Logical Transductions for the Typology of Ditransitive Prosody

Mai Ha Vu  Aniello De Santo  Hossep Dolatian

m.h.vu@iln.uio.no
aniello.desanto@utah.edu
hossep.dolatian@alumni.stonybrook.edu

SIGMORPHON 2022
Phonological processes can refer to domains larger than words.

These domains form hierarchical layers (prosodic constituents).

But: Prosodic constituency cannot be read directly from syntactic constituency.

Also: Little existing work on the computation of phrase-level phonology (Yu 2021).

### In This Talk

**Computational requirements for the syntactic/prosody mapping?**

- Using logical tree transductions
- A case study: Ditransitives in SVO languages
Prosodic Constituency

- Prosodic domains form hierarchical layers
- Consider the internal arguments of a ditransitive verb...

\[ i \]
\[ p \]
\[ w \]
\[ w \]
\[ p \]
\[ w \]

I gave Mary books
Prosodic Constituency

- Prosodic domains form hierarchical layers
- Consider the internal arguments of a ditransitive verb...

```
I gave Mary books
```

```
Intonational Phrase
```
Prosodic Constituency

- Prosodic domains form hierarchical layers
- Consider the internal arguments of a ditransitive verb...

I gave Mary books
Prosodic Constituency

- Prosodic domains form hierarchical layers
- Consider the internal arguments of a ditransitive verb...

I gave Mary books
Syntax to Prosody?

How does the syntactic parse map to the prosodic parse?

\[
\begin{align*}
\text{CP}_9 & \quad \text{vP}_8 \\
& \quad \text{VP}_7 \\
& \quad \text{V'} \\
& \quad \text{NP}_5 \\
& \quad \text{N}_2 \\
& \quad \text{V}_3 \\
& \quad \text{NP}_6 \\
& \quad \text{N}_4 \\
\text{gave} & \quad \text{Mary} & \quad \text{books}
\end{align*}
\]
Syntax to Prosody?

How does the syntactic parse map to the prosodic parse?
Syntax to Prosody?

How does the syntactic parse map to the prosodic parse?

Mismatches in the size of an XP and its prosodic phrase
Ambiguity wrt input-output correspondences
Syntax/Prosody Mappings: Ewe

▶ SVO ditransitive phrases: four types of prosodic parses
(Kalivoda 2018)

Input Syntax
[V [N N]]

Separated
(V)(N)(N)

\[\text{gave Mary books}\]
Syntax/Prosody Mappings: Chimwiini

- SVO ditransitive phrases: four types of prosodic parses
  (Kalivoda 2018)

Input Syntax

\[
[V [N N]]
\]

Closest-merged

\[(VN)(N)\]
Syntax/Prosody Mappings: Kimatuumbi

▶ SVO ditransitive phrases: four types of prosodic parses (Kalivoda 2018)

Input Syntax

```
[V [N N]]
```

Recursive

```
((VN)(N))
```

gave Mary books
Syntax/Prosody Mappings: Zulu

- SVO ditransitive phrases: four types of prosodic parses (Kalivoda 2018)

Input Syntax

\[[V \ [N \ N]]\]

```
CP_9
  ...
    vP_8
      VP_7
        V'
          NP_5
            v_1
              N_2
                V_3
                  NP_6
                    N_4
```

gave Mary books

All-merged (VNN)

```
i_9'
  p_8'
    w_1'
    w_2'
    w_4'

gave Mary books
```
A Typological Overview (Kalivoda 2018)

Questions

- What is the complexity of these mappings?
- What syntactic information is relevant?
A Typological Overview (Kalivoda 2018)

Input Syntax

[\text{\text{V [N N]}}]

Separated

Ewe

(V)(N)(N)

Closest-merged

Chimwiini

(VN) (N)

Recursive

Kimatuumbi

((VN)N)

All-merged

Zulu

(VNN)

Questions

- What is the complexity of these mappings?
- What syntactic information is relevant?
Logical Tree Transductions

- Take a mapping that changes root labels from $a$ to $b$

- With logical transductions, the input tree model is defined in terms of a signature $\langle D, R \rangle$

### Tree Model

**Domain** $D = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$

**Unary relations** $L \subset R$:
- $a(x) = \{\varepsilon, 0, 01, 12\}$
- $b(x) = \{1, 00\}$
- $c(x) = \{11\}$

**Binary relations** in $R$:
- $\langle x, y \rangle = \{(\varepsilon, 0), (\varepsilon, 1), (0, 00), (0, 01), (1, 10), (1, 11)\}$
- $\langle x, y \rangle = \{(0, 1), (00, 01), (10, 11)\}$
Logical Tree Transductions

- Take a mapping that changes root labels from \( a \) to \( b \)

\[ 
\begin{array}{c}
\text{a} \\
\text{a} \\
\text{b} \\
\text{b} \\
\text{a} \\
\text{a} \\
\text{c} \\
\text{c}
\end{array} \quad \begin{array}{c}
\text{b} \\
\text{b} \\
\text{a} \\
\text{a} \\
\text{a} \\
\text{a} \\
\text{c} \\
\text{c}
\end{array}
\]

- With logical transductions, the input tree model is defined in terms of a signature \( \langle D, R \rangle \)

Tree Model

Domain \( D = \{ \varepsilon, 0, 1, 00, 01, 10, 11 \} \)

Unary relations \( L \subset R \):
- \( a(x) = \{ \varepsilon, 0, 12 \} \)
- \( b(x) = \{ 1, 00 \} \)
- \( c(x) = \{ 11 \} \)

Binary relations in \( R \):
- \( \prec(x, y) = \{ (\varepsilon, 0), (\varepsilon, 1), (0, 00), (0, 01), (1, 10), (1, 11) \} \)
- \( \prec(x, y) = \{ (0, 1), (00, 01), (10, 11) \} \)
Logical Tree Transductions [cont.]

- Take a mapping that changes root labels from $a$ to $b$

\[
\begin{array}{c}
\text{a} \\
\text{a} \quad \text{b} \\
\text{b} \quad \text{a} \quad \text{a} \quad \text{c} \\
\end{array}
\begin{array}{c}
\text{b} \\
\text{a} \quad \text{a} \quad \text{a} \quad \text{c} \\
\end{array}
\]

- Predicated define properties of the input segments
- Output functions define output segments wrt input segments

**Tree transduction**

\[
\begin{align*}
\text{root}_{a}(x) & \overset{\text{def}}{=} a \land \exists y[<(y, x)] \\
<(x', y') & \overset{\text{def}}{=} <(x, y) \\
\phi a(x') & \overset{\text{def}}{=} a(x) \land \neg \text{root}_{a}(x) \\
\phi b(x') & \overset{\text{def}}{=} b(x) \lor \text{root}_{a}(x) \\
\phi c(x') & \overset{\text{def}}{=} c(x)
\end{align*}
\]

Formalizing Syntax/Prosody Mappings

What Information Matters?

- Pronounced vs unpronounced nodes
  ⇒ prosody works over overt or pronounced terminal items

- Headedness
  ⇒ can be reconstructed from local geometry of the tree

- Tree geometry
  ⇒ sensitivity to sisterhood and c-command

- Argument structure
  ⇒ two configurations: with and without head-movement

- Linearity
  ⇒ the verb is phrased with its closest argument

- Category labels
  ⇒ syntax/prosody mappings generally blind to category labels
Formalizing Syntax/Prosody Mappings

What Information Matters?

- Pronounced vs unpronounced nodes
  ⇒ prosody works over overt or pronounced terminal items

- Headedness
  ⇒ can be reconstructed from local geometry of the tree

- Tree geometry
  ⇒ sensitivity to sisterhood and c-command

- Argument structure
  ⇒ two configurations: with and without head-movement

- Linearity
  ⇒ the verb is phrased with its nearest argument

- Category labels
  ⇒ syntax/prosody mappings generally blind to category labels
Formalizing Syntax/Prosody Mappings

What Information Matters?

- Pronounced vs unpronounced nodes
  ⇒ prosody works over overt or pronounced terminal items
- Headedness
  ⇒ can be reconstructed from local geometry of the tree
- Tree geometry
  ⇒ sensitivity to sisterhood and c-command
- Argument structure
  ⇒ two configurations: with and without head-movement
- Linearity
  ⇒ the verb is phrased with its closest argument
- Category labels
  ⇒ syntax/prosody mappings generally blind to category labels
Formalizing Syntax/Prosody Mappings

What Information Matters?

- Pronounced vs unpronounced nodes
  ⇒ prosody works over overt or pronounced terminal items

- Headedness
  ⇒ can be reconstructed from local geometry of the tree

- Tree geometry
  ⇒ sensitivity to sisterhood and c-command

- Argument structure
  ⇒ two configurations: with and without head-movement

- Linearity
  ⇒ the verb is phrased with its closest argument

- Category labels
  ⇒ syntax/prosody mappings generally blind to category labels
Formalizing Syntax/Prosody Mappings

What Information Matters?

- Pronounced vs unpronounced nodes
  ⇒ prosody works over overt or pronounced terminal items

- Headedness
  ⇒ can be reconstructed from local geometry of the tree

- Tree geometry
  ⇒ sensitivity to sisterhood and c-command

- Argument structure
  ⇒ two configurations: with and without head-movement

- Linearity
  ⇒ the verb is phrased with its closest argument

- Category labels
  ⇒ syntax/prosody mappings generally blind to category labels
Formalizing Syntax/Prosody Mappings

What Information Matters?

- Pronounced vs unpronounced nodes
  \(\Rightarrow\) prosody works over overt or pronounced terminal items

- Headedness
  \(\Rightarrow\) can be reconstructed from local geometry of the tree

- Tree geometry
  \(\Rightarrow\) sensitivity to sisterhood and c-command

- Argument structure
  \(\Rightarrow\) two configurations: with and without head-movement

- Linearity
  \(\Rightarrow\) the verb is phrased with its closest argument

- Category labels
  \(\Rightarrow\) syntax/prosody mappings generally blind to category labels
Summing Up

**Broad Result**

First-order Tree Transductions derive the alignment mismatches between syntactic and prosodic constituents!

**General Takeaways**

- Usually unspecified mapping details matter!
  - Head-movement and locality
  - Predictions from category Blindness
  - Complexity of the mappings

- Tree transductions to refine long-standing theoretical questions

- Inspect theoretical assumptions about linguistic representations across sub-domains
Summing Up

**Broad Result**

First-order Tree Transductions derive the alignment mismatches between syntactic and prosodic constituents!

**General Takeaways**

- Usually unspecified mapping details matter!
  - Head-movement and locality
  - Predictions from category Blindness
  - Complexity of the mappings
- Tree transductions to refine long-standing theoretical questions
- Inspect theoretical assumptions about linguistic representations across sub-domains

Thank you!