Fusional Reduction

& the Logic of Ranking Arguments

Adrian Brasoveanu, UC Santa Cruz
Alan Prince, Rutgers University

abrsvn@gmail.com prince@ruccs.rutgers.edu

University of Pennsylvania – February 25, 2007

The Setting

- **OT**: a grammar *decides* between alternatives.
- **Constraints**: criteria of decision.
- **Penalties**: a constraint detects only flaws.
- **Better Than** on a single constraint: *fewer* flaws on it.

When Constraints Collide

- **Disagreements**: btw constraints, about what's *worse*:
  - $(CVC)_\sigma$ *vs*. C→Ø codas *vs*. deletions
  - C→Ø *vs*. Ø→V delete *vs*. insert

- **Better Than**: over the *entire conflicting mass of criteria* ?
  - Need to resolve all discord

- **Rank them all**.
  - Impose a linear order on the constraint set

Optimality

- **Better Than** over a ranked constraint set
  - **A** is *better than* **B** on a ranking

    iff **A** is better than **B** on the highest-ranked constraint

    that distinguishes them.

- **Optimal**. **A** is optimal over a ranked constraint set

  iff **A** is *better than every* distinct alternative **B**

  over that ranking.

- The optimal alternative is the grammar's choice.

Two Analytical Challenges

- A **Grammar** is a linear order on a constraint set
  - Two analytical problems then arise:

- **The Selection Problem**
  - Given the ranking order,

    which candidate is optimal?

- **The Ranking Problem**
  - Given a (desired) optimum

    which rankings will produce it?

Solving the Selection Problem

- A simple sequential filtration finds the optima.
  - Take the best, ignore the rest.
    - *Slogan due to Gigerenzer & Goldstein* 1996
  - Start with 1st constraint & continue down the hierarchy
    - Taking the best among the previous best, and so on.

- Easily represented in a violation tableau (VT)
  - Annotated at the point of suboptimum demise

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	**Ident/O:Voi**	***ObVoi**	**Ident:Voi**
berg → berk	0	1	1
berg → berg	0	2	0
berg → perk	1	0	2

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
berg $\rightarrow$ berk	0	1	1
berg $\rightarrow$ berg	0	2	0
berg $\rightarrow$ perk	1	0	2

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
berg → berk	0	1	1
berg → berg	0	2	0
berg → perk	1 W!	0	2

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
berg → berk	0	1	1
berg → berg	0	2	0
berg → perk	1 **W!**	0	2

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
berg → berk	0	1	1
berg → berg	0	2	0
berg → perk	1 **W!**	0	2

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	**Ident/O:Voi**	***ObVoi**	**Ident:Voi**
berg → berk	0	1	1
berg → berg	0	2 **W!**	0
berg → perk	1 **W!**	0	2

Selecting the Optimum

- A violation tableau (VT). Assume ranked.

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
✓ berg → berk	0	1	1
berg → berg	0	2 W!	0
berg → perk	1 W!	0	2

The Ranking Problem

- We have a winner (in mind): the ‘desired optimum’
    - From observation & linguistic analysis
- And we have:
    - The constraints
    - The alternative candidates
- Which rankings choose the desired winner?

Solving the Ranking Problem

- What rankings will make **A** better than **B** ?
  - Assume no ranking known:

/berg/	**Ident/O:Voi**	***ObVoi**	**Ident:Voi**
want: **A**: berg → berk	0	1	1
B: berg → perk	1	0	2

Finding the Friends of A

- If **A** is *better than* **B** on a ranking

  Some constraint preferring **A** to **B**

  ranks above *all* constraints preferring **B** to **A**.

- In any such ranking,
  - the highest-ranked constraint distinguishing A and B

    decides in favor of A and against B.

    .

- What do we need to know about the constraints

  to sort this out?

The Constraint's Eye View

- A constraint views the A/B competition in one of 3 ways:
- **W**: **A** is better than **B**
  - The desired optimum wins. *B is worse in violations*.
- **L**: **B** is better than **A**.
  - The desired optimum *loses*! *B has less violation*!
- **e**: No decision.
  - A and B are violationwise identical.

Solving the Ranking Problem

- Calculate how each constraint views A vs. B
  - Assume no ranking known:

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
***want*: A**: berg → berk	0	1	1
B: berg → perk	1 **W**	0 **L**	2 **W**

The Comparative Tableau

- Eliminate violation data
  - Its work is done. We only care about *more* vs. *less*

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
A ~ B	W	L	W

The Elementary Ranking Condition

- ERC: Some W must dominate all L's.

  - holds of each tableau row

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
A ~ B	W	L	W

ERC: **Ident/O** >>***ObVoi** -OR- **Ident**>>***ObVoi**

The Candidate Set, Revisited

- Annotating the VT

/berg/	**Ident/O:Voi**	***ObVoi**	**Ident:Voi**
***want*:A**: berg → berk	0	1	1
C: berg → berg	0	2 **W**	0 **L**
B: berg → perk	1 **W**	0 **L**	2 **W**

The Candidate Set, Revisited

- The full CT

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
A ~ B	W	L	W
A ~ C		W	L

The Candidate Set, Revisited

- Conclusion: necessarily **Ident/O:Voi >> *ObVoi >> *Ident:Voi**

/berg/	Ident/O:Voi	*ObVoi	Ident:Voi
A ~ B	W	L	W
A ~ C		W	L

Issue: Out of many, One

T4	C_1	C_2	C_3
r_4	W	L	L
r_2	e	W	L

T2	C_1	C_2	C_3
r_1	W	L	W
r_2	e	W	L

T3	C_1	C_2	C_3
r_3	W	L	e
r_2	e	W	L

These all designate the same ranking:

$$C_1 \gg C_2 \gg C_3$$

Issue: Out of many, Yet More

	C_1	C_2	C_3
r_1	**W**	**L**	**W**
r_3	**W**	**L**	e
r_4	**W**	**L**	**L**
r_2	e	**W**	**L**

This also designates the same ranking:

$$C_1 >> C_2 >> C_3$$

- CT's like this often arise ecologically, in the course of analysis

Of All the CTs in the World …

- We want the most concise, most informative representation of the ranking conditions inherent in the data.

- The learner is happy with a ranking that works.
  - Sufficient conditions for success will suffice.

- The analyst must know more: both nec. & suff.
  - The structure of grammars lies in the exact conditions
  - E.g. when the *necessary* conditions for one optimum

    entails the *sufficient* conditions for another.

    the presence of the 1st entails the presence of the 2nd.

The Basis

- For any set Σ of ERCs

  a Basis B for Σ is a *minimum cardinality* set of ERCs such that B defines the same rankings as Σ.

- Any Basis for C_1 >> C_2 >> C_3 has *two* ERCS

- Several such exist.

All the Bases

T1	C_1	C_2	C_3
r_4	W	L	L
r_2	e	W	L

T2	C_1	C_2	C_3
r_1	W	L	W
r_2	e	W	L

T3	C_1	C_2	C_3
r_3	W	L	e
r_2	e	W	L

Bases for $C_1 >> C_2 >> C_3$

Not a Basis

	C_1	C_2	C_3
r_1	**W**	**L**	**W**
r_3	**W**	**L**	e
r_4	**W**	**L**	**L**
r_2	e	**W**	**L**

Flabbily designates the same ranking:

$C_1 \gg C_2 \gg C_3$

- Heavily redundant: only *one* of top 3 rows r_1, r_3, r_4 is needed

All Bases are not created equal

T1	C_1	C_2	C_3
r_4	**W**	**L**	**L**
r_2	e	**W**	**L**

Basis for $C_1 >> C_2 >> C_3$

Most Informative Basis (MIB)

- Gives *total* domination info for each row's W-set
- Most L's, minimal # of W's

All Bases are not created equal

T2	C_1	C_2	C_3
r_4	W	L	W
r_2	e	W	L

Basis for $C_1 >> C_2 >> C_3$

Least Informative Basis

- As many spurious local disjunctions (**W**'s) as can be tolerated
- The most W's
- Who needs it?

All Bases are not created equal

T2	C_1	C_2	C_3
r_4	**W**	**L**	e
r_2	e	**W**	**L**

Basis for $C_1 \gg C_2 \gg C_3$

Skeletal Basis

- Eliminates all info derived from transitivity of ranking order
- The most *e*’s, minimal W’s
- Nice: but best approached through the MIB

Admirable Qualities of the MIB

- The MIB is minimal in size, lacks all redundancy.
- Each MIB row contains a unique W-set
- The total ranking info for that W-set is displayed
- Every disjunctive W in a row represents a ranking option that is realized in some licit linearization
- The MIB ties *ranking conditions* to the motivating *data*.

Obtaining the MIB

- How does this desirable object emerge from data?

- Related work from different perspectives includes
  - Hayes, B. 2003, Four Rules of Inference
    - On this, see Prince 2006, ROA-882
  - Riggle, J. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory

FRed Obtains the MIB

Fusion

Fusion (Prince 2002a, b):

Fusion	W	e	L
W	W	W	L
e	W	e	L
L	L	L	L

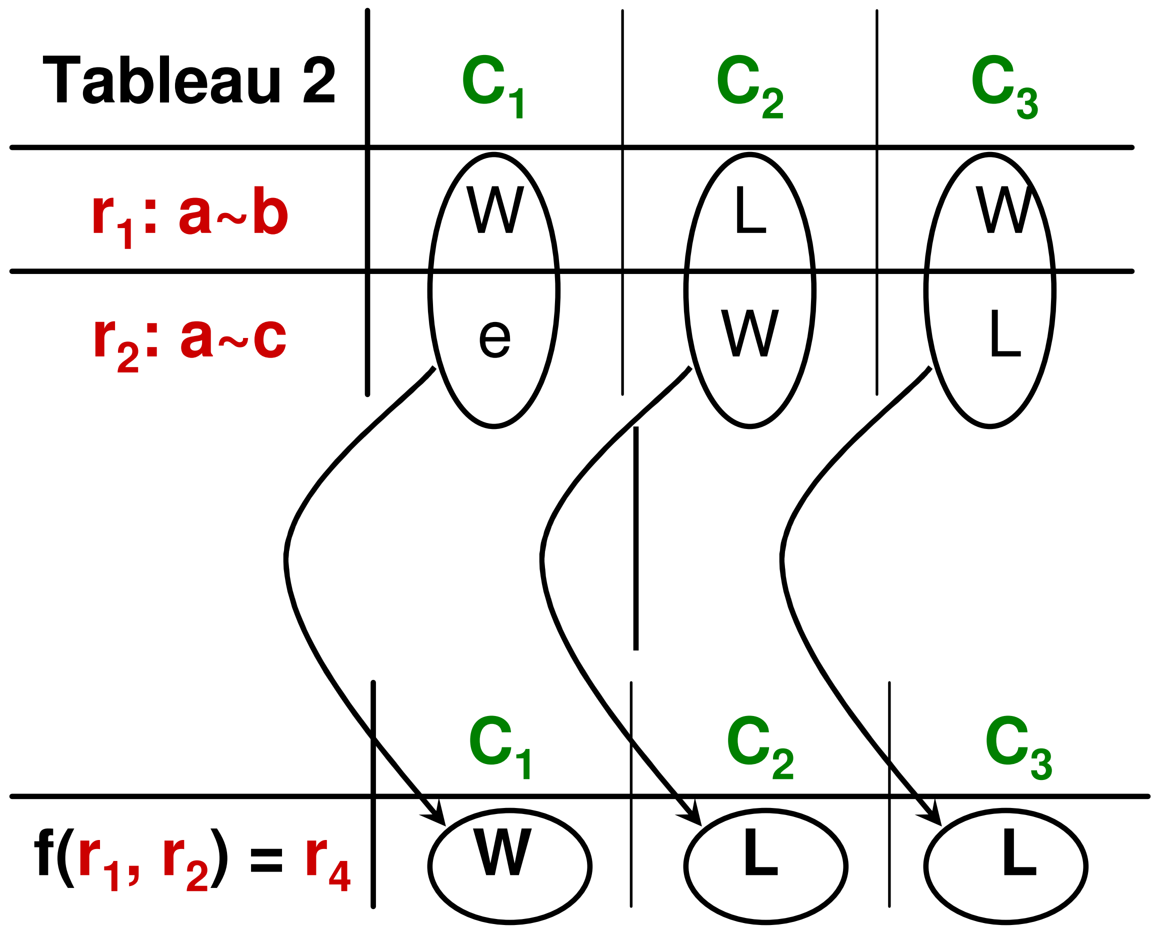
Fusion: select the minimal element relative to the order **L**<**W**<**e**.

Fusion

Fusion: select the minimal element relative to the order **L**<**W**<**e**, that is …

- **L** is dominant: for any X in {**W**, **e**, **L**}, f(**L**, X)=**L**
- **e** is identity: for any X in {**W**, **e**, **L**}, f(**e**, X)=X
- f(**W**, **W**)= **W**
- fusion of two rows r_1 and r_2 is obtained by **constraintwise** fusion and is abbreviated as: f(r_1, r_2), for example …

Fusion



Fusion

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L

	C_1	C_2	C_3
f(r_3, r_2) = r_4	W	L	L

Fusion

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L
$f(r_1, r_2)=r_4$	**W**	**L**	**L**

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L
$f(r_3, r_2)=r_4$	**W**	**L**	**L**

The fusion $f(r_1, r_2)=f(r_3, r_2)$ retains all the ranking information in r_1 / r_3 and strengthens it,

i.e. it locally **maximizes information** in rows r_1 and r_3 based on the rest of the tableau:

we require not only that $C_1 \gg C_2$ (as r_3 does), but also that $C_1 \gg C_3$.

Fusion

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L
f(r_1, r_2)=r_4	**W**	**L**	**L**

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L
f(r_3, r_2)=r_4	**W**	**L**	**L**

Thus, we will use **fusion** to obtain the **Most Informative Basis** (**MIB**) of Tableau 2 / Tableau 3.

But:

fusion is not enough!

Fusion

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L
$f(r_1, r_2)=r_4$	**W**	**L**	**L**

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L
$f(r_3, r_2)=r_4$	**W**	**L**	**L**

Fusion is not enough because we want the MIB to be **equivalent** to the initial tableau, but …

… the ranking information provided by r_2 (i.e. C_2>>C_3) is lost in the fusion $f(r_1, r_2)=f(r_3, r_2)$,

which requires only that C_1>>C_2 and C_1>>C_3.

Fusion

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L
$f(r_1, r_2)=r_4$	**W**	**L**	**L**

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L
$f(r_3, r_2)=r_4$	**W**	**L**	**L**

So:

we keep row r_2 together with the fusion $r_4=f(r_1, r_2)=f(r_3, r_2)$,

which yields Tableau 4!

Fusion

Tableau 4	C_1	C_2	C_3
$f(r_1, r_2)=r_4$	W	L	L
r_2: a~c	e	W	L

Fusion

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L

	C_1	C_2	C_3
$f(r_1, r_2)=r_4$	W	L	L

	C_1	C_2	C_3
r_2: a~c	e	W	L

We keep row r_2 together with the fusion $r_4=f(r_1, r_2)$ …

Fusion

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L

	C_1	C_2	C_3
$f(r_1, r_2)=r_4$	W	L	L

	C_1	C_2	C_3
r_2: a~c	e	W	L

Tableau 4	C_1	C_2	C_3
$f(r_1, r_2)=r_4$	W	L	L
r_2: a~c	e	W	L

… which yields Tableau 4.

The Fusional Reduction Algorithm

The **Fusional Reduction** (**FRed**) algorithm generalizes this two-step strategy, namely:

- **use fusion** to obtain maximally informative rows
- **retain** all the rows that contain **information lost in the fusion**

The Fusional Reduction Algorithm

The question is:

How do we identify the rows that lose information in the fusion?

Answer:

By identifying **info loss configurations** …

The Fusional Reduction Algorithm

And an **info loss configuration** is …

a constraint (i.e. a column in a tableau) that fuses to **W** and contains some **e**’s.

That is,

an info loss configuration is any column in a tableau that contains **only** **e**'s and **W**'s – and **at least** one **e** and one **W**.

The Fusional Reduction Algorithm

Info loss configuration: a constraint that fuses to **W** and contains some **e**'s.

For example:

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L
$f(r_1, r_2)=r_4$	W	L	L

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L
$f(r_3, r_2)=r_4$	W	L	L

The Fusional Reduction Algorithm

And the information that is lost in the fusion is contributed by the rows with an **e**.

The **e** rows in an info loss configuration form the **info residue** of that info loss configuration.

For example:

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L
$f(r_1, r_2)=r_4$	W	L	L

Tableau 3	C_1	C_2	C_3
r_3: d~d'	W	L	e
r_2: a~c	e	W	L
$f(r_3, r_2)=r_4$	W	L	L

The Fusional Reduction Algorithm

The basic Fusional Reduction (FRed) algorithm.

To obtain the **Most Informative Basis** (**MIB**):

- **whole fusion:** fuse all tableau rows and construct a branch for the fusion
- **info loss:** identify all the info loss configurations; for each info loss configuration, construct a branch with the info residue

The Fusional Reduction Algorithm

And this is exactly what we did before with Tableau 2.

whole fusion

info loss config

Tableau 2	C_1	C_2	C_3
r_1: a~b	W	L	W
r_2: a~c	e	W	L

info residue

	C_1	C_2	C_3
f(r_1, r_2)=r_4	W	L	L

	C_1	C_2	C_3
r_2: a~c	e	W	L

The Fusional Reduction Algorithm

There are two further issues:

- cases in which the whole fusion is useless
- cases in which the info residue consists of more than one row

Let's examine them in turn …

The Fusional Reduction Algorithm

First issue: the whole fusion is useless.

For example,

when the info residues repeat the entire initial tableau.

In this case, **we discard the whole fusion**.

The Fusional Reduction Algorithm

Tableau 5	C_1	C_2	C_3
r_5	W	e	L
r_2	e	W	L

	C_1	C_2	C_3
$f(r_5, r_2)$	W	W	L

	C_1	C_2	C_3
r_2	e	W	L

	C_1	C_2	C_3
r_5	W	e	L

The Fusional Reduction Algorithm

Second issue: the info residue consists of more than one row.

For example …

The Fusional Reduction Algorithm

Tableau 6	C_1	C_2	C_3	C_4
r_6	W	L	e	e
r_7	e	W	L	W
r_8	e	e	W	L

	C_1	C_2	C_3	C_4
$f(r_6, r_7, r_8)$	W	L	L	L

	C_1	C_2	C_3	C_4
r_7	e	W	L	W
r_8	e	e	W	L

In this case, **we recurse on the info residue**.

The Fusional Reduction Algorithm

Tableau 6	C_1	C_2	C_3	C_4
r_6	W	L	e	e
r_7	e	W	L	W
r_8	e	e	W	L

	C_1	C_2	C_3	C_4
f(r_6, r_7, r_8)	W	L	L	L

	C_1	C_2	C_3	C_4
r_7	e	W	L	W
r_8	e	e	W	L

	C_1	C_2	C_3	C_4
f(r_7, r_8)	e	W	L	L

	C_1	C_2	C_3	C_4
r_8	e	e	W	L

The Fusional Reduction Algorithm

The Fusional Reduction (FRed) algorithm.

To obtain the **Most Informative Basis (MIB)**:

- **whole fusion:** fuse all tableau rows and construct a branch for the fusion
- **info loss:** identify all the **info loss configurations**; for each info loss configuration, construct a branch with the **info residue**
- **check** if the **whole fusion** has more **L**'s than **the fusion of all the info residues**: keep it if it does, discard it if it doesn't
- **recurse** on each of the info residues, i.e., for each of them, go through the above steps (whole fusion, info loss, check)

FRed Reduces

- Till now, FRed has only *fused* the world
- But FRed can also drastically reduce it

Demo time!

Acknowledgments

The authors, whose names are arranged alphabetically, would like to thank Jane Grimshaw, Naz Merchant, Paul Smolensky, and Bruce Tesar for useful discussion; and, for helpful comments, audiences at the Rutgers Optimality Research Group (April 2004, Nov. 2005), HUMDRUM (May 2004), and the LSA 2005 Summer Institute. 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, misprisions, or infelicities that may have crept into the text.

References

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

Brasoveanu, Adrian. 2003. Minimal Fusion Normal Form. 2003. Ms. Rutgers University, NB. http://www.rci.rutgers.edu/~abrsvn/pdfs/OT/MFNF.pdf.

Brasoveanu, Adrian and Alan Prince. 2004. Fusional Normal Form. Talk given at HUMDRUM, Rutgers University, New Brunswick.

Brasoveanu, Adrian and Alan Prince. 2005. Ranking and Necessity. Part I, ROA 794.

Gigerenzer, Gerd and Daniel G. Goldstein. 1996. Reasoning the Fast and Frugal Way: Models of Bounded Rationality. *Psychological Review* 1996, Vol. 103, No. 4, 650-669.

Gigerenzer, Gerd, Peter M. Todd. and the ABC Research Group. 1999. *Simple heuristics that make us smart*.Oxford University Press. New York.

Grimshaw, Jane. 1997. Projection, Heads, and Optimality. LI 28.4, 373-422; ROA-68.

Hayes, B. 2003, Four Rules of Inference, UCLA ms.

Merchant, Nazarré. 2004. FRed: a Java implemenation. Ms. Rutgers University, New Brunswick

Meyer, Robert K. 1975. Chapters 29.3 and 29.12 in Anderson & Belnap 1975.

Parks, R. Zane. 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.

References

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

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

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

Prince, Alan. 2003. The logic of Optimality Theory. Invited talk, SWOT, UAz, Tucson. http://ling.rutgers.edu/gamma/SWOT.pdf.

Prince, Alan. 2005. LSA 238 Lectures. http://ruccs.rutgers.edu/~prince/ot.html

Prince Alan. 2006. No More Than Necessary, ROA-882.

Riggle, J. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory, PhD dissertation, UCLA.

Samek-Lodovici, Vieri and Alan Prince. 2002. Fundamental Properties of Harmonic Bounding. 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 and Paul Smolensky. 1993. The Learnability of Optimality Theory: an Algorithm and some Basic Complexity Results. ROA-2.

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

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