1 Introduction

Harmonic Serialism (Prince & Smolensky, 1993/2004; McCarthy, 2000)

• Serial variant of Optimality Theory (Prince & Smolensky, 1993/2004)
• GEN only produces candidates that differ from the input via the application of one operation
• Inputs may pass multiple times through GEN and EVAL before derivation converges

Local vs. global optima

• Parallel OT always reaches the global optimum of harmony; HS can get stuck in local optima
• In many cases, local optima are a better typological fit (McCarthy, 2008a)

Advantages of serial footing

• Accounts for stress-syncope interactions which are difficult to capture in parallel OT (McCarthy, 2008b)
• Avoids global processes that depend on whether an input is odd- or even-parity (Pruitt, 2010, 2012)
• Disallows vowel epenthesis from enforcing binary feet (Moore-Cantwell, 2016)
2 Footing in HS (Pruitt, 2010, 2012)

2.1 Operations

(2) Candidate set for /σσσσ/

| σ → (σ) | Faithful: σσσσ |
| σσ → (σσ) | Monosyllabic foot: (σ)σσσ | σ(σ)σσ | σσ(σ)σσ | σσσ(σ)σσ |
| σσ → (σ′σ) | Disyllabic trochee: (σσ)σσ | σ(σσ)σσ | σσ(σσ)σσ |
| σσ → (σ′σ) | Disyllabic iamb: (σ′σ)σσ | σ(σ′σ)σσ | σσ(σ′σ)σσ |

2.2 Constraints

PARSE-σ: Assign one violation mark for every syllable that is not a member of some foot.

σσσσ × σσ(σσ) × σ(σσ)σσ × σσ(σσ)σσ

ALLFT-L: For each foot in a word assign one violation mark for every syllable separating it from the left edge of the word.

σσ(σ) × σσ(σσ) × σσ(σσ)σσ × σσ(σσ)σσ

ALLFT-R: For each foot in a word assign one violation mark for every syllable separating it from the right edge of the word.

(σ)σσσ × (σσ)σσ × (σσ)σσσ × (σσ)σσ

TROCHEE: Assign one violation mark for a foot whose head is not aligned with its left edge.

(σ′σ)σσ × (σσ)σσ × (σσ)σσ

IAMB: Assign one violation mark for a foot whose head is not aligned with its right edge.

(σσ)σσ × (σσ)σσ × (σσ)σσ

FTBIN: Feet are binary at some level of analysis (μ, σ).

(σ)σσσ × (σσ)σσ × (σσ)σσ

σσσσ
2.3 Illustration

- The alignment constraints ALLFt-L and ALLFt-R determine whether feet are parsed and where.

**Exhaustive left-to-right trochees**: /LLLLL/ → (LL)(LL)(L)

- Ranked below Parse-σ, ALLFt-L pushes feet as far to the left as possible

\[(3) /LLLLL/ \rightarrow (\text{LL})(\text{LL})(\text{L}): \text{Step 1}\]

<table>
<thead>
<tr>
<th>/LLLLL/</th>
<th>Parse-σ</th>
<th>ALLFt-L</th>
<th>Trochee</th>
<th>ALLFt-R</th>
<th>Iamb</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. LLLL</td>
<td>W 5</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>→ b. (LL)LLL</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. (L)LLL</td>
<td>W 1</td>
<td>W 4</td>
<td>L</td>
<td>W 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. (L)LLL</td>
<td>W 4</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. LLL(LL)</td>
<td>3</td>
<td>W 3</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[(4) /LLLLL/ \rightarrow (\text{LL})(\text{LL})(\text{L}): \text{Step 2}\]

<table>
<thead>
<tr>
<th>(LL)LLL</th>
<th>Parse-σ</th>
<th>ALLFt-L</th>
<th>Trochee</th>
<th>ALLFt-R</th>
<th>Iamb</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. (LL)LLL</td>
<td>W 3</td>
<td>L 1</td>
<td>4 2 W 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. (LL)(LL)</td>
<td>W 2</td>
<td>2 1 4 L 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. (LL)(L)LL</td>
<td>W 2</td>
<td>W 5 L 1 W 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. (LL)L(LLL)</td>
<td>W 3 W 3</td>
<td>L 3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[(5) /LLLLL/ \rightarrow (\text{LL})(\text{LL})(\text{L}): \text{Step 3}\]

<table>
<thead>
<tr>
<th>(LL)(LL)</th>
<th>Parse-σ</th>
<th>ALLFt-L</th>
<th>Trochee</th>
<th>ALLFt-R</th>
<th>Iamb</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. (LL)(LL)</td>
<td>W 1</td>
<td>L 2 4 2 L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>→ b. (LL)(LL)</td>
<td>W 6</td>
<td>4 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. (LL)(L)LL</td>
<td>W 2</td>
<td>1 4 L 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. (LL)L(LLL)</td>
<td>W 3</td>
<td>L 3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Non-iterative stress**: /LLLLL/ → (LL)LLL

- Ranked above Parse-σ, ALLFt-L prevents any misaligned feet from being parsed

\[(6) /LLLLL/ \rightarrow (\text{LL})LLL: \text{Step 1}\]

<table>
<thead>
<tr>
<th>/LLLLL/</th>
<th>Parse-σ</th>
<th>ALLFt-L</th>
<th>Trochee</th>
<th>ALLFt-R</th>
<th>Iamb</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. LLLL</td>
<td>W 5</td>
<td>L 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>→ b. (LL)LLL</td>
<td>3</td>
<td>3 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. (L)LLL</td>
<td>W 1</td>
<td>W 4</td>
<td>L 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. (L)LLL</td>
<td>W 4</td>
<td>L 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. LLL(LLL)</td>
<td>W 3 W 3</td>
<td>L 3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[(7) /LLLLL/ \rightarrow (\text{LL})LLL: \text{Step 2}\]

<table>
<thead>
<tr>
<th>(LL)LLL</th>
<th>Parse-σ</th>
<th>ALLFt-L</th>
<th>Trochee</th>
<th>ALLFt-R</th>
<th>Iamb</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. (LL)LLL</td>
<td>W 2</td>
<td>3 3 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. (LL)(LL)</td>
<td>W 2</td>
<td>L 1 4 2 W 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. (L)LLL</td>
<td>W 2</td>
<td>W 4 1 W 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. (L)LLL</td>
<td>W 2</td>
<td>W 5 1 W 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. LLL(LLL)</td>
<td>W 3</td>
<td>3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.4 Factorial Typology

- To my knowledge, there isn’t a published factorial typology of this system, so I wrote a python script to calculate one

- 16 languages were found which uniquely assigned outputs to two inputs: the even-parity /LLLLLL/ and the odd-parity /LLLLLLLL/
### Parsing feet in Directional Harmonic Serialism PhoNE - April 13

<table>
<thead>
<tr>
<th>8)</th>
<th>/LLLLLL/</th>
<th>/LLLLLLL/</th>
<th>Trochees</th>
<th>Exhaustive</th>
<th>Inexhaustive</th>
<th>Iterative</th>
<th>Non-iterative</th>
</tr>
</thead>
</table>

### (9)

<table>
<thead>
<tr>
<th>Parse-σ</th>
<th>FtBin</th>
<th>Parse-σ</th>
<th>FtBin</th>
<th>Parse-σ</th>
<th>FtBin</th>
<th>Parse-σ</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllFt-L Trochee</td>
<td>AllFt-L Trochee</td>
<td>AllFt-R Iamb</td>
<td>AllFt-R Iamb</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>FtBin</td>
<td>Parse-σ</td>
<td>AllFt-L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Parse-σ</td>
<td>AllFt-L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>AllFt-R Iamb</td>
<td>AllFt-R Iamb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>AllFt-L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>AllFt-L Trochee</td>
<td>AllFt-L Trochee</td>
<td>AllFt-L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Parse-σ</td>
<td>Parse-σ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>AllFt-R Iamb</td>
<td>AllFt-R Iamb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>AllFt-L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Iterative**: Parse-σ ≫ AllFt-L, AllFt-R
- **Non-iterative**: AllFt-L ≫ Parse-σ or AllFt-R ≫ Parse-σ
- **Left-to-right**: AllFt-L ≫ AllFt-R
- **Right-to-left**: AllFt-R ≫ AllFt-L
- **Trochees**: Trochee ≫ Iamb
- **Iambs**: Iamb ≫ Trochee
- **Monosyllabic**: Trochee, Iamb ≫ Parse-σ
- **Exhaustive**: Parse-σ ≫ FtBin
- **Inexhaustive**: FtBin ≫ Parse-σ
3 Problems with alignment

**Distance-sensitive** (McCarthy & Prince, 1993; Hyde, 2012, 2016)

- For every foot F, assign one violation for every syllable between F and the left/right edge of the word

\[
\begin{array}{|c|c|c|}
\hline
\text{Foot} & \text{Violation} & \text{Foot} \\
\hline
\text{a. (LL)LLL} & 4 & \text{b. L(LL)LLL} \\
\text{b. L(LLLLLLL)} & 1 & \text{c. LL(LLLLLL)} \\
\text{c. LL(LLLLLLL)} & 2 & \text{d. LLLL(LLLL)} \\
\text{d. LLLL(LLLL)} & 4 & \text{e. LLLL(LLLL)} \\
\hline
\end{array}
\]

**Distance-insensitive** (Eisner, 1998; Kager, 2001; McCarthy, 2003; Buckley, 2009)

- For every foot F, assign one violation if F is not at the left/right edge of the word

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Violation} \\
\hline
\text{a. (LL)LLL} & 1 \\
\text{b. L(LLLLLLL)} & 1 \\
\text{c. LL(LLLLLLL)} & 1 \\
\text{d. LLLL(LLLL)} & 1 \\
\text{e. LLLL(LLLL)} & 1 \\
\hline
\end{array}
\]

- Distance-insensitive alignment produces largely the same typology, but without non-iterative stress

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{a. (LL)LLL} & \text{Trochee} \\
\text{b. L(LLLLLLL)} & \text{Trochee} \\
\text{c. LL(LLLLLLL)} & \text{Iamb} \\
\text{d. LLLL(LLLL)} & \text{Iamb} \\
\text{e. LLLL(LLLL)} & \text{Both} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{1. (LL)(LL)(LL)} & \text{Trochee} \\
\text{2. (LL)(LL)(LL)(L)} & \text{Trochee} \\
\text{3. (LL)(LL)(LL)(LL)} & \text{Trochee} \\
\text{4. (LL)(LL)(LL)(LL)(L)} & \text{Trochee} \\
\text{5. (LL)(LL)(LL)(LL)} & \text{Iamb} \\
\text{6. (LL)(LL)(LL)(LL)(LL)} & \text{Iamb} \\
\text{7. (LL)(LL)(LL)(LL)(LL)(L)} & \text{Both} \\
\text{8. (LL)(LL)(LL)(LL)(LL)(LL)} & \text{Both} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{1. *Ft} & \text{Trochee} \\
\text{2. *Ft} & \text{Trochee} \\
\text{3. *Ft} & \text{Trochee} \\
\text{4. *Ft} & \text{Trochee} \\
\text{5. *Ft} & \text{Iamb} \\
\text{6. *Ft} & \text{Iamb} \\
\text{7. *Ft} & \text{Iamb} \\
\text{8. *Ft} & \text{Iamb} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{9. } & \text{ Both} \\
\text{10. } & \text{ Both} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{11. } & \text{ Trochee} \\
\text{12. } & \text{ Trochee} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{13. } & \text{ Trochee} \\
\text{14. } & \text{ Trochee} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Foot} & \text{Type} \\
\hline
\text{15. } & \text{ Trochee} \\
\text{16. } & \text{ Trochee} \\
\hline
\end{array}
\]
3.1 Why alignment is necessary

- At each step, alignment determines where feet are parsed
- Without this control, divergent ties (Pruitt, 2009) produce different outputs for a given input

\[(14) \quad /LLLL/ \rightarrow (LL)(LL) \sim L(LLL)\]

<table>
<thead>
<tr>
<th>/LLLL/</th>
<th>FtBin</th>
<th>Parse-σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. LLLL</td>
<td>W 4</td>
<td></td>
</tr>
<tr>
<td>→ b. (LL)LL</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>→ c. L(LLL)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>→ d. LL(LLL)</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L(LLL)</th>
<th>FtBin</th>
<th>Parse-σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ a. L(LLL)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>b. (LL)(LL)</td>
<td>W 1 L 1</td>
<td></td>
</tr>
<tr>
<td>c. L(LL)(LL)</td>
<td>W 1 L 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LL(LL)</th>
<th>FtBin</th>
<th>Parse-σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. LL(LL)</td>
<td>L 2</td>
<td></td>
</tr>
<tr>
<td>→ b. (LL)(LL)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(LL)(LL)</th>
<th>FtBin</th>
<th>Parse-σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ a. (LL)(LL)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- There are multiple local optima, some better than others
- Derivations in HS cannot look ahead to later results. In Step 1, all feet are equally promising

\[(15)\]

3.2 Problem 1: distance-sensitive alignment is computationally complex

- Loci of violation grow quadratically in the length of the input
- This cannot be represented by a finite-state transducer (Eisner, 1997a,b; Biró, 2003)
- Opens the door to innumerable pathologies (Lamont, 2018, 2019c)

\[(16)\] Violations of AllFt-L (Language 9)  \[(17)\] Violations of AllFt-R (Language 9)
• If alignment only regulates some feet, the rest are subject to ties

• ALIGNWd-R: Assign one violation mark for a word that does not have a foot at its right edge.

(18) /LLLLLL/ → *(LL)(LL)(LL) ~ L(LL)L(LLL): Step 1


3.3 Problem 2: neurotic parsing (Language 16)

• When both alignment constraints dominate Parse-σ, only short words are parsed into feet

• This happens with both distance-sensitive and distance-insensitive alignment

(20) /LL/ → *(LL) (Language 16): Step 1


(21) /LLL/ → LLL (Language 16): Step 1


4 Directional Harmonic Serialism

• Traditionally, output constraints compare candidates in terms of how many violations they incur

• Under directional evaluations, constraints compare candidates in terms of where violations occur (Eisner, 2000; Finley, 2008, 2009), not their total number

• Left-to-right evaluation disprefer violations towards the left edge; right-to-left evaluation disprefer violations towards the right edge

• This works very well for modeling iterative processes in HS (Lamont, 2019a,b,c)

(22) /mawas/ → [māwās]: Step 1


(23) /mawas/ → [māwās]: Step 2


4.1 Directional evaluation does away with alignment

- Where to parse feet is built into the preferences of Parse-$\sigma$
- Constraints only regulate the locations of violations

**Parse-$\sigma$:** Assign one violation mark for every syllable that is not a member of some foot.

**Left-to-right**

\[
\sigma\sigma\sigma \prec \sigma\sigma('\sigma) \prec \sigma('\sigma)\sigma \prec \sigma\sigma(\sigma'\sigma) \prec \sigma('\sigma)\sigma \prec \sigma(\sigma)\sigma \prec \sigma(\sigma'\sigma) \prec \sigma(\sigma)\sigma
\]

**Right-to-left**

\[
\sigma\sigma\sigma \prec \sigma(\sigma)\sigma \prec \sigma(\sigma'\sigma) \prec \sigma(\sigma'\sigma)\sigma \prec \sigma(\sigma)\sigma \prec \sigma(\sigma)\sigma \prec \sigma(\sigma)\sigma
\]

**Trochee:** Assign one violation mark for a foot whose head is not aligned with its left edge.

**Left-to-right**

\[
(\sigma'\sigma)\sigma \prec \sigma(\sigma'\sigma) \prec \sigma(\sigma'\sigma) \prec (\sigma'\sigma)\sigma \prec (\sigma'\sigma)\sigma\sigma \prec \sigma(\sigma'\sigma)\sigma \prec \sigma(\sigma)\sigma
\]

**Right-to-left**

\[
\sigma\sigma(\sigma'\sigma) \prec \sigma(\sigma'\sigma) \prec (\sigma'\sigma)\sigma \prec (\sigma'\sigma)\sigma\sigma \prec \sigma(\sigma'\sigma)\sigma \prec \sigma(\sigma)\sigma
\]

**Iamb:** Assign one violation mark for a foot whose head is not aligned with its right edge.

**Left-to-right**

\[
(\sigma\sigma)\sigma \prec \sigma(\sigma\sigma) \prec \sigma(\sigma'\sigma) \prec \sigma(\sigma'\sigma)\sigma \prec \sigma(\sigma'\sigma)\sigma\sigma \prec \sigma(\sigma'\sigma)\sigma\sigma \prec \sigma(\sigma)\sigma
\]

**Right-to-left**

\[
\sigma\sigma(\sigma\sigma) \prec \sigma(\sigma\sigma) \prec \sigma(\sigma'\sigma) \prec \sigma(\sigma'\sigma)\sigma \prec \sigma(\sigma'\sigma)\sigma\sigma \prec \sigma(\sigma'\sigma)\sigma\sigma \prec \sigma(\sigma)\sigma
\]
FtBin: Feet are binary at some level of analysis (μ, σ).

Left-to-right

\[(\sigma)\sigma\sigma < \sigma(\sigma)\sigma < \sigma\sigma(\sigma) < (\sigma)\sigma\sigma \prec (\sigma\sigma)\sigma \prec \sigma(\sigma')\sigma \prec \sigma\sigma(\sigma')\sigma\]

Right-to-left

\[\sigma\sigma(\sigma) < \sigma(\sigma)\sigma < \sigma(\sigma)\sigma < (\sigma)\sigma\sigma \prec (\sigma\sigma)\sigma \prec (\sigma\sigma)\sigma \prec (\sigma\sigma)\sigma \prec (\sigma\sigma)\sigma \prec (\sigma\sigma)\sigma\]

4.2 Illustration

Exhaustive left-to-right trochees: /LLLLL/ \(\rightarrow\) (LL)(LL)(L)

- PARSE-σ evaluated left-to-right, motivating directional parsing
- No other constraint’s directionality matters
- Winner-~ loser status of IAMB depends on its directionality

(26) /LLLLL/ \(\rightarrow\) (LL)(LL)(L): Step 1

<table>
<thead>
<tr>
<th>/LLLLL/</th>
<th>PARSE-σ(_{L\rightarrow R})</th>
<th>TROCHEE</th>
<th>IAMB</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. LLLL</td>
<td>W σ σ σ σ σ</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\rightarrow) b. (LL)LLL</td>
<td></td>
<td>σ σ σ</td>
<td>(σσ)</td>
<td></td>
</tr>
<tr>
<td>c. L(LL)LL</td>
<td>W σ σ σ</td>
<td>W~L</td>
<td>(σσ)</td>
<td></td>
</tr>
<tr>
<td>d. LL(LL)L</td>
<td>W σ σ σ</td>
<td>W~L</td>
<td>(σσ)</td>
<td></td>
</tr>
<tr>
<td>e. LLL(LL)</td>
<td>W σ σ σ</td>
<td>W~L</td>
<td>(σσ)</td>
<td></td>
</tr>
<tr>
<td>f. (L)LLLL</td>
<td>σ σ σ W (σσ) L</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g. (L)LLL</td>
<td>W σ σ σ σ</td>
<td>W (σ)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(27) /LLLLL/ \(\rightarrow\) (LL)(LL)(L): Step 2

<table>
<thead>
<tr>
<th>(LL)LLL</th>
<th>PARSE-σ(_{L\rightarrow R})</th>
<th>TROCHEE</th>
<th>IAMB</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. (LL)LLL</td>
<td>W σ σ σ</td>
<td>L (σσ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\rightarrow) b. (LL)(LL)L</td>
<td></td>
<td>σ</td>
<td>(σσ) (σσ)</td>
<td></td>
</tr>
<tr>
<td>c. (LL)(LL)LL</td>
<td>W σ</td>
<td>W~L (σσ) (σσ)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(28) /LLLLL/ \(\rightarrow\) (LL)(LL)(L): Step 3

<table>
<thead>
<tr>
<th>(LL)(LL)L</th>
<th>PARSE-σ(_{L\rightarrow R})</th>
<th>TROCHEE</th>
<th>IAMB</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. (LL)(LL)</td>
<td>W σ</td>
<td>(σσ) (σσ) L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\rightarrow) b. (LL)(LL)(L)</td>
<td></td>
<td>(σσ) (σσ)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3 Factorial typology

- Calculated with both versions of each constraint present in CON
- Where not specified, the direction of the constraint does not matter
- 10 languages found; largely matches (8), but without non-iterative stress and Language 16
### 4.4 No neurotic parsing

- Language 17 is very similar to Language 16, except that it doesn’t parse short words into feet
- No constraint’s directionality matters here
- To avoid confusion with the notation, violating syllables are represented with $\sigma$

#### (31) /LL/ → LL (Language 17): Step 1

<table>
<thead>
<tr>
<th>/LL/</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>→ a. LL</td>
<td>Trochee</td>
<td>Iamb</td>
<td>FtBin</td>
<td>Parse-σ</td>
</tr>
</tbody>
</table>
|   |   |   |   | $\sigma$ | $\sigma$
| b. (LL) | W | $\sigma$ | $\sigma$ | L |
| c. (LL) | W | $\sigma$ | $\sigma$ | L |
| d. (LL) | W | $\sigma$ | L | $\sigma$
| e. L(L) | W | $\sigma$ | L | $\sigma$

#### (32) /LLL/ → LLL (Language 17): Step 1

<table>
<thead>
<tr>
<th>/LLL/</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>→ a. LL</td>
<td>Trochee</td>
<td>Iamb</td>
<td>FtBin</td>
<td>Parse-σ</td>
</tr>
</tbody>
</table>
|   |   |   |   | $\sigma$ | $\sigma$ | $\sigma$
| b. (LL) | W | $\sigma$ | $\sigma$ | L | $\sigma$
| c. L(LL) | W | $\sigma$ | L | $\sigma$
| d. (LL) | W | $\sigma$ | L | $\sigma$
| e. L(LL) | W | $\sigma$ | L | $\sigma$
| f. (LL) | W | $\sigma$ | L | $\sigma$
| g. (L)LL | W | $\sigma$ | L | $\sigma$ | $\sigma$
| h. L(L) | W | $\sigma$ | L | $\sigma$ | $\sigma$
| i. L(L) | W | $\sigma$ | L | $\sigma$ | $\sigma$
### 4.5 Non-iterative stress

*Foot*: Assign one violation mark for every foot.

**HaveStress**: Assign one violation mark for every word without a stress.  

<table>
<thead>
<tr>
<th>(33)</th>
<th>/LLLLLLL/</th>
<th>/LLLLLLL/</th>
<th>Trochees</th>
<th>Iamb</th>
<th>Both</th>
<th>Both</th>
<th>Exhaustive</th>
<th>Inexhaustive</th>
<th>Iterative</th>
<th>Non-iterative</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>(34)</th>
<th>Parse-σ→R Trochee</th>
<th>FtBin</th>
<th>Parse-σ→R Trochee</th>
<th>FtBin</th>
<th>Parse-σ→L Trochee</th>
<th>FtBin</th>
<th>Parse-σ→L Trochee</th>
<th>FtBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. After Lx#Pr (Prince & Smolensky, 1993/2004:45)
5 Conclusion

- Footing has been argued to be local, in the sense that feet are considered one-by-one, not all at once
- Serial approaches require a way to specify where to place feet
- Directional HS offers such a mechanism while avoiding the problems of alignment constraints

Acknowledgments

This work has greatly benefited from discussions with Gaja Jarosz, John McCarthy, Joe Pater, Kristine Yu, and participants in the UMass Sound Workshop. All remaining errors are of course my own.

References


