Fragments of First-Order Logic for Linguistic Structures

Thomas Graf

Stony Brook University
mail@thomasgraf.net
http://thomasgraf.net

UConn Logic Colloquium
Apr 19, 2017
The Talk in a Nutshell

Narrow Goal

Find the smallest fragment of first-order logic that is sufficiently expressive for natural language structures.

Why it Matters

- discover new parallels between phonology, morphology, syntax, even semantics
- explain typological gaps
- new empirical predictions
- simplify the learning problem
- benefit NLP applications
The Talk in a Nutshell

Narrow Goal

Find the smallest fragment of first-order logic that is sufficiently expressive for natural language structures.

Why it Matters

- discover new parallels between phonology, morphology, syntax, even semantics
- explain typological gaps
- new empirical predictions
- simplify the learning problem
- benefit NLP applications
Bigger Picture for...

- **Semanticists/Philosophers of Language**
  - Logic is not limited to natural language meaning.
  - It is just as useful for studying natural language structures.

- **Linguists in General**
  - Mathematical abstraction is a good thing.
  - It captures insights that are usually lost among the details.

- **Logicians**
  - Very weak logics are very relevant.
  - There are tons of problems to be solved.
Outline

1 Logics for Phonology
   - Logic and Linguistic Structures
   - Application to Phonology
   - TSL: Relativized Precedence

2 Beyond TSL Phonology
   - TSL Morphology
   - TSL Morpho-Semantics
   - TSL Syntax

3 Open Problems
   - Better Formal Understanding of TSL
   - Mappings Between Structures
Linguistic Structures

Linguists distinguish three major levels of structure:

1. **Phonology** = sound structure
   - word-final devoicing: rad → rat
   - primary stress: axiom

2. **Morphology** = word structure
   - inflection: she run+s
   - derivation: in+decipher+able

3. **Syntax** = sentence structure
   - John does not like Mary.
   - *Not John does like Mary.
Linguistic Structures

Linguists distinguish three major levels of structure:

1. **Phonology** = sound structure
   - word-final devoicing: \( \text{rad} \rightarrow \text{rat} \)
   - primary stress: \( 'aks.i.əm, \) not \( 'aks.i.'əm \)

2. **Morphology** = word structure
   - inflection: she run+s
   - derivation: in+decipher+able

3. **Syntax** = sentence structure
   - John does not like Mary.
   - *Not John does like Mary.
The Big Linguistic Questions

- What are the laws that govern each structural level?
- How complex are these laws? How hard are they to compute?
- Do we find typological gaps, i.e. patterns that should exist but don’t appear in any language?
- What can we infer about human cognition?

The Computational Program

- Computer scientists have figured out a lot about complexity, so let’s apply their ideas to language.
- Formal language theory and logic greatly deepen our understanding of language.
The Big Linguistic Questions

- What are the laws that govern each structural level?
- How complex are these laws? How hard are they to compute?
- Do we find typological gaps, i.e. patterns that should exist but don’t appear in any language?
- What can we infer about human cognition?

The Computational Program

- Computer scientists have figured out a lot about complexity, so let’s apply their ideas to language.
- Formal language theory and logic greatly deepen our understanding of language.
A Familiar Picture: The Chomsky Hierarchy

- The perceivable output of language is strings (sequences of sound waves, words, sentences).
- The complexity of string languages is measured by the (extended) Chomsky hierarchy. (Chomsky 1956, 1959)
A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of subregular languages
(McNaughton and Papert 1971; Rogers et al. 2010)

Regular

∪

Locally

Threshold Testable

∪

Star Free

Locally

Testable

∪

Piecewise

Strictly

Testable

∪

Strictly

Piecewise

Strictly

Local
# A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of **subregular languages**  
(McNaughton and Papert 1971; Rogers et al. 2010)

<table>
<thead>
<tr>
<th>Regular</th>
<th>Monadic Second-Order Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>Union</td>
</tr>
<tr>
<td>Locally Threshold Testable</td>
<td>Star Free</td>
</tr>
<tr>
<td>Union</td>
<td>Union</td>
</tr>
<tr>
<td>Locally Testable</td>
<td>Piecewise Testable</td>
</tr>
<tr>
<td>Union</td>
<td>Union</td>
</tr>
<tr>
<td>Strictly Local</td>
<td>Strictly Piecewise</td>
</tr>
</tbody>
</table>
A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of subregular languages
(McNaughton and Papert 1971; Rogers et al. 2010)
A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of subregular languages
(McNaughton and Papert 1971; Rogers et al. 2010)
A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of subregular languages
(McNaughton and Papert 1971; Rogers et al. 2010)
A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of **subregular languages**
(McNaughton and Papert 1971; Rogers et al. 2010)

<table>
<thead>
<tr>
<th></th>
<th>Regular</th>
<th>Monadic Second-Order Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locally Testable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Piecewise Testable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strictly Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conjunction of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative Literals</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ S/\triangleleft \]
A Different Picture: The Subregular Hierarchy

Often forgotten: hierarchy of **subregular languages**
(McNaughton and Papert 1971; Rogers et al. 2010)

<table>
<thead>
<tr>
<th>Locally Threshold Testable</th>
<th>Regular</th>
<th>Monadic Second-Order Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>∪</td>
<td>First-Order Logic</td>
</tr>
<tr>
<td>Locally Testable</td>
<td></td>
<td>Propositional Logic</td>
</tr>
<tr>
<td>U</td>
<td></td>
<td>Conjunction of Negative Literals</td>
</tr>
<tr>
<td>Strictly Testable</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>S/◁</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Star Free</th>
<th>Piecewise Testable</th>
<th>Strictly Piecewise</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤</td>
<td>≤</td>
<td>≤</td>
</tr>
</tbody>
</table>

S/◁ ⊂ ⊂ ⊂ ⊂ ⊂ ⊂ ⊂
Example: Word-Final Devoicing is SL

- Captured by forbidding voiced segments at the end of a word
- **German**: Don’t have $z$ or $v$ (where $\$ = word edge).

### Corresponding Logical Formula with $\langle$ $\rangle$

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg z$</td>
<td>$\neg (z \land \langle $)</td>
<td>$\neg(\exists x, y [z(x) \land $ (y) \land x \triangleleft y])$</td>
</tr>
<tr>
<td></td>
<td>$\land$</td>
<td>$\land$</td>
</tr>
<tr>
<td>$\neg v$</td>
<td>$\neg (v \land \langle $)</td>
<td>$\neg(\exists x, y [v(x) \land $ (y) \land x \triangleleft y])$</td>
</tr>
</tbody>
</table>

### Example

- alalas: $a \land al \land la \land as \land s$
- *alalaz: $a \land al \land la \land as \land z$
Example: Word-Final Devoicing is SL

- Captured by forbidding voiced segments at the end of a word
- **German**: Don’t have $z$ or $v$ (where $\$ = word edge).

### Corresponding Logical Formula with $\triangleleft$

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg z$</td>
<td>$\neg (z \wedge \triangleleft $)$</td>
<td>$\neg (\exists x, y[z(x) \wedge $ (y) \wedge x \triangleleft y])$</td>
</tr>
<tr>
<td>$\wedge$</td>
<td>$\wedge$</td>
<td>$\wedge$</td>
</tr>
<tr>
<td>$\neg v$</td>
<td>$\neg (v \wedge \triangleleft $)$</td>
<td>$\neg (\exists x, y[v(x) \wedge $ (y) \wedge x \triangleleft y])$</td>
</tr>
</tbody>
</table>

### Example

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>alalas</td>
<td>$a \wedge al \wedge la \wedge as \wedge s$</td>
</tr>
<tr>
<td>*alalaz</td>
<td>$a \wedge al \wedge la \wedge as \wedge z$</td>
</tr>
</tbody>
</table>
Example: Word-Final Devoicing is SL

- Captured by forbidding voiced segments at the end of a word
- **German**: Don’t have $z$ or $v$ (where $\triangle$ = word edge).

### Corresponding Logical Formula with $\triangle$

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg z\triangle$</td>
<td>$\neg (z \land \triangle)$</td>
<td>$\neg (\exists x, y [z(x) \land \triangle(y) \land x \triangle y])$</td>
</tr>
<tr>
<td>$\land$</td>
<td>$\land$</td>
<td>$\land$</td>
</tr>
<tr>
<td>$\neg v\triangle$</td>
<td>$\neg (v \land \triangle)$</td>
<td>$\neg (\exists x, y [v(x) \land \triangle(y) \land x \triangle y])$</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>alalas</th>
<th>$a \land al \land la \land as \land s\triangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>*alalaz</td>
<td>$a \land al \land la \land as \land z\triangle$</td>
</tr>
</tbody>
</table>
Example: Intervocalic Voicing is SL

- Captured by forbidding voiceless segments between vowels
- **Suppose:**
  - \([-\text{voice}] = \{s, \ddash\}\]
  - \(V = \{a, i, u\}\)
- **Then:** don’t have \(asa, a\ddash a, asi, a\ddash i, \ldots\)

### Corresponding Logical Formula with \(\triangleright\)

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\neg asa)</td>
<td>(\neg(a \land \triangleright s \land \triangleright \triangleright a))</td>
<td>((\exists x, y, z[a(x) \land s(y) \land a(z) \land x \triangleright y \land y \triangleright z]))</td>
</tr>
<tr>
<td>(\land)</td>
<td>(\land)</td>
<td>(\land)</td>
</tr>
<tr>
<td>(\neg a\ddash a)</td>
<td>(\neg(a \land \triangleright \ddash \land \triangleright \ddash \land \ddash a))</td>
<td>((\exists x, y, z[a(x) \land \ddash(y) \land a(z) \land x \triangleright y \land y \triangleright z]))</td>
</tr>
<tr>
<td>(\land)</td>
<td>(\land)</td>
<td>(\land)</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
</tbody>
</table>
Example: Sibilant Voicing Harmony is SP

- If multiple sibilants ($s, z, \check{3}, \check{3}$) occur in the same word, they must all be voiceless ($s, \check{3}$) or voiced ($z, \check{3}$).
- In other words: Don’t mix purple and teal.
- But: Sibilants can be arbitrarily far away from each other!

### Corresponding Logical Formula with $\triangleleft^+$

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg z s$</td>
<td>$\neg (z \land \triangleleft^+ s)$</td>
<td>$\neg (\exists x, y [z(x) \land s(y) \land x \triangleleft^+ y])$</td>
</tr>
<tr>
<td>$\land$</td>
<td>$\land$</td>
<td>$\land$</td>
</tr>
<tr>
<td>$\neg z \check{3}$</td>
<td>$\neg (z \land \triangleleft^+ \check{3})$</td>
<td>$\neg (\exists x, y [z(x) \land \check{3}(y) \land x \triangleleft^+ y])$</td>
</tr>
<tr>
<td>$\land$</td>
<td>$\land$</td>
<td>$\land$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Example

- salalas: $s \land a \land l \land $ $s$ $\land$ sa $\land$ sl $\land$ ss $\land$ s$ $\land$ al $\land$ aa $\land$ ...  
- *zalalas: $z \land a \land l \land $ $s$ $\land$ $s$ $\land$ za $\land$ zl $\land$ zs $\land$ z$ $\land$ al $\land$ aa $\land$ ...
Example: Sibilant Voicing Harmony is SP

- If multiple sibilants \((s, z, f, 3)\) occur in the same word, they must all be voiceless \((s, f)\) or voiced \((z, 3)\).
- In other words: Don’t mix purple and teal.
- But: Sibilants can be arbitrarily far away from each other!

Corresponding Logical Formula with \(\triangleleft^+\)

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\neg zs)</td>
<td>(\neg(z \land \triangleleft^+ s))</td>
<td>(\neg(\exists x, y [z(x) \land s(y) \land x \triangleleft^+ y]))</td>
</tr>
<tr>
<td>(\land)</td>
<td>(\land)</td>
<td>(\land)</td>
</tr>
<tr>
<td>(\neg z)</td>
<td>(\neg(z \land \triangleleft^+ f))</td>
<td>(\neg(\exists x, y [z(x) \land f(y) \land x \triangleleft^+ y]))</td>
</tr>
<tr>
<td>(\land)</td>
<td>(\land)</td>
<td>(\land)</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
</tbody>
</table>

Example

- salalas: \(s \land a \land l \land s\) \(\land \ldots\)
- *zalalas: \(z \land a \land l \land s\) \(\land \ldots\)
Example: Sibilant Voicing Harmony is SP

- If multiple sibilants (s, z, f, 3) occur in the same word, they must all be voiceless (s, f) or voiced (z, 3).
- In other words: Don’t mix purple and teal.
- But: Sibilants can be arbitrarily far away from each other!

Corresponding Logical Formula with $\triangleright^+$

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬zs</td>
<td>¬(z ∧ $\triangleright^+ s$)</td>
<td>¬(∃x, y[z(x) ∧ s(y) ∧ x $\triangleright^+ y])</td>
</tr>
<tr>
<td>∧</td>
<td>∧</td>
<td>∧</td>
</tr>
<tr>
<td>¬zf</td>
<td>¬(z ∧ $\triangleright^+ f$)</td>
<td>¬(∃x, y[z(x) ∧ f(y) ∧ x $\triangleright^+ y])</td>
</tr>
<tr>
<td>∧</td>
<td>∧</td>
<td>∧</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Example

Salalas: $s ∧ a ∧ l ∧ $$ ∧ sa ∧ sl ∧ ss ∧ s$ ∧ al ∧ aa ∧ ...

*Zalalas: $z ∧ a ∧ l ∧ $s ∧ $$ ∧ za ∧ zl ∧ z$ ∧ z$ ∧ al ∧ aa ∧ ...
Example: Primary Stress is LTT

▶ Every word has exactly one primary stress.
▶ SL, SP, and LT are too weak:

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>SP</th>
<th>LT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>at most one stress</strong></td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td><strong>at least one stress</strong></td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

▶ SL fails because this is a non-local dependency.
▶ SP fails because it can only forbid presence, not absence.
▶ LT only distinguishes “exactly 0” and “strictly more than 0”.

▶ We need LTT, i.e. FO with successor.

Corresponding Logical Formula (no relation required)

\[ \exists ! x [ \bigvee \alpha(x) \mid \alpha \text{ a segment with primary stress} ] \]
Problem 1: LTT is Too Powerful

First-order logic can combine restrictions too freely
⇒ massive overgeneration

If there is a vowel $V$ at the end, and there is a sibilant $S$ such that either $S$ is word-initial or there are exactly seven consonants, then intervocalic voicing is enforced iff

$$\forall V, \forall S \left[ \bigvee_{\alpha \text{ a vowel}} \alpha(V) \land \exists y [y(y) \land V \prec y] \land (s(S) \lor z(S) \lor f(S) \lor 3(S)) \land \neg \left( \exists x [x(x) \land x \prec S] \leftrightarrow \forall i \leq 7 \left[ \bigwedge_{1 \leq i \leq 7} \left( \bigvee_{\alpha \text{ a consonant}} \alpha(x_i) \right) \right] \right) \right]$$

the voicing value of $S$ is

$$\left( \left( \neg V s V \land \neg V f V \right) \leftrightarrow (\neg \text{voice}(S) \land [+\text{round}](V)) \lor ([+\text{voice}](S) \land [\neg\text{round}](V)) \right)$$
Problem 1: LTT is Too Powerful

First-order logic can combine restrictions too freely
\[ \Rightarrow \text{massive overgeneration} \]

If there is a vowel \( V \) at the end, and there is a sibilant \( S \) such that either \( S \) is word-initial or there are exactly seven consonants, then intervocalic voicing is enforced iff

\[
\forall V, \forall S \left( \bigvee_{\alpha \text{ a vowel}} \alpha(V) \land \exists y[\$(y) \land V \prec y] \land (s(S) \lor z(S) \lor f(S) \lor 3(S)) \land \neg(\exists x[\$(x) \land x \prec S] \leftrightarrow \exists! 7 x_i \left( \bigwedge_{1 \leq i \leq 7} (\bigvee_{\alpha \text{ a consonant}} \alpha(x_i))) \right) \rightarrow (\neg VsV \land \neg VfV) \leftrightarrow (\lnot \text{voice}(S) \land \text{+round}(V)) \lor (\text{+voice}(S) \land \lnot \text{round}(V))) \right)
\]
Problem 2: Scattered Distribution

Phonological dependencies seem to be **weirdly distributed**.

<table>
<thead>
<tr>
<th>Locally Threshold Testable</th>
<th>Regular</th>
<th>Monadic Second-Order Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\cup$</td>
<td>$\cup$</td>
</tr>
<tr>
<td></td>
<td>$\cup$</td>
<td>$\cup$</td>
</tr>
<tr>
<td>Locally Testable</td>
<td>Star Free</td>
<td>First-Order Logic</td>
</tr>
<tr>
<td></td>
<td>$\cup$</td>
<td></td>
</tr>
<tr>
<td>Strictly Local</td>
<td>Piecewise Testable</td>
<td>Propositional Logic</td>
</tr>
<tr>
<td></td>
<td>$\cup$</td>
<td></td>
</tr>
<tr>
<td>$S/\triangleleft$</td>
<td>Strictly Piecewise</td>
<td>Conjunction of Negative Literals</td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\subset$</td>
<td></td>
</tr>
</tbody>
</table>
Problem 2: Scattered Distribution

Phonological dependencies seem to be **weirdly distributed**.

<table>
<thead>
<tr>
<th>Locally Threshold Testable</th>
<th>Regular</th>
<th>Monadic Second-Order Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>U</td>
<td>First-Order Logic</td>
</tr>
<tr>
<td>Locally Testable</td>
<td>U</td>
<td>Propositional Logic</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>Conjunction of Negative Literals</td>
</tr>
<tr>
<td>Strictly Local</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>S/⟨</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strictly Piecewise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; /⟨⁺</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A New Challenger: Tier-Based Strictly Local

First defined in Heinz et al. (2011)
TSL is a minimal expansion of SL.
TSL replaces $\triangleleft$ by a relativized version of $\triangleleft^+$, denoted $\triangleleft_T$.
Inspired by phonological tiers.

Defining Tier-Precedence $\triangleleft_T$

Given alphabet $\Sigma$, a tier is some $T \subseteq \Sigma$.

$$x \triangleleft_T y \iff T(x) \land T(y) \land x \triangleleft^+ y \land \neg \exists z [T(z) \land x \triangleleft^+ z \land z \triangleleft^+ y]$$

$$T(x) \iff \bigvee_{t \in T} t(x)$$
Example: Sibilant Voicing Harmony Revisited

- Reminder: Don’t mix purple and teal.
- T := \{s, z, f, ʒ\}
- Don’t allow zs, zf, sz, sʒ, ... 

Corresponding Logical Formula with \(\triangleleft_T\)

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>\neg zs</td>
<td>\neg (z \land \triangleleft_T s)</td>
<td>\neg (\exists x, y [z(x) \land s(y) \land x \triangleleft_T y])</td>
</tr>
<tr>
<td>\land</td>
<td>\land</td>
<td>\land</td>
</tr>
<tr>
<td>\neg zf</td>
<td>\neg (z \land \triangleleft_T f)</td>
<td>\neg (\exists x, y [z(x) \land f(y) \land x \triangleleft_T y])</td>
</tr>
<tr>
<td>\land</td>
<td>\land</td>
<td>\land</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example: Sibilant Voicing Harmony Revisited

- Reminder: Don’t mix purple and teal.
- $T := \{s, z, \acute{s}, \acute{z}\}$
- Don’t allow $zs, z\acute{s}, sz, s\acute{z}, \ldots$

<table>
<thead>
<tr>
<th>$s$</th>
<th>$s$</th>
<th>$s$</th>
<th>$z$</th>
<th>$s$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$s$</td>
<td>$e$</td>
<td>$l$</td>
<td>$a$</td>
<td>$s$</td>
</tr>
<tr>
<td>$a$</td>
<td>$z$</td>
<td>$e$</td>
<td>$l$</td>
<td>$a$</td>
<td>$s$</td>
</tr>
</tbody>
</table>

Corresponding Logical Formula with $\triangleleft_T$

<table>
<thead>
<tr>
<th>CNL</th>
<th>Modal</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg zs$</td>
<td>$\neg (z \land \triangleleft_T s)$</td>
<td>$\neg (\exists x, y [z(x) \land s(y) \land x \triangleleft_T y])$</td>
</tr>
<tr>
<td>$\triangleleft$</td>
<td>$\triangleleft$</td>
<td>$\land$</td>
</tr>
<tr>
<td>$\neg z\acute{s}$</td>
<td>$\neg (z \land \triangleleft_T \acute{s})$</td>
<td>$\neg (\exists x, y [z(x) \land \acute{s}(y) \land x \triangleleft_T y])$</td>
</tr>
<tr>
<td>$\triangleleft$</td>
<td>$\triangleleft$</td>
<td>$\land$</td>
</tr>
<tr>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>
Example: Primary Stress Revisited

- Every word has exactly one primary stress.
- We don’t need LTT, **TSL is sufficient:**

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>SP</th>
<th>LT</th>
<th>LTT</th>
<th>TSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>at most one stress</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>at least one stress</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

- \( T \) contains all stressed segments \( \breve{o} \).
- **At most one stress:** don’t have \( \breve{o}\breve{o} \)
- **At least one stress:** don’t have $$

\[
\begin{array}{cccccc}
$ & $ & $ \acute{a} & \acute{a} & $ & $ \\
\mid & \mid & \mid & \mid & \mid & \\
$ a & l & a & $ & $ \acute{a} & l & \acute{a} & $ & $ \acute{a} & l & a & $ \\
\end{array}
\]
A Single Locus in the Subregular Hierarchy

Phonological dependencies now neatly fit into a **contiguous subregion** of the subregular hierarchy.

\[
\text{Locally Threshold Testable} \cup \text{Locally Testable} \cup \text{Strictly Local} \subset S/\triangleleft \subset TSL \subset 1,
\]

**Regular**

\[
\cup
\]

**Star Free**

\[
\cup
\]

**Piecewise Testable**

\[
\cup
\]

**Strictly Piecewise**

\[
< \triangleleft^+\
\]

**Monadic Second-Order Logic**

**First-Order Logic**

**Propositional Logic**

**Conjunction of Negative Literals**
Outline

1 Logics for Phonology
   - Logic and Linguistic Structures
   - Application to Phonology
   - TSL: Relativized Precedence

2 Beyond TSL Phonology
   - TSL Morphology
   - TSL Morpho-Semantics
   - TSL Syntax

3 Open Problems
   - Better Formal Understanding of TSL
   - Mappings Between Structures
Going Beyond Phonology

TSL provides a good fit for phonological dependencies.

The $10^6$ Question

Is TSL also a good fit for other linguistic structures?

- Morphology?
- Syntax?
Joint work with Alëna Aksënova and Sophie Moradi.

It seems that morphology is also TSL.

(Aksënova et al. 2016)
Example: Circumfixation in Indonesian

- Indonesian has circumfixation with no upper bound on the distance between the two parts of the circumfix.

(1) maha siswa
big pupil
‘student’

(2) *(ke-) maha siswa *(an)
NMN-big pupil -NMN

‘student affairs’

- Requirements: exactly one *ke- and exactly one *an

T contains all NMN affixes
$ an ke ke$

forbidden
$an, ke$, keke, anan

$ an m s ke ke$
Example: Circumfixation in Indonesian

- Indonesian has circumfixation with no upper bound on the distance between the two parts of the circumfix.

1. maha siswa
   big pupil
   'student'

2. *(ke-) maha siswa *(an)
   Nmn- big pupil -Nmn
   'student affairs'

- Requirements: exactly one ke- and exactly one -an

\[
\begin{array}{c c c c c c c}
T & \text{contains all } N_{MN} \text{ affixes} & $ & \text{an} & \text{ke} & \text{ke} & $ \\
\text{forbidden} & $\text{an}, \text{ke}\$, \text{keke}, \text{anan} & | & | & | & | & | \\
\end{array}
\]
Explaining a Typological Gap

▶ In general, affixation can be unbounded.

\[ \text{morgen} \quad \text{tomorrow} \]
\[ \text{über+morgen} \quad \text{the day after tomorrow} \]
\[ (\text{über+})^n \text{morgen} \quad \text{(the day after)}^n \text{ tomorrow} \]

▶ This pattern is SL and hence TSL.

▶ But **circumfixation cannot be unbounded** (e.g. Ilocano).

\[ \text{bigát} \quad \text{tomorrow} \]
\[ \text{ka+bigát+an} \quad \text{the day after tomorrow} \]
\[ * \text{ka+ka+bigát+an+an} \quad \text{the day after the day after tomorrow} \]

**Explanation**

▶ The pattern would be \( \text{ka}^n + \text{bigát} + \text{an}^n \).

▶ This is not first-order definable and hence not TSL.
Explaining a Typological Gap

- In general, affixation can be unbounded.

  \[ \text{morgen} \quad \text{tomorrow} \]
  \[ \text{über+morgen} \quad \text{the day after tomorrow} \]
  \[ (\text{über+})^n \text{morgen} \quad (\text{the day after})^n \text{ tomorrow} \]

- This pattern is SL and hence TSL.

- But **circumfixation cannot be unbounded** (e.g. Ilocano).

  \[ \text{bigát} \quad \text{tomorrow} \]
  \[ \text{ka+bigát+an} \quad \text{the day after tomorrow} \]
  \[ * \text{ka+ka+bigát+an+an} \quad \text{the day after the day after tomorrow} \]

**Explanation**

- The pattern would be \( \text{ka}^n + \text{bigát} + \text{an}^n \).

- This is not first-order definable and hence not TSL.
Explaining a Typological Gap

▶ In general, affixation can be unbounded.

\[
\begin{align*}
morgen & \quad \text{tomorrow} \\
\text{über}+\text{morgen} & \quad \text{the day after tomorrow} \\
(\text{über}+)^n \text{morgen} & \quad (\text{the day after})^n \text{ tomorrow}
\end{align*}
\]

▶ This pattern is SL and hence TSL.
▶ But \textbf{circumfixation cannot be unbounded} (e.g. Ilocano).

\[
\begin{align*}
\text{bigát} & \quad \text{tomorrow} \\
\text{ka}+\text{bigát}+\text{an} & \quad \text{the day after tomorrow} \\
* \text{ka}+\text{ka}+\text{bigát}+\text{an}+\text{an} & \quad \text{the day after the day after tomorrow}
\end{align*}
\]

**Explanation**

▶ The pattern would be \(\text{ka}^n+\text{bigát}+\text{an}^n\).
▶ This is not first-order definable and hence not TSL.
TSL Morpho-Semantics?

The importance of TSL for word structure seems to extend even into semantics.

Case Study: Generalized Quantifiers (Graf 2017b)

A generalized quantifier may have a monomorphemic realization only if its quantifier language is TSL.

- Let’s take this step by step:
  - Monomorphemic?
  - Generalized quantifier?
  - Quantifier language?
The importance of TSL for word structure seems to extend even into semantics.

**Case Study: Generalized Quantifiers (Graf 2017b)**

A generalized quantifier may have a monomorphemic realization only if its quantifier language is TSL.

- Let’s take this step by step:
  - Monomorphemic?
  - Generalized quantifier?
  - Quantifier language?
## Typology of Generalized Quantifiers

Monomorphemic  not assembled from smaller parts

<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Can be monomorphemic?</th>
</tr>
</thead>
<tbody>
<tr>
<td>every</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>some</td>
<td>yes</td>
</tr>
<tr>
<td>not all</td>
<td>no</td>
</tr>
<tr>
<td>two</td>
<td>yes</td>
</tr>
<tr>
<td>all but one</td>
<td>no</td>
</tr>
<tr>
<td>an even number</td>
<td>no</td>
</tr>
<tr>
<td>a third of</td>
<td>no</td>
</tr>
<tr>
<td>most</td>
<td>???</td>
</tr>
</tbody>
</table>
Reminder: Generalized Quantifiers

Generalized quantifier $Q(A, B)$:
- two sets $A$ and $B$ as arguments
- returns truth value $(0, 1)$

Example

(3) Every student cheated.

- $\text{every}(A, B) = 1$ iff $A \subseteq B$
- $\text{student}$: John, Mary, Sue
- $\text{cheat}$: John, Mary
- $\text{student} \not\subseteq \text{cheat} \Rightarrow \text{every}(\text{student}, \text{cheat}) = 0$
- “Every student cheated” is false.
Binary Strings

- The language of $A$ is the set of all permutations of $A$.

**Example**

<table>
<thead>
<tr>
<th>$L(\text{student})$</th>
<th>John, Mary, Sue</th>
</tr>
</thead>
<tbody>
<tr>
<td>student</td>
<td>John Mary Sue, John Sue Mary</td>
</tr>
<tr>
<td></td>
<td>Mary John Sue, Mary Sue John</td>
</tr>
<tr>
<td></td>
<td>Sue John Mary Sue Mary John</td>
</tr>
</tbody>
</table>

- Now replace every $a \in A$ by a truth value:
  - 1 if $a \in B$
  - 0 if $a \notin B$
- The result is the **binary string language** of $A$ under $B$.

**Example**

<table>
<thead>
<tr>
<th>student</th>
<th>John, Mary, Sue</th>
</tr>
</thead>
<tbody>
<tr>
<td>cheat</td>
<td>John, Mary</td>
</tr>
<tr>
<td>binary strings</td>
<td>110, 101, 011</td>
</tr>
</tbody>
</table>
Binary Strings

- The language of $A$ is the set of all permutations of $A$.

### Example

<table>
<thead>
<tr>
<th>student</th>
<th>John, Mary, Sue</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L(\text{student})$</td>
<td>John Mary Sue, John Sue Mary, Mary John Sue, Mary Sue John, Sue John Mary, Sue Mary John</td>
</tr>
</tbody>
</table>

- Now replace every $a \in A$ by a truth value:
  
  1 if $a \in B$
  
  0 if $a \notin B$

- The result is the binary string language of $A$ under $B$.

### Example

<table>
<thead>
<tr>
<th>student</th>
<th>John, Mary, Sue</th>
</tr>
</thead>
<tbody>
<tr>
<td>cheat</td>
<td>John, Mary</td>
</tr>
<tr>
<td>binary strings</td>
<td>110, 101, 011</td>
</tr>
</tbody>
</table>
Quantifier Languages

- Quantifier accepts only binary strings of specific shape
- This is its **quantifier language**.

**Example: every**

- $\text{every}(A, B)$ holds iff $A \subseteq B$
- So every element of $A$ must be mapped to 1.
- All strings must be sequences of 1.
- $L(\text{every}) = \{1\}^*$

**Example: some**

- $\text{some}(A, B)$ holds iff $A \cap B \neq \emptyset$
- Some element of $A$ must be mapped to 1.
- $L(\text{some}) = \{0, 1\}^* 1 \{0, 1\}^*$
Quantifier Languages

- Quantifier accepts only binary strings of specific shape
- This is its \textit{quantifier language}.

\textbf{Example: every}

- \texttt{every}(A, B) holds iff $A \subseteq B$
- So every element of $A$ must be mapped to 1.
- All strings must be sequences of 1.
- $L(\text{every}) = \{1\}^*$

\textbf{Example: some}

- \texttt{some}(A, B) holds iff $A \cap B \neq \emptyset$
- Some element of $A$ must be mapped to 1.
- $L(\text{some}) = \{0, 1\}^* 1 \{0, 1\}^*$
Quantifier Languages

- Quantifier accepts only binary strings of specific shape
- This is its **quantifier language**.

**Example: every**

- $\text{every}(A, B)$ holds iff $A \subseteq B$
- So every element of $A$ must be mapped to 1.
- All strings must be sequences of 1.
- $L(\text{every}) = \{1\}^*$

**Example: some**

- $\text{some}(A, B)$ holds iff $A \cap B \neq \emptyset$
- Some element of $A$ must be mapped to 1.
- $L(\text{some}) = \{0, 1\}^* 1 \{0, 1\}^*$
Overview of Quantifier Languages

If a quantifier language is not TSL, then its quantifier cannot be monomorphemic in any language.

<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Constraint</th>
<th>TSL Description</th>
<th>Mono.</th>
</tr>
</thead>
<tbody>
<tr>
<td>every</td>
<td>no 0</td>
<td>$T := {0, 1}, \neg 0$</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>no 1</td>
<td>$T := {0, 1}, \neg 1$</td>
<td>yes</td>
</tr>
<tr>
<td>some</td>
<td>one or more 1</td>
<td>$T := {1}, \neg$</td>
<td>yes</td>
</tr>
<tr>
<td>not all</td>
<td>one or more 0</td>
<td>$T := {0}, \neg$</td>
<td>no</td>
</tr>
<tr>
<td>(at least) two</td>
<td>two or more 1</td>
<td>$T := {1}, \neg$ $\land \neg1$</td>
<td>yes</td>
</tr>
<tr>
<td>(at most) two</td>
<td>two or fewer 1</td>
<td>$T := {1}, \neg111$</td>
<td>yes</td>
</tr>
<tr>
<td>all but one</td>
<td>exactly one 0</td>
<td>$T := {0}, \neg$ $\land \neg0$</td>
<td>no</td>
</tr>
<tr>
<td>even number</td>
<td>even 1</td>
<td>impossible</td>
<td>no</td>
</tr>
<tr>
<td>most</td>
<td>more 1 than 0</td>
<td>impossible</td>
<td>???</td>
</tr>
</tbody>
</table>
Two Important Remarks

- There is good semantic evidence that “most” is internally complex and hence not monomorphemic. (Hackl 2009)
- If we stipulate that $0 \in T$ implies $1 \in T$, only monomorphemic quantifiers are left!

<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Constraint</th>
<th>TSL Description</th>
<th>Mono.</th>
</tr>
</thead>
<tbody>
<tr>
<td>every</td>
<td>no 0</td>
<td>$T := {0, 1}, \neg 0$</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>no 1</td>
<td>$T := {0, 1}, \neg 1$</td>
<td>yes</td>
</tr>
<tr>
<td>some</td>
<td>one or more 1</td>
<td>$T := {1}, \neg $$</td>
<td>yes</td>
</tr>
<tr>
<td>not all</td>
<td>one or more 0</td>
<td>$T := {0}, \neg $$</td>
<td>no</td>
</tr>
<tr>
<td>(at least) two</td>
<td>two or more 1</td>
<td>$T := {1}, \neg $$ \land \neg 1$</td>
<td>yes</td>
</tr>
<tr>
<td>(at most) two</td>
<td>two or fewer 1</td>
<td>$T := {1}, \neg 111$</td>
<td>yes</td>
</tr>
<tr>
<td>all but one</td>
<td>exactly one 0</td>
<td>$T := {0}, \neg $$ \land \neg 0$</td>
<td>no</td>
</tr>
<tr>
<td>even number</td>
<td>even 1</td>
<td>impossible</td>
<td>no</td>
</tr>
<tr>
<td>most</td>
<td>more 1 than 0</td>
<td>impossible</td>
<td>no</td>
</tr>
</tbody>
</table>
Two Important Remarks

- There is good semantic evidence that “most” is internally complex and hence not monomorphemic. (Hackl 2009)
- If we stipulate that $0 \in T$ implies $1 \in T$, only monomorphemic quantifiers are left!

<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Constraint</th>
<th>TSL Description</th>
<th>Mono.</th>
</tr>
</thead>
<tbody>
<tr>
<td>every</td>
<td>no 0</td>
<td>$T := {0, 1}, \neg 0$</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>no 1</td>
<td>$T := {0, 1}, \neg 1$</td>
<td>yes</td>
</tr>
<tr>
<td>some</td>
<td>one or more 1</td>
<td>$T := {1}, \neg $$</td>
<td>yes</td>
</tr>
<tr>
<td>not all</td>
<td>one or more 0</td>
<td>$T := {0}, \neg $$</td>
<td>no</td>
</tr>
<tr>
<td>(at least) two</td>
<td>two or more 1</td>
<td>$T := {1}, \neg $$ \land \neg $$ 1$</td>
<td>yes</td>
</tr>
<tr>
<td>(at most) two</td>
<td>two or fewer 1</td>
<td>$T := {1}, \neg 1 1 1$</td>
<td>yes</td>
</tr>
<tr>
<td>all but one</td>
<td>exactly one 0</td>
<td>$T := {0}, \neg $$ \land \neg 0 0$</td>
<td>no</td>
</tr>
<tr>
<td>even number</td>
<td>even 1</td>
<td>impossible</td>
<td>no</td>
</tr>
<tr>
<td>most</td>
<td>more 1 than 0</td>
<td>impossible</td>
<td>no</td>
</tr>
</tbody>
</table>
TSL Syntax

▶ Every sentence hides a very elaborate tree structure.
▶ Linguists assume two structural notions:
  - Dependency encodes functor-argument relations
    \(\approx\) semantics
  - Move displaces subtrees \(\approx\) word order

(4) John likes this girl.
(5) This girl, John likes.

▶ In addition, all movements are triggered by features.
TSL Syntax

- Every sentence hides a very elaborate tree structure.
- Linguists assume two structural notions:
  - **Dependency** encodes functor-argument relations
    (≈ semantics)
  - **Move** displaces subtrees (≈ word order)

(4) John likes this girl.

(5) This girl, John likes.

- In addition, all movements are triggered by **features**.
TSL Syntax

- Every sentence hides a very elaborate tree structure.
- Linguists assume two structural notions:
  - Dependency encodes functor-argument relations (≈ semantics)
  - Move displaces subtrees (≈ word order)

(4) John likes this girl.

(5) This girl, John likes.

- In addition, all movements are triggered by features.
More Complex Movement Configurations

A single sentence can contain multiple movements. Movers always target the closest matching node.

(6) Which girl did John tell which picture he took.

\[
\begin{align*}
\text{did} & \quad [\text{wh}^+] \\
\text{tell} & \\
\text{John} & \quad \text{which} \quad [\text{wh}^-] \quad \text{took} \quad [\text{wh}^+] \\
& \quad \text{girl} \quad \text{he} \quad \text{which} \quad [\text{wh}^-] \\
& \quad \quad \text{picture}
\end{align*}
\]
Unbounded Movement

The length of movement steps is unbounded (\(\triangleleft\) won’t be enough).

(7) This girl, John seems to be likely to appear to deny to like.
Exact Matching

And feature polarities must line up one-to-one.

(8) John was attacked.
(9) * Was attacked John.

\[
\begin{array}{c}
\text{was attacked} & [\text{nom}^+] \\
\downarrow & \downarrow \\
\text{John} & [\text{nom}^-]
\end{array}
\]

\[
\begin{array}{c}
\text{was attacked} & \text{was attacked} & [\text{nom}^+] \\
\downarrow & \downarrow & \\
\text{John} & \text{John} & [\text{nom}^-]
\end{array}
\]
An Abstract Example

Is the tree below well-formed? We can solve it with TSL!
An Abstract Example

Is the tree below well-formed? We can solve it with TSL!

\[
\begin{array}{c}
\text{\$} \\
\mid \\
A_+ \\
\mid \\
\mid \\
C_+ \\
\mid \\
\mid \\
G^- \\
\mid \\
\mid \\
\mid \\
\mid \\
E \\
\mid \\
\mid \\
F \\
\mid \\
G \\
\mid \\
\mid \\
J \\
\mid \\
K \\
\mid \\
L \\
\mid \\
M \\
\mid \\
N \\
\mid \\
\mid \\
B \\
\mid \\
C [f^+, g^+, f^-] \\
\mid \\
D \\
\mid \\
A [f^+] \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
\mid \\
A \\
An Abstract Example

Is the tree below well-formed? We can solve it with TSL!
An Abstract Example

Is the tree below well-formed? We can **solve it with TSL!**

Ban everything with locally unmatched $-$ or $+$!
An Abstract Example

Is the tree below well-formed? We can solve it with TSL!

Ban everything with locally unmatched − or +!
An Abstract Example

Is the tree below well-formed? We can solve it with TSL!

Ban everything with locally unmatched − or +!
An Abstract Example

Is the tree below well-formed? We can **solve it with TSL!**

Ban everything with locally unmatched − or +!
An Abstract Example
Is the tree below well-formed? We can solve it with TSL!

Ban everything with locally unmatched − or +!
Why Syntax is TSL

- Syntactic structures encode
  - head-argument dependencies,
  - movement dependencies.
- Movement is controlled by a precise feature calculus.
- Given a tree, we can easily project a “tree tier”
  for each type of movement feature.
- Those tiers greatly reduce the complexity of the problem:
  movement dependencies hold between adjacent nodes.
- Hence we can block illicit local configurations as usual.
Interim Summary

Logical View of TSL

\[ \text{TSL} = \text{Conjunction of Negative Literals with } \triangleleft_T \]

- Phonology and morphology only have TSL dependencies (with a few exceptions).
- TSL plays a role even in morpho-semantics.
- The core of syntax (dependencies, movement) is TSL, too.

Strong Parallelism Hypothesis

All linguistic structures only involve dependencies that are TSL (or at most a minor extension of TSL).
Interim Summary

Logical View of TSL

\[ \text{TSL} = \text{Conjunction of Negative Literals with } \lnot T \]

- Phonology and morphology only have TSL dependencies (with a few exceptions).
- TSL plays a role even in morpho-semantics.
- The core of syntax (dependencies, movement) is TSL, too.

Strong Parallelism Hypothesis

All linguistic structures only involve dependencies that are TSL (or at most a minor extension of TSL).
Outline

1. Logics for Phonology
   - Logic and Linguistic Structures
   - Application to Phonology
   - TSL: Relativized Precedence

2. Beyond TSL Phonology
   - TSL Morphology
   - TSL Morpho-Semantics
   - TSL Syntax

3. Open Problems
   - Better Formal Understanding of TSL
   - Mappings Between Structures
The Open Problems

1. We do not understand TSL well.
2. TSL undergenerates slightly.
3. We also need mappings between structures.
Open Questions About TSL

- Right now, dependencies are shown to be TSL by providing a formula/grammar. A more abstract technique would be much more efficient.
- Non-TSL is cumbersome to prove:
  - consider all possible tiers
  - show that none work

What We Need

- pumping lemma
- decomposition theorems (if $L \not\in X \cap Y$, then $L \not\in \text{TSL}$)
- smallest-counterexample results
- ...
There are a few, very rare phenomena that require slightly more expressivity than TSL:

1. local blocking of unbounded sibilant harmony
2. RHOL-like stress patterns
3. unbounded tone plateauing
4. unbounded circumambient processes

(1) and (2) are handled with minimal extensions of TSL. (Baek 2017; De Santo 2017; De Santo and Graf 2017)

(3) and (4) require an extension of SP:
FO formulas of the form $\Delta \rightarrow \Phi$, where

$\Delta$ is a domain formula
$\Phi$ is an SP formula (≡ CNL formula with $\triangleleft^+$)

(Graf 2016)
Mappings and Transductions

- Linguists care deeply about **mappings between structures**.

**Example: Word-Final Devoicing**

- It is not enough that voiced consonants are forbidden at the end of a word.

- There are principled alternations that need to be captured:
  
  \[
  \begin{align*}
  [\text{raːt}] & \quad \text{wheel or advice} \\
  [\text{raːtə}] & \quad \text{advice.Pl} \\
  [\text{reːdə}] & \quad \text{wheel.Pl}
  \end{align*}
  \]

  \[\iffalse\]
  
  \[
  \begin{align*}
  /\text{raːt}/ & \quad \text{advice} \\
  /\text{raːd}/ & \quad \text{wheel}
  \end{align*}
  \]

- Logical models of mappings (＝ transductions) exist, but are too powerful.
A First-Order Transduction

- A logical transduction operates by representing one structure inside another.

Example: String Reversal as an FO transduction

Input Relation: ◁
Output Relation: ◀

\[ x ◀ y \iff y ◁ x \]

- No natural language mapping is capable of reversal.
- Like in the case of LTT, full FO is too much.
A First-Order Transduction

- A logical transduction operates by representing **one structure inside another**.

**Example: String Reversal as an FO transduction**

**Input Relation**: \(<\)

**Output Relation**: \(\triangleleft\)

\[ x \triangleleft y \iff y \triangleleft x \]

- No natural language mapping is capable of reversal.
- Like in the case of LTT, full FO is too much.
Important Questions About Transductions

1. What are linguistically reasonable fragments of FO for transductions?
2. Are they closed under composition?
   (A linguistic grammar usually is a sequence of mappings.)
3. Do they preserve definability in TSL/FO/MSO?
4. What is the strongest class of transductions that preserves TSL?
Weak string transductions have been studied, but their **connection to logic is unclear**. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)

![Jane Chandlee](image)

**Snapshot of Ongoing Research**

Finite-State Transductions

Left Subsequential

Left Output Strictly Local

Right Subsequential

Input Strictly Local

First-Order Logic

Right Output Strictly Local

MSO = 2-DFST
Weak string transductions have been studied, but their **connection to logic is unclear**.
(Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Weak string transductions have been studied, but their **connection to logic is unclear**. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Weak string transductions have been studied, but their **connection to logic is unclear**. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Weak string transductions have been studied, but their **connection to logic is unclear**. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Phonology Beyond TSL Phonology Open Problems Conclusion

Snapshot of Ongoing Research

Weak string transductions have been studied, but their connection to logic is unclear. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)

Jane Chandlee
Weak string transductions have been studied, but their connection to logic is unclear.
(Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Weak string transductions have been studied, but their connection to logic is unclear. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Weak string transductions have been studied, but their connection to logic is unclear. (Mohri 1997; Engelfriet and Hoogeboom 2001; Courcelle and Engelfriet 2012; Chandlee 2014)
Conclusion

- When viewed from a logical perspective, language is **surprisingly weak**:
  - Even propositional logic is too much.
  - We need CNL with relativized $\langle^+\rangle$.
- This weakness holds **across language modules**.
  - phonology
  - morphology
  - morpho-semantics
  - syntax
- This has major empirical and theoretical implications.
- But many open questions remain.
- In order to address those questions, we will need an **alliance of linguists, logicians, and computer scientists**.
Resources and Readings

1. **Survey papers**: Pullum and Rogers (2006); Heinz (2011a,b, 2015); Rogers and Pullum (2011); Chandlee and Heinz (2016)

2. **TSL and its extensions**: Heinz et al. (2011); McMullin (2016); Baek (2017); De Santo (2017); De Santo and Graf (2017); Graf (2016)

3. **TSL morphology**: Aksënova et al. (2016); Graf (2017a)

4. **TSL morpho-semantics**: Graf (2017b)

5. **TSL syntax**: Graf (2012); Graf and Heinz (2016)


7. **Learnability**: Heinz (2010); Kasprzik and Kötzing (2010); Heinz et al. (2012); Jardine et al. (2014); Lai (2015); Jardine and Heinz (2016); Jardine and McMullin (2017)


References III


References IV


References V


