Stringency and Anti-Paninian Hierarchies

Alan Prince
LSA Institute 6/26/97

(1) Logic of Scales, Hierarchies, and Multiplicities

-Scales: if you fall 150 ft, you have fallen 100 ft.
if you can carry 20 kg, you can carry/have carried 10kg.
maintain voicing for a 200msec closure, & you have done/can do so for 100msec.

-Hierarchies:
if an event occurs in a subordinate clause, it has also occured in a whole sentence.
if a Foot contains a stressed high vowel, so does a Prosodic Word.

-Multiplicities
if a word contains 2 r's, it contains an r.

In every case we can imagine a nested set of intervals anchored at 0: ((((0) 1) 2)....) .

Banning these sets generates a stringency hierarchy: *(0,1) is more stringent (general) than *(0).

(2) **Phony Stringency**:
Multiplicity of violation upsets the apple cart, if we think only of the violation profiles of the elements constraints are defined on. So we must be sure that the constraints stand in the appropriate relation with respect to the actual candidates.

Example. Consider *C]σ vs. *CC]σ

	*C]σ	*CC]σ
CVC.CVC	**	
CVCC.	*	*

On such forms, these constraints conflict directly. So when one invokes properties of stringency one must be sure that the constraints do indeed stand in the stringency relationship!

(3) What has stringency to do with Elsewhere/ Paninian **Contradiction**? The constraints don't conflict. But they disagree, and in interaction with other constraints, this distinguishes different mappings.

(4) **Stringency**: Boolean (pass, fail; two-valued). *S ⇒ *G

/a/	S	G
a		
b		*
c	*	*

(5) Ranking Background: Agreement, Disagreement, Conflict.

Ranking — A, B directly crucially rankable if they conflict on (opt, subopt). Indirectly rankable by transitivity, e.g. $\exists$ T A>>T and T>>B.

(6) Ranking Properties of Constraints in Stringency Relationship.

(7) **Adjacency.** $G{\frown}S \equiv S{\frown}G$.

Pf. G and S do not conflict, so when adjacent, their mutual ranking is not crucial

(8) Activity.

Observe that one or the other or both may be active on different candidate sets. E.g on the above, {a,b} - only G active

{b,c} - only S active

{a,b,c} - if S>>G, then both

(9) **Activity Inhibition** (Analog of "Panini's Theorem" P&S 1993.)

If G>>S, then $G^+ \Rightarrow S^-$. Equivalently $S^+ \Rightarrow G^-$ in the same situation (G>>S).

(10) Observations about activity:

I. If G>>S , S can still be active!! Namely, on some candidate set where G is not active.

2. If S>>G, both can be active. Though when $S{\frown}G$, S's activity is inessential, by Properties I and II above, since $S{\frown}G \equiv G{\frown}S$ and $G^+ \Rightarrow S^-$. So $S^+ {\frown} G^+ \equiv G$. But when S>>T>>G, both can be active and essential. (Construct a case!).

(11) **A generalization to n-ary constraints**. Suppose the order imposed by G on î, call it î/G, *refines* the order î/S. Then the properties cited above still hold.

We do not need or want complete refinement, though. In particular, the bottom class $S^\perp$ can be excluded from consideration so long as $S^\perp \subseteq G^\perp$.

Notice that in the Boolean Case, G refines $S - S^\perp$.

Observe that S and G do not conflict when G refines S. Refinement means that $a >_S b \Rightarrow a >_G b$ so we can't have the conflicting $b >_G a$.

(12) Now suppose the î/G refines î/$S - S^\perp$

(13) **Adjacency.** $G{\frown}S \equiv S{\frown}G$.

Pf. G and S do not conflict, so when adjacent, their mutual ranking is not crucial.

(14) Activity Inhibition Property.

If G>>S, then $G^+ \Rightarrow S^-$. Equivalently $S^+ \Rightarrow G^-$ in the same situation (G>>S).

Discussion. If G is active on some candidate set î, then G(î) $\subseteq G^k$ for some non-bottom k. But everything in G^k belongs to some single class S^m , so S cannot further divide the candidate set after G has winnowed it, and S is therefore inactive.

(15) **A note on Anti-constraints .** One might have naively expected that G>>S would deactivate S completely. But we now know that inactive G opens the way for S to be active. Under what conditions does this unconditional deactivation property emerge?

(16) **Total Deactivation Property**. Suppose A and B have exactly the same violation classes — regardless of the ordering A and B place on these classes. I.e.,, there is a 1-1 map from î/A to î/B for some given î.

Then, if A>>B, B is completely inactive on î. Further, î cannot be used to rank B with respect to *any* other constraint.

Pf. HA(î) $\subseteq A^k = B^{m.}$. So H_1AH_2T(î) $\subseteq B^m$ and H_1AH_2**TB**(î) $\equiv$ H_1AH_2**BT**(î) $\subseteq B^m$.

(Since, below A, everything that remains is in B^m , B can't be ranked crucially with any constraint below A.)

(17) Corollary. **No Duplication**. If a constraint appears twice in a hierarchy only the first instance does anything.

Pf. Since C has the same violation classes as C, for all î, the upper instance of C completely deactivates the lower one.

(18) Corollary. **Pseudo-Parametrization**. Let B be boolean, and let B* assign the opposite values to every form. Then if B>>B*, B is complete invisible and can have no effect whatsoever.

Similarly, if B is n-ary, and B′ has the same violation classes as B on every input, then B>>B′ renders B′ totally ineffective.

(19) Corollary. (Grimshaw).**L vs. R Alignment**. Gradient Alignment to edge E on fixed-length forms deactivates the gradient alignment constraint to the opposite edge.

Pf. For a form of length n, class k of ALIGN-L = class (n−k−1) of ALIGN-R. So the classes are the same.

	Align-α-L	Align-α-R
α x y	0	2
x α y	1	1
x y α	2	0

But if the length of the candidates is not fixed, no go:

	Align-L	Align-R
α x y	0	2
α x	0	1
α	0	0
x α y	1	1
x α	1	0

Now Align-R distinguishes the members of various violation classes of Align-L and all kinds of activity is possible.

I. Entirely Faithful.

In	Out	Max-C	*C]	*CC]
CVCC	☞ CVCC		*	*
	CVC-	* !	*	
	CV--	* ! *		
CVC	☞ CVC		*	
	CV-	*!		

Ranking I:

Max-C
/ \
*C] *CC]

Maps:
CVCC ↣ CVCC
CVC ↣ CVC

II. Paninian Intervention

In	Out	*CC]	Max-C	*C]
CVCC	CVCC	* !		*
	☞ CVC-		*	*
	CV--		** !	
CVC	☞ CVC			*
	CV-		* !	

Ranking II:

*CC] (Spec)
|
Max-C
|
*C] (General)

Maps:
CVC**C** ↣ CVC
CVC ↣ CVC

III. Strict CV

In	Out	*C]	Max-C	*CC]
CVCC	CVCC	* !		*
	CVC-	* !	*	
	☞ CV--		**	
CVC	CVC	* !		
	☞ CV-		*	

Ranking III:

*C] *CC]
|
Max-C

Maps:
CV**CC** ↣ CV
CV**C** ↣ CV

Missing:

the chain shift,
CVCC ↣ CVC
CVC ↣ CV

the gapped system,
CVCC ↣ CVCC
CVC ↣ CV

the crossover.
CVCC ↣ CV
CVC ↣ CVC

CV	III	II	I
|			
CVC		II	I
|			
CVCC			I

IA. Entirely Faithful

In	Out	Max-C	*C]	*CC]	FtBin
CVCC	☞ CVCC		*	*	
	CVC-	* !	*		
	CV--	* ! *			*
CVC	☞ CVC		*		
	CV-	* !			*
CVCCta	☞ CVCCta		*	*	
	CVC - ta	* !	*		
	CV- - ta	* ! *			
CVCta	☞ CVCta		* !		
	CV- ta				

Ranking IA

```
  Max-C      FtBin
   /   \
*C]   *CC]
```

Maps:

CVCC ↦ CVCC
CVC ↦ CVC

IIA. Paninian Intervention - partially simplifying

In	Out	*CC]	Max-C	*C]	FtBin
CVCC	CVCC	* !		*	
	☞ CVC-		*	*	
	CV--		** !		*
CVC	☞ CVC			*	
	CV-		* !		*
CVCCta	CVCCta	*!		*	
	☞ CVC - ta		*	*	
	CV- - ta		** !		
CVCta	☞ CVCta			*	
	CV- ta		* !		

Ranking IIA

```
*CC]      FtBin
  |
Max-C
  |
 *C]
```

Maps:

CVC**C** ↦ CVC
CVC ↦ CVC

IIIA. Strict CV - completely simplifying

In	Out	*C]	FtBin	Max-C	*CC]
CVCC	CVCC	* !			*
	CVC-	* !		*	
	☞ CV--		*	**	
CVC	CVC	* !			
	☞ CV-		*	*	
CVCCta	CVCCta	* !			*
	CVC - ta	* !		*	
	☞ CV- - ta			**	
CVCta	CVCta	* !			
	☞ CV- ta			*	

Ranking IIIA

*C]   *CC]
/   \
Max-C   FtBin

Maps

CV**CC** ↦ CV
CV**C** ↦ CV

IV. Anti-Paninian

In	Out	FtBin	*C]	Max-C	*CC]
CVCC	☞ CVCC		*		*
	CVC-		*	* !	
	CV--	*!		* ! *	
CVC	☞ CVC		*		
	CV-	* !		*	
CVCCta	CVCCta		* !		*
	CVC - ta		* !	*	
	☞ CV- - ta			**	
CVCta	CVCta		* !		
	☞ CV- ta			*	

Ranking IV

FtBin
|
*C]   (General)
|
Max-C
|
*CC]   (Special)

Maps:

#CVCC# ↦ CVCC
#CVC# ↦ CVC (cf **I**)

Otherwise —
CV**CC** ↦ CV
CV**C** ↦ CV   (cf **III**)

V. Monosyllable Partial Simplifying (nb. *CC] >> Max-C)

In	Out	*CC]	FtBin	*C]	Max-C
CVCC	CVCC	* !		*	
	☞ CVC-			*	*
	CV--		* !		**
CVC	☞ CVC			*	
	CV-		* !		*
CVCCta	CVCCta	* !		*	
	CVC - ta			* !	*
	☞ CV- - ta				**
CVCta	CVCta			* !	
	☞ CV- ta				*

Ranking V

```
FtBin
  |
 *C]      *CC]
   \      /
    Max-C
```

Maps:

#CVC**C#** ↦ CVC
#CVC# ↦ CVC (cf **II**)

Otherwise,
CVCC ↦ CV (cf **III)**
CVC ↦ CV

(20) An element hierarchy *a >> *b >> *c .... is equivalent to a Paninian inclusion hierarchy *a >> *{a,b} >> *{a, b, c} ...., so long as a∩b=Ø.

Reason why. Consider *a >> *b vs. *a >> *{a,b}.

	*a	*b	*{a,b}
1. $a^n b^m$	n	m	n+m
2. $a^k b^p$	k	p	k+p

3 cases:

•If n<k, candidate (1) wins on *a.

•If n>k, candidate (2) wins on *a.

‣ If **n=k** then n+**m** > k+**p** is the same as n+**m** > n+**p** and it holds iff m > p (etc.).

I.e. When the decision is passed down the hierarchy by virtue of a tie on *a, the relation of m to p, examined by *b, is the same as the relation of n+m to k+p, since n=k.

Probe Question: why then isn't *{a,b} >> *{a} equivalent to *{a,b} >> *{a,b} ????

(21) Any fixed universal element hierarchy, a la Prince & Smolensky 1993, can then be reinterpreted as a Paninian hierarchy of inclusion. (The basic idea of using such inclusion hierarchies was first put forth in Kiparsky 1993, 1994, though the assumption there may have been that they were descriptively equivalent to fixed element hierarchies, due to an overhasty application of "Panini's Theorem".)

(22) What are the consequences of so doing? The Inclusion Hierarchy theory is much richer, since it replaces a fixed k-element hierarchy with many distinct rankings.

▸What do the additional Anti-Paninian rankings look like?

(23) Consider the interaction between Align-Peak-L (main stress on first syllable) and a peak prominence hierarchy: 1′ ≺ *2′ ≺ *3′≺ ... where 1, 2, 3,... name degrees of intrinsic prominence of syllables from weakest (1) to stronger..., as e.g. $|\sigma_{\mu}| < |\sigma_{\mu\mu}| < |\sigma_{\mu\mu\mu}| < ....$ (Cf Hayes 1995 for extensive discussion, Prince & Smolensky 1993, Walker 1996, Baković 1996, 1997 for disc.)

(24) The conflict is between initial stress (Pk-L) and stressing the weightiest syllable, which need not be initial. (Isomorphic to McCarthy's discussion of Nakanai reduplication 6/24/97, where the conflict is between copying the first vowel and copying the most prominent vowel.)

(25) The crucial cases: those with weight contrast, with heavier element in noninitial position. Limiting ourselves to bisyllables, and a 3-way scale:

12 ↦ **1′** 2 *or* 1 **2′**
23 ↦ **2′** 3 *or* 2 **3′**
23 ↦ **2′** 3 *or* 2 **3′**

(26) The Paninian Rankings

(27) Pk-L >> *{1′} , *{1′, 2′} . Clearly Stress is always initial.

(28) *{1′} >> Pk-L >> *{1′, 2′}. Stress flees from initial 1 to heavier σ if such there is, else initial.

	*{1′}	Pk-L	*{1′, 2′}
1′ 2	* !		*
☞ 1 **2′**		*	*
1′ 3			
☞ 1 **3′**		* !	
☞ **2′** 3			*
2 **3′**		* !	

(29) *{1′}, *{1′, 2′} >> Pk-L. Both 1 and 2 yield stress to stronger σ, else initial.

	*{1′}	*{1′, 2′}	Pk-L
1′ 2	* !	*	
☞ 1 **2′**		*	*
1′ 3	* !	*	
☞ 1 **3′**			*
2′ 3		* !	
☞ 2 **3′**			*

(30) **Anti-Paninian Ranking.** 3 beats 1 and 2. Else initial.

	*{1′, 2′}	Pk-L	*{1′}
☞ **1′** 2	*		*
1 **2′**	*	* !	
1′ 3	* !		
☞ 1 **3′**		*	
2′ 3	* !		
☞ 2 **3′**		*	

```
(31) Paninian:   1 2 3     3   2         3
                            \ /          |
                             1           2
                                         |
                                         1
(32) Anti-Paninian:       3
                         / \
                        2   1
(33) Extended to 4-scale.
       a.Paninian:   1 2 3 4       2 3  4        3   4       4
                                    \ | /         \ /        |
                                                         1            2            3
                                                   |             |
                                                   1             2
                                                                 |
                                                                 1
```

```
b. Anti-P        4            4            4            4  3
                /|\          / \           |            \  /
               2 3 1        2   3          3            /  \
                             \ /          / \          2    1
                              1          2   1
```

(34) Non pathology of AP rankings. All rankings **respect the scale**. — in this sense:
if $k > j$, then $n>k \Rightarrow n > j$.
if $k > j$, then $m < j \Rightarrow k > m$.

(35) Thus, strict domination and inclusion hierarchies fit rather well together. Though free ranking increases the number of predicted systems (= variant implementations of the same underlying scale), they all have the desirable basic property of preserving the basic sense of the scale.

Notation and vocabulary — some standard, some ad hoc

$\Rightarrow$ 'impies' E.g. $p \Rightarrow q$ means 'p implies q', 'if p then q'

$\Leftrightarrow$ if and only if

$\mapsto$ 'maps to' . E.g $a \mapsto b$ means G(a) = b for G the hierarchy under discussion

A, B, C capital letters: constraints

a,b,c lower case: individual forms or form classes

H_k a subhierarchy of constraints.

ABC A>>B>>C

A>>B A dominates B. The hierarchy looks like $H_1AH_2BH_3$

A⁀B A dominates B directly; no other constraint intervenes

î a candidate set. This may be the full Gen(i) for some i, or just what's left after this has been processed by a subhierarchy.

A^k The kth violation class of A.. A^k(î) = those elements of î that violate A k times.

A(î) the output of applying A to the candidate set î — the topmost violation class, wrt constraint A, in î.

î/A The set of violation classes of î wrt A, with the order A^k(î) < A^m(î) for k<m .

A^+ A is active on some candidate set î . That is, A divides î into distinct, nonempty violation classes.

A^- A is inactive on some candidate set.

‖ is not ordered with respect to. a‖b 'a and b are not ordered wrt each other'

'order' The familiar binary relation; here crucially *transitive* — x>y and y>z implies that x>z.

'partial order' In a partial order on a set S, not every pair of elements need stand in the order relation. Consider the relation 'descendant of' — for many pairs of people a,b it is the case that neither 'a is a descendant of b' nor 'b is a descendant of a'.

'refines' An order O_2 refines an order O_1 if $a \geq_{O1} b \Rightarrow a \geq_{O2} b$. That is, O_2 may have more structure than O_1 , but the relations of O_1 are still respected in O_2 . Examples. Suppose we order words by their first letter only. Then **a**nt > **b**at, but **a**nt‖**a**ardvark. Regular alphabetical order is a **refinement** of this order. Suppose we order individuals by military rank: General > Colonel > Major>... Now order the same individuals by age. This is **not** a refinement of the rank order, because some Generals are younger than some Colonels, and vice versa.

Panini **-** Indian grammarian who first introduced a version of the Elsewhere Condition/ Proper Inclusion Precedence Principle, roughly 'specific takes precedence over general'

Paninian order **-** here we refer to a ranking as being in Paninian order if more specific (less stringent) constraints are ranked above more general (more stringent) constraints. Observe that the Paninian order is just one of many rankings of a set of such constraints.

An example in which the subordinated Special constraint is active.

In	Out	FtBin	$*C]_\sigma$	$\mathscr{F}$(q)	$*CC]_\sigma$	$\mathscr{F}$(d)	Comments
fa**q**ta	faq.ta		* !				q→Ø to avoid $C]_\sigma$
	☞ fa.ta			*			
fa**qq**ta	faqq.ta		* !		*		q→Ø to avoid $C]_\sigma$
	faq.ta		* !	*			
	☞ fa.ta			**			
fa**qq**	☞ fá**qq**.		*		*		q→q
	fáq.		*	* !			
	fá.	* !					
fa**dd**	fádd.		*		* !		d→Ø
	☞ fá**d**.		*			*	
	fá.	* !					