The Logic of Optimality Theory

Alan Prince
Rutgers University
April 26, 2003. SWOT

(1) **The Constraint as comparator**, decision-maker, evaluator of relative harmony.

COMPARING candidate ω with candidate z, a constraint C offers one of 3 evaluations:

a. C: ω ≻ z   ω is **better than** z

b. C: ω ≈ z   ω is **the same as** z   ❖*concisifying*: :'more harmonic than' ⇛ 'better than'

c. C: z ≻ ω   z is **better than** ω   ❖NB '≻' is transitive and asymmetric (and so, irreflexive).

(2) **Win, Lose, or Draw**. Writing the comparison as ω~z, "(desired) Winner ~ (desired) Loser", we record the possible outcomes in this fashion:

A.

W~ L	C	Interp
ω ~ z_1	**W**	**ω** ≻ z_1
ω ~ z_2	**e**	ω ≈ z_2
ω ~ $\mathbf{z_3}$	**L**	$\mathbf{z_3}$ ≻ ω

B.

W ~ L	ONS	Remark
.pa .o. ~ .a .o.	**W**	1 ~ 2
.pa .o. ~ .a .to	**e**	1 ~ 1
.pa .o. ~ .pa .to	**L**	1 ~ 0

(3) **Interface with Violation Data**

- C:ω≻z iff #C(ω)< #C(z)
- C:ω≈z iff #C(ω) = #C(z)

where #C(*x*) = number of violations of C in *x*.

	ONS	
.pa.o	1	
.pa.o ~ .a .o.	**W**	2
.pa.o ~ .a .to	**e**	1
.pa.o ~ .pa .to	**L**	0

(4) **Making a difference.** C 'distinguishes' ω from z when C:ω≻ z or C:z≻ω. Then, call C 'polar', else 'neutral'.

(5) **The Constraint Hierarchy as comparator.** Given a hierarchy of constraints H, a total ordering on a set of constraints, we extend 'better than' in the following way, defining **'better than' for a hierarchy** in terms of 'better than' for a constraint.

H:ω≻z iff **C: ω≻ z**, where C is the highest-ranked constraint in H distinguishing ω from z.

(6) **Remark**: exactly this is recursively defined in P&S: Ch. 5, p. 81. The concise formulation is Grimshaw's.

(7) **Def. Optimal**. ω is *optimal* iff it beats all competitors (that are distinct from it), i.e. never loses.

- ω∈K, a candidate set, is optimal iff ∀z∈K we have H:ω≻z or H:ω≈z.
- ω∈K is optimal iff ¬∃z∈K such that z≻ω.
- ω is a maximal element in the H:≻ order.

(8) **Summary.**

a. **'Better than'** on a single constraint C

a,b∈K, C:a≻b iff #C(a)< #C(b)   ❖SLP/ERA: C({a,b}) = {a}

b. **'Better than'** on an entire hierarchy H.

H:a≻b iff C:a≻b, where C is the highest-ranked constraint in H distinguishing a,b.

c. **'Best'.** ω∈K is optimal in K over H iff ¬∃z∈K s.t. H:z≻ω.

(9) **What can we learn from a single pairwise comparison?** Imagine a constraint set Σ={A,…,G} and the following list of comparative evaluations, given as a comparative tableau, with no assumption of domination order.

	A	B	C	D	E	F	G
ω ~ z		**L**	**W**		**W**	**L**	**L**

(10) **Aim**: to find the rankings on Σ that ensure H:ω≻z.

❖Notation: we reserve the right to omit *e*, as above.We will also write such tableaux as (e,L,W,e,W,L,L).

(11) **Requirement.** From the definition of 'better than'on a hierarchy, we know that the highest-ranked constraint of the polar subset {B,C,E,F,G} must be **either** C **or** E, if we are to have H:ω≻z.

(12) **ERC**. **Elementary Ranking Condition.**

- Each constraint assessing L must be dominated by some constraint assessing W.
- Equivalently: Some constraint assessing W must dominate every constraint assessing L.

(Equivalent because, by total order, some W-constraint will dominate all the other W-constraints.)

❖ Every comparison ω~z is associated with an ERC.

❖ This restates the Cancellation/Domination Lemma of P&S: §8.2.6, p. 142.

(13) **ERC Sets.** The art of justifying ranking relations is precisely the art of juggling sets of ERCs.

(14) The **ERC** is disjunctive in W and conjunctive in L. **All L**'s must be subordinated. But only **one of the W**'s need be superordinate.

(**C**>>*B* & **C**>>*F* & **C**>>*G*) ∨ (**E**>>*B* & **E**>>*F* & **E**>>*G*)

(15) **The disjunctive tangle**. ERC = disjunction of conjunctions. Set of ERCs = conjunction of disjunction of conjunctions. May appear hard to unravel.

(16) **Example.** Suppose α= **(W,L,W)** and β= **(W,W, L)**. What does it take to satisfy ARG = {α,β} ?

Q: Must C_1>>C_2 ? α is disjunctive on the point and β completely uninformative.

Q: Must C_1>>C_3 ? β is disjunctive here and α uninformative.

α&β = (C_1>>C_2 ∨ C_3>>C_2) & (C_1>>C_3 ∨ C_2>>C_3)

= (C_1>>C_2 & C_1>>C_3) ∨ (C_1>>C_2 & C_2>>C_3) ∨ (C_3>>C_2 & C_1>>C_3) ∨ (C_3>>C_2 *and* C_2>>C_3) [distrib. law]

= ¿ … **??** >> **??** … ?

(17) **Storming the PC barricades**. A head-on assault using the resources of Prop Calc is unappetizing.

O From α we know that *either* C_1>>C_2 *or* C_3>>C_2. From β, we know that *either* C_1>>C_3 *or* C_2>>C_3. Conjoining these arguments produces four cases to consider. Of these, we may dismiss one immediately as inconsistent: 'C_3>>C_2 *and* C_2>>C_3'. The remaining three cases are [1] C_1>>C_2 *and* C_1>>C_3 , [2] C_1>>C_2 *and* C_2>>C_3, [3] C_3>>C_2 *and* C_1>>C_3. Observe that by transitivity of '>>', case [2] is equivalent to [2'] C_1>>C_2 *and* C_2>>C_3 *and* C_1>>C_3. Similarly, case [3] is equivalent to [3'] C_3>>C_2 *and* C_1>>C_3 *and* C_1>>C_2. Now observe that [1], [2'], and [3'] all contain the expression [1]. Therefore, since p∨(p∧q)∨(p∧r) is equivalent to p, we have [1]. **QED**. ❖v. ERA:3.

Such-like drives Hayes 1997/2003 to remark: "I will suggest that arguments from ordinary human reasoning are (probably) not generally as reliable as arguments produced by algorithm." [if the algorithm is well-reasoned! -AP]

(18) **GOALS**: To develop the logical tools for these essential analytical projects:

- Evaluate the necessity and sufficiency of ranking claims
- Find redundancies, non-obvious consequences, and contradictions in sets of ranking arguments
- Determine harmonic bounding relations, so as to extract predictions of linguistic impossibility
- Ultimately, to assess the explanatory role of individual constraints in particular analyses.

(19) **Practicum: from data to argument.** Reducing a data tableau to a comparative tableau.

- The comparative tableau emerges from the calculation that is required to assess the 'C:≻' relation.
- A data tableau may easily be annotated in situ to reflect this.

  P Leave the 'desired optimum' rows alone. They only provide the standard for judging the competitors.

  - ▸**W**. Mark a des. suboptimum cell **W** if it contains more violations than the desired optimum. ["**W**orse"]
  - ▸ **L**. Mark a des. suboptimum cell **L** if it contains fewer violations than the des. optimum. ["**L**ess"]
  - ▸ **e**. Leave a des. suboptimum cell alone if it contains the same # of violations as the des. optimum.

A.

	C_1	C_2	C_3
ω	**	****	***
z	*	****	****

→

B.

	C_1	C_2	C_3
ω	**	****	***
~ z	**L** *	****	**W** ****

An equivalent and more perspicuous superimposition of the violation data and comparison structure:

	C_1	C_2	C_3
ω	2	4	3
z	**L** 2~1	4~4	**W** 3~4

Arithmetically: determine Δ = #C(z) −#C(ω). Then, Δ>0 ➝W. Δ<0 ➝ L. Δ=0 ➝ e

(20) **IGNORANCE = BLISS**. Under what conditions may we safely *disregard* constraints in a ranking argument?

(21) **Example.** Author X argues, in an important and valuable piece of work, that necessarily C >>D because:

	C	D
ω		**
z	* !	*

— *i.e.*➝

	C	D
ω ~ z	**W**	**L**

(22) But *seven* constraints are under direct consideration in his analysis, and over the whole group we actually have:

	A	B	C	D	E	F	G
ω ~ z			**W**	**L**	**W**	**W**	**W**

■From this we may deduce only that **ONE** of the W-set {C, E, F, G} dominates E.

(23) **Polar constraints cannot be ignored.** This follows immediately from the def. of 'better than on H'.

a. **False Assertion**. To ignore C_k assessing W leads to simple **falsity**, as above. See P&S:§7.2.1, p. 118.

It is *not true* in (21) that [ω~z] over {C,D} necessitates the ranking C>>D. [❖ asserting a disjunct]

b. **Incomplete assertion.** To ignore C_k assessing L can lead to a true but incomplete assertion.

E.g. to assert (W,L,e), when (W,L,L) holds, omits (W,e,L). But beware. [❖fa iling to assert a conjunct]

(24) Ignoring constraints from the whole set requires justification.

a. C neutral wrt a given comparison ➝ C may be ignored.

b. C polar wrt a given comparison must be accounted for.

- Either included in the ERC, or shown to be irrelevant (e.g. already ranked below some L).

(25) **Candidates** can be meaningfully compared in pairs, yielding an ERC. **Constraints** cannot, in general.

❖ The belief that constraints can is phps a left-over from earlier descriptive practice. Even true of rules then?

❖ That *candidates* can is due to the very simple way OT constraints evaluate, with '≈' an equivalence relation.

Entailment & Inconsistency in ERC sets

(26) One ERC may entail the validity of another. An ERC set may entail the validity of further ERCs.

a. (W,L,L) entails (W,L,e), *i.e.* (A>>B) & (A>>C) entails (A>>B). ❖ $p\&q \vdash p$

b. (e,W,L) entails (W,W,L), *i.e.* (B>>C) entails (B>>C) ∨ (A>>C) ❖ $p \vdash p \vee q$

c. (W,L,e) , (e,W,L) entail (W,e,L) *i.e.* {(A>>B), (B>>C)} entails (A>>C) ❖ *transitivity of* '>>'

(27) Sets of ERCs may be inconsistent, in that there is no ranking that validates the set.

a. (L,L) is *false* or *invalid*, in the sense that no ranking will choose the desired optimum as optimal.

b. (W,L), (L,W) is inconsistent. *i.e.* $\vdash \neg$(A>>B & B>>A) ❖ *asymmetry of* '>>'

(28) **Why entailment is interesting.**

a. Entailed propositions are *redundant*. [efficiency]

b. Important entailments may be *imperspicuous*, hard to see. [correctness]

• ''Mate in 10.' 'π^2 is irrational'. 'The butler did it'.

(29) **Why entailment is interesting in OT: necessity.** Within any ERC set ARG there lies a minimal nonredundant subset that determines the ranking conditions necessary to validate ARG.

❖We want to know *exactly which facts* determine the ranking, as analysts and learners.

(30) **Why entailment is interesting in OT: sufficiency.**

The **sufficiency** of a hierarchy — the actual optimality of desired optima — is determined by entailment.

a. Let ARG be a set of ERCs [ω~z_i], over hierarchy H, for $\omega \in K$, a candidate set, and for various $z_i \in K$.

• Say H validates ARG. Then, if $\forall$**z**$\in$**K, ARG** $\vdash$**[ω~z]**, we may conclude that ω is optimal.

b. This tells us we can prove optimality from the candidates z_i *alone*, if we can show that all further possible ERCs on ω are *entailed* by ARG = {[ω~z_i]}. ❖Proof technique: 'method of mark eliminability', P&S:127ff.

(31) **Entailments of an ERC.** Part I. **L-retraction.**

Given an ERC vector **α**, replace any L by **e** to get **β** Then **α** $\vdash$ **β**. ❖*cf.* $p\&q \vdash p$

	A	B	C	D
α	**W**	**L**	**W**	**L**

➔

	A	B	C	D
β	**W**	**L**	**W**	

(32) **Entailments of an ERC.** Part II. **W-extension**.

Given an ERC vector **α**, replace any **e** by W to get **β**. Then **α** $\vdash$ **β**. ❖*cf.* $p \vdash p \vee q$

	A	B	C	D
α	**W**	**L**	**W**	

➔

	A	B	C	D
β	**W**	**L**	**W**	**W**

(33) **Terminology. 'Nontrivial'**

a. A **nontrivial ERC** contains at least one W and at least one L. Else, trivial.

A trivial ERC is either true under all rankings, like (W,e) or (e,e).

or false under all rankings, like (L,e) or (L,L).

b. A **nontrivial entailment** involves only nontrivial ERCs.

Trivially, *ex falso quodlibet*: (L,e) $\vdash$ α, for any α. Likewise, *verum ex quodlibet*: ∀α, α$\vdash$ (W,e).

(34) **ENTAILMENTS of an ERC. COMPLETE.**

Every nontrivial entailment from an ERC α follows by a sequence of L-retractions and W-extensions. ❖Prop. 1.1, ERA:6.

Valid coordinate-wise moves: L → e, W

e → W

(35) **General Entailment Problem**. Get entailments jointly from sets of ERCs, not from individual members of the set.

Trans. {A>>B, B>>C} ⊢ A>>C — but A>>C is entailed by neither individually.

Asym. {A>>B, B>>C, C>>A} is inconsistent — but OK individually and pairwise.

(36) **Fusion**. The key to simplifying such arguments is a method of ERC combination: *fusion.*

Combine two ERC vectors α,β coordinate-by-coordinate to produce an ERC vector α∘β as follows. X∈ {W,e,L}

X∘L	= L∘X = L	*Dominance of L*
X∘e	= e∘X = X	e *is Identity*
X∘X	= X	*Idempotence*

(37) **Finding dominations-from-transitivity.** ❖In α∘β, we lose B>>C from β, but we gain **A>>C**.

	A	B	C
α	**W**	**L**	e
β	e	**W**	**L**
α∘β	**W**	**L**	**L**

(38) **Resolving disjunctions: Does A>>B ? Does A>>C ?**

	A	B	C
α	**W**	**L**	**W**
β	**W**	**W**	**L**
α∘β	**W**	**L**	**L**

(39) **Do they or do they not?** Indeed: fusion tells me so.

❖Thus we slice the disjunctive knot of (16): (A>>B ∨ C>>B) & (A>>C ∨ B>>C) ⊢ ????

(40) **Fusion needn't be informative.** Here α⊢α∘β simply by W-extension.

	A	B	C	D
α	**W**	**L**	e	**L**
β	e	**L**	**W**	e
α∘β	**W**	**L**	**W**	**L**

(41) **But it never lies**. **And it tells all.**

(42) **THM. EGR. Every ERC** entailed by ARG, a set of ERCs, **follows from the fusion** of some subset of ARG.

ARG⊢α iff ∃Ψ⊆ARG such that *f*Ψ ⊢ α ❖Prop 2.5, ERA:14.

(43) Fusion therefore reduces the general entailment problem to single ERC entailment.

(44) **Basic properties of fusion**

a. α&β⊢α∘β , *i.e.* α,β⊢ α∘β — and more generally, Ψ⊢*f*Ψ ❖Prop. 2.1, ERA:10.

b. α∘β ⊢ α∨β — and more generally, *f*Ψ⊢∨Ψ ❖Prop. 2.2, ERA:11.

c. Unlike conjunction, there is **no guarantee** that α∘β⊢α.

d. Unlike disjunction, there is **no guarantee** that α⊢α∘β. ❖ v. ERA §3, 15-20.

(45) **Why inconsistency is interesting.** Inconsistency/invalidity is the logical flip-side of entailment.

$p \vdash q$ iff $\{p, \neg q\}$ is inconsistent, *i.e.* $p \wedge \neg q$ is invalid.

❖ Example: If an integer is divisible by 4, then it is **even**. No integer is divisible by 4 *and* **odd**.

(46) **Why inconsistency is interesting in OT.**

a. **Failure.** *Tout court*. Analysts want to know when their constraint system cannot produce desired optima.

Harmonic Bounding. A candidate never optimal under any ranking is a prediction of impossibility.

b. **Ranking** exists precisely to control inconsistency.

(47) **Notation.** An invalid ERC vector contains no W's and at least one L. Call the set of such: L^+. ❖(e,L,e)∈ L^+

(48) **Fusion detects inconsistency**. If ARG cannot be satisfied over Σ, a set of constraints, by any ranking, then some subset of ARG fuses to L^+.

THM. ARG is inconsistent iff there is a $\Psi \subseteq$ARG such that $f\ \Psi \in \mathsf{L}^+$. ❖Prop. 2.4, ERA:11.

(49) **Ecological example**. Author P once encountered the following as class-time approached:

	NONFIN	HD-F-RT	RT-MAIN	PARSE-σ	Max-μ
α	L	L		W	W
β	W		L	W	L
γ		W	W	L	
δ	W			W	L
α∘β∘γ∘δ	**L**	**L**	**L**	**L**	**L**

(50) **WHY IT WORKS**. Coarsely put, fusion preserves necessary subordination (L) and possible domination (W).

(51) **Harmonic Bounding**. Many are called: few chosen.

- The vast majority of candidates from any input ('almost all') can never be optimal under any ranking.
- They are perpetual *losers*. Whatever the ranking, a *loser* is always beaten: another candidate is always better.

(52) **Example from Alignment theory**

	PARSE-σ	ALL-FEET-LEFT
a. (σσ)(σσ)(σσ)	0	6
b. (σσ)(σσ)σσ	2	2
c. (σσ)σσσσ	4	0

Prs-σ	AFL
a	c
b	**b**
c	a

❖ Cand. (b) is harmonically bounded by {a,c} over these constraints. PRS-σ>>AFL yields (a). AFL>>PRS-σ yields (c).

(53) Try to make (b) optimal.

	PARSE-σ	ALL-FEET-LEFT
b ~ a	L 2~0	W 2~6
b ~ c	W 2~4	L 2~0
[b~a] ∘ [b~c]	**L**	**L**

(54) **General Conditions for Harmonic Bounding** (SLP 1999) are easily ascertained via fusional ERC theory.

(55) **Consider z, the loser** over Σ and K. Examine ERCs of the form $\alpha_i = [z\sim a_i]$, for z, $a_i \in K$

a. If no ranking of Σ works, then there must be a collection ARG = $\{\alpha_i\}_{i\in I}$ which is *inconsistent*.

b. By Thm. (48), $\exists \Psi \subseteq$ ARG such that $f\Psi \in \mathsf{L}^+$.

(56) Examine the columns of $f\Psi$. Since $f\Psi$ fuses to L^+, the columns fall into three types.

a. C[e] all e all the time.	❖ C: $z \approx a_i$, for *all* a_i			C[W,L]
b. C[L] all L all the time.	❖ C:$a_i \succ z$, for *all* a_i			a_k
c. C[W,L] both W and L present.	❖❖ **C:z≻a_i for some a_i**	◂α_i supplies W		**z**
	❖❖ **BUT C:a_k≻z for some other a_k**	◂α_k supplies L		a_i

(57) **Reciprocity Condition.**

a. A set of candidates **Q** has *reciprocity* wrt z, iff

whenever **z** beats some **a**∈Q on some C, there's a **b**∈Q that beats **z** on that C.

b. Q⊆K, a set of candidates meets reciprocity wrt z∈K, iff

$\forall C \in \Sigma \;\; \forall a \in Q \;\; \exists b \in Q$ **if C:z≻a then C:b≻z.** ❖NB C(z~a)=W & C(z~b) =L.

(58) **Distinctness**. A set of candidates Q⊆K is *distinct* from z∈K if it contains some q that differs from z on some constraint. I.e. not all the candidates in Q are violation-copies of z.

(59) **Harmonic Bounding.** A candidate z∈K is never optimal over Σ iff there is Q(z)⊆K meeting these conditions.

a. **Reciprocity.** Q(z) meets the reciprocity condition wrt z.

b. **Distinctness.** Q(z) is distinct from z.

(60) **Simple Harmonic Bounding.** Suppose Q(z) is a unit set, {q}. Say Q satisfies reciprocity & distinctness wrt z.

a. **Reciprocity**. Satisfaction must be achieved vacuously: z can never beat q (there being no reciprocator in Q).

b. **Distinctness.** z≉q for some C∈Σ. Therefore, on such C we have **C:q≻z**.

(61) Example.

/patok/	ONS	NOCODA	Dep	Max
z. pat **.** ok	1	2	0	0
q. pa **.** tok	0	1	0	0
z~q	**L** 1~0	**L** 2~1	e 0~0	e 0~0

(62) **Harmonic Bounding and Entailment.** HB is a great provoker of entailment.

a. Simple case. If **q** h-bounds **z**, then for any ω, ω~**q** ⊢ ω~**z**

b. General case. If **Q**(z) h-bounds **z**, then {ω~**Q**}⊢ ω~**z** and f{ω~**Q**}⊢ ω~**z**

(63) **Why?** For the simple case it is sufficient to note that q beats z on *any* ranking. Therefore on those in which ω beats q, we also have ω beating z. (Similarly for ω≈q.) For further discussion, see ERA:§6, 35-46.

(64) **Bounding Free.** If a set ARG is free of entailments among its members, it is also free of harmonic bounding among the set of candidates involved in the comparisons in ARG.

❖ Coarsely: HB entails entailment, therefore by contraposition, lack of entailment entails lack of bounding.

(65) **RCD and Fusion**. Recursive Constraint Demotion (Tesar 1995, Tesar & Smolensky 2000). A highly efficient method of constructing a ranking (class of rankings) satisfying ARG, if such a ranking exists.

(66) **How to do it.**

a. Collect those constraints that can be top-ranked, and remove them to a new stratum.
b. Find the ERCs thereby solved, and remove them — they are satisfied.
c. Continue with the set of as-yet-unsolved ERCs and the as-yet-unstratified constraints.
  - If some ERCs are ineliminable, then the ERC set can't be solved by ranking Σ.

(67) Given ARG and Σ, which C∈Σ *cannot be* top-ranked? — Exactly those which in *f*ARG fuse to L.

The rankable C's are exactly Σ − **L**(*f*ARG). ❖writing **L**(*f* S) for those objects in S fusing to L

- Which ERCs in Arg are thereby solved? Those corresponding to W's in the rankable C's.

The solved ERCs are exactly **W**(*f* RankableC) ❖ now fusing *constraints* rather than ERC vectors!

(68) **RCD(ARG, Σ)**

```
If ARG=Ø then RCD(ARG, Σ) :=  Σ                                # ARG=Ø ends it.
  else                                                          #  ARG≠Ø continues it
    RankableC :=  Σ −L(ƒARG).                                   # collect rankable C's
    If RankableC=Ø then RETURN (ARG "inconsistent over" Σ)      # no rankables ends it badly
      else                                                      # RankableC≠Ø continues it
      | UnsolvedARG  :=   ARG −W(ƒRankableC)                    # collect unsolved residue of ARG
      | UnrankedC    :=   Σ − RankableC                         # collect unranked residue of Σ
      | RCD(ARG, Σ)  :=  RankableC >> RCD(UnsolvedARG, UnrankedC)  # rank and recurse
```

(69) **Example [See Appendix for REVISED VIEW]**

I	**A**	B	C	D	E	***f*{A}**
α	Ⓦ	L	L			**W**
β		W		L		
γ			W	L		
δ				W	L	
α∘β∘γ∘δ	**W**	**L**	**L**	**L**	**L**	

❖ RankableC = W(α∘β∘γ∘δ) = {A}. Solved args = W(A)= {α}

II	**B**	**C**	D	E	**B∘C**
β	Ⓦ		L		**W**
γ		Ⓦ	L		**W**
δ			W	L	
β∘γ∘δ	**W**	**W**	**L**	**L**	

❖ RankableC = W(β∘γ∘δ) = {B,C}. Solved args = W(B∘C)= {β,γ}

III	**D**	E
δ	Ⓦ	L

❖ RankableC =W(δ) = {D}. Solved args = W(D)= {δ}. UnsolvedARG = Ø.

(70) **Finding Entailments.** Given ARG and φ, how do we efficiently determine whether ARG⊢φ ?
We know that ∃Ψ⊆ ARG such that. fΨ∈L$^+$.
O But there are $2^{|\text{ARG}|}$ subsets to rake through!

(71) **Inconsistency to the rescue**. Define the 'negative' −φ of an ERC **ω~z** to be **z~ω**. Observe the following:

φ: ω ~ z	W	L	e
−φ: z ~ ω	L	W	e
φ ∘ −φ	**L**	**L**	**e**

❖ Negative as negation. −φ is the logical negation of φ, except when C(φ) = e.

(72) **Entailment and Inconsistency**. ARG⊢φ iff ARG ∪ {−φ} is inconsistent (for φ not all e). ❖ERA:13.

(73) **RCD efficiently calculates entailment.** To find whether ARG⊢φ over Σ, we apply RCD to ARG∪{−φ}.
❖ If RCD fails, we have entailment. (We also get the subset of ERCs that entails.)

(74) **Example. Necessary domination.** RCD stratification does not mean necessary domination. Constraint A may be in a higher stratum than constraint D not because A has any necessary relation, or any relation at all, to D.

	Stratum I			Stratum II
	A	B	C	D
α	e	W	W	L
β	W	e	e	e

(75) **Necessary Domination is an ERC**. φ= "A>>D in all succesful rankings" = (W,e,e,L). Revealed by ARG∪{−φ}.
❖In addition, if we alter Σ to Σ′ by replacing {A, D} with A∘D, then □A>>D iff ARG is inconsistent over Σ′.
• Thus is extremely easy to extract necessary-domination information from the MSH resulting from RCD.

(76) **A LOGIC.** With the introduction of the negative '−', we have a full-blown logic.
Identtifying W = T, L = F, and *e* with a third truth value, we recognize it as the implication-negation fragment of the relevance logic RM3 originally explored by Soboćinski. A ranked constraint hierarchy on N constraints can assign 2N+1 truth values (value = rank-from-bottom×RM3 valuation); it reflects the logic RM.
❖v. ERA:§7, 47-80, Anderson & Belnap 1975.

(77) **Implication in RM3 is ERC entailment**, coordinate-wise.
Define: $\alpha \rightarrow \beta =_{df} -(\alpha \circ -\beta)$ ❖ Cf. standard PC: $A \supset B =_{df} \neg(A \,\&\, \neg B)$

α→β when valid	α⊢β coordinatewise
F → F, e, T	L → L, e, W
e → e, W	e → e, W
T → T	W →W

❖❖❖

Appendix: A Better View of RCD through Fusion

Remark: It is unnecessary to remove constraints from the tableau. RCD is really about removing ERCs.

I	**A**	B	C	D	E	***f*{A}**
α	ⓦ	L	L			**W**
β		W		L		
γ			W	L		
δ				W	L	
α∘β∘γ∘δ	**W**	**L**	**L**	**L**	**L**	

❖ **ARG={α,β,γ, δ}**. RankableC = W(α∘β∘γ∘δ) = {A}. Solved args = W(A)= {α}

II	**A**	**B**	**C**	D	E	**B∘C**
β		ⓦ		L		**W**
γ			ⓦ	L		**W**
δ				W	L	
β∘γ∘δ		**W**	**W**	**L**	**L**	

❖ ARG={β,γ, δ} RankableC = W(β∘γ∘δ) = {B,C}. Solved args = W(B∘C)= {β,γ}

III	A	B	C	**D**	E	***f*{D}**
δ				ⓦ	L	**W**
***f*{δ}**				**W**	**L**	

❖ ARG={δ}. RankableC =W(δ) = {D}. Solved args = W(D)= {δ}. UnsolvedARG = Ø.

IV	A	B	C	**D**	E

❖ ARG = Ø.

Notations and Abbreviations

ERC	elementary ranking condition.
ERC vector	list of W,L,e from which the ERC is derived. ERC & ERC vector not carefully distinguished here.
ω~z	The ERC or ERC vector resulting from comparison of candidate ω with candidate z, aiming for ω≻z.
$\alpha \vdash \beta$	α entails β. No distinction observed between syntax and semantics here.
α,β,...	ERCs or ERC vectors
ARG	set of ERCs
$\alpha \circ \beta$	fusion of α and β
ƒ S	fusion over the whole set S
−α	the negative of α. $-[a\sim b] =_{df} [b\sim a]$
Σ	set of constraints
K	a set of candidates
a,b,…q,z,ω	candidates
A,B,C,…,G	constraints
#C(z)	number of violations constraint C assesses of candidate z
H	constraint hierarchy
❖	remark follows

Acknowledgment

Thanks to Paul Smolensky, crucial methodologist of OT, and to Vieri Samek-Lodovici, Bruce Tesar, Seth Cable, Jane Grimshaw and John McCarthy, for invaluable discussions on various key points.

References

P&S: Prince & Smolensky 1993/2002
SLP: Samek-Lodovici & Prince 1999
ERA: Prince 2002a.
ROA = http:/roa.rutgers.edu

Anderson, Alan Ross, and Nuel D. Belnap, Jr. 1975. *Entailment: the logic of relevance and necessity*. Volume 1. Princeton University Press.

Besnard, Ph., G. Fanselow, and T. Schaub. 2003. Optimality Theory as a family of cumulative logics. *Journal of Logic, Language, and Information* 12:153-182.

Grimshaw, Jane. 1999/2001. Optimal Clitic Positions and the Lexicon in Romance Clitic Systems. In Legendre, Géraldine, Jane Grimshaw, and Sten Vikner, eds., *Optimality-Theoretic Syntax*. Also ROA-374.

Hammond, Michael. 2000. The Logic of Optimality Theory. ROA-390.

Hayes, Bruce. 1997/2003. Four rules of inference for ranking argumentation. Ms. UCLA. http://www.linguistics.ucla.edu/people/hayes/otsoft/argument.pdf.

Karttunen, Lauri. The proper treatment of Optimality Theory in computational phonology. ROA-258.

Parks, R. Zane. 1972. A note on R-mingle and Sobociński's three-valued logic. NDJFL 13:227-228.

Potts, Christopher and Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. Ms. UCSC.

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

Prince, Alan. 2002b. Arguing Optimality. ROA-562.

Prince, Alan & Paul Smolensky. 1993/2002. *Optimality Theory: Constraint Interaction in Generative Grammar*. ROA-537.

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*. Ph.D. Dissertation, Univ. of Colorado. ROA-90

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