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
The Talk in One Slide

- 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 gave Mary books
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...

\[
\begin{array}{c}
\text{i} \\
p \\
w \\
\text{w} \\
\text{w} \\
\text{w} \\
\text{Phonological Phrase} \\
\text{Intonational Phrase}
\end{array}
\]

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?

```
CP_9
   ...
   vP_8
         ...
         VP_7
             ...
             V'
                 ...
                 NP_5
                 V_3
                 NP_6

v_1    N_2    V_3    N_4

gave    Mary    t    books
```

```
i_9'
   ...
   p_5'/8/e'
       ...
       p_6'
           ...
           w_1'
           w_2'
           w_4'

gave    Mary    books
```
Syntax to Prosody?

How does the syntactic parse map to the prosodic parse?

Prosodic Constituency
A Prosodic Typology
Logical Transductions
Generalizations

Prosodic Constituency
A Prosodic Typology
Logical Transductions
Generalizations
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)

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))

[47x192]Recursive
((VN)(N))

... vP₈

vP₈

V₇

NP₅

NP₆

V' N₂ V₃ N₄

V' w₁' w₂' w₄'

p₅'/e'

i₉'

p₈'

gave Mary books

input syntax

[SVO ditransitive phrases: four types of prosodic parses
(Kalivoda 2018)]
Syntax/Prosody Mappings: Zulu

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

Input Syntax

\[
[V [N N]]
\]

All-merged

(VNN)

\[
\begin{align*}
i_{9'} & \quad p_{8'} \\
w_{1'} & \quad w_{2'} \quad w_{4'}
\end{align*}
\]

gave Mary books

\[
gave \quad Mary \quad t \quad books
\]
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

[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 \( b \)

\[
\begin{align*}
\text{a} & \quad \text{b} \\
\text{a} & \quad \text{b} \\
\text{b} & \quad \text{a} & \quad \text{a} & \quad \text{c} \\
\text{b} & \quad \text{a} & \quad \text{a} & \quad \text{c}
\end{align*}
\]

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

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

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

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

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

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

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

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

**Tree transduction**

\[
\begin{align*}
\text{root}_a(x) & \overset{\text{def}}{=} a \land \exists y[\langle y, x \rangle] \\
\langle x', y' \rangle & \overset{\text{def}}{=} \langle x, y \rangle \\
\phi_a(x') & \overset{\text{def}}{=} a(x) \land \neg \text{root}_a(x) \\
\phi_b(x') & \overset{\text{def}}{=} b(x) \lor \text{root}_a(x) \\
\phi_c(x') & \overset{\text{def}}{=} c(x)
\end{align*}
\]
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 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
  ⇒ 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
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!