
1. Structure of *SPE* grammar (ignoring interaction with morphology)
   a. An **ordered set of rewrite rules** \{R_1, R_2, ..., R_{n-1}, R_n\} of the form A → B / X__Y.
   b. A **lexicon** L of underlying representations \{UR_1, UR_2, ...\}.
   c. **Derivation** of a form: R_n ∘ R_{n-1} ∘ ... ∘ R_2 ∘ R_1(UR_k) = surface representation SR_k.

2. Basic rule notation and terminology
   a. A → B / X__Y = XAY → XBY.
   b. A is the **focus** or target or **affected segment** of the rule.
   c. (A →) B is the change made by the rule; XBY is its **structural change**.
   d. X__Y is the **context** or environment of the rule; XAY is its **structural description**.
   e. A rule R is **applicable** to any string S iff S is **nondistinct** from the structural description of R.¹
      (More on the definition of nondistinctness can be found in 6 on the next page.)
   f. Application of R to S may be **vacuous**, meaning (nothing more than) that XAY = XBY.
      - e.g., [–sonorant] → [–voice] / [–voice] __ # applies vacuously to /mæks/ ‘Max’.

3. A: the rule focus
   a. A can be a featural description of a class of segments (a ‘feature matrix’) or ∅.²
   b. If A is ∅, you have an insertion/epenthesis rule (changing ‘nothing’ into something).
   c. Sometimes, if A is meant to pick out a single sound, we use a phonetic (e.g., IPA) symbol
      instead: u → [–high] / (C)#. This is a good idea for the sake of readability, but keep in
      mind that the ‘real’ rule replaces the symbol with the smallest feature matrix that distin-
      guishes that sound from others (at the derivational point at which the rule applies).
      - What does the [u] above abbreviate if the vowel inventory at the derivational point at
        which the rule applies is i, a, u? What if it’s i, a, u, o? What if it’s i, y, a, u, o?

   d. Sometimes we also use C to abbreviate [–syl] or V to abbreviate [+syl]. Again, this is
      good for readability. Be careful when you read, though, because some authors, following
      *SPE*, use C and V to abbreviate {–voc}, [+cons]} and [+voc, –cons].

4. B: the rule change
   a. B can also be a feature matrix or ∅.
   b. If B is ∅, then the segment that A matches is deleted.
   c. If B is a feature matrix, then any of the affected segment’s features that are mentioned in
      B are changed to the value given in B. **All other features are left alone**.
      - What does [+syl] [–low] → [+high] do to [o]? What about [u]? (And what about [a]?)

¹ R actually applies to any applicable string S iff S is in the immediate derivational input to R. The distinction between the applicability and application of rules is particularly helpful in figuring out necessary orders between rules.
² This is true of *SPE* proper, where the only representational units were feature matrices and word/morpheme boundaries (which were themselves thought to be feature matrices, with *ad hoc* features like ‘[–segment]’). Once other representational units were introduced into the theory (e.g. syllables, feature class nodes), these also became possible instances of A, B, X, or Y. But around the time (mid-to-late 1970s through the 1980s) that these representational enhancements were being proposed, the forms and functions of phonological rules were also being significantly rethought to accommodate those enhancements; as John McCarthy (1988: 84) famously (but perhaps too optimistically) put it, “if the representations are right, then the rules will follow.”
d. Again, we sometimes use a phonetic symbol as an abbreviation for all the feature changes necessary to change anything that could match A into the feature matrix represented by that symbol: $[-\text{low}] \rightarrow i / ~ #$. (Considerations noted previously apply equally here.)

- What does the [i] above abbreviate if the vowel inventory at the derivational point at which the rule applies is $i, a, u$? What if it’s $i, a, u, o$? What if it’s $i, y, a, u, o$?

e. If A is $\emptyset$, then a phonetic symbol most conveniently abbreviates all of the features needed to pick the inserted segment out of the relevant inventory: $\emptyset \rightarrow i / C ~ C#$.

5. X__Y: the rule context

a. X and Y are strings made up of three types of units: feature matrices (or abbreviatory symbols), the word (#) and morpheme (+) boundaries, or specific category boundaries.

- Feature matrices in X and Y match segments in the same way that A does (i.e., they match a segment if not distinct from it). Phonetic symbols also work the same way.

- The boundaries # and + are treated in SPE as feature matrices (recall fn. 2):$^3$

$$# \text{ is } \begin{bmatrix} -\text{seg} \\ -\text{FB} \\ +\text{WB} \end{bmatrix} \quad \text{and is } \begin{bmatrix} -\text{seg} \\ +\text{FB} \\ -\text{WB} \end{bmatrix}$$

- Category boundaries (labeled brackets) like $\text{[}_\text{Noun}\text{]}$ and $\text{[}_\text{Verb}\text{]}$ can also be used, but only at the edges of X__Y (and if both edges have them, the labels have to match). By convention, contexts like $\text{[}_\text{VC#}]_\text{Noun}$ can be abbreviated as $\text{[}_\text{VC}]_\text{Noun}$.

b. Contexts are assumed to be local to the rule target. SPE interprets locality strictly and horizontally (e.g., strings immediately adjacent to A). Much later work noted in fn. 2 dispensed with the idea of boundary matrices altogether, requiring that there also be a vertical notion of locality (e.g., the category or constituent immediately containing A).$^4$

6. Nondistinctness

a. If A is a feature matrix, then the rule is in principle applicable to any segment that is nondistinct from that matrix. Two feature matrices are distinct iff there is some feature whose value is different in the two matrices. This means that if A doesn’t mention some feature f, it “doesn’t care” about it — that part of the rule matches segments that are $[+f]$, segments that are $[-f]$, or even segments that fail to have a value for $[\pm f]$.

- Which of the following are distinct from $[+\text{syll}][-\text{low}]$,$\begin{bmatrix} [+\text{syll}] \\ [+\text{low}] \end{bmatrix}$,$\begin{bmatrix} [-\text{low}] \\ [+\text{back}] \end{bmatrix}$,$\begin{bmatrix} [-\text{low}] \\ [-\text{round}] \end{bmatrix}$,$\begin{bmatrix} [+\text{syll}] \\ [+\text{low}] \end{bmatrix}$,$\begin{bmatrix} [+\text{high}] \\ [+\text{round}] \end{bmatrix}$?

b. Here’s how we extend the definition of nondistinctness from pairs of units to pairs of strings, to accommodate the entire structural description (rule focus + context): XAY matches (is nondistinct from) some string S iff X and S have the same number of units n, and the ith unit of XAY matches (is not distinct from) the ith unit of S for all $1 \leq i \leq n$.$^5$

---

$^3$ SPE also proposes a third boundary type, =, which has the features $[-\text{FB},-\text{WB}]$ and is more or less the boundary between non-productive or nontransparent affixes and stems (e.g., English per=mit); you won’t see this one much. Also, $\#$ indicates both what we (pretheoretically) think of as a word boundary — roughly, the space between two written words — but this boundary is also found between members of compounds (e.g., blackboard), between stems and some affixes (e.g., happiness), and elsewhere.

$^4$ ‘Tiered’ (multi-dimensional) representations expanded the interpretation of locality even more. We’ll discuss some of this later.

$^5$ ‘Special’ if + is included in X and V, then it is required: $V \rightarrow \emptyset / _+ VC$ does not apply to $ibauk$ because $V+VC$ does not match $auk$. But $V \rightarrow \emptyset / _- VC$ does apply to $ibauk$ because $V-C$ matches any of $\{VVC, V+VC, VV+C, V+V+C\}$. 

Fall 2016

1. Structure of OT grammar
   a. \( \text{Gen}(\text{In}_k) \rightarrow \{\text{Out}_1, \text{Out}_2, \ldots\} \)
      i. \( \text{In}_k \): a specific input (possibly but not necessarily an underlying representation).
      ii. Gen: a function that generates a set of candidate outputs (\( \text{Out}_1 \), etc.) for \( \text{In}_k \).
         - The candidate set is assumed to be theoretically infinite but algorithmically finite.
   b. \( \text{H-Eval}(\text{Out}_i, 1 \leq i \leq \infty) \rightarrow \text{Out}_{real} \)
      i. H-Eval: a function that evaluates the harmony of each \( \text{Out}_i \) relative to \( \text{In}_k \).
      ii. \( \text{Out}_{real} \): the highest-harmony/optimal/grammatical \( \text{Out}_i \) for \( \text{In}_k \) w.r.t. H-eval.

2. Harmony and optimality
   a. Harmony is a measure of how well an output satisfies a single constraint or a hierarchy of constraints. (In what follows, \( x \succ y \) stands for \( x \) is more harmonic than \( y \).)
   b. Single constraint. \( x \succ y \) according to constraint \( C \) iff \( x \) violates \( C \) fewer times than \( y \) does.
      For any two \( x, y \) and any \( C \) we have exactly one of \( x \succ y \), \( y \succ x \), or \( x \equiv y \).
   c. Constraint hierarchy. \( x \succ y \) according to hierarchy \( H \) iff the highest-ranked constraint that distinguishes between \( x \) and \( y \) is better-satisfied by \( x \) (from Grimshaw 1997, \textit{LI}).
   d. An output \( x \) is optimal for a given input according to hierarchy \( H \) iff there is no output \( y \) for the same input such that \( y \succ x \) according to \( H \) (i.e., iff \( x \) is ‘unbeaten’).

3. Violaible well-formedness constraints\(^6\)
   a. \textbf{Markedness} constraints (roughly equivalent to SPE structural descriptions) assign violations to structures to specified structures in (or absent from) an output. Some examples:
      i. \textsc{Onset}. A syllable must begin with a consonant.
      ii. \textsc{NoCoda}. A syllable must not end with a consonant.
      iii. \textsc{NoVoCODA}. A voiced obstruent must not occur syllable-finally.
      iv. \textsc{FTBin}. Feet are binary at some level of analysis (\( \mu, \sigma \)).
   b. \textbf{Faithfulness} constraints (roughly equivalent to SPE structural anti-changes) demand that an output be identical to its input in a particular way. Some examples:
      i. \textsc{Max-C}. A consonant in the input must have a correspondent in the output.
      ii. \textsc{Dep-V}. A vowel in the output must have a correspondent in the input.
      iii. \textsc{Contiguity}. Adjacent input segments have adjacent output correspondents.
      iv. \textsc{Linearity}. If \( a \) precedes \( b \), then \( a \)'s correspondent must precede \( b \)'s correspondent.
      v. \textsc{Ident(VoI)}. Corresponding segments must have the same value of \( [\pm \text{voi}] \).

4. Strong OT claim: grammars are constructed from \textbf{violable} constraints that are \textbf{ranked} in strict domination hierarchies. Some examples of core issues raised by this strong claim:
   a. Content of the constraints (e.g., more vs. less substantive/formal).
   b. Origin of the constraints (e.g., learned vs. innate; learning mechanisms; lg. specificity).
   c. Method of interaction (e.g., strict domination vs. numerical weighting).
   d. Input-output mapping (e.g., parallel evaluation vs. serial derivation).

\(^6\) The examples below are some relatively well-established constraints; names and formal subtleties of definition may vary. \textit{Nota benc}: constraints are very often stated using informal language like you see here, but you should demand more formal definitions for evaluating them and you should use more formal definitions when proposing them yourself. A useful way to start is to think of a constraint algorithmically: ‘assign a violation when such-and-such is but shouldn’t be / isn’t but should be / changes’.
5. Some analytical strategies (or desiderata) in OT
   a. Constraints are maximally general. In particular, “do such-and-such only when so-and-so” (triggering) and “do such-and-such except when so-and-so (blocking) should be derived to the extent possible from the interaction between conflicting constraints.
   b. Putative universals should also be derived (again, to the extent possible) from the possibilities of constraint interaction, rather than via stipulations about Gen.

6. Digression on generality/simplicity: maximally general rules are also analytical desiderata in SPE (recall ‘smallest feature matrix’ from 3.c on p. 1); in fact, the ink-saving virtues of maximum generality were (and in some circles still are) the foundation of an evaluation metric for selecting the shortest (and supposedly psychologically real) grammatical description from among a set of descriptions equally compatible with the data. This of course assumes that we start from all the right assumptions: the right features, the right formal notation, the right conventions, etc. (And what happens if there’s a tie for shortest description?) More plausibly, we want to find independent evidence as to which grammatical description is the psychologically real one, and make sure our theory explains how/why the learner would choose that one over others — and this is a much harder task! Also, we should be careful not to confuse conciseness with generality: the structural description of NO-SIM (*C_iC_j, where C_i = C_j but ignoring [±voi]) is more general than, but also less concise than, that of NO-GEM (*C,C).

7. Ranking and factorial typology
   a. Many (but certainly not all) of the more interesting results in OT come from considerations of factorial typology: given a set of interacting constraints, how many distinct grammars (In→Out maps) can be described by that set, and what are those grammars?
   b. n constraints → n! possible rankings, where n! = n × (n – 1) × (n – 2) × … × (n – n + 1).
   c. 2! = 2, 3! = 6, 4! = 24, 5! = 120, 6! = 720, 7! = 5040, 8! = 40320, 9! = 362880 … whew!
   d. Does this mean that there are n! distinct grammars described by n constraints? No, because some constraints do not interact with each other (either inherently, because they don’t conflict, or under certain rankings with respect to other conflicting constraints).

8. Some issues of possible contention
   a. Is constraint ranking total, or (crucially) partial? (Is a crucial “tie” possible?)
   b. Are all constraints universal? (If not, what’s the use of factorial typology?)
   c. Does every grammar (= language) consist of a single ranking of constraints? (What about affix classes, lexical classes, exceptions, interactions with other modules, etc.?)

9. Some core principles (and strict interpretations thereof)
   a. There are (only) violable well-formedness constraints.
      • There are no rules, no inviolable constraints, etc.
   b. There is (only) a total, strict domination hierarchy of constraints.
      • There is no numerical weighting, no constraint ties, etc.
   c. There are (only) universal constraints.
      • All languages have the same constraints, with no language-specific information, etc.
   d. Ranking is language-particular (and only ranking is language-particular).
      • There are no language-particular constraints on inputs, etc.
   e. Output candidates for a given input are (only) compared in parallel.
      • There is no serial derivation, no intermediate representations, etc.

---