
(1) Winners and losers

\[ \text{loser} = \text{candidate that loses under every ranking} \]
\[ \text{winner} = \text{candidate that wins under some ranking} \]

- How does telling winners from losers save us from infinity?
- How can Recursive Constraint Demotion (RCD) be used to tell if a candidate is a loser?

(2) Constraints

- previous view: constraint is a function from candidate to number
- new view: constraint is a function from a candidate set to subset of that set.

(3) Comparative tableaux

- rows are candidate pairs
- cells tell you which member of the pair is preferred by the constraint (if no cells with candidate \( z \), it’s a loser—but not the converse).

(4) Relative harmony

(p. 38): reminds us that a ranking doesn’t just select an optimal candidate; it orders all the candidates.
This fact is exploited in Colin Wilson’s work on opacity.
The same is true of any sub-part of the ranking.

(5) A constraint defines a partial ordering of candidates

Poset (partially ordered set) vs. stratified hierarchy

\[ \text{strata: } \neg(a > b \lor b > a) \rightarrow ((a > x \rightarrow b > x) \& (y > a \rightarrow y > b)) \]
\[ \text{poset: } \text{set ordered by a reflexive, anti-symmetric, transitive relation} \]

Strata work for candidates as ordered by a constraint, but posets (more general) may be necessary for constraints as ordered by a ranking.

(6) Review: properties of relations

reflexive: \( (\forall a) (aRa) \)
irreflexive: \( (\forall a) (\neg aRa) \)
nonreflexive: \((\exists a) (\neg \text{Ra})\)

symmetric: \((\forall a)(\forall b) (\text{aRb} \rightarrow \text{bRa})\)

asymmetric: \((\forall a)(\forall b) (\text{aRb} \land \neg \text{bRa})\)

antisymmetric: \((\forall a)(\forall b) ((\text{aRb} \land \text{bRa}) \rightarrow a=b)\)

nonsymmetric: \((\exists a)(\exists b) (\text{aRb} \land \neg \text{bRa})\)

transitive: \((\forall a)(\forall b)(\forall c) ((\text{aRb} \land \text{bRc}) \rightarrow \text{aRc})\)

intransitive: \((\forall a)(\forall b)(\forall c) ((\text{aRb} \land \text{bRc}) \rightarrow \neg \text{aRc})\)

An equivalence relation is one that is reflexive, symmetric, and transitive.

(7) Composition
\[
f \circ g (x) = f(g(x))
\]
The composition of functions \(f\) and \(g\) is itself a function. It’s the result of applying \(f\) to the result of applying \(g\).
In other words, to apply the composition of \(f\) and \(g\) to \(x\), first apply \(g\) to \(x\), then apply \(f\) to the result. Order matters!

(8) Harmonic bounding
A candidate \(a \in K\) is harmonically bounded relative to the constraint set \(\Sigma\) if there exists some other candidate \(b \in K\) that does at least as well as \(a\) on all constraints, and better than \(a\) on at least one.

(9) Bounding set
= set of candidates that collectively bounds some candidate
A set of candidates \(B \subseteq K\) is a bounding set for \(a \in K\) relative to a constraint set \(\Sigma\) iff:
\* Strictness: Every member of \(B\) is better than \(a\) on at least one constraint in \(\Sigma\)
\* Reciprocity: If \(a\) is better than some member of \(B\) on a certain constraint \(C \in \Sigma\), then some other member of \(B\) beats \(a\) on \(C\).
(a one-member bounding set is the degenerate case)
A candidate is a loser iff it has a non-null bounding set.

• Discuss p. 10: Misleadingly universalized weak bounding is too strong, and pseudo-reciprocity is too weak.

(10) Minimal bounding set
If \(B\) is a bounding set for \(a\) on \(\Sigma\), then \(B\) is a minimal bounding set for \(a\) on \(\Sigma\) iff no strict, non-empty subset of \(B\) is a bounding set for \(a\) on \(\Sigma\).
(11) **Bounding Theorem** (p. 11)
Every loser has a non-null bounding set (and every winner’s only bounding set is null).

(12) **Finding a bounding set**
It would take too long too check all possible sets.

(13) **Aside: Covering set**
$K$ is a covering set for candidate $a$ iff for all constraint rankings, some candidate $x \in K$ is more harmonic than $a$.
The covering set must contain within it a bounding set.
• When is a covering set not itself a bounding set? (p. 12)

(14) **S-L&P’s basic strategy**
Check candidates to exclude them from the bounding set.

(15) **Favoring**
A constraint $C$ favors a candidate $a$ with respect to a set of candidates $K$ if $a$ does as least as well on $C$ as any other member of $K$.

(16) **Constructing the favoring hierarchy** (p. 22)
1. First stratum: all constraints that favor $a$ with respect to the candidate set $K$ (the set of favoring constraints of $a$ in $K$).

   Eliminate from the possible bounding set any candidates that are not favored by all the constraints in the stratum—that is, take the intersection of the top strata of all the constraints in the stratum.

2. Next stratum: all remaining constraints that favor $a$ with respect to the candidates that have not been eliminated.

   Eliminate from the remaining possible bounding set any candidates that are not favored by all the constraints in the stratum.

3. Repeat 2 until you run out of constraints (then $a$ is a winner), or until none of the remaining constraints favor $a$ (then $a$ is a loser and the remaining candidates are the bounding set).

   The favoring hierarchy turns out to be the hierarchy that the RCD would get in trying to construct a ranking that chooses $a$.

• Let’s try some examples

• Compare to RCD (pp. 25-26)
(17) **Residual candidates**
Candidates that are better than $a$ on some residual constraint.
The residual candidates are a bounding set

- **strictness**: by definition
- **reciprocity**: $a$ is not better than any of the residual candidates on any non-residual constraints

This puts an *upper bound* on the size of the minimal bounding set.

(18) **Disfavoring system**
A disfavoring system for $a$ is a collection of constraints in which, for each constraint, $a$ loses to some other candidate

(19) **Blocking set**
For each constraint in a disfavoring system for $a$, pick one candidate that beats $a$.
The blocking set will have the same size as the number of constraints in the disfavoring system.

A blocking set is a bounding set (p. 31).

(20) **Size for the bounding set**
The residual constraints are a disfavoring system for $a$, so a blocking set constructed from them is a bounding set for $a$.

So, we can always construct a bounding set for $a$ that is no bigger than the number of residual constraints for $a$.

- Let’s look at the examples we worked above

*Aside:* If all the constraints are binary, then the smallest possible bounding set for any loser has just one member (p. 12).

(21) **Discussion**
- This gets us a finite number of candidates to worry about (for learning, generation), but of course the initial procedure deals with an infinite set of candidates, so we still need to finite-ize that.

- Can we adapt this procedure to identify not just loser candidates, but loser languages?

- It would be fun to make an implementation of this.
Jason Eisner (1997). What Constraints Should OT Allow?

Basics of OTP (= Primitive OT)

(22) Representations

= temporally aligned tiers (represent prosody and gestures)

edges: [, ]
interiors: F

ŋk:

\[
\begin{array}{c}
\text{voi} \\
\text{nas} \\
\text{cl} \\
\text{vel}
\end{array} = \begin{array}{c}
\text{voi} \\
\text{nas} \\
\text{cl} \\
\text{vel}
\end{array}
\]

• Gen requires matched pairs of brackets & no nesting on a tier
• Input has its own tier(s) (underlined)—floating input material may be placed anywhere on the tier
• Otherwise, anything goes

(23) Correspondence

I-O Correspondence = alignment of input and output tiers
B-R Correspondence = alignment of R tiers and copy of B tiers (Gen is required to make a perfect copy of the base)

How about O-O Correspondence?

(24) Constraints

Refer to temporal relationships among elements of the representation

require simultaneity: \( \alpha \rightarrow \beta \quad (\forall \alpha) \quad (\exists \beta) \quad (\alpha \text{Coincide} \beta) \)

one violation for every \( \alpha \) without a \( \beta \)

forbid simultaneity: \( \alpha \perp \beta \quad (\forall \alpha) \quad \neg(\exists \beta) \quad (\alpha \text{Coincide} \beta) \)

one violation for every overlapping \( \alpha-\beta \) pair

\( \alpha \) and \( \beta \) can be edges, interiors, and conjunctions and disjunctions thereof
(25) Examples: MacEachern 1999

\text{IDENT}[^{\text{LAR}}] \text{ must be broken down into a set of constraints (this is what MacEachern}
\text{ does too)}

\begin{align*}
\text{c}g & \rightarrow \text{c}g \quad \text{DEP}[CG] \\
\text{c}g & \rightarrow \text{c}g \quad \text{MAX}[CG] \\
\text{s}g & \rightarrow \text{s}g \quad \text{DEP}[SG] \\
\text{s}g & \rightarrow \text{s}g \quad \text{DEP}[SG]
\end{align*}

\text{IDENT}[^{\text{PLACE}}] \text{ again, break down into a set of constraints}

\begin{align*}
\text{l}a\text{b} & \rightarrow \text{l}a\text{b} \quad \text{DEP}[LAB] \\
\text{l}a\text{b} & \rightarrow \text{l}a\text{b} \quad \text{MAX}[LAB]
\end{align*}

\text{etc.}

*\text{LARSIM} \text{ this is trickier—works if the Cs are adjacent on a C tier, but not otherwise}

\text{BEIDENTICAL} \text{ similarly. Unless we assume pseudoreplicative correspondence.}

(26) \text{Are these “bad” constraints unrepresentable in OTP?}

\text{PALINDROMIC} \\
\text{FTQUINT} \\
\text{MEMBEROF}(\text{A, AARDVARK, AARDVARKS, AARDWOLF, AARDWOLVES, AARON}…) \\
\text{MATCHESOUTPUTOFSP}\text{E}

(27) \text{How about these “questionable” constraints?}

\text{FTBIN} \\
\text{ALIGN-L(FOOT, PRWD)}

(28) \text{Discussion}

\begin{itemize}
  \item Existential vs. universal faithfulness (Struijke, Feng)
  \item Harmony
  \item \text{REDUP} (Zuraw), long-distance consonantal correspondence (Walker & Rose)
  \item \text{OCP}
  \item Is the set of possible constraints finite in OTP?
  \item Some “bad” constraints are still allowed.
  \item Adaptability to syntax?
\end{itemize}
Next Time
  • Generation

For next time
  • Read
    o Tesar & Smolensky ch. 8
  • Homework
    o Construct the favoring hierarchy for each candidate in some tableau—perhaps one of the tableaux from an example you used in a previous homework. Show intermediate steps, and don’t forget to include the tableau.
    o Using the same tableau if possible, rewrite the constraints in Eisner’s Primitive OT (see pp. 6-13 for examples), and draw the candidates using Eisner’s timeline notation (see 15 on p 4). Bring 5 copies to class.