbf{S}$, with the restriction that every member of \mathbf{A} is essential to the entailment (i.e., cannot be removed from \mathbf{A} without losing the relation), then every member of \mathbf{A} has rank less than or equal to \mathbf{b} in any hierarchy that satisfies \mathbf{S} .

From this, taken with the ranking–testing interpretation of FRed, it will follow that if $f\mathbf{A}_K$ is asymmetrically entailed, it is entailed by a set of vectors in the total residue of \mathbf{A}_K , since these include *all those* of lower rank than $f\mathbf{A}_K$ in the ranking sequences developed through \mathbf{A}_K and its residues. (Prince and Brasoveanu 2010 fills in the details.) This means that the extreme locality of FRed’s general entailment-check step—flying in the face of the potentially daunting ramifications of the FRed tree, which has far-flung nodes that might be imagined relevant—is fully justified. The very same observation applies to the checking of atomic entailments: since everything that could be involved in asymmetric entailment of $f\mathbf{A}_K$ is located in its residues, it can’t happen that we’re missing any atomic entailers. For present purposes, it shows once again how the informational interpretations of FRed and RCD are mirrored directly in ranking interpretations, leading to considerable mutual illumination.

5 The worst-case complexity of FRed

As a proxy measure of the potential complexity of FRed, let us count the total number of fusions of sets of vectors that it can involve, summing over the entire computation.

To obtain the worst possible case, imagine that each of n constraints produces its own info loss residue. Still looking for the worst, assume that each such first-order residue itself produces a maximum number of its own residues, and so on. Since each of these n first-order residues has one constraint that is all e ’s, each will maximally produce $(n - 1)$ further residues. We therefore have $n(n - 1)$ second-order residues. Continuing in this fashion, each of the $n!/(n - k)!$ k^{th} -order residues produces $n - k$ further residues, adding $n(n - 1) \dots (n - k)$ more residues to the total, until we reach the last possible level of residues, the $(n - 1)^{\text{st}}$. (Each residue of degree k neuters out

k constraints, and we want one constraint left at the end, so as to obtain the absolutely maximal extension of the process.) Summing the terms, we find that the total number of ERC sets computed, each of which is fused, including the initial one, is given by this formula:

$$(147) \quad \text{Total} = 1 + n + n(n - 1) + n(n - 1)(n - 2) + \dots + n!$$

To estimate this quantity, we observe first that it may be rewritten as follows:

$$(148) \quad \begin{aligned} \text{Total} &= \frac{n!}{n!} + \frac{n!}{(n - 1)!} + \frac{n!}{(n - 2)!} + \dots + \frac{n!}{1!} \\ &= n! \sum_{m=1}^n \frac{1}{k!} \end{aligned}$$

The summation term in this expression is recognizable as a truncation of the series expansion for $e - 1$, where e is the base of the natural logarithms (2.718281828459045...).

$$(149) \quad e - 1 = \sum_{k=1}^{\infty} \frac{1}{k!}$$

Because $1/k!$ gets small fast, the summation term in the Total is well-approximated by the infinite series, so we have:

$$(150) \quad \text{Total} \approx (e - 1)n!$$

In fact, the approximation is so good that the Total is *always* equal to the greatest integer in the approximating expression, its integer ‘floor’:

$$(151) \quad \text{Total} = \text{floor}[(e - 1)n!]$$

Thus, in the worst case, the number of fusions grows factorially in the number of constraints. This situation amounts to an exhaustive listing of all possible rankings on the constraint set—with duplications.

The worst case only arises, of course, when there is an unnaturally high degree of symmetry. We’d need n constraints where each row and each column contains a single W, with the rest e ’s. (Imagine a square array with W’s along the diagonal.) This particular arrangement won’t present itself often in practice. More realistically, we could have $n + 1$ constraints, with n of them as described and the $(n + 1)^{\text{st}}$ all L’s. (In this case, the machinations of FRed are not particularly productive, since we’re starting out with a **MIB**.) Furthermore, our reckoning of the maximal bushiness of the FRed tree must seriously overestimate the amount of computational work that needs to be done in even the worst case. There is no reason to reprocess a residue, and the number of possible residues is bounded by the number of subsets of the ERC vectors under consideration. Keeping track of where the algorithm has been will have a favorable impact on processing time whenever there are duplications.

The absolute worst case is unlikely to be encountered ecologically, but we can still expect a certain amount of complexity in naturalistic applications of the algorithm, as

is confirmed by work with RUBOT (Prince and Tesar 2008), which uses depth-first traversal of the FRed tree. Observe that these complexity considerations should not be taken as a rhetorical talking point by would-be objectors, since they apply to a construction at the meta-level, where the theorist is trying to understand the theory, rather than at the level where the learner is manipulating language data. Worst-case analyses should not be fetishized, in any case, since they do not have immediate empirical relevance.

The best case arises when the ERC set demands a linear order. At each application, there is only one info loss residue. Including the original set, only $n - 1$ set fusions are calculated, one for each required ordering relation.

6 Related algorithms

Here we mention four algorithms related to FRed, one with a different functionality but similar structure; the others with similar goals but different modes of approach.

6.1 An abstract parallel

In “Fundamental Properties of Harmonic Bounding,” Samek-Lodovici and Prince 2005 present an algorithm that takes as input a set of violation profiles and produces as output a set of ‘bounding minima’, profiles that are sufficient to ‘simply bound’ everything that is bounded, simply or collectively, by the original set. This algorithm, which was developed in complete independence of FRed and for very different purposes, has a structure that resembles FRed’s in a number of respects. The columns of a matrix (of violations) are examined for a certain criterial property (minimal violation value), and a summary of the matrix, a derived profile of violations, analogous to fusion, is constructed with respect to the minimal values. Associated with each column is a submatrix defined as those rows that carry the minimal value in the column; these submatrices are gathered for further recursive processing, like the information loss residues, except that every column has one.

There are strong direct connections between harmonic bounding and ERC entailment (ERA: 35–46). Single ERC entailment is simple bounding between ERC vectors under the $L < e < W$ order, and further abstract relations exist between set entailment and collective bounding (Prince 2006b: 28). These arise from shared order properties, since both involve coordinate-wise order in their respective vectorial domains. The goals of the two related algorithms remain distinct, even viewed abstractly—not just because the bounding reduction algorithm operates in the domain of violation profiles, sorting out relations between candidates rather than between constraints, but because its scope is ultimately narrower than that of FRed. It seeks to reduce the problem of bounding entirely to simple bounding, which parallels the pursuit of the Minimal Atomic Generator (MAG). FRed handles the derivation of both the MIB and the MAG (Sect. 3.3), looking either for entailments from sets of any size or specifically for atomic entailments. As with the MAG, bounding reduction needs only a simple comparison across two submatrices, parent and child in the tree, to determine whether a product of the algorithm should be retained. Such parallels suggest the value of further scrutinizing the relation between the two algorithms.

6.2 A shared goal

The matter of ridding an ERC set of logical dependencies and W-compliant sets is taken up in *ERA*: 31–34. The technique proposed there for dealing with entailments is based on the observation that ERC logic supports a form of negation, there called the ‘negative’, which is obtained by permuting the terms of the comparison, so that the desired winner becomes the desired loser. The formal effect is to switch W and L (*ERA*: 12ff.). For all ERCs except the ‘degenerate’ one consisting entirely of *e*’s, the negative works exactly like classical negation. In particular, we have the familiar relationship between inconsistency/unsatisfiability and entailment:

(152) **Entailment and ERC negative.** For **a** nondegenerate, $\mathbf{A} \models \mathbf{a}$ iff $\mathbf{A} \cup \{-\mathbf{a}\}$ is unsatisfiable.

Proof. *ERA*: 13, Lemma (23).

(This result is used to prove that fusion delivers all nontrivial entailments, *ERA*: 14, Prop. 2.5.) The tractable behavior of the ERC negative gives us a very efficient way to determine if any $\mathbf{a} \in \mathbf{A}$ is entailed by other vectors in **A**. We simply replace **a** with $-\mathbf{a}$ in **A** and ask if the result is inconsistent. If it is, then **a** is entailed; if not, then not. RCD makes this a quick calculation, since it is highly efficient and fully capable of detecting inconsistency. To rid **A** of entailments, we simply go through it, ERC by ERC, checking in this fashion and discarding the entailed ERCs that we find.

In the *ERA* algorithm, the entailment check is to be preceded by a winnowing of the W-compliant subsets (*ERA*: 32–33). In many cases, the result of this two-barreled procedure will be identical to that of FRed, but there is one key case where they diverge: where a subset is *fusionally reducible* in the context of the whole, but not locally W-compliant. Example (68), repeated here for convenience, provides an example:

(153) **Fusionally reducible but not internally W-compliant.**

A	1	2	3	4	ERC
a	W	L	<i>e</i>	W	$\mathbf{1} \gg \mathbf{2} \vee \mathbf{4} \gg \mathbf{2}$
b	W	<i>e</i>	L	<i>e</i>	$\mathbf{1} \gg \mathbf{3}$
c	<i>e</i>	W	<i>e</i>	L	$\mathbf{2} \gg \mathbf{4}$
a ◦ b ◦ c	W	L	L	L	$\mathbf{1} \gg \{\mathbf{2}, \mathbf{3}, \mathbf{4}\}$

There are no W-compliant subsets, yet {**a**,**b**} is fusionally reducible, as its members are pairwise W-compliant with the fusion of the whole.

Thus, the *ERA* algorithm will not always produce a basis. For the limited goal of determining whether an individual ERC is entailed by a specific set of vectors, however, the efficient *ERA* technique remains unrivalled.

6.3 Test & verify: an alternative route

The iterated-testing style of the *ERA* algorithm can be used to skeletalize an ERC set directly (Prince 2008b). Instead of looking to see whether each ERC is entailed *as is*, we can test possible modifications of its {W,L}-content, assessing their feasibility.

This approach is patterned after the Horn clause minimization algorithm of Hammer and Kogan 1993: 135, which similarly rests on the ease of checking entailment in the Horn clause setting.

Suppose we wish to know whether a given ERC vector is in skeletal form; that is, whether it has as few W's and L's as it can possibly have in the context of the overall reference set. To test this, we need merely turn one of its polar values to *e*. We first test to see whether the modified vector is entailed by the reference set, and is therefore a candidate for membership in a set that is equivalent to the reference set. (If unentailed, it says something that the reference set does not.) If it is, replace the original vector with this partially skeletalized derivative, and check to see whether the modified reference set entails the original vector. If it does, then we have replaced a polar value whose content follows from the other vectors in the set; that value was inessential and can be omitted. If not, then the tested value was necessary, and the original coordinate could not be emptied while preserving the ranking content of the set. To skeletalize, we continue in the same mode, working our way through each polar coordinate, replacing the vector with its reduction whenever skeletalization is successful, iterating until the absolute polar minimum is reached. If *all* coordinates are replaced, reducing original vector to degeneracy, then it must have been entailed by the set. If we throw away degenerate vectors obtained by this process, we are guaranteed that the end result of the algorithm is a logically independent set, one that is free of internal entailments. Let's call this algorithm 'Test and Verify', T&V for short.

If every polar coordinate in the reference set is subjected to the T&V algorithm, the polar content of each vector is definitively minimized, and all entailed vectors are reduced to degeneracy. The set of nondegenerate vectors in the output of T&V is not guaranteed to be a *basis*, because it may contain fusionally reducible sets. Example (153) again provides a case. T&V will find that the W in constraint 4 of ERC **a** is inessential. But that's it. The result will look like the following, writing **a'** for the skeletalized version of **a**:

(154) **T&V skeletalization of A (153).**

T&V(A)	1	2	3	4	ERC
a'	W	L	<i>e</i>	<i>e</i>	1 » 2 ∨ 4 » 2
b	W	<i>e</i>	L	<i>e</i>	1 » 3
c	<i>e</i>	W	<i>e</i>	L	2 » 4

The set {**a'**,**b**} is fusionally reducible, but unreduced, meaning that basis status eludes T&V(A).

With skeletalization, the nature of the reduction problem has shifted. Observe that ERC vectors **a'** and **b** are now simply W-compliant; no reference to **c** is needed to obtain reducibility. T&V removes any block to internal W-compliance by neutralizing to *e* (bolded) the very W that blocks W-compliance in {**a**,**b**}, which fusion of the whole set would obliterate with L. This is a general fact: a skeletalized form will be much better behaved with respect to W-compliance than an arbitrary ERC set. We may therefore proceed to the Skeletal Basis, if we wish, by fusing only the W-compliant sets, without concern that other fusionally-reducible sets have been missed.

The skeletalization part of this algorithm is tractable, in the sense that it is linear in both the number of constraints in the target ERC set and in the number of polar values in that set. Two factors determine the algorithm’s ultimate complexity: the inherent complexity of the entailment check step, and the number of times it must be run. Observe first that checking entailment requires a run of RCD. Counting its complexity by the number of set fusions it requires, we see that it is almost appallingly efficient: for n constraints, there are at most n strata (with one constraint in each), and therefore at most $n - 1$ set fusions can be required. In the worst case, we must test each of the p polar values in the reference set: this tells us how many times we must run the entailment check. We must check for two entailments: one of the target skeletalization by the reference set, the other of the target ERC itself by the modified reference set. The total number of set fusions required in total is merely $2p(n - 1)$, with no exponents greater than one anywhere in sight. We can bound p by the maximum number of polar values, which cannot be greater than the number of constraints times the number of ERC vectors, call it $|\mathbf{A}|$ for a set \mathbf{A} , yielding $\max(p) \leq n|\mathbf{A}|$. Substituting this in, we find that the absolute maximum number of fusions required to skeletalize a set \mathbf{A} is $2|\mathbf{A}|n(n - 1)$, or strictly less than $2|\mathbf{A}|n^2$.

To get to a basis, we must fuse W -compliant subsets in the output of T&V. The general problem is messy, because a set can be W -compliant as a whole, yet fail to have any W -compliant subsets. Here’s an example:

(155) **Global-only W -compliance.**

A	1	2	3	4
a	W	<i>e</i>	L	W
b	W	W	<i>e</i>	L
c	W	L	W	<i>e</i>
aoboc	W	L	L	L

The whole of \mathbf{A} is W -compliant, as may be seen immediately from the fusion. But no pair is; shading in the columns marks the loci of pairwise failure. A similar construction may be provided for an ERC set of arbitrary size, rendering the whole W -compliant while every proper subset fails to be so.

This phenomenon also disappears under skeletalization, and for exactly the same reason that fusional reducibility simplifies to mere W -compliance. To get the global-only effect, all columns that contain some internal noncompliance must fuse to L . But this means that any W ’s in those columns are eliminable by skeletalization. When those W ’s go, internal noncompliance goes with them. In a skeletalized ERC set, every subset of a W -compliant set is also W -compliant. Therefore, after skeletalization, we may perform fusional reduction in a strictly pairwise, iterative fashion. This is equivalent to reducing out the repetitions in a list. For m items, this takes less than $m(m - 1)/2$ comparisons. Here the analog of a repeated pair of items is a pair of W -compliant vectors, whose compliance we can test by a fusion. Since there are $|\mathbf{A}|$ ERCs on hand, the number of required fusions will be strictly less than $|\mathbf{A}|^2$.

Putting this together with our first bound on skeletalization, we find that the entire process of producing the Skeletal Basis by the T&V algorithm accompanied by post-

T&V collapse of W -compliant subsets requires strictly less than $2|A|^3n^2$ fusional steps.²⁸

This polynomial term counts as tractable in standard terms, while the expressions $n!$ and $2^{|A|}$, which show up in the analysis of FRed, do not. The question arises as to why one should proceed with FRed. Two considerations, we believe, are compelling:

- (1) First, complexity in the event and complexity in the worst-case limit are far from equivalent. FRed responds to the number of residues and T&V to the number of polar specifications; in any given case, their actual distribution can conspire to favor one or the other. For example, in an ERC set incorporating a square block with a diagonal of W 's amid a field of e 's, FRed will face a maximally large number of residues, while T&V looks only at the W 's. If we reverse figure and ground, placing a diagonal of e 's in a square field of W 's, T&V will be grinding through each W while FRed swiftly marches through the e 's.
- (2) Secondly, and far more important for the fundamental goal of the enterprise, FRed gives us a window into the structure of the data-theory relation. The skeletalizing step in FRed, as noted in Sect. 3.2 above, points directly toward the collection of ERCs that are responsible for the ranking relations incorporated in a basis element. T&V, by contrast, works quite globally with the entire mass of ERCs, and only asks whether skeletalization can be achieved in that undifferentiated context. Further structure would have to be built on top of it to try to dig out more specific relations. We conclude that FRed retains its unique attractiveness as a general solution to the Ranking Problem.

6.4 Emin

Another ERC-minimizing algorithm, 'emin', has been developed by Jason Riggle, and is alluded to in Riggle 2007. The perspective is computational and the explicit motivation for minimization is that the potential size of ERC sets grows rapidly with the number constraints. The algorithm appears to produce the **MIB**, but emin works by pairwise fusion of ERCs, a different tactic than the one employed here, which is to process the whole set recursively. We defer for a future occasion the exploration of its relation to FRed.

7 Conclusion

The Ranking Problem in OT asks for the necessary and sufficient ranking conditions imposed by data in which the desired optima have been previously identified. This problem plays out in the realm of comparative tableaux: these display the results of the fundamental calculation that extracts the relative performance of each optimum~suboptimum pair on every constraint. At the heart of the comparative tableau are the vectors of W , L , and e values recording the relative successes, fail-

²⁸It is then possible to produce the **MIB**, if desired, by testing each e in the Skeletal Basis to see whether it can be replaced with L , preserving equivalence to the reference set.

ures, and no-contests involving the desired optima. They support a calculus that enables computation of the desired logical properties.

Because it is the maximally concise and maximally informative equivalent to a reference set, the **MIB** or 'Most Informative Basis' provides a definitive and complete answer to the Ranking Problem. The Skeletal Basis, which can be derived from **MIB** elements, gives another maximally concise representation, which is superior for certain uses, which range from constructing diagrams to connecting ranking restrictions with the data that supports them. The Minimal Atomic Generator (**MAG**), also constructible by FRed in essentially the same way as the **MIB**, gives up absolute conciseness to obtain simplicity of the entailment relation. The perspective developed here also sheds new light on RCD, the algorithm that efficiently delivers a ranking (often a set of rankings) which suffices to generate a set of data, should any such ranking exist; RCD takes its place within the fusion-based family of algorithms as one that involves a kind of lossy compression of the constraint set, in contrast to FRed which extracts and reveals every scrap of ranking information.

With FRed in hand, OT analysts can undertake their investigations with the absolute certainty that they completely control the ranking content of any data set; and that upon that base they can build a thorough understanding of the actual relation between empirical generalizations and the theory proposed to explain them. It is to be hoped that these results will lead researchers to ask for analogous levels of certainty and depth in the analysis of other theories, including variants of OT itself.

Acknowledgements The authors, whose names are arranged alphabetically, would like to thank Jane Grimshaw, Junko Ito, Naz Merchant, Paul Smolensky, and Bruce Tesar for useful discussion; and, for helpful comments, Ryan Bennett, Sara O'Neill, audiences at the Rutgers Optimality Research Group (April 2004, Nov. 2005), HUMDRUM (May 2004), the LSA 2005 Summer Institute, and three astute but anonymous *NLLT* reviewers. For support during aspects of the investigation, we thank the John Simon Guggenheim Memorial Foundation, the National Science Foundation (Grant No. BCS-0083101), and the National Institutes of Health (NRSA Training Grant No. 1-T32-MH-19975-05). *The opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or of the National Institutes of Health.* The authors also claim responsibility for any errors, omissions, misprints, or infelicities that may have crept into the text.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- ROA = Rutgers Optimality Archive. <http://roa.rutgers.edu>.
- Anderson, Alan Ross, and Nuel D. Belnap Jr., eds. 1975. *Entailment: the logic of relevance and necessity*, Vol. 1. Princeton: Princeton University Press.
- Baković, Eric. 2000. Harmony, dominance, and control. PhD dissertation, Rutgers University, New Brunswick. ROA-360: 25–50.
- Brasoveanu, Adrian. 2003. Minimal fusion normal form. Manuscript, Rutgers University. <http://people.ucsc.edu/~abrsvn/MFNF.pdf>.
- Brasoveanu, Adrian, and Alan Prince. 2005. Ranking and necessity, Part I (first version). ROA-794.
- Brasoveanu, Adrian, and Alan Prince. 2007. Fusional reduction and the logic of ranking arguments in OT. Handout. In 31st Penn linguistics colloquium. University of Pennsylvania.
- Gigerenzer, Gerd, and Daniel G. Goldstein. 1996. Reasoning the fast and frugal way: models of bounded rationality. *Psychological Review* 103(4): 650–669.

- Gigerenzer, Gerd, Peter M. Todd, and the ABC Research Group. 1999. *Simple heuristics that make us smart*. New York: Oxford University Press.
- Grimshaw, Jane. 1997. Projection, heads, and optimality. *Linguistic Inquiry* 28(4): 373–422. ROA-68.
- Hammer, Peter, and Alexander Kogan. 1993. Optimal compression of propositional Horn knowledge bases: complexity and approximation. *Artificial Intelligence* 64: 131–145.
- Hayes, Bruce. 2004. Phonological acquisition in optimality theory: the early stages. In *Constraints in Phonological Acquisition*, eds. René Kager, Joe Pater, and Wim Zonneveld, 158–203. Cambridge: Cambridge University Press. Also as ROA-327.
- Hayes, Bruce, Bruce Tesar, and Kai Zuraw. 2004. OTSoft. <http://www.linguistics.ucla.edu/people/hayes/otsoft/>.
- Lombardi, Linda. 1999. Positional faithfulness and voicing assimilation in optimality theory. *Natural Language & Linguistic Theory* 17: 267–302.
- Merchant, Nazarré. 2008. Discovering underlying forms: contrast pairs and ranking. PhD dissertation, Rutgers University, New Brunswick. ROA-964.
- Meyer, Robert K. 1975. Chapters 29.3 and 29.12. In Vol. 1 of *Entailment: the logic of relevance and necessity*, eds. Alan Ross Anderson and Nuel D. Belnap Jr. Princeton: Princeton University Press.
- Parks, Zane R. 1972. A note on R-mingle and Sobociński's three-valued logic. *Notre Dame Journal of Formal Logic* 13: 227–228.
- Prince, Alan. 1998. A proposal for the reformation of tableaux. ROA-288.
- Prince, Alan. 2000. Comparative Tableaux. ROA-376.
- Prince, Alan. 2002a. *Entailed ranking arguments*. ROA-500.
- Prince, Alan. 2002b. Arguing optimality. ROA-562.
- Prince, Alan. 2006a. Lectures on optimality theory, Università degli Studi di Verona. Lecture 2.
- Prince, Alan. 2006b. Implication & impossibility in grammatical systems. ROA-880.
- Prince, Alan. 2006c. No more than Necessary: beyond the Four Rules, and a bug report. ROA-882.
- Prince, Alan. 2008a. The proper treatment of ranking in OT. Manuscript, Rutgers University.
- Prince, Alan. 2008b. ERC minimization is tractable. Manuscript, Rutgers University.
- Prince, Alan. 2008c. OTWorkplace. Freeware. <http://ling.rutgers.edu/people/faculty/prince.html>.
- Prince, Alan. 2009. RCD—the movie. ROA-1057.
- Prince, Alan, and Adrian Brasoveanu. 2010. The formal structure of ranking arguments in OT. Manuscript, Rutgers University and UC Santa Cruz.
- Prince, Alan, and Paul Smolensky. 2004. *Optimality theory: constraint interaction in generative grammar*. Oxford: Blackwell. Revised from ROA-537 (1993 version).
- Prince, Alan, and Bruce Tesar. 2004. Learning phonotactic distributions. In *Constraints in phonological acquisition*, eds. René Kager, Joe Pater, and Wim Zonneveld, 245–291. Cambridge: Cambridge University Press. Also as ROA-353 and RUCCS-TR-54.
- Prince, Alan, and Bruce Tesar. 2008. RUBOT. Freeware, Dept of Linguistics, Rutgers University.
- Riggle, Jason. 2007. Efficiently Computing OT Typologies. Talk at Linguistic Society of America Annual Meeting. Anaheim, CA. Abstract available at http://clml.uchicago.edu/~max/pdf/abstract-efficiently_computing_ot_typologies.pdf.
- Samek-Lodovici, Vieri. 1992. Universal constraints and morphological gemination: a crosslinguistic study. Revised as “A Unified Analysis of Cross-linguistic Morphological Gemination,” 1996: ROA-149.
- Samek-Lodovici, Vieri, and Alan Prince. 1999. Optima. RuCCS-TR-57. ROA-363.
- Samek-Lodovici, Vieri, and Alan Prince. 2005. Fundamental Properties of Harmonic Bounding. RUCCS-TR-71: http://ruccs.rutgers.edu/tech_rpt/harmonicbounding.pdf. Corrected 2005 as ROA-785.
- Sobociński, Bolesław. 1952. Axiomatization of a partial system of three-valued calculus of propositions. *The Journal of Computing Systems* 1: 23–55.
- Tesar, Bruce. 1995. Computational optimality theory. PhD Dissertation, University of Colorado at Boulder. ROA-90.
- Tesar, Bruce. 1997a. Multi-recursive constraint demotion. ROA-197.
- Tesar, Bruce. 1997b. Using the mutual inconsistency of structural descriptions to overcome ambiguity in language learning. In *Proceedings of the North East Linguistic Society* 28, eds. Pius N. Tamanji and Kiyomi Kusumoto, 469–483. Amherst: GLSA, University of Massachusetts.
- Tesar, Bruce, and Alan Prince. 2005. Using phonotactics to learn phonological alternations. In *Proceedings of the thirty-ninth conference of the Chicago Linguistics Society*, Vol. II. The Panels. ROA-620.
- Tesar, Bruce, and Paul Smolensky. 1993. The learnability of optimality theory: an algorithm and some basic complexity results. ROA-2.
- Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in optimality theory*. Cambridge: MIT Press. Also ROA-156.