Morphotactics and phonology as subregular languages

Alëna Aksënova

Stony Brook University

MIT LingLunch
October 20, 2016
Truth can be stated in a thousand different ways, yet each one can be true.

Swami Vivekananda
Brief outline of ideas

Truth can be stated in a thousand different ways, yet each one can be true.

Swami Vivekananda

- Phonology: SPE, OT, ...
- Morphology: PFM, DM, ...
- Unified treatment: FSM as a model of both phonology and morphology
Brief outline of ideas

Truth can be stated in a thousand different ways, yet each one can be true.

*Swami Vivekananda*

- Phonology: SPE, OT, ...
- Morphology: PFM, DM, ...
- Unified treatment: FSM as a model of both phonology and morphology
- FSM is too powerful for both of them
- They both are *subregular*
Outline

1. Regular languages
2. (Sub)regular phonology
   - Regular phonology
   - SL patterns
   - TSL patterns
   - SP patterns
   - Unaccounted typological gaps
3. (Sub)regular morphotactics
   - Regular morphotactics
   - SL patterns
   - TSL patterns
   - Unaccounted typological gaps
4. Results
5. Future research
The Chomsky Hierarchy of String Languages

- **reg (finite)**
- **regular**
- **context-free**
- **mildly context-sensitive**
- **context-sensitive**
- **recursively enumerable**
Class of regular languages

Definition (Regular languages)

= language accepted by a Finite State Machine (FSM)
Class of regular languages

**Definition (Regular languages)**

= language accepted by a Finite State Machine (FSM)

**Definition (Finite State Machine)**

= an automaton with states, some of those are final or initial, and transitions between them
Class of regular languages

Definition (Regular languages)
= language accepted by a Finite State Machine (FSM)

Definition (Finite State Machine)
= an automaton with states, some of those are final or initial, and transitions between them

Example (FSM for language \((ab)^*\))
Class of regular languages

Definition (Finite State Machine)

= an automaton with states, some of those are final or initial, and transitions between them
Definition (Finite State Machine)

= an automaton with states, some of those are final or initial, and transitions between them

Example (FSM for language $a(ba)^+$)
Regular phonology

Kaplan and Kay (1994)

[...] only regular devices are required for the analysis and interpretation of such [phonological] systems.
Regular phonology

Kaplan and Kay (1994)

[...] only regular devices are required for the analysis and interpretation of such [phonological] systems.

- Regular power is *enough* for phonology
- Does phonology use the whole power of regular languages?
Regular phonology: example I

Example (Turkish vowel harmony)

- Vowels within a word are either [+front]: i, ü, e, ö, or [-front]: ı, u, a, o
- Adjacent non-high vowels harmonize in [round]: i, ü and ı, u
Vowels within a word are either [+front]: i, ü, e, ö, or [-front]: i, u, a, o

Adjacent non-high vowels harmonize in [round]: i, ü and i, u
Regular phonology: example II

Example (Anticipatory obstruent assimilation in Russian)

- Obstruents in a cluster must agree in voicing
- If the last obstruent is voiced, others are also voiced:
  ot dveri ‘from door’ → o[ddv]eri
- If the last obstruent is voiceless, others are also voiceless:
  iz korobki ‘out of box’ → i[sk]oro[pk]i
Regular phonology: example II

Example (Anticipatory obstruent assimilation in Russian)

- Obstruents in a cluster must agree in voicing
- If the last obstruent is voiced, others are also voiced:
  \textit{ot dveri} ‘\textit{from.door}’ $\rightarrow$ o[ddv]eri
- If the last obstruent is voiceless, others are also voiceless:
  \textit{iz korobki} ‘\textit{out.of.box}’ $\rightarrow$ i[sk]oro[pk]i
Regular phonology: predicted pattern I

- Can FSMs over-predict?
Regular phonology: predicted pattern I

- Can FSMs over-predict?

**Example (First-Last Harmony)**

- Harmony happens only between the first and the last vowels
Regular phonology: predicted pattern I

- Can FSMs over-predict?

**Example (First-Last Harmony)**

- Harmony happens only between the first and the last vowels
- Predicted but unattested
Regular phonology: predicted pattern I

- Can FSMs over-predict?

**Example (First-Last Harmony)**

- Harmony happens only between the first and the last vowels
- Predicted but unattested
Regular phonology: predicted pattern II

- Can FSMs over-predict?
Regular phonology: predicted pattern II

- Can FSMs over-predict?

**Example (Sour Grapes Harmony)**

- Harmony happens only if there is no blocker (#) in a word
Regular phonology: predicted pattern II

- Can FSMs over-predict?

Example (Sour Grapes Harmony)

- Harmony happens only if there is no blocker (#) in a word
- Predicted but unattested
Can FSMs over-predict?

Example (Sour Grapes Harmony)

- Harmony happens only if there is no blocker (#) in a word
- Predicted but unattested
Regular phonology: too much?

Is the whole power of regular languages needed for phonology? (See Heinz 2015 for more elaborate discussion.)
Regular phonology: too much?

Is the whole power of regular languages needed for phonology? (See Heinz 2015 for more elaborate discussion.)

- **Expressivity**: class of regular languages seems to be sufficiently expressive.
Regular phonology: too much?

Is the whole power of regular languages needed for phonology? (See Heinz 2015 for more elaborate discussion.)

- **Expressivity**: class of regular languages seems to be sufficiently expressive.
- **Restrictiveness**: it doesn’t seem to be sufficiently restrictive. Regular languages can analyze patterns we are incapable of producing.
Subregular phonology

Phonology is less then regular. What is the better fit for it?
Subregular phonology

Phonology is less than regular. What is the better fit for it?

This case is being investigated by

Jeff Heinz  Jane Chandlee  Adam Jardine  Thomas Graf
... and others
not full power of finite-state machinery is being exploited ⇒ subregular hierarchy
not full power of finite-state machinery is being exploited ⇒ **subregular hierarchy**

### Strong Subregular Hypothesis

All **phonological dependencies** are
- strictly local (SL)
- tier-based strictly local (TSL)
- strictly piecewise (SP)
SL patterns: assimilation and devoicing

SL languages

Strictly Local grammars are ones that capture local dependencies.
SL patterns: assimilation and devoicing

### SL languages

Strictly Local grammars are ones that capture local dependencies.

### Example (Assimilation and word-final devoicing in Russian)

- **Anticipatory obstruent assimilation:**
  - $C_{+voi}C_{-voi}$: *ot dveri ‘from.door’* → o[ddv]eri
  - $C_{-voi}C_{+voi}$: *iz korobki ‘out.of.box’* → i[sk]oro[pk]i
SL patterns: assimilation and devoicing

**SL languages**

Strictly Local grammars are ones that capture local dependencies.

**Example (Assimilation and word-final devoicing in Russian)**

- **Anticipatory obstruent assimilation:**
  - $*C_{+voi}C_{-voi}$: ot dveri ‘from door’ $\rightarrow$ o[ddv]eri
  - $*C_{-voi}C_{+voi}$: iz korobki ‘out of box’ $\rightarrow$ i[sk]oro[pk]i

- **Obstruent word final devoicing:**
  - $*C_{+voi} \times$: moroz ‘frost’ $\rightarrow$ moro[s]
    - moroz̆ ‘frost’ $\rightarrow$ moro[z]̆
SL patterns: assimilation and devoicing

SL languages

Strictly Local grammars are ones that capture local dependencies.

Example (Assimilation and word-final devoicing in Russian)

- **Anticipatory obstruent assimilation:**
  
  *$C_{+\text{voi}}C_{-\text{voi}}$: ot dveri ‘from door’ $\rightarrow$ o[ddv]eri*
  
  *$C_{-\text{voi}}C_{+\text{voi}}$: iz korobki ‘out of box’ $\rightarrow$ i[sk]oro[pk]i*

- **Obstruent word final devoicing:**
  
  *$C_{+\text{voi}}\times$: moroz ‘frost’ $\rightarrow$ moro[s]*
  
  moroz$\overline{\varepsilon}$ ‘frosts’ $\rightarrow$ moro[z]$\varepsilon$

- **$G = \{ *C_{+\text{voi}}C_{-\text{voi}}, *C_{-\text{voi}}C_{+\text{voi}}, *C_{+\text{voi}}\times \} $**
SL patterns: assimilation and devoicing

SL languages
Strictly Local grammars are ones that capture local dependencies.

Example (Assimilation and word-final devoicing in Russian)

- Anticipatory obstruent assimilation:
  *$C_{+\text{voi}}C_{-\text{voi}}$: ot dveri ‘from.door’ → o[ddv]eri
  *$C_{-\text{voi}}C_{+\text{voi}}$: iz korobki ‘out.of.box’ → i[sk]oro[pk]i

- Obstruent word final devoicing:
  *$C_{+\text{voi}}\bowtie$: moroz ‘frost’ → moro[s]
  moroz$\varnothing$ ‘frosts’ → moro[z]$\varnothing$

- $G = \{ *C_{+\text{voi}}C_{-\text{voi}}, *C_{-\text{voi}}C_{+\text{voi}}, *C_{+\text{voi}}\bowtie \}$

- mozg ‘brain’ → mo[sk]

- *$\bowtie$mozg$\bowtie$ *$\bowtie$mosg$\bowtie$ *$\bowtie$mozk$\bowtie$ ok $\bowtie$mosk$\bowtie$
SL patterns: intervocalic voicing

Example (Intervocalic voicing in German)

- **Intervocalic [s] voicing:**
  - Faser ‘fiber’ → fa[z]er
  - reisen ‘to.travel’ → rei[z]en
SL patterns: intervocalic voicing

Example (Intervocalic voicing in German)

- **Intervocalic [s] voicing:**
  - Faser ‘fiber’ → fa[z]er
  - reisen ‘to.travel’ → rei[z]en
- **Wasser ‘water’ → wa[ss]er**
  - reiste ‘traveled’ → rei[s]te
SL patterns: intervocalic voicing

Example (Intervocalic voicing in German)

- **Intervocalic [s] voicing:**
  - Faser ‘fiber’ → fa[z]er
  - reisen ‘to.travel’ → rei[z]en

- **Wasser ‘water’ → wa[ss]er**
  - reiste ‘traveled’ → rei[s]te

- \( G = \{*VC_{-\text{voi}}V\} \)
SL patterns: intervocalic voicing

Example (Intervocalic voicing in German)

- **Intervocalic [s] voicing:**
  - Faser ‘fiber’ → fa[z]er
  - reisen ‘to.travel’ → rei[z]en
- **Wasser ‘water’ → wa[ss]er**
  - reiste ‘traveled’ → rei[s]te

- \( G = \{ *\text{VC}_\text{voi} \_ \text{V} \} \)
- \( *\text{rei}[s]\text{en} \) \( \text{ok} \) \( \text{rei[z]}\text{en} \)
SL patterns: intervocalic voicing

Example (Intervocalic voicing in German)

- **Intervocalic [s] voicing:**
  - Faser ‘fiber’ → fa[z]er
  - reisen ‘to.travel’ → rei[z]en
- **Wasser ‘water’** → wa[ss]er
  - reiste ‘traveled’ → rei[s]te

- \( G = \{ *VC_{-\text{voi}}V \} \)

- \(*re[i][s]en \quad \text{\textit{ok}} \quad rei[z]en\)

- This does not affect *reiste*: ‘ist’ and ‘ste’ are not blocked.
SL is not enough for phonology

Example (Vowel harmony in Lakaa, Niger-Congo)

- Non-high vowels harmonize in ATR (Akinlabi 2009)
  
  [+ATR]: èsìsòn ‘smoke’, lèjìmè ‘matriclan’
  
  [-ATR]: èsìsòn ‘housefly’, òtú:má ‘need’
SL is not enough for phonology

Example (Vowel harmony in Lakaa, Niger-Congo)

- Non-high vowels harmonize in ATR (Akinlabi 2009)
  \[ +\text{ATR} \]: \text{è}sìsòn ‘smoke’, lèjìmè ‘matriclan’
  \[ -\text{ATR} \]: \text{è}sìsòn ‘housefly’, òtúmá ‘need’

- This pattern is not SL: there can be unbounded amount of \[ +\text{hi} \] vowels between the two \[ -\text{hi} \] ones, at some moment we will lose track of the last \[ -\text{hi} \] vowel.
SL is not enough for phonology

Example (Vowel harmony in Laka, **Niger-Congo**)

- **Non-high vowels harmonize in ATR** (Akinlabi 2009)
  - [+ATR]: èsìsòn ‘smoke’, lèjìmè ‘matriclan’
  - [-ATR]: ësìsòn ‘housefly’, ëtú:má ‘need’

- **This pattern is not SL**: there can be unbounded amount of [+hi] vowels between the two [-hi] ones, at some moment we will loose track of the last [-hi] vowel.

- SL languages can capture only *local* dependencies.
TSL phonology: vowel harmony

TSL languages

Tier-based Strictly Local grammars capture non-local dependencies by analyzing them as local over a certain tier.
TSL phonology: vowel harmony

**TSL languages**

Tier-based Strictly Local grammars capture non-local dependencies by analyzing them as local *over a certain tier*.

**Example (Vowel harmony in Lakaa, Niger-Congo)**

- Project tier of non-high vowels
- Block illicit combinations on this tier

\[ G_{T(non\text{-}high)} = \{ \star \alpha ATR \beta ATR \} \]
TSL phonology: vowel harmony

TSL languages
Tier-based Strictly Local grammars capture non-local dependencies by analyzing them as local *over a certain tier*.

Example (Vowel harmony in Lakaa, **Niger-Congo**)
- Project tier of non-high vowels
- Block illicit combinations on this tier

\[ G_{T(non-high)} = \{ \star \alpha ATR \ \beta ATR \} \]

\[ \overset{ok}{\text{èsi}s\circ n} : \]

- \[ \text{è s i s ò n} \]

- tier of [-hi] vowels
TSL phonology: vowel harmony [cont.]

Example (Vowel harmony in Lakaa, Niger-Congo)

- $G_{T(non-high)} = \{*\alpha ATN \beta ATN\}$

ok èsìsìòì:

<table>
<thead>
<tr>
<th>×</th>
<th>è</th>
<th>ç</th>
<th>×</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>tier of [-hi] vowels</td>
</tr>
<tr>
<td>×</td>
<td>è</td>
<td>s</td>
<td>i</td>
</tr>
</tbody>
</table>
Example (Vowel harmony in Lakaa, Niger-Congo)

- \( G_{T(\text{non-high})} = \{ *\alpha \text{ATR} \beta \text{ATR} \} \)

\( \overset{ok}{\text{èssìson}:} \)
- \( \times \quad \varepsilon \quad \circ \quad \times \)
- \( \times \quad \varepsilon \quad s \quad \overset{\text{tier of [-hi] vowels}}{\circ} \quad s \quad \overset{\text{on}}{\circ} \quad n \quad \times \)

\( \overset{*}{\text{èssìson}:} \)
- \( \times \quad \varepsilon \quad \circ \quad \times \)
- \( \times \quad \varepsilon \quad s \quad \overset{\text{tier of [-hi] vowels}}{\circ} \quad s \quad \overset{\text{èssìson}}{\circ} \quad n \quad \times \)
TSL phonology: stress

Example (Stress assignment)

- There must be exactly one stress in a word.
- Not SL: not always there is a limit on the amount of the material before or after the stressed syllable.
Example (Stress assignment)

- There must be exactly one stress in a word.
- Not SL: not always there is a limit on the amount of the material before of after the stressed syllable.
- \( G_{T}(\text{stress-vow}) = \{*\times \times, \,*\text{áá}\} \)
TSL phonology: stress

Example (Stress assignment)

- There must be exactly one stress in a word.
- Not SL: not always there is a limit on the amount of the material before of after the stressed syllable.
- \[ G_T(stress-vow) = \{*, \times \times, *áá\} \]

\[ \text{aabába:} \]
\[ \times \quad \text{á} \quad \times \]
\[ \text{tier of stressed vowels} \]
\[ \times \quad a \quad a \quad b \quad á \quad b \quad a \quad \times \]
Example (Stress assignment)

- There must be exactly one stress in a word.
- Not SL: not always there is a limit on the amount of the material before of after the stressed syllable.
- $G_T(stress-vow) = \{*\times\times, *\dagger\dagger\}$

**Ok**: aabába:

\[
\begin{array}{cccc}
\times & \dagger & \dagger & \times \\
\hline
\times & a & a & b & \dagger & b & a & \times
\end{array}
\]

***aababa**:  

\[
\begin{array}{cccc}
\times & \dagger & \dagger & \times \\
\hline
\times & a & a & b & a & b & a & \times
\end{array}
\]
TSL phonology: dissimilation in Latin

Example (-ālis/-āris alternation in Latin)

- **Liquid dissimilation in Latin** (Müller 2013)

- -ālis is chosen when the previous liquid is ‘r’:
  - spīritus ‘breath’ → spīritālis ‘spiritual’
  - lībra ‘pound’ → lībrālis ‘weighing one pound’

- -āris is chosen when the previous liquid is ‘l’:
  - mīles ‘soldier’ → mīlitāris ‘military’
  - cōnsul ‘consul’ → cōnsulāris ‘consular’
TSL phonology: dissimilation in Latin

Example (-ālis/-āris alternation in Latin)

- **Liquid dissimilation in Latin** (Müller 2013)
  - -ālis is chosen when the previous liquid is ‘r’:
    - spīritus ‘breath’ → spīritālis ‘spiritual’
    - lībra ‘pound’ → lībrālis ‘weighing one pound’
  - -āris is chosen when the previous liquid is ‘l’:
    - mīles ‘soldier’ → mīlitāris ‘military’
    - cōnsul ‘consul’ → cōnsulāris ‘consular’
  - Not SL: unbounded amount of material between the two liquids.
TSL phonology: dissimilation in Latin

Example (-ālis/-āris alternation in Latin)

- **Liquid dissimilation in Latin** (Müller 2013)
  - -ālis is chosen when the previous liquid is ‘r’:
    - spīritus ‘breath’ → spīritālis ‘spiritual’
    - lībra ‘pound’ → lībrālis ‘weighing one pound’
  - -āris is chosen when the previous liquid is ‘l’:
    - mīles ‘soldier’ → mīlitāris ‘military’
    - cōnsul ‘consul’ → cōnsulāris ‘consular’

- Not SL: unbounded amount of material between the two liquids.
  \[ G_T(\text{liquids}) = \{ *\ll, *rr \} \]
Example (-ālis/-āris alternation in Latin)

\[ G_T(\text{liquids}) = \{*ll, *rr\} \]
Example (-ālis/-āris alternation in Latin)

- \( G_T(\text{liquids}) = \{*\text{ll}, *\text{rr}\} \)

\( o^k \text{lībrālis:} \)

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c}
\times & l & r & l & \times \\
\hline
\text{tier of liquids} & & & & & & & & & \\
\times & l & \text{i} & b & r & \text{ā} & l & i & s & \times \\
\end{array}
\]
Example (-ālis/-āris alternation in Latin)

- $G_{T(liquids)} = \{*ll, *rr\}$

$^ok$ lībrālis:

<table>
<thead>
<tr>
<th></th>
<th>l</th>
<th>r</th>
<th>l</th>
<th></th>
</tr>
</thead>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* lībrāris:

<table>
<thead>
<tr>
<th></th>
<th>l</th>
<th>r</th>
<th>r</th>
<th></th>
</tr>
</thead>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

 tier of liquids
Example (-ālis/-āris alternation in Latin)

\[ G_T(liquids) = \{ *ll, *rr \} \]
Example (-ālis/-āris alternation in Latin)

\[ G_{T(liquids)} = \{ *ll, *rr \} \]

\[ \text{\textasciitilde m\textbar l\textbar i\textbar t\textbar ā\textbar r\textbar i\textbar s} \text{\textbar tier of liquids} \]

TSL phonology: dissimilation in Latin [cont.]
TSL phonology: dissimilation in Latin [cont.]

Example (-ālis/-āris alternation in Latin)

\[ G_{T(\text{liquids})} = \{ *\text{ll}, *\text{rr} \} \]

\[ \text{\textit{ok} mīlitāris:} \]
\[
\begin{array}{cccccc}
\times & l & r & \times \\
\times & m & i & t & ā & r & i & s & \times \\
\end{array}
\]

\[ \text{tier of liquids} \]

\[ \text{\textit{*mīlitālis:} \}
\[
\begin{array}{cccccc}
\times & l & l & \times \\
\times & m & i & t & ā & l & i & s & \times \\
\end{array}
\]

\[ \text{tier of liquids} \]
Non-TSL phonology: tone plateauing

Example (Tone plateauing in Luganda (Bantu))

- Underlyingly, TBUs are either H or $\emptyset$.
- L is inserted after H on a final syllable.
- All $\emptyset$ tones between two H tones become H.
Non-TSL phonology: tone plateauing

**Example (Tone plateauing in Luganda (BANTU))**

- Underlyingly, TBUs are either H or ∅.
- L is inserted after H on a final syllable.
- All ∅ tones between two H tones become H.

**Single words:**

- `/mu-tund-a/ → mutunda ‘seller’`
- `/mu-tém-a/ → mutèma ‘chopper’`
- `/bi-kó-po/ → bikópo ‘cups’`
- `/ki-si-kí/ → kisikî ‘logs’`
Non-TSL phonology: tone plateauing

Example (Tone plateauing in Luganda (Bantu))

- Underlyingly, TBUs are either H or ∅.
- L is inserted after H on a final syllable.
- All ∅ tones between two H tones become H.

**Single words:**

- /mu-tund-a/ → mutunda ‘seller’
- /mu-tém-a/ → mutèma ‘chopper’
- /bi-kó-po/ → bikópo ‘cups’
- /ki-si-kí/ → kisikî ‘logs’

**Compounds:**

- /mutunda + bikópo/ → mutunda-bikópo ‘cup seller’
- /mutéma + ki-si-kí/ → mutémá-kísikî ‘log-chopper’
Non-TSL phonology: tone plateauing

Example (Tone plateauing in Luganda (BANTU))

- Underlyingly, TBUs are either H or $\emptyset$.
- L is inserted after H on a final syllable.
- All $\emptyset$ tones between two H tones become H.

Single words:
- $/\text{mu-tund-a}/ \rightarrow \text{mutunda} \ 'seller'$
- $/\text{mu-tém-a}/ \rightarrow \text{mutèma} \ 'chopper'$
- $/\text{bi-kó-po}/ \rightarrow \text{bikópo} \ 'cups'$
- $/\text{ki-si-kí}/ \rightarrow \text{kisikî} \ 'logs'$

Compounds:
- $/\text{mutunda} + \text{bikópo}/ \rightarrow \text{mutunda-bikópo} \ 'cup seller'$
- $/\text{mutéma} + \text{ki-si-kí}/ \rightarrow \text{mutémá-kísíkí} \ 'log-chopper'$

Hyman & Katamba (2010) on Luganda data and discussion
Jardine (2015) on elaborate computational analysis
McPherson (2012) on underspecified tones in Bantu languages
Non-TSL phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (BANTU))

- Projecting only H tones on a tier:
Non-TSL phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (BANTU))

- Projecting only H tones on a tier:
  - $\times$ $\emptyset$ won’t be seen

- $\times$ C H C $\emptyset$ ... $\emptyset$ C H $\times$
Non-TSL phonology: tone plateauing [cont.]

**Example (Tone plateauing in Luganda (BANTU))**

- Projecting only H tones on a tier:
  - ![Diagram](image)
  - $\times \emptyset$ won’t be seen

- Projecting H and $\emptyset$ on a tier:
Example (Tone plateauing in Luganda (Bantu))

- Projecting only H tones on a tier:
  - \( \times \) \( \emptyset \) won’t be seen
  - \( \times \) C H C \( \emptyset \) \( \ldots \) \( \emptyset \) C H \( \times \)

- Projecting H and \( \emptyset \) on a tier:
  - \( \times \) unbounded amount of \( \emptyset \) between two H locality cannot be achieved
  - \( \times \) C H C \( \emptyset \) \( \ldots \) \( \emptyset \) C H \( \times \)
Non-TSL phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (Bantu))

- No H∅+H sequences
- Projecting only H tones on a tier:  
  × ∅ won’t be seen
- Projecting H and ∅ on a tier:  
  × unbounded amount of ∅ between two H  
  locality cannot be achieved
Non-TSL phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (Bantu))

- No H∅+H sequences

- Projecting only H tones on a tier:
  - ∅ won’t be seen

- Projecting H and ∅ on a tier:
  - Unbounded amount of ∅ between two H
    - Locality cannot be achieved

- This pattern is not TSL
Non-TSL phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (Bantu))

- No H∅+H sequences
- Projecting only H tones on a tier:
  - ∅ won’t be seen
- Projecting H and ∅ on a tier:
  - ∅ won’t be seen
  - unbounded amount of ∅ between two H
  - locality cannot be achieved
- This pattern is not TSL

- (T)SL languages work with *locality*, not with ‘*memory*’.
SP phonology: tone plateauing

SP languages

Strictly Piecewise grammars capture non-local dependencies by blocking (allowing) certain subsequences.
SP phonology: tone plateauing

SP languages

Strictly Piecewise grammars capture non-local dependencies by blocking (allowing) certain subsequences.

Example (Tone plateauing in Luganda (BANTU))

- No H∅+H sequences
SP phonology: tone plateauing

SP languages

Strictly Piecewise grammars capture non-local dependencies by blocking (allowing) certain subsequences.

Example (Tone plateauing in Luganda (Bantu))

- No $H\emptyset^+H$ sequences
- A prosodic word is out if there are $\emptyset$ tone(s) between two $H$ tones
- $G_{SP} = \{*V_H V_\emptyset V_H, *V_H V_\emptyset V_{HL}\}$
SP phonology: tone plateauing

SP languages

Strictly Piecewise grammars capture non-local dependencies by blocking (allowing) certain subsequences.

Example (Tone plateauing in Luganda (BANTU))

- No $H\emptyset^+H$ sequences
- A prosodic word is out if there are $\emptyset$ tone(s) between two $H$ tones
- $G_{SP} = \{ *V_HV_\emptyset V_H, *V_HV_\emptyset V_{HL} \}$
- ok mutunda-bikópo
  Subsequences (only vowels are listed):
  uua, uui, uuó, uuo, uai, uaó, uao, uió, uio, uóo, aió, aio, aóo, ióo
SP phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (Bantu))

- A prosodic word is out if there are $\emptyset$ tone(s) between two H tones
- $G_{SP} = \{ *V_H V_\emptyset V_H, *V_H V_\emptyset V_{HL} \}$
SP phonology: tone plateauing [cont.]

Example (Tone plateauing in Luganda (Bantu))

- A prosodic word is out if there are \( \emptyset \) tone(s) between two H tones
- \( G_{SP} = \{ *V_H V_\emptyset V_H, *V_H V_\emptyset V_{HL} \} \)
- \( ok \) mutémá-kísíkî
  Subsequences (only vowels are listed):
  ueá, uéí, uēî, éáî, éáî, áii, áîî, íî
Example (Tone plateauing in Luganda (Bantu))

- A prosodic word is out if there are \( \emptyset \) tone(s) between two H tones
- \( G_{SP} = \{ *V_H V_\emptyset V_H, *V_H V_\emptyset V_{HL} \} \)
- \( \text{ok} \) mutémá-kísíkî
  Subsequences (only vowels are listed):
  uéá, uéí, uêî, éáí, éáî, áïí, áïî, íîí
- *mutéma-kisikî
  Subsequences (only vowels are listed):
  uéa, uéî, uêî, éai, éaî, ...
Example (Sibilant harmony in Navajo)

- All sibilants in one words must agree in interiority

Martin (2005), Heinz&Rogers (2010)
SP phonology: sibilant harmony

Example (Sibilant harmony in Navajo)

- All sibilants in one words must agree in interiority
  - Martin (2005), Heinz & Rogers (2010)
  - tʃa ‘ear’
  - nééz ‘long’
  - tsanééz ‘mule’
  - *tʃanééz
SP phonology: sibilant harmony

Example (Sibilant harmony in Navajo)

- All sibilants in one words must agree in interiority
  Martin (2005), Heinz&Rogers (2010)
  \[
  \text{tʃa} \quad \text{‘ear’} \quad \text{tsanéez} \quad \text{‘mule’}
  \]
  \[
  \text{nééz} \quad \text{‘long’} \quad \text{*tʃanééz}
  \]
- This language is both TSL and SP
- \[ G_{SP} = \{ *S^\alpha_{ant}S^\beta_{ant} \} \]
SP phonology: sibilant harmony

Example (Sibilant harmony in Navajo)

- All sibilants in one words must agree in interiority
  Martin (2005), Heinz&Rogers (2010)

<table>
<thead>
<tr>
<th>tʃa</th>
<th>‘ear’</th>
<th>tsanéez</th>
<th>‘mule’</th>
</tr>
</thead>
<tbody>
<tr>
<td>néez</td>
<td>‘long’</td>
<td>*tʃanéez</td>
<td></td>
</tr>
</tbody>
</table>

- This language is both TSL and SP
- \( G_{SP} = \{ *S_\alpha ant S_\beta ant \} \)
- \( ok \) tsanéez
  No blocked subsequences.
SP phonology: sibilant harmony

Example (Sibilant harmony in Navajo)

- All sibilants in one words must agree in interiority
  - Martin (2005), Heinz&Rogers (2010)
  - ꞏ�a ‘ear’
  - ꞏ�eˊez ‘mule’
  - ꞏ�eˊez ‘long’
  - ꞏ�anˊeˊez

- This language is both TSL and SP
- $G_{SP} = \{ *S_{\alpha ant}S_{\beta ant} \}$
  - ok tsanˊeˊez
    - No blocked subsequences.
  - *tʃanˊeˊez
    - Subsequence tʃz is found.
Unexpected patterns

Example (First-Last Harmony)

- Harmony happens only between the first and the last vowels
Unexpected patterns

**Example (First-Last Harmony)**

- Harmony happens only between the first and the last vowels.
- **Not (T)SL**: the first and the last vowels can be arbitrary far from each other, locality cannot be achieved.
- **Not SP**: in terms of subsequences, no difference between first, last and intermediate vowels.
Unexpected patterns

Example (First-Last Harmony)

- Harmony happens only between the first and the last vowels
- Not (T)SL: the first and the lost vowels can be arbitrary far from each other, locality cannot be achieved
- Not SP: in terms of subsequences, no difference between first, last and intermediate vowels

Example (Sour Grapes Harmony)

- Harmony happens only if there is no blocker in a word
Unexpected patterns

Example (First-Last Harmony)

- Harmony happens only between the first and the last vowels
- **Not (T)SL**: the first and the lost vowels can be arbitrary far from each other, locality cannot be achieved
- **Not SP**: in terms of subsequences, no difference between first, last and intermediate vowels

Example (Sour Grapes Harmony)

- Harmony happens only if there is no blocker in a word
- **Not (T)SL**: the blocker can be arbitrary far from the edge of a word
- **Not SP**: blocked cannot be forced to appear in every possible substring, it will be simply not noticed
Interim summary: Phonology

Language classes phonology falls into:

- **Strictly Local**: enforce local dependencies word-final devoicing, voicing assimilation within a cluster, intervocalic voicing, etc.

- **Tier-based Strictly Local**: capture non-local dependencies by projecting elements of a certain type on a tier harmonies, long-distance dissimilation or assimilation processes, etc.

- **Strictly Piecewise**: check wellformedness of substructures of a string unbounded harmony processes, tone plateauing
Regular Morphology


Morphological patterns can be depicted with Finite State Automata, therefore morphology is a set of regular relations, i.e. it is regular.
# Subregular morphology

*Received view*

- **Phonology**
  - regular
  - Kaplan & Kay (1994)

- **Morphology**
  - regular
  - Beesley & Karttunen (2003)

*Recent research*

- **Phonology**
  - subregular
  - Heinz (2015)

- **Morphology**
  - ?
Subregular morphology

**Received view**

- **Phonology**
  - regular
  - Kaplan & Kay (1994)

- **Morphology**
  - regular
  - Beesley & Karttunen (2003)

**Recent research**

- subregular
  - Heinz (2015)

Maybe, morphology is also subregular?

Thomas Graf  Sophie Moradi

... and me
not full power of finite-state machinery is being exploited ⇒ **subregular hierarchy**

**Strong Subregular Hypothesis**

All **phonological dependencies** are
- strictly local (SL)
- tier-based strictly local (TSL)
- strictly piecewise (SP)
Subregular Phonology and Morphology

not full power of finite-state machinery is being exploited ⇒ **subregular hierarchy**

**Strong Subregular Hypothesis**

All **phonological dependencies** are
- strictly local (SL)
- tier-based strictly local (TSL)
- strictly piecewise (SP)

**Subregular Morphotactics**

All **morphotactic** dependencies are
- strictly local (SL)
- tier-based strictly local (TSL)
Example (Affixation in English)

- *un-* is a prefix: unholy, undo
- *-able* is a suffix: drinkable, moveable
Example (Affixation in English)

- *un-* is a prefix: unholy, undo
- *-able* is a suffix: drinkable, moveable
- $G = \{\text{*able-stem, *stem-un}\}$ blocks improper ordering and predicts that there will be words where these two affixes co-occur.
Example (Affixation in English)

- **un-** is a prefix: unholy, undo
- **-able** is a suffix: drinkable, moveable

\[ G = \{ \text{*able-stem, *stem-un} \} \] blocks improper ordering and predicts that there will be words where these two affixes co-occur.

- Indeed, it is correct: undoable, unlockable
Non-SL morphology: circumfixation

- English un-...-able are prefix and suffix that can co-occur
- However, two parts of a *circumfix* cannot occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- Abstract nominalizer, circumfix 'ke-...-an':
  - tinggi 'high'
  - mahasiswa 'student (big pupil)'
  - ketinggian 'altitude'
  - kemahasiswaan 'student affairs'

\[ G = \{ \text{stem} \}, \text{ke} \]

Locality cannot be achieved: there can be unbounded amount of stems between ke- and -an.

This pattern is not SL.
Non-SL morphology: circumfixation

- English *un-*...-*able* are prefix and suffix that can co-occur
- However, two parts of a *circumfix* cannot occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- “Abstract” nominalizer, circumfix ‘*ke-*...-*an*’:
  - tinggi ‘high’
  - ketinggian ‘altitude’
  - mahasiswa ‘student (big pupil)’
  - kemahasiswaan ‘student affairs’
  - *ketinggi*
  - *ketinggian*
  - *mahasiswa*
  - *mahasiswaan*
Non-SL morphology: circumfixation

- English un-...-able are prefix and suffix that can co-occur
- However, two parts of a circumfix cannot occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- “Abstract” nominalizer, circumfix ‘ke-...-an’:
  - tinggi ‘high’ mahasiswa ‘student (big pupil)’
  - ketinggian ‘altitude’ kemahasiswaan ‘student affairs’
  - *ketinggi
  - *mahasiswaan

- \( G = \{ \text{*an-stem, *stem-ke} \} \) doesn’t enforce co-occurrence of the two parts of the circumfix.
Non-SL morphology: circumfixation

- English un-...-able are prefix and suffix that can co-occur.
- However, two parts of a circumfix cannot occur independently.

Example (Indonesian circumfixation, Sneddon (1996))

- “Abstract” nominalizer, circumfix ‘ke-...-an’:
  - tinggi ‘high’ mahasiswa ‘student (big pupil)”
  - ketinggian ‘altitude’ kemahasiswaan ‘student affairs’
  - *ketinggi *mahasiswaan

- $G = \{*\text{an-stem}, *\text{stem-ke}\}$ doesn’t enforce co-occurrence of the two parts of the circumfix.
- Locality cannot be achieved: there can be unbounded amount of stems between ke- and -an.
Non-SL morphology: circumfixation

- English un-...-able are prefix and suffix that can co-occur
- However, two parts of a circumfix cannot occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- “Abstract” nominalizer, circumfix ‘ke-...-an’:
  - tinggi ‘high’ mahasiswa ‘student (big pupil)’
  - ketinggian ‘altitude’ kemahasiswaan ‘student affairs’
- G = {*an-stem, *stem-ke} doesn’t enforce co-occurrence of the two parts of the circumfix.
- Locality cannot be achieved: there can be unbounded amount of stems between ke- and -an.
- This pattern is not SL.
Example (Indonesian circumfixation)

- This pattern is TSL
- Elements of the circumfix is projected on a tier
- \( G = \{ *\text{an-ke}, *\text{ke} \times, * \times \text{an}, *\text{an-an}, *\text{ke-ke} \} \)
TSL morphology: circumfixation

Example (Indonesian circumfixation)

- This pattern is TSL
- Elements of the circumfix is projected on a tier
- \( G = \{ *\text{an-ke}, *\text{ke\-tinggi-an}, *\text{ke-ke}\} \)

\( ok \) ke-tinggi-an:

\[
\begin{array}{c|c|c|c}
\text{ke} & \text{an} & \times \\
\hline
\text{ke} & \text{tinggi} & \text{an} & \times \\
\end{array}
\]
Example (Indonesian circumfixation)

\[ G = \{ *an-ke, *ke \times, *\times an, *an-an, *ke-ke \} \]
TSL morphology: circumfixation [cont.]

Example (Indonesian circumfixation)

- $G = \{ *an-ke, \ bke, \ *\times an, \ *an-an, \ *ke-ke \}$

```
ok maha-siswa:
  \times \hspace{0.5cm} \times
  \hspace{1.5cm} \text{tier of circumfix}
  \times \hspace{0.5cm} \times
  maha siswa \times
```
TSL morphology: circumfixation [cont.]

Example (Indonesian circumfixation)

\( G = \{ ^*\text{an-ke}, ^*\text{ke} \times, ^*\times \text{an}, ^*\text{an-an}, ^*\text{ke-ke} \} \)

\( ^ok \) maha-siswa:

- \( \times \) maha siswa \( \times \)

\( ^ok \) ke-maha-siswa-an:

- \( \times \) ke \( \times \) an 

- \( \times \) ke maha siswa an \( \times \)
Example (Indonesian circumfixation)

- \( G = \{*\text{an-ke}, *\text{ke} \times, *\times\text{an}, *\text{an-an}, *\text{ke-ke}\} \)
Example (Indonesian circumfixation)

- $G = \{ \text{*an-ke, ke} \times \times, \text{*an}, \text{*an-an, ke-ke}\}$

*ke-maha-siswa:

\[
\begin{array}{c}
\times & \text{ke} & \times \\
\times & \text{ke maha siswa} & \times \\
\end{array}
\]
Example (Indonesian circumfixation)

$G = \{ \ast an-ke, \ast ke\times, \ast \times an, \ast an-an, \ast ke-ke \}$

*ke-maha-siswa:

$\times \begin{array}{l}
\times ke \\
ke maha siswa \\
\end{array}$

*an-ke, *ke, *\times an, *an-an, *ke-ke

*maha-siswa-an:

$\times \begin{array}{l}
\times an \\
maha siswa an \\
\end{array}$
Example (Floating affix in Swahili)

- In Swahili, -vyo is a floating affix (Stump 2016)
TSL morphology: floating affixes

Example (Floating affix in Swahili)

- In Swahili, -vyo is a floating affix (Stump 2016)
  1. a-vi-soma-vyo
     SBJ-OBJ-read-REL
     ‘reads’
TSL morphology: floating affixes

Example (Floating affix in Swahili)

- In Swahili, -vyo is a floating affix (Stump 2016)
  1. a-vi-soma-vyo
     SBJ-OBJ-read-REL
     ‘reads’
  2. a-si-vyo-vi-soma
     SBJ-NEG-REL-OBJ-read
     ‘doesn’t read’
TSL morphology: floating affixes

Example (Floating affix in Swahili)

- In Swahili, -vyo is a floating affix (Stump 2016)
  1. a-vi-soma-vyo
     SBJ-OBJ-read-REL
     ‘reads’
  2. a-si-vyo-vi-soma
     SBJ-NEG-REL-OBJ-read
     ‘doesn’t read’
  3. *a-vyo-vi-soma
  4. *a-vyo-vi-soma-vyo
  5. *a-si-vyo-vi-soma-vyo
  6. *a-si-vi-soma-vyo
### Example (Floating affix in Swahili)

- **In Swahili, vyo is a floating marker (Stump 2016)**
  1. a-vi-soma-vyo
     - SBJ-OBJ-**read-REL**
     - ‘reads’
  2. a-si-vyo-vi-soma
     - SBJ-**NEG-REL-OBJ-read**
     - ‘doesn’t read’
Example (Floating affix in Swahili)

In Swahili, **vyo** is a floating marker (Stump 2016)

1. a-vi-soma-vyo  
   SBJ-OBJ-**read**-REL  
   ‘reads’
2. a-si-vyo-vi-soma  
   SBJ-**NEG**-REL-OBJ-**read**  
   ‘doesn’t read’

- **vyo** can be used just once
- **vyo** is *prefix* if the negation **si** is present
- **vyo** is *suffix* in other cases
TSL morphology: floating affixes [cont.]

<table>
<thead>
<tr>
<th>Example (Floating affix in Swahili)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Swahili, <strong>vyo</strong> is a floating marker <em>(Stump 2016)</em></td>
</tr>
<tr>
<td>1. a-<strong>vi</strong>-soma-vyo</td>
</tr>
<tr>
<td>SBJ-OBJ-<strong>read</strong>-REL</td>
</tr>
<tr>
<td>‘<strong>reads</strong>’</td>
</tr>
<tr>
<td>2. a-<strong>si</strong>-vyo-<strong>vi</strong>-soma</td>
</tr>
<tr>
<td>SBJ-NEG-<strong>REL</strong>-OBJ-<strong>read</strong></td>
</tr>
<tr>
<td>‘<strong>doesn’t read</strong>’</td>
</tr>
<tr>
<td><strong>vyo</strong> can be used just once</td>
</tr>
<tr>
<td><strong>vyo</strong> is <em>prefix</em> if the negation <strong>si</strong> is present</td>
</tr>
<tr>
<td><strong>vyo</strong> is <em>suffix</em> in other cases</td>
</tr>
<tr>
<td>This language is not SL: stem is unbounded in length.</td>
</tr>
<tr>
<td>It is TSL with <strong>vyo</strong>, <strong>si</strong> and stem boundaries (#, #) on the tier.</td>
</tr>
</tbody>
</table>
TSL morphology: floating affixes [cont.]

Example (Floating affix in Swahili)

\[ G = \{ \} \]
Example (Floating affix in Swahili)

\[ G = \{ \]

\[ \bullet \text{vyo can be used just once:} \]

\[ *\text{vyo-vyo, vyo-#-#-vyo} \]

\[ \} \]
Example (Floating affix in Swahili)

\[ G = \{ \]

- \textit{vyo} can be used \textit{just once}:
  \[ *\text{vyo-vyo}, *\text{vyo-#-#-vyo} \]

- \textit{vyo} is prefix if the negation is present:
  \[ *\text{vyo-si}, *\text{si-#-#-vyo}, *\text{vyo-#-#-si} \]

\[ \} \]
Example (Floating affix in Swahili)

\[ G = \{ \]

- vyo can be used just once:
  \[ *vyo-vyo, *vyo-\#-\#-vyo \]

- vyo is prefix if the negation is present:
  \[ *vyo-si, *si-\#-\#-vyo, *vyo-\#-\#-si \]

- vyo is suffix in other cases:
  \[ *\times-vyo-\#-\# \]

\[ \}

TSL morphology: floating affixes [cont.]
Example (Floating affix in Swahili)

\[ G = \{ *vyo-vyo, *vyo-\#-\#-vyo, *vyo-si, *si-\#-\#-vyo, *vyo-\#-\#-si, *\times-vyo-\#-\# \} \]
Example (Floating affix in Swahili)

\[ G = \{ *\text{vyo-vyo}, *\text{vyo-}##\text{-}##\text{-vyo}, *\text{vyo-si}, *\text{si-}##\text{-}##\text{-vyo}, \\ *\text{vyo-}##\text{-}##\text{-si}, *\times\text{-vyo-}##\text{-}## \} \]

\( \overset{ok}{\text{a-vi-soma-vyo}}: \)

\[
\begin{array}{cccccc}
\times & \# & \# & \text{vyo} & \times \\
& & & & & \text{tier} \\
\times & \text{a} & \text{vi} & \# & \text{soma} & \# & \text{vyo} & \times
\end{array}
\]
Example (Floating affix in Swahili)

\[ G = \{ *vyo-vyo, *vyo-\#-\#-vyo, *vyo-si, *si-\#-\#-vyo, *vyo-\#-\#-si, *\times-vyo-\#-\# \} \]

\[ \text{ok} \quad a-vi-soma-vyo: \]

\[
\begin{array}{c c c c}
\times & \# & \# & vyo \times \\
\hline
\times & a & vi & soma & \# & vyo \times \\
\end{array}
\]

\[ \text{*a-vyo-vi-soma:} \]

\[
\begin{array}{c c c c c}
\times & vyo & \# & \# & \times \\
\hline
\times & a & vyo & vi & soma & \# & \times \\
\end{array}
\]
Example (Floating affix in Swahili)

\[ G = \{ *\text{vyo-vyo}, *\text{vyo-\#-\#-vyo}, *\text{vyo-si}, *\text{si-\#-\#-vyo}, \\
*\text{vyo-\#-\#-si}, *\text{\#-\#-vyo}, *\text{\#-\#-si}, *\text{\#-\#-vyo} \} \]
Example (Floating affix in Swahili)

\[
G = \{ *\text{vyo-vyo}, *\text{vyo-\#-\#-vyo}, *\text{vyo-si}, *\text{si-\#-\#-vyo}, *\text{vyo-\#-\#-si}, *\text{\#-\#-vyo} \}
\]

\( ok \) a-si-vyo-vi-soma:

\( \times \quad \text{si} \quad \text{vyo} \quad \# \quad \# \quad \times \)

\[ \text{tier} \]

\( \times \quad \text{a} \quad \text{si} \quad \text{vyo} \quad \text{vi} \quad \# \quad \text{soma} \quad \# \quad \times \)
Example (Floating affix in Swahili)

\[ G = \{ *vyo-vyo, *vyo-\#-\#-vyo, *vyo-si, *si-\#-\#-vyo, *vyo-\#-\#-si, *\times-vyo-\#-\# \} \]

\( o_k \) a-si-vyo-vi-soma:

\[
\begin{array}{ccccccc}
\times & si & vyo & \# & \# & \times \\
\end{array}
\]

\[
\begin{array}{ccccccc}
\times & a & si & vyo & vi & \# & soma & \# & \times \\
\end{array}
\]

\( *a-si-vyo-vi-soma-vyo: \)

\[
\begin{array}{ccccccc}
\times & si & vyo & \# & \# & vyo & \times \\
\end{array}
\]

\[
\begin{array}{ccccccc}
\times & a & si & vyo & vi & \# & soma & \# & vyo & \times \\
\end{array}
\]
Typological gaps

Basic Logic of Argument

- Suppose all morphotactic patterns are at most TSL.
- If the combination of two attested TSL patterns is not TSL, it cannot be generated.
- We predict a typological gap.
Typological gaps

Basic Logic of Argument

- Suppose all morphotactic patterns are at most TSL.
- If the combination of two attested TSL patterns is not TSL, it cannot be generated.
- We predict a typological gap.

Some predicted gaps:

- No embedded circumfixation;
- No cases when amount of prefixes depends on the amount of suffixes;
- In general, no $a^n b^n$ pattern and its derivatives.
Typological gap I: Impossible compounding

Example (Compounding patterns)

- **Compounding in Russian**: *(stem-o)*-stem
  - vodovoz ‘water carrier’
  - vodovozovoz ‘carrier of water carriers’
Typological gap I: Impossible compounding

Example (Compounding patterns)

- **Compounding in Russian: (stem-o)*-stem**
  - vodovoz ‘water carrier’
  - vodovozovoz ‘carrier of water carriers’

- **Compounding in Turkish: stem-(stem⁺-sI)**
  - bahçe kapı-sI ‘garden gate’
  - türk kahve-sI ‘Turkish coffee’
  - türk bahçe kapı-sI(-*sI) ‘Turkish garden gate’
Typological gap I: Impossible compounding

Example (Compounding patterns)

- **Compounding in Russian:** \((\text{stem-o})*\text{-stem}\)
  - vodovoz ‘water carrier’
  - vodovozovoz ‘carrier of water carriers’

- **Compounding in Turkish:** \(\text{stem-}(\text{stem}^+\text{-si})\)
  - bahçe kapı-si ‘garden gate’
  - türk kahve-si ‘Turkish coffee’
  - türk bahçe kapı-si(-si) ‘Turkish garden gate’

- **Russian + Turkish pattern:** \(\text{stem-}(\text{stem}^n\text{-marker}^n)\)
  - amount of compounding affixes = amount of added stems
Typological gap I: Impossible compounding

Example (Compounding patterns)

- **Compounding in Russian:** \((\text{stem-o})^*\)-stem
  
  vodovoz ‘water carrier’
  
  vodovozovoz ‘carrier of water carriers’

- **Compounding in Turkish:** \(\text{stem-} (\text{stem}^+ - \text{si})\)
  
  bahçe kapı-si ‘garden gate’
  
  türk kahve-si ‘Turkish coffee’
  
  türk bahçe kapı-si(-*si) ‘Turkish garden gate’

- **Russian + Turkish pattern:** \(\text{stem-} (\text{stem}^n - \text{marker}^n)\)
  
  amount of compounding affixes = amount of added stems

- This pattern is not TSL/regular and appears to be non-existent
Typological gap II: Recurrent affixation

<table>
<thead>
<tr>
<th>Example (multiple affix application)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In German, prefix über- ‘after’ can be iterated: über*-stem</td>
</tr>
<tr>
<td>morgen ‘tomorrow’</td>
</tr>
<tr>
<td>über-morgen ‘the day after tomorrow’</td>
</tr>
<tr>
<td>über-über-morgen ‘the day after the day after tomorrow’</td>
</tr>
</tbody>
</table>
Typological gap II: Recurrent affixation

Example (multiple affix application)

- In German, prefix über- ‘after’ can be iterated: über*-stem
  morgen ‘tomorrow’
  über-morgen ‘the day after tomorrow’
  über-über-morgen ‘the day after the day after tomorrow’

- In Illocano, circumfix ka-...-an ‘next’ cannot: ka-stem-an
  bigát ‘morning’
  ka-bigát-an ‘the next morning’
  *ka-ka-bigát-an-an
Typological gap II: Recurrent affixation

Example (multiple affix application)

- In German, prefix über- ‘after’ can be iterated: über*-stem
  morgen ‘tomorrow’
  über-morgen ‘the day after tomorrow’
  über-über-morgen ‘the day after the day after tomorrow’

- In Illocano, circumfix ka-...-an ‘next’ cannot: ka-stem-an
  bigát ‘morning’
  ka-bigát-an ‘the next morning’
  *ka-ka-bigát-an-an

- The resulting pattern of iteration of the circumfix would be crc^n-stem-mfx^n
Typological gap II: Recurrent affixation

Example (multiple affix application)

- In German, prefix über- ‘after’ can be iterated: über*-stem
  morgen ‘tomorrow’
  über-morgen ‘the day after tomorrow’
  über-über-morgen ‘the day after the day after tomorrow’

- In Illocano, circumfix ka-...-an ‘next’ cannot: ka-stem-an
  bigát ‘morning’
  ka-bigát-an ‘the next morning’
  *ka-ka-bigát-an-an

- The resulting pattern of iteration of the circumfix would be
  \[crc^n-\text{stem-mfx}^n\]

- This pattern is not TSL/regular, therefore it does not exist
Language classes morphotactics falls into:

- **Strictly Local**: enforce local dependencies sequential order of morphemes (mostly)

- **Tier-based Strictly Local**: capture non-local dependencies by projecting elements of a certain type on a tier circumfixation, variable ordering affixes, etc.

For further details see Aksënova et al (2016).
SL, TSL and SP are learnable in the limit from positive text. Very little data is needed: the acquisition shouldn’t be a problem.

- **Strictly Local**
  learning = memorizing finite number of $k$-grams
  (see Heinz (2007))
SL, TSL and SP are learnable in the limit from positive text. Very little data is needed: the acquisition shouldn’t be a problem.

- **Strictly Local**
  learning = memorizing finite number of $k$-grams
  (see Heinz (2007))

- **Tier-Based Strictly Local**
  learning = memorizing finite number of $k$-grams + tier induction (see Jardine&Heinz (2016))
Learning

SL, TSL and SP are learnable in the limit from positive text. Very little data is needed: the acquisition shouldn’t be a problem.

- **Strictly Local**
  learning = memorizing finite number of $k$-grams
  (see Heinz (2007))

- **Tier-Based Strictly Local**
  learning = memorizing finite number of $k$-grams + tier induction
  (see Jardine&Heinz (2016))

- **Strictly Piecewise**
  learning = inferring possible precedence relations
  (see Heinz (2010))
**Conclusion**

- **Phonology** is TSL and SP
- **Morphotactics** is (T)SL
- Set of typological gaps can be explained due to the subregular nature of morphology
- Same formal tools can be used for morphology and phonology
- This approach can give new perspectives on acquisition
Future work

- Look at more typologically diverse languages
- Morphology: how to represent?
  - order of morpheme application (derivation history)
  - surface form complexity
- Cooperation of how many tiers can be needed?
- Extend to mappings from underlying to surface forms: interaction of phonological and morphological systems
Future work

The lesson is this: study the language and do some old-fashioned pure linguistics modeling before jumping into coding. Your programs will never be better than the linguistic model behind them.

Beesley & Kartunnen 2003, p.283.

Thank you!
References

Neutral vowels in Lokaa harmony.

Aksenova, Alena, Thomas Graf and Sedigheh Moradi (2016)
Morphotactics as Tier-Based Strictly Local Dependencies.
In *Proceedings of SIGMorPhon 2016*.

Finite state morphology.
Stanford, CA: CSLI Publications.

Hippocrene Books Inc., U.S.

Heinz, Jeffrey (2007)
The Inductive Learning of Phonotactic Patterns.
PhD thesis, UCLA.

Heinz, Jeffrey (2010)
Learning long-distance phonotactics.

Heinz, Jeffrey and James Rogers (2010)
Estimating Strictly Piecewise Distributions
Heinz, Jeffrey (2015)
The computational nature of phonological generalizations.
Manuscript, available online.

Hyman, Larry and Francis X. Katamba (2010)
Tone, syntax and prosodic domains in Luganda.

Computationally, tone is different.
Manuscript (long version), available online.

Jardine, Adam and Jeffrey Heinz (2016)
Learning Tier-based Strictly 2-Local Languages.
Transactions of the Association for Computational Linguistics 4, 87 – 98.

Kaplan, Ronald M. and Martin Kay (1994)
Regular Models of Phonological Rule Systems.

Martin, Andrew Thomas (2005)
The Effects of Distance on Lexical Bias: Sibilant Harmony in Navajo Compounds
Master's Thesis. UCLA.

McPherson, Laura (2012)
Underspecified tone in Tommo So (Dogon, Mali).
Müller, Daniela (2013)
Liquid dissimilation with a special regard to Latin.

Sneddon, James Neil (1996)
Indonesian Comprehensive Grammar.

Stump, Greg (2016)
Rule composition in an adequate theory of morphotactics.
Manuscript.
More than (T)SL or SP?

- Regular
- Star-Free
- LTT
- LT
- SL
- TSL

Closure properties inheritance:
- Concatenation
- Complement
- Union
- Intersection
More than (T)SL or SP?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English.
More than (T)SL or SP?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗
More than (T)SL or SP?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗

- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian.
More than (T)SL or SP?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗

- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian. ✗
More than (T)SL or SP?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ❌
- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian. ❌
- **Closure under relative complement**: Hsilgne contains all words that are ill-formed in English.
More than (T)SL or SP?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗

- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian. ✗

- **Closure under relative complement**: Hsilgne contains all words that are ill-formed in English. ✗
More than (T)SL or SP?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian that they share in common.
More than (T)SL or SP?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian that they share in common. ✓

  (spoken in Northern Norway, 18th-19th centuries)
More than (T)SL or SP?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian that they share in common. ✓
  (spoken in Northern Norway, 18th-19th centuries)

**Example (Closure under intersection)**

- A language allows complex nuclei and blocks codas (Supyire)
- A language forbids complex nuclei and allows codas (Russian)
More than (T)SL or SP?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian that they share in common. ✓
  (spoken in Northern Norway, 18th-19th centuries)

**Example (Closure under intersection)**

- A language allows complex nuclei and blocks codas (Supyire)
- A language forbids complex nuclei and allows codas (Russian)
- Then there will be a language that blocks complex nuclei and codas (Hawaiian, Senufo)
More than (T)SL or SP?

- Closure under concatenation
- Closure under union
- Closure under relative complement
- Closure under intersection
More than (T)SL or SP?

- Closure under concatenation
- Closure under union
- Closure under relative complement
- Closure under intersection

Diagram:

- Regular → Star-Free → LTT → LT → SL → TSL
- PT → SP
- concatenation
- complement union
- intersection

Closure properties inheritance