Syntactic tiers for movement and agreement

Day 1: Representations and the status of tiers

Thomas Graf

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

KU Leuven Lecture Series
Decmber 5–7, 2023

The big linguistic questions

- What are the laws that govern each structural level?
- **Why** are those the laws?
- How **complex** are these laws? How hard are they to compute?
- How are they **learned**?
- 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 larger program behind syntactic tiers

- grounded in computation
- insights across subfields (e.g. parallels between phonology and syntax)
- take insights from generative grammar, but interpret them in new ways

The big linguistic questions

- What are the laws that govern each structural level?
- **Why** are those the laws?
- How **complex** are these laws? How hard are they to compute?
- How are they **learned**?
- 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 larger program behind syntactic tiers

- grounded in computation
- insights across subfields (e.g. parallels between phonology and syntax)
- take insights from generative grammar, but interpret them in new ways

Outline

SL & TSL: (T)ier-Based Strictly Local

- Locality is perhaps the most central property of language.
- But locality can be measured in many ways.
- SL and TSL provide very strict notions of locality that work well empirically.

(Tier-Based) Strictly Local Dependencies

- All patterns described by markedness constraints that are
  - inviolable,
  - locally bounded,
  - formalized as n-grams.
- Non-local dependencies are **local over tiers**.
  (Goldsmith 1976; "but there is another. . .")
- **Linguistic core idea:**
  Dependencies are local over the right structure.

SL & TSL: (T)ier-Based Strictly Local

- Locality is perhaps the most central property of language.
- But locality can be measured in many ways.
- SL and TSL provide very strict notions of locality that work well empirically.

(Tier-Based) Strictly Local Dependencies

- All patterns described by markedness constraints that are
  - inviolable,
  - locally bounded,
  - formalized as n-grams.
- Non-local dependencies are **local over tiers**.
  (Goldsmith 1976; "but there is another...")
- **Linguistic core idea:**
  Dependencies are local over the right structure.

Example: Word-Final Devoicing is SL-2

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

Example: German

* $ r a **d** $

***z$**

***v$**

***d$**

Example: Word-Final Devoicing is SL-2

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

Example: German

	*z$
* $ r a d $	*v$
	*d$

Example: Word-Final Devoicing is SL-2

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

Example: German

$*\boxed{\$\ r}\ a\ d\ \$$

*z$

*v$

*d$

Example: Word-Final Devoicing is SL-2

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

Example: German

* **$** $\boxed{\text{r a}}$ **d** **$**

***z$**

***v$**

***d$**

Example: Word-Final Devoicing is SL-2

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

Example: German

* $ r $\boxed{\text{a d}}$ $

***z$**

***v$**

***d$**

Example: Word-Final Devoicing is SL-2

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

Example: German

* $ r a $\boxed{\text{d \$}}$

***z$**

***v$**

***d$**

Example: Word-Final Devoicing is SL-2

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

Example: German

* $ r a $\boxed{\text{d \$}}$ → *d$

*z$

*v$

*d$

Example: Word-Final Devoicing is SL-2

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

Example: German

	*z$	
* $ r a **d** $	*v$	$ r a t $
	*d$	

Example: Word-Final Devoicing is SL-2

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

Example: German

* **$** r a **d** **$**

***z$**
***v$**
***d$**

$\boxed{\$\ r}$ a t **$**

Example: Word-Final Devoicing is SL-2

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

Example: German

*$ r a **d** $

*z$
*v$
*d$

$ r a t $

Example: Word-Final Devoicing is SL-2

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

Example: German

* $ r a **d** $

***z$**
***v$**
***d$**

$ r a t $

Example: Word-Final Devoicing is SL-2

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

Example: German

* $ r a **d** $

***z$**
***v$**
***d$**

$ r a $\boxed{\text{t \$}}$

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

* $ a s o l a $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

* $ **a s o** l **a** $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$\boxed{\text{\$ a s}}$ o l a \$

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

* \$ **a s o** l **a** \$

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

*$ a s o l a $ $ a z o l a $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ a s o l a $ $ a z o l a $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have V[−voice]V
- **Suppose**:
  - [−voice] = {s,ʃ}
  - V = {a,i,o,u}
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

*$ **a s o** l **a** $ $ **a** z **o** l **a** $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ a s o l a $ $ a $\boxed{\text{z o l}}$ a $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ a s o l a $ $ a z o l a $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ a s o l a $ $ a z o $\boxed{\text{l a \$}}$

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*\$ a s o l a \$ \$ a z o l a \$ \$ a + s o c i a l e \$

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ **a** **s** **o** l **a** $ $ **a** z **o** l **a** $ $\boxed{\$\ \mathbf{a}\ +}$ **s** **o** c **i** **a** l **e** $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ a s o l a $ $ a z o l a $ $ a + s o c i a l e $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

*$ a s o l a $ $ a z o l a $ $ a $\boxed{+ \text{ s o}}$ c i a l e $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ **a s o** l **a** $ $ **a** z **o** l **a** $ $ **a** + $\boxed{\textbf{s o}\ \mathrm{c}}$ **i a** l **e** $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

*$ a s o l a $ $ a z o l a $ $ a + s $\boxed{\text{o c i}}$ a l e $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

*$ a s o l a $ $ a z o l a $ $ a + s o c i a l e $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\mathrm{voice}] = \{$s,ʃ$\}$
  - $\mathrm{V} = \{$a,i,o,u$\}$
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, ...

Example: Northern Italian

*$ a s o l a $ $ a z o l a $ $ a + s o c i a l e $

Example: Intervocalic Voicing is SL-3

- Captured by forbidding voiceless segments between vowels
- **Description:** don't have V[−voice]V
- **Suppose**:
  - [−voice] = {s,ʃ}
  - V = {a,i,o,u}
- **Compiled out**: don't have **asa**, **aʃa**, **asi**, **aʃi**, . . .

Example: Northern Italian

*$ a s o l a $ $ a z o l a $ $ a + s o c i $\boxed{\text{a l e}}$ $

Test Your Might

Exercise

Suppose that a language with CV as the only syllable template exhibits vowel harmony without any neutral vowels or any blockers.

1. What is the complexity of vowel harmony in this language?
2. What if CVC syllables are also possible?

Exercise

Can you give an example of an SL string dependency in (morpho)syntax?

Outline

Not all of phonology is strictly local

Culminativity

Every phonological word has **exactly one** syllable that carries primary stress.

Culminativity (Rephrased)

Every phonological word has

- **at most one** syllable that carries primary stress, and
- **some** syllable that carries primary stress.

Exercise

Explain why culminativity cannot be SL.

Another Problem: Samala Sibilant Harmony

- If multiple sibilants occur in the same word, they must all be [+anterior] (**s**,**z**) or [−anterior] (**ʃ**,**ʒ**).
- In other words: Don't mix **purple** and **teal**.

  ***sʃ** ***sʒ** ***zʃ** ***zʒ**
  ***ʃs** ***ʒs** ***ʃz** ***ʒz**

- **But:** Sibilants can be arbitrarily far away from each other!

Example: Samala (Applegate 1972)

* $ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Another Problem: Samala Sibilant Harmony

- If multiple sibilants occur in the same word, they must all be [+anterior] (s,z) or [−anterior] (ʃ,ʒ).
- In other words: Don't mix **purple** and **teal**.

*sʃ *sʒ *zʃ *zʒ
*ʃs *ʒs *ʃz *ʒz

- **But:** Sibilants can be arbitrarily far away from each other!

Example: Samala (Applegate 1972)

*$ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Another Problem: Samala Sibilant Harmony

- If multiple sibilants occur in the same word, they must all be [+anterior] (s,z) or [−anterior] (ʃ,ʒ).
- In other words: Don't mix **purple** and **teal**.

  *sʃ *sʒ *zʃ *zʒ
  *ʃs *ʒs *ʃz *ʒz

- **But:** Sibilants can be arbitrarily far away from each other!

Example: Samala (Applegate 1972)

* $ h a $\boxed{\text{s x i n t i l a w a ʃ}}$ $

$ h a ʃ x i n t i l a w a ʃ $

Another Problem: Samala Sibilant Harmony

- If multiple sibilants occur in the same word,
  they must all be [+anterior] (s,z) or [−anterior] (ʃ,ʒ).
- In other words: Don't mix **purple** and **teal**.

  *sʃ *sʒ *zʃ *zʒ
  *ʃs *ʒs *ʃz *ʒz
- **But:** Sibilants can be arbitrarily far away from each other!

Example: Samala (Applegate 1972)

* $ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Another Problem: Samala Sibilant Harmony

- If multiple sibilants occur in the same word, they must all be [+anterior] (**s**,**z**) or [−anterior] (**ʃ**,**ʒ**).
- In other words: Don't mix **purple** and **teal**.

  ***sʃ** ***sʒ** ***zʃ** ***zʒ**
  ***ʃs** ***ʒs** ***ʃz** ***ʒz**

- **But:** Sibilants can be arbitrarily far away from each other!

Example: Samala (Applegate 1972)

* $ h a **s** x i n t i l a w a **ʃ** $

$ h a **ʃ** x i n t i l a w a **ʃ** $

* $ **s** t a j a n o w o n w a **ʃ** $

Another Problem: Samala Sibilant Harmony

- If multiple sibilants occur in the same word, they must all be [+anterior] (**s**,**z**) or [−anterior] (**ʃ**,**ʒ**).
- In other words: Don't mix **purple** and **teal**.

  *sʃ *sʒ *zʃ *zʒ
  *ʃs *ʒs *ʃz *ʒz
- **But:** Sibilants can be arbitrarily far away from each other!

Example: Samala (Applegate 1972)

*$ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

*$ s t a j a n o w o n w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

*$ h a s x i n t i l a w a ʃ $ $ h a ʃ x i n t i l a w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

*$ h a s x i n t i l a w a ʃ $ $ h a ʃ x i n t i l a w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

```
  $    s               ʃ $
  |    |               | |
*$ h a s x i n t i l a w a ʃ $        $ h a ʃ x i n t i l a w a ʃ $
```

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

```
 $    s              ʃ $      $   ʃ               ʃ $
 |    |              | |      |   |               | |
*$ h a s x i n t i l a w a ʃ $      $ h a ʃ x i n t i l a w a ʃ $
```

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

$ s ʃ $

* $ h a s x i n t i l a w a ʃ $

$ ʃ ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

$ s ʃ $ $ ʃ ʃ $

*$ h a s x i n t i l a w a ʃ $ $ h a ʃ x i n t i l a w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

Tier:	$ s ʃ $	$ ʃ ʃ $
String:	* $ h a s x i n t i l a w a ʃ $	$ h a ʃ x i n t i l a w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

$ s ʃ $

*$ h a s x i n t i l a w a ʃ $

$ ʃ ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**.
  (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

1. Project sibilant tier
2. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

$ s ʃ $ $\quad\quad$ $ ʃ ʃ $

*$ h a s x i n t i l a w a ʃ $ $\quad\quad$ $ h a ʃ x i n t i l a w a ʃ $

Another TSL Boon: Blocking

- TSL can also handle blocking effects.
- **Slovenian sibilant harmony with blocking**
  1. **[−ant]** ... **[+ant]** is illicit,
  2. unless **t** or **d** intervenes.
- **TSL-2 account**
  1. project all **[−ant]**, **[+ant]**, **t**, and **d**
  2. don't have **[−ant] [+ant]**

Example: Slovenian (Jurgec 2011; McMullin 2016)

*$ **s** p i **ʃ** $ $ **z** i **d** a **ʃ** $

Another TSL Boon: Blocking

- TSL can also handle blocking effects.
- **Slovenian sibilant harmony with blocking**
  1. **[−ant]** ... **[+ant]** is illicit,
  2. unless **t** or **d** intervenes.
- **TSL-2 account**
  1. project all **[−ant]**, **[+ant]**, **t**, and **d**
  2. don't have **[−ant]** **[+ant]**

Example: Slovenian (Jurgec 2011; McMullin 2016)

\$ **s** **ʃ** \$

*\$ **s** p i **ʃ** \$ \$ **z** i **d** a **ʃ** \$

Another TSL Boon: Blocking

- TSL can also handle blocking effects.
- **Slovenian sibilant harmony with blocking**
  1. **[−ant]** . . . **[+ant]** is illicit,
  2. unless **t** or **d** intervenes.
- **TSL-2 account**
  1. project all **[−ant]**, **[+ant]**, **t**, and **d**
  2. don't have **[−ant] [+ant]**

Example: Slovenian (Jurgec 2011; McMullin 2016)

$ **s** ʃ $ $ **z** **d** ʃ $

*$ **s** p i ʃ $ $ **z** i **d** a ʃ $

Tiers have many interpretations

1 Separate structure

$$\$ \xrightarrow{\lhd} \mathsf{z} \xrightarrow{\lhd} \mathsf{d} \xrightarrow{\lhd} \int \xrightarrow{\lhd} \$$$

2 Additional relation

$$\begin{tikzcd}
\$ \arrow[r, "\lhd"] & \mathsf{z} \arrow[r, "\lhd"] \arrow[rr, bend left, "\lhd_T"] & \mathsf{i} \arrow[r, "\lhd"] & \mathsf{d} \arrow[r, "\lhd"] \arrow[rr, bend left, "\lhd_T"] & \mathsf{a} \arrow[r, "\lhd"] & \int \arrow[r, "\lhd"] & \$
\end{tikzcd}$$

3 Substructures

- without tier: $z, zi, id, da, aʃ, ʃ$
- with tier: $z, zd, dʃ, ʃ$

4 Memory configurations

Input	$	z	i	d	a	ʃ	$
SL-2 memory	-	$	z	i	d	a	ʃ
TSL-2 memory	-	z	z	d	d	ʃ	ʃ

Epistemological modesty: what are tiers?

- Don't get too attached to a specific interpretation of tiers!
- Tiers are a visual metaphor for a computational mechanism:
  1. Memorize a finite amount of symbols **of a specific type**
  2. Check the current symbol against the memory configuration
- Nothing about this is specific to strings $\Rightarrow$ **tree tiers!**

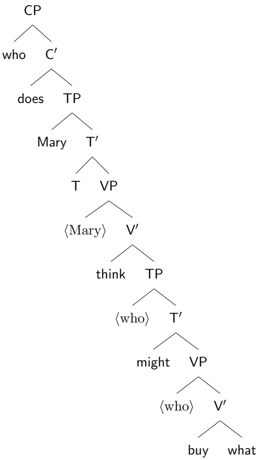
Outline

1 Strict locality (SL)

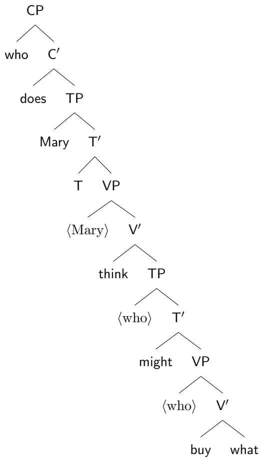
2 Tier-Based Strictly Local (TSL)

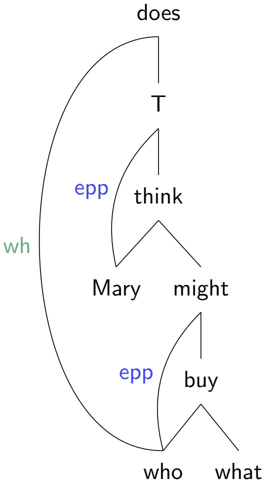
3 Subregular Syntax: SL & TSL over Trees
- Representing syntactic computations
- Merge is SL
- Move is TSL
- Islands $\equiv$ Blocking

Freeing syntax from its output

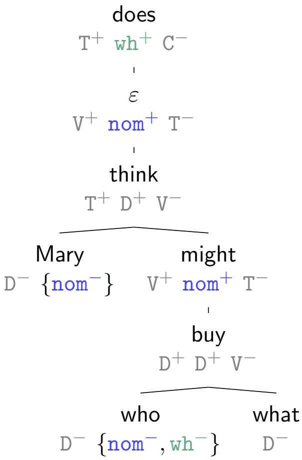


Freeing syntax from its output





Using diacritics to arborize dependency graphs



- **Terminology**
  - Category ~~feature~~ $\mathtt{X}^-$
  - Selector ~~feature~~ $\mathtt{X}^+$
  - Licensee ~~feature~~ $\mathtt{x}^-$
  - Licensor ~~feature~~ $\mathtt{x}^+$
- **Diacritics $\neq$ features**

  feature: potential to form a dependency

  diacritic: fully formed dependency

Original inspiration: Minimalist Grammars

Ed Stabler

- Diacritics inspired by Minimalist grammars (MGs)
- MG: formalization of Minimalist syntax (Stabler 1997, 2011)

A Detailed Merge Example

(1) John [$_{\mathrm{VP}}$ t laughed at Bill].

Sequence of Merge steps:

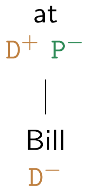
1. **at** selects DP (**Bill**)
2. **laughed** selects PP (**at**)
3. **laughed** selects DP (**John**)

A Detailed Merge Example

(1) John [$_{\mathrm{VP}}$ t laughed at Bill].

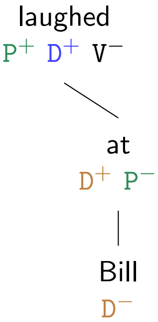
Sequence of Merge steps:

1. **at** selects DP (**Bill**)
2. **laughed** selects PP (**at**)
3. **laughed** selects DP (**John**)



A Detailed Merge Example

(1) John [$_{\text{VP}}$ t laughed at Bill].



Sequence of Merge steps:

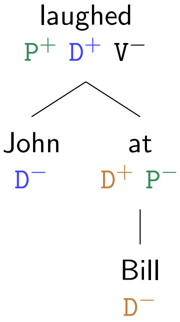
1. **at** selects DP (**Bill**)
2. **laughed** selects PP (**at**)
3. **laughed** selects DP (**John**)

A Detailed Merge Example

(1) John [$_{\text{VP}}$ t laughed at Bill].

Sequence of Merge steps:

1. **at** selects DP (**Bill**)
2. **laughed** selects PP (**at**)
3. **laughed** selects DP (**John**)



Merge is SL-2

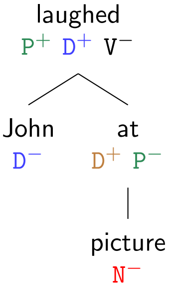
Merge is SL-2 over trees because we only need to look at

1. the mother and
2. its daughters

Merge is SL-2

Merge is SL-2 over trees because we only need to look at

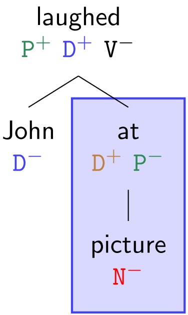
1. the mother and
2. its daughters



Merge is SL-2

Merge is SL-2 over trees because we only need to look at

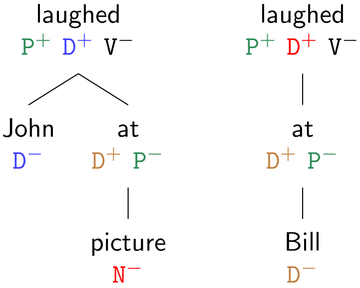
1. the mother and
2. its daughters



Merge is SL-2

Merge is SL-2 over trees because we only need to look at

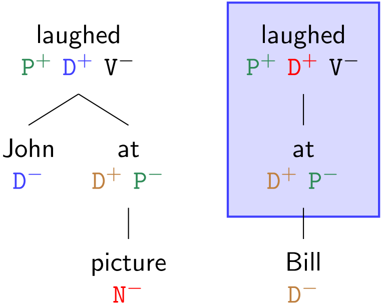
1. the mother and
2. its daughters



Merge is SL-2

Merge is SL-2 over trees because we only need to look at

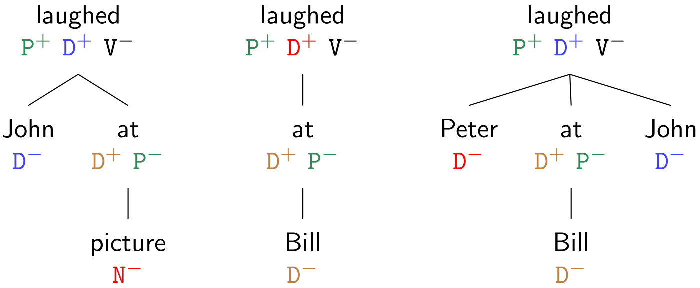
1. the mother and
2. its daughters



Merge is SL-2

Merge is SL-2 over trees because we only need to look at

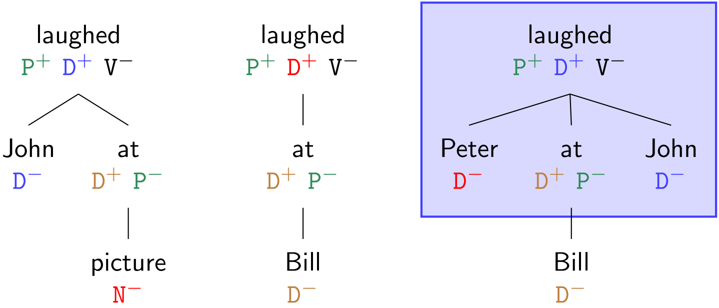
1. the mother and
2. its daughters



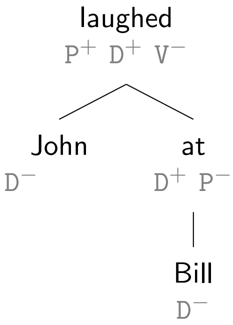
Merge is SL-2

Merge is SL-2 over trees because we only need to look at

1. the mother and
2. its daughters



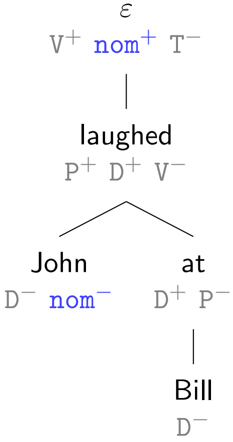
Movement is Local over Tree Tiers



Movement is Local over Tree Tiers

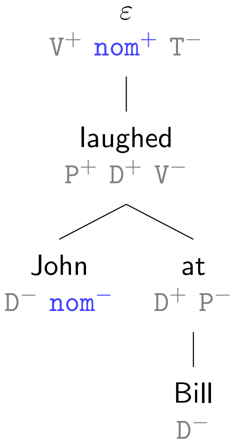
```
            laughed
           P+ D+ V-
          /        \
     John            at
  D- nom-          D+ P-
                     |
                    Bill
                     D-
```

Movement is Local over Tree Tiers

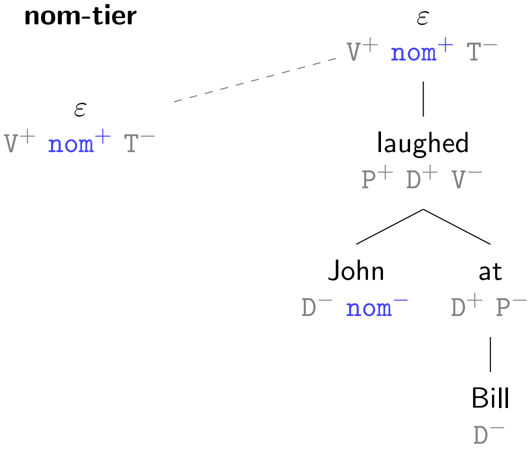


Movement is Local over Tree Tiers

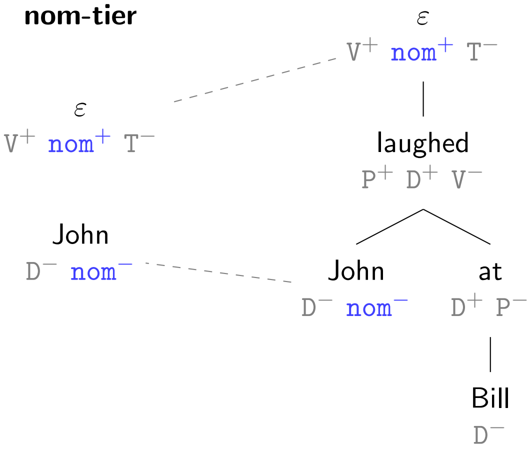
nom-tier



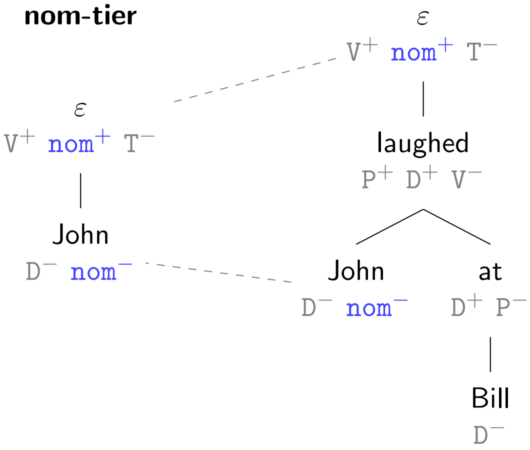
Movement is Local over Tree Tiers



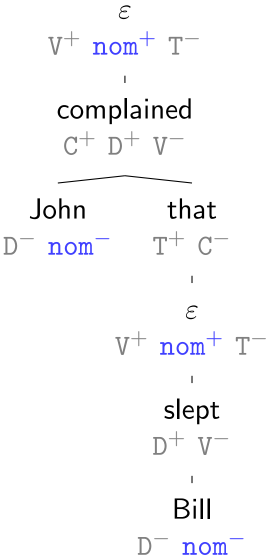
Movement is Local over Tree Tiers



Movement is Local over Tree Tiers

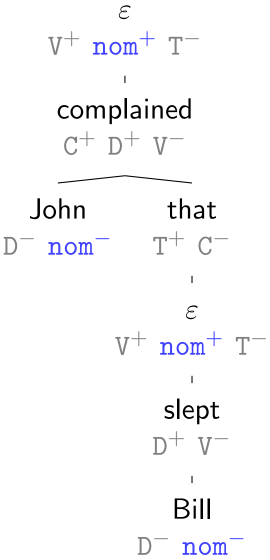


Tier with Multiple Movers



Tier with Multiple Movers

nom-tier

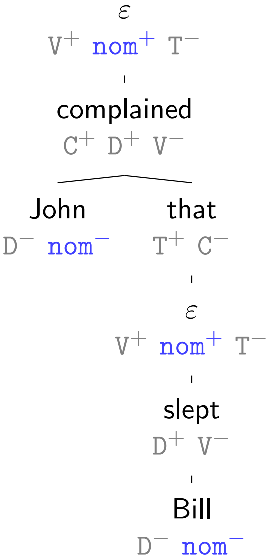


Tier with Multiple Movers

nom-tier

ε

$\texttt{V}^{+}\ \texttt{nom}^{+}\ \texttt{T}^{-}$



Tier with Multiple Movers

nom-tier

ε
$\mathtt{V}^{+}\ \mathtt{nom}^{+}\ \mathtt{T}^{-}$

John
$\mathtt{D}^{-}\ \mathtt{nom}^{-}$

ε
$\mathtt{V}^{+}\ \mathtt{nom}^{+}\ \mathtt{T}^{-}$

complained
$\mathtt{C}^{+}\ \mathtt{D}^{+}\ \mathtt{V}^{-}$

John
$\mathtt{D}^{-}\ \mathtt{nom}^{-}$

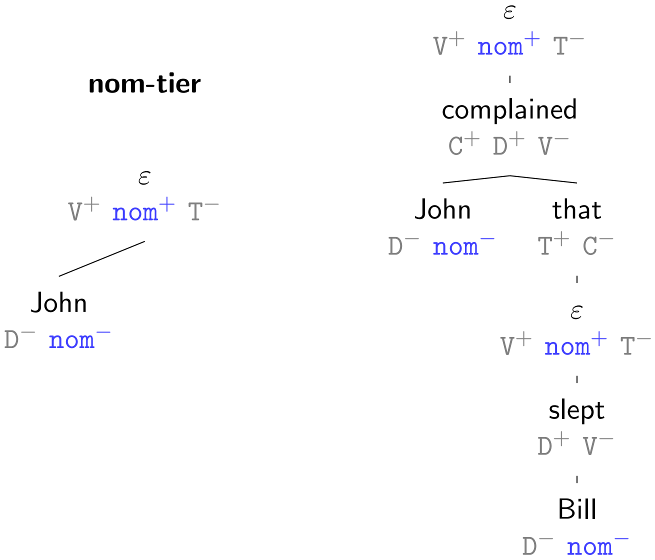
that
$\mathtt{T}^{+}\ \mathtt{C}^{-}$

ε
$\mathtt{V}^{+}\ \mathtt{nom}^{+}\ \mathtt{T}^{-}$

slept
$\mathtt{D}^{+}\ \mathtt{V}^{-}$

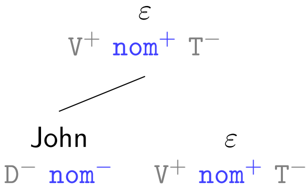
Bill
$\mathtt{D}^{-}\ \mathtt{nom}^{-}$

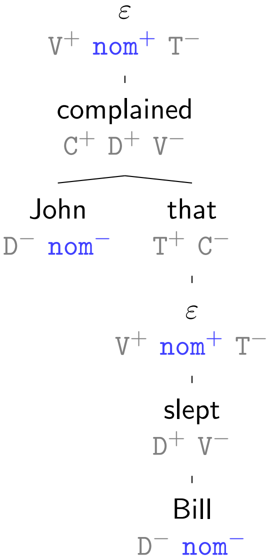
Tier with Multiple Movers



Tier with Multiple Movers

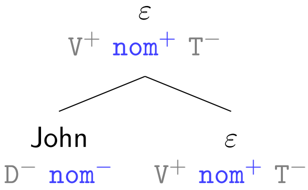
nom-tier

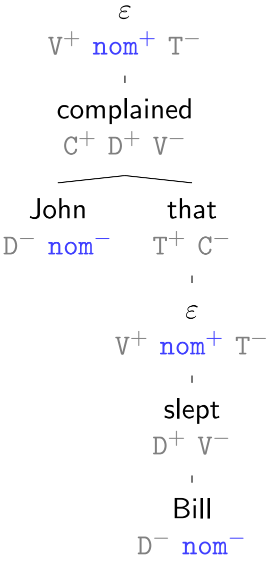




Tier with Multiple Movers

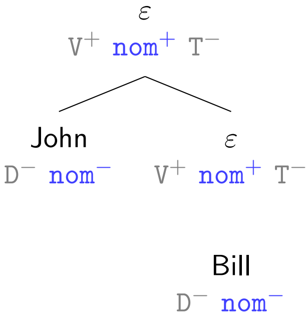
nom-tier

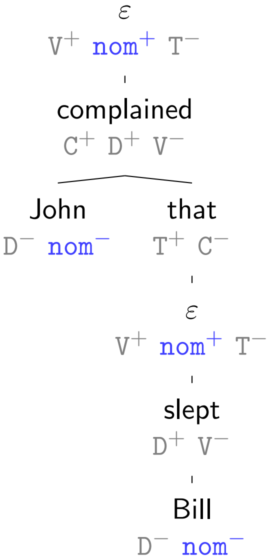




Tier with Multiple Movers

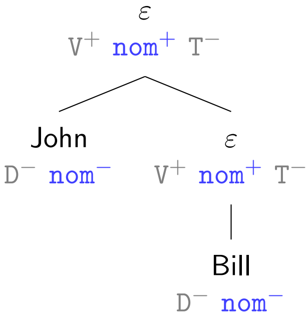
nom-tier

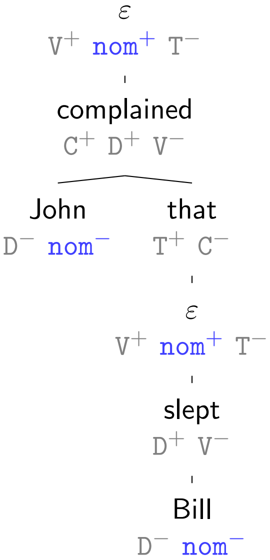




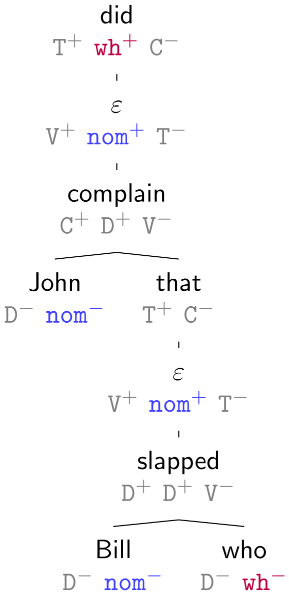
Tier with Multiple Movers

nom-tier



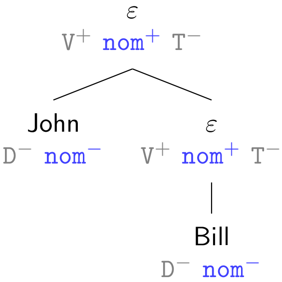


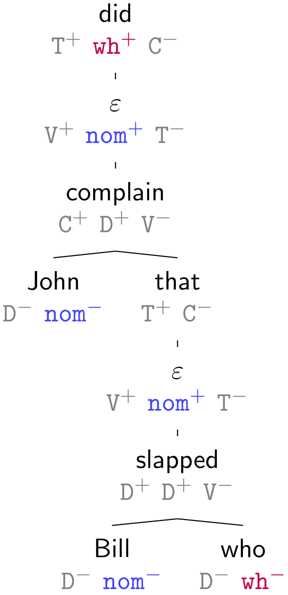
Separate Tier for Each Movement Type



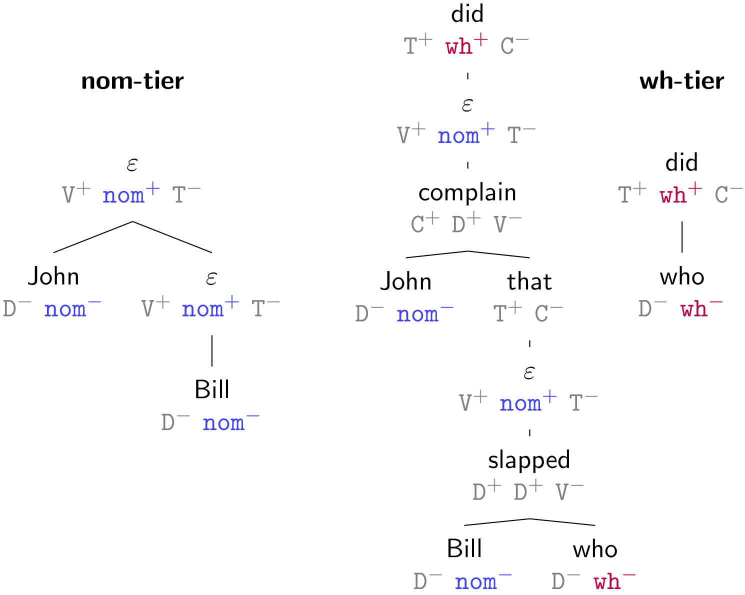
Separate Tier for Each Movement Type

nom-tier





Separate Tier for Each Movement Type



Move is TSL-2

- We now know how to construct movement tiers.
- Licit movement only creates tiers of a specific shape.
- **Move is TSL-2 over trees**:
  1. Every $\mathtt{f}^-$ must have an $\mathtt{f}^+$ mother.
  2. Every $\mathtt{f}^+$ has exactly one $\mathtt{f}^-$ among its daughters.

Cognitive parallelism

	Phonology	Syntax
SL	local dependencies	Merge
TSL	non-local dependencies	Move

Move is TSL-2

- We now know how to construct movement tiers.
- Licit movement only creates tiers of a specific shape.
- **Move is TSL-2 over trees**:
  1. Every $\mathtt{f}^-$ must have an $\mathtt{f}^+$ mother.
  2. Every $\mathtt{f}^+$ has exactly one $\mathtt{f}^-$ among its daughters.

Cognitive parallelism

	Phonology	**Syntax**
SL	local dependencies	Merge
TSL	non-local dependencies	Move

Excursus: making Merge more general

- Dependency format compatible with n-ary Merge
  Merge(laughed,John, at) instead of
  Merge(John, Merge(laughed, at))
- Later on, we'll be able to even accommodate unbounded Merge for flat coordination
  perhaps an evolutionary step between Merge and Move?
- Category & selector features might be inferable from context and thus redundant

Islands Come for Free

Two Fundamental Questions of Syntax

- Why do islands exist?
- Why do island exceptions exist?

A computational argument

1. Movement requires the power of TSL-2.
2. TSL-2 can model islands as blocking effects.
3. The cognitive ability for movement
   entails the cognitive ability for islands.

Islands Examples

(2) * Which car did John complain [**because** Bill damaged t].

(3) * Which car did John deny [the **fact** that Bill damaged t].

(4) Which car did John drive Mary crazy [**while** trying to fix t].

Islands Examples

(2) * Which car did John complain [**because** Bill damaged t].

(3) * Which car did John deny [the **fact** that Bill damaged t].

(4) Which car did John drive Mary crazy [**while** trying to fix t].

did
$\mathtt{T}^{+}$ $\mathtt{wh}^{+}$ $\mathtt{C}^{-}$

|

because
$\mathtt{T}^{+}$ $\mathtt{V}^{\sim}$

|

which
$\mathtt{D}^{-}$ $\mathtt{wh}^{-}$

Islands Examples

(2) * Which car did John complain [**because** Bill damaged t].

(3) * Which car did John deny [the **fact** that Bill damaged t].

(4) Which car did John drive Mary crazy [**while** trying to fix t].

did	did
$\mathtt{T}^{+}$ $\mathtt{wh}^{+}$ $\mathtt{C}^{-}$	$\mathtt{T}^{+}$ $\mathtt{wh}^{+}$ $\mathtt{C}^{-}$
|	|
because	fact
$\mathtt{T}^{+}$ $\mathtt{V}^{\sim}$	$\mathtt{C}^{+}$ $\mathtt{N}^{-}$
|	|
which	which
$\mathtt{D}^{-}$ $\mathtt{wh}^{-}$	$\mathtt{D}^{-}$ $\mathtt{wh}^{-}$

Islands Examples

(2) * Which car did John complain [**because** Bill damaged t].

(3) * Which car did John deny [the **fact** that Bill damaged t].

(4) Which car did John drive Mary crazy [**while** trying to fix t].

did $\mathtt{T}^{+}\ \mathtt{wh}^{+}\ \mathtt{C}^{-}$	did $\mathtt{T}^{+}\ \mathtt{wh}^{+}\ \mathtt{C}^{-}$	did $\mathtt{T}^{+}\ \mathtt{wh}^{+}\ \mathtt{C}^{-}$
|	|	|
because $\mathtt{T}^{+}\ \mathtt{V}^{\sim}$	fact $\mathtt{C}^{+}\ \mathtt{N}^{-}$	which $\mathtt{D}^{-}\ \mathtt{wh}^{-}$
|	|	
which $\mathtt{D}^{-}\ \mathtt{wh}^{-}$	which $\mathtt{D}^{-}\ \mathtt{wh}^{-}$	

Impossible Islands

- Islands arise when a blocker is projected onto a tier.
- Tier projection only considers lexical item itself,
  not its structural context
- TSL-2 theory of islands hence rules out:
  - **Gang-up islands**
    "A mover can escape n islands, but not more than that."
  - **Configurational islands**
    "An adjunct is an island iff it is inside an embedded clause."
  - **Cowardly islands**
    "An adjunct is an island iff
    there are at least two adjuncts in the clause."
  - **Rationed islands**
    "Only one adjunct per clause can be an island."
  - **Discerning islands**
    "Adjuncts only block movers that contain an adjective."

Recap: Core concepts

- Theory of syntax $\approx$ theory of computational locality

  SL strictly local (substructure of fixed size)

  TSL strictly local over tiers

- Tiers are abstract, don't reify them!
- Derivation as fundamental syntactic representation
- Diacritics (not features!) for encoding dependencies

Recap: Empirical insights

- **Cognitive parallelism**

	Phonology	**Syntax**
SL	local dependencies	Merge
TSL	non-local dependencies	Move

- blocking $\approx$ islands
- Islands...
  - arise naturally from movement,
  - are correctly predicted to vary across languages,
  - are correctly predicted to be typologically limited.

Acknowledgments

This work is supported by the National Science Foundation under Grant No. BCS-1845344.

References I

Applegate, Richard B. 1972. *Ineseño Chumash grammar*. Doctoral Dissertation, University of California, Berkeley.

Goldsmith, John. 1976. *Autosegmental phonology*. Doctoral Dissertation, MIT.

Heinz, Jeffrey, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 58–64. URL `http://www.aclweb.org/anthology/P11-2011`.

Jurgec, Peter. 2011. *Feature spreading 2.0: A unified theory of assimilation*. Doctoral Dissertation, University of Tromsø.

McMullin, Kevin. 2016. *Tier-based locality in long-distance phonotactics: Learnability and typology*. Doctoral Dissertation, University of British Columbia.

Stabler, Edward P. 1997. Derivational Minimalism. In *Logical aspects of computational linguistics*, ed. Christian Retoré, volume 1328 of *Lecture Notes in Computer Science*, 68–95. Berlin: Springer. URL `https://doi.org/10.1007/BFb0052152`.

Stabler, Edward P. 2011. Computational perspectives on Minimalism. In *Oxford handbook of linguistic Minimalism*, ed. Cedric Boeckx, 617–643. Oxford: Oxford University Press.

Syntactic tiers for movement and agreement

Day 2: A whirlwind tour of syntactic phenomena

Thomas Graf

Stony Brook University
`mail@thomasgraf.net`
`https://thomasgraf.net`

KU Leuven Lecture Series
Decmber 5–7, 2023

A little less conversation, a little more action please

- We now have a good idea of how syntactic tiers work.
- Let's apply them to some phenomena!

Outline

Capturing variation

- Island effects vary in unexpected ways:
  - across languages
  - across lexical items
- Syntactic tiers can handle both because the tier projection can determine for each lexical item whether it projects.
- **Flip side**: need a story for why island constraints are still pretty uniform across languages

Accounting for gradience

- There's a **huge debate about gradience** in island effects.
  - **Option 1**: island constraints are purely a performance phenomenon
  - **Option 2**: islands are in the grammar, gradience is a performance phenomenon
  - **Option 3**: island are gradient constraints in the grammar
- Syntactic tiers are compatible with all three options.
- In particular, syntactic tiers make it easy to combine gradience with standard analyses.
- How? **Weighted/Probabilistic tier projection**

Gradience

- Tier projection can be made probabilistic to capture gradience (Mayer 2021; Torres et al. 2023)

Calculating gradience with probabilistic wh-tier

1. construct all possible versions of the wh-tier
2. filter out illicit tiers
3. sum up probabilities of remaining tiers

Example calculation

(1) * Who did Bill complain after having a meeting with ⟨who⟩ because he's always angry.

Item	Probability
any $\mathtt{wh}^+$	1
any $\mathtt{wh}^-$	1
because	.9
after	.8

- **Possible tiers**
  1. Project $\mathtt{wh}^+$, $\mathtt{wh}^-$, because, after: .72, illicit
  2. Project $\mathtt{wh}^+$, $\mathtt{wh}^-$, because: .18, licit
  3. Project $\mathtt{wh}^+$, $\mathtt{wh}^-$, after: .08, illicit
  4. Project $\mathtt{wh}^+$, $\mathtt{wh}^-$: .02, licit
- Overall probability: .2

Probe horizons VS the Ban on Improper Movement

- Tier-based story for islands also captures probe horizons (Keine 2016, 2019)

Example

- **Probe horizons**: A-movement features cannot probe into CPs
- **Syntactic tiers**: all C-heads project onto all A-movement tiers

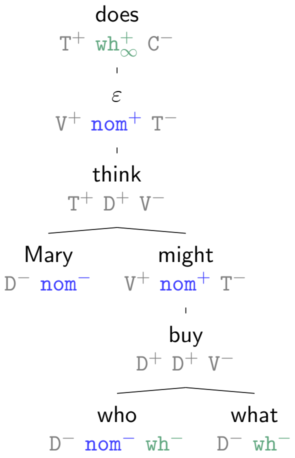
- Syntactic tiers also show us that probe horizons are preferable to the Ban on Improper Movement (BoIM).

Ruling out improper movement via...

Probe horizons : no new tiers needed

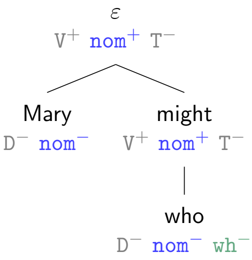
BoIM : up to $\frac{n^2-n}{2}$ new tiers
(where n is the number of movement types)

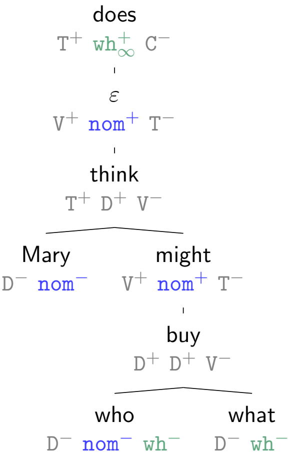
Multiple wh-movement is also TSL



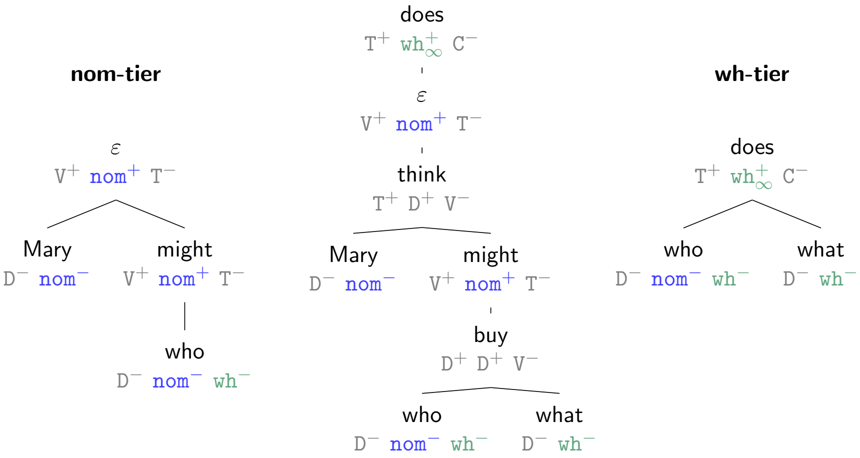
Multiple wh-movement is also TSL

nom-tier

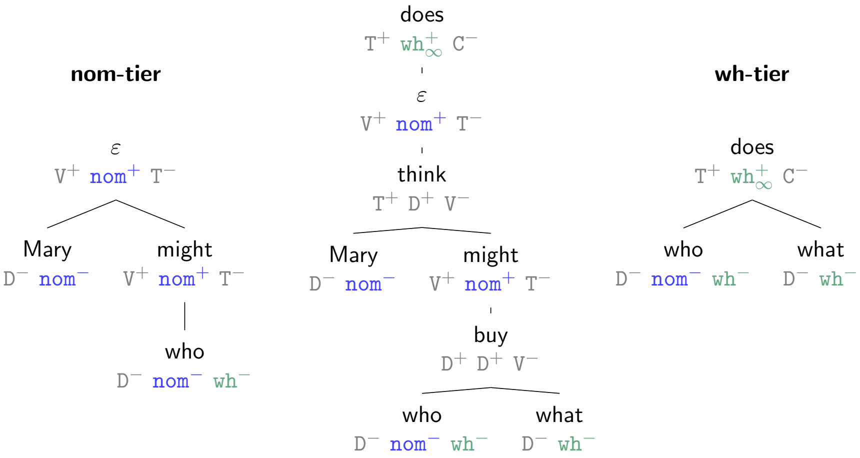




Multiple wh-movement is also TSL



Multiple wh-movement is also TSL



Persistent feature checking as a constraint on tiers

On each $\texttt{f}$-tier, every $\texttt{f}^{+}_{\infty}$ has **at least 1** $\texttt{f}^{-}$ among its daughters.

Multiple wh-movement is not special

From the TSL-perspective, multiple wh-movement is already part of standard movement:

- Standard movement: every $\mathtt{f}^+$ has exactly one $\mathtt{f}^-$ daughter
- **TSL:** "exactly one" $=$ "at least one" $+$ "at most one"

multiple wh-movement standard movement - "at most one"

standard movement multiple wh-movement $+$ "at most one"

Work to be done...

That's a nice starting point, but we need to account for...

- why there is no multiple nom-movement
- linearization,
- superiority effects/c-command,
- multiple movers with distinct targets,
- the 2-by-2 typology of wh-movement

Multiple wh	Superiority	Example language
-	-	Spanish
-	+	English
+	-	Russian
+	+	Bulgarian

Superiority effects

- Some languages have a limited superiority effect: the highest wh-mover must be first in the wh-cluster
- This is just a distributional constraint on `wh` VS $\texttt{wh}_{\infty}$
- That would immediately tell us why superiority and multiple-wh are independent parameters

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each $\mathtt{f}$-tier

1. Every $\mathtt{f}^+$ has exactly 1 $\mathtt{f}^-$ among its daughters.
2. Every $\mathtt{f}^-$ has $\mathtt{f}^+$ as its mother.

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each `f`-tier

1. Every **Positive** has exactly 1 **Negative** among its daughters.
2. Every **Negative** has **Positive** as its mother.

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each `f`-tier

1. Every **Positive** has exactly 1 **Negative** among its daughters.
2. Every **Negative** has **Positive** as its mother.

	Positive on `f`-tier	**Negative** on `f`-tier
$\mathtt{f}^+$	True	False
$\mathtt{f}^-$	False	True

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each `f`-tier

1. Every **Positive** has exactly 1 **Negative** among its daughters.
2. Every **Negative** has **Positive** as its mother.

	Positive on `f`-tier	**Negative** on `f`-tier
$\texttt{f}^{+}$	True	False
$\texttt{f}^{-}$	False	True
	False	False

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each `f`-tier

1. Every **Positive** has exactly 1 **Negative** among its daughters.
2. Every **Negative** has **Positive** as its mother.

	Positive on `f`-tier	**Negative** on `f`-tier
$\mathtt{f}^{+}$	True	False
$\mathtt{f}^{-}$	False	True
	False	False
	True	True

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each `f`-tier

1. Every **Positive** has exactly 1 **Negative** among its daughters.
2. Every **Negative** has **Positive** as its mother.

	Positive on `f`-tier	**Negative** on `f`-tier
$\mathtt{f}^{+}$	True	False
$\mathtt{f}^{-}$	False	True
islands	False	False
	True	True

A 2-by-2 typology of tier nodes

Feature checking as a constraint on each `f`-tier

1. Every **Positive** has exactly 1 **Negative** among its daughters.
2. Every **Negative** has **Positive** as its mother.

	Positive on `f`-tier	**Negative** on `f`-tier
$\mathtt{f}^{+}$	True	False
$\mathtt{f}^{-}$	False	True
islands	False	False
wh-agreement	True	True

Islands as tier blockers (neither positive nor negative)

(2) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?

did :: $\mathtt{T}^+\mathtt{wh}^+\mathtt{C}^-$

ε :: $\mathtt{V}^+\mathtt{nom}^+\mathtt{T}^-$

complain :: $\mathtt{P}^+\mathtt{D}^+\mathtt{V}^-$

John :: $\mathtt{D}^-$ $\{\mathtt{nom}^-\}$ about :: $\mathtt{D}^+\mathtt{P}^-$

the :: $\mathtt{N}^+\mathtt{D}^-$

fact :: $\mathtt{C}^+\mathtt{N}^-$

that :: $\mathtt{T}^+\mathtt{C}^-$

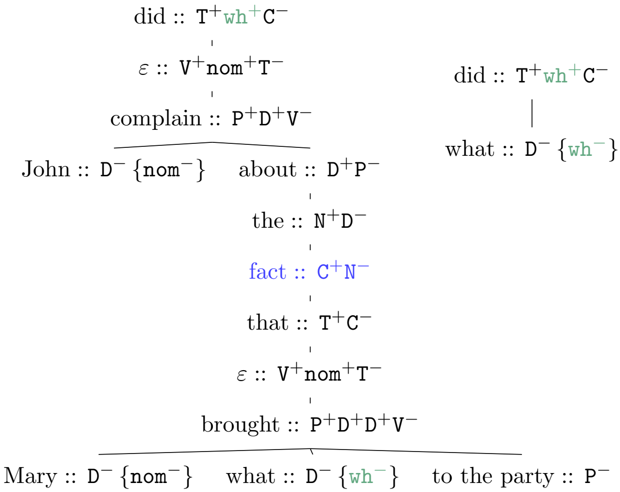
ε :: $\mathtt{V}^+\mathtt{nom}^+\mathtt{T}^-$

brought :: $\mathtt{P}^+\mathtt{D}^+\mathtt{D}^+\mathtt{V}^-$

Mary :: $\mathtt{D}^-$ $\{\mathtt{nom}^-\}$ what :: $\mathtt{D}^-$ $\{\mathtt{wh}^-\}$ to the party :: $\mathtt{P}^-$

Islands as tier blockers (neither positive nor negative)

(2) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?



Islands as tier blockers (neither positive nor negative)

(2) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?

did :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$

ε :: $\mathtt{V}^{+}\mathtt{nom}^{+}\mathtt{T}^{-}$

complain :: $\mathtt{P}^{+}\mathtt{D}^{+}\mathtt{V}^{-}$

John :: $\mathtt{D}^{-}$ {$\mathtt{nom}^{-}$} | about :: $\mathtt{D}^{+}\mathtt{P}^{-}$

the :: $\mathtt{N}^{+}\mathtt{D}^{-}$

fact :: $\mathtt{C}^{+}\mathtt{N}^{-}$

that :: $\mathtt{T}^{+}\mathtt{C}^{-}$

ε :: $\mathtt{V}^{+}\mathtt{nom}^{+}\mathtt{T}^{-}$

brought :: $\mathtt{P}^{+}\mathtt{D}^{+}\mathtt{D}^{+}\mathtt{V}^{-}$

Mary :: $\mathtt{D}^{-}$ {$\mathtt{nom}^{-}$} | what :: $\mathtt{D}^{-}$ {$\mathtt{wh}^{-}$} | to the party :: $\mathtt{P}^{-}$

did :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$

\+ −

what :: $\mathtt{D}^{-}$ {$\mathtt{wh}^{-}$}

Islands as tier blockers (neither positive nor negative)

(2) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?

did :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$

ε :: $\mathtt{V}^{+}\mathtt{nom}^{+}\mathtt{T}^{-}$

complain :: $\mathtt{P}^{+}\mathtt{D}^{+}\mathtt{V}^{-}$

John :: $\mathtt{D}^{-}$ {$\mathtt{nom}^{-}$}   about :: $\mathtt{D}^{+}\mathtt{P}^{-}$

the :: $\mathtt{N}^{+}\mathtt{D}^{-}$

fact :: $\mathtt{C}^{+}\mathtt{N}^{-}$

that :: $\mathtt{T}^{+}\mathtt{C}^{-}$

ε :: $\mathtt{V}^{+}\mathtt{nom}^{+}\mathtt{T}^{-}$

brought :: $\mathtt{P}^{+}\mathtt{D}^{+}\mathtt{D}^{+}\mathtt{V}^{-}$

Mary :: $\mathtt{D}^{-}$ {$\mathtt{nom}^{-}$}   what :: $\mathtt{D}^{-}$ {$\mathtt{wh}^{-}$}   to the party :: $\mathtt{P}^{-}$

did :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$

+ −

what :: $\mathtt{D}^{-}$ {$\mathtt{wh}^{-}$}

did :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$

+

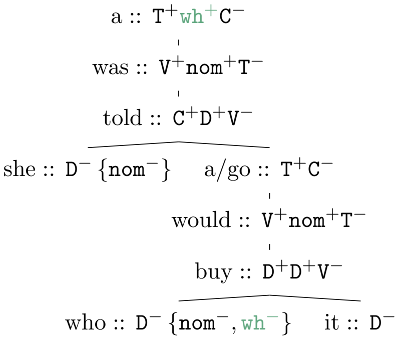
fact :: $\mathtt{C}^{+}\mathtt{N}^{-}$

−

what :: $\mathtt{D}^{-}$ {$\mathtt{wh}^{-}$}

Wh-agreement via tier conduits (positive & negative)

(3) Cé **a**/*go dúradh léithi **a**/*go cheannódh é?
who C-wh/C was-said with-her C-wh/C would-buy it
'Who was she told would buy it?'
(Irish, McCloskey 2001:p.94)



Wh-agreement via tier conduits (positive & negative)

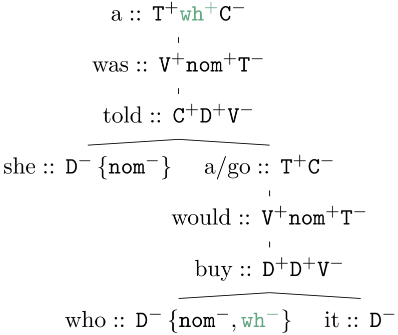
(3) Cé **a**/*go dúradh léithi **a**/*go cheannódh é?
who C-wh/C was-said with-her C-wh/C would-buy it
'Who was she told would buy it?'
(Irish, McCloskey 2001:p.94)

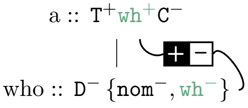
- a :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$
  - was :: $\mathtt{V}^{+}\mathtt{nom}^{+}\mathtt{T}^{-}$
    - told :: $\mathtt{C}^{+}\mathtt{D}^{+}\mathtt{V}^{-}$
      - she :: $\mathtt{D}^{-}\ \{\mathtt{nom}^{-}\}$
      - a/go :: $\mathtt{T}^{+}\mathtt{C}^{-}$
        - would :: $\mathtt{V}^{+}\mathtt{nom}^{+}\mathtt{T}^{-}$
          - buy :: $\mathtt{D}^{+}\mathtt{D}^{+}\mathtt{V}^{-}$
            - who :: $\mathtt{D}^{-}\ \{\mathtt{nom}^{-}, \mathtt{wh}^{-}\}$
            - it :: $\mathtt{D}^{-}$

- a :: $\mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$
  - who :: $\mathtt{D}^{-}\ \{\mathtt{nom}^{-}, \mathtt{wh}^{-}\}$

Wh-agreement via tier conduits (positive & negative)

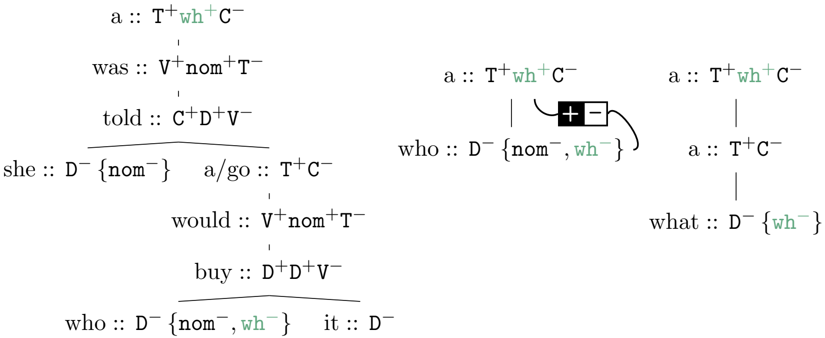
(3) Cé **a**/*go dúradh léithi **a**/*go cheannódh é?
who C-wh/C was-said with-her C-wh/C would-buy it
'Who was she told would buy it?'
(Irish, McCloskey 2001:p.94)





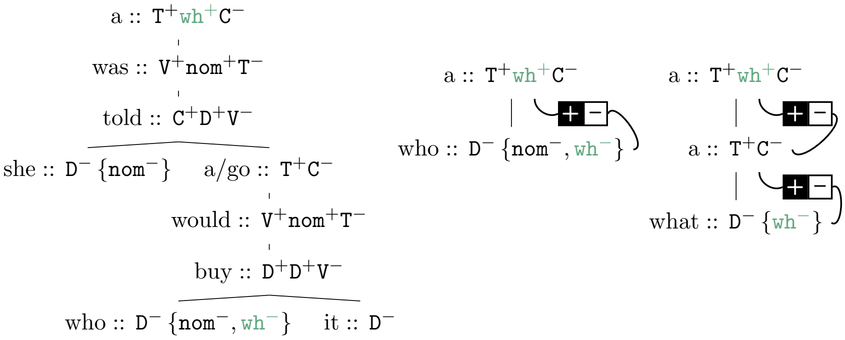
Wh-agreement via tier conduits (positive & negative)

(3) Cé **a**/*go dúradh léithi **a**/*go cheannódh é?
who C-wh/C was-said with-her C-wh/C would-buy it
'Who was she told would buy it?'
(Irish, McCloskey 2001:p.94)



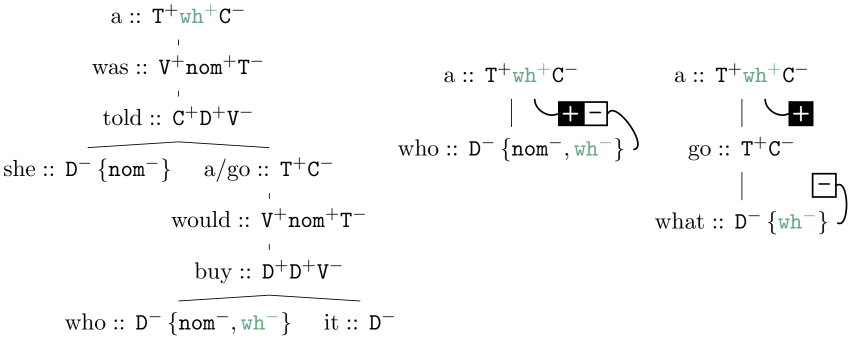
Wh-agreement via tier conduits (positive & negative)

(3) Cé **a**/*go dúradh léithi **a**/*go cheannódh é?
who C-wh/C was-said with-her C-wh/C would-buy it
'Who was she told would buy it?'
(Irish, McCloskey 2001:p.94)



Wh-agreement via tier conduits (positive & negative)

(3) Cé **a**/*go dúradh léithi **a**/*go cheannódh é?
who C-wh/C was-said with-her C-wh/C would-buy it
'Who was she told would buy it?'
(Irish, McCloskey 2001:p.94)



Extraction morphology: An example from Wolof u-chains

Extraction morphology
morphology conditioned by the presence of movement

- **Wolof u-chains** (Niger-Congo)
  - (covert) wh-phrase moves to matrix Spec,CP
  - highest C-head *u* along the movement path must agree with wh-phrase in Class
  - intermediate C-heads may agree with wh-phrase in Class

(4) [ε **k**-u Kumba wax [ne **k**/**l**-u Isaa defe [
Q CL-u Kumba say FRC CL/EXPL-u Isaa think
ne **k**/**l**-u Maryam dóór *t*]]]
FRC CL/EXPL-u Maryam hit
'Who did Kumba say that Isaa thought that Maryam hit?'
(Torrence 2012:1171)

Why care about extraction morphology?

- **Highly relevant**
  poster child for **successive cyclic movement**
- **Good case study**
  data is robust, varied, typologically diverse

A typology of extraction morphology

1. What is agreeing?
   - T **T**arget of movement (Wolof u-chains)
   - O some **O**ther head in the clause of the landing site (Duala *no*-marking on T)
2. What clauses display agreement?
   - F the clause of the **F**inal landing site (Chamorro C-agreement)
   - N clauses with **N**on-final landing sites (Kiitharaka focus marking on verb)
3. agreement is. . .
   - X mandatory for X
   - (X) optional for X

based on Georgi (2017), a terrific paper

Example

Wolof u-chains are TF(N).

Extraction morphology and tiers

- Tiers work well for extraction morphology.
- The various options fall out nicely for T-patterns.

Central idea

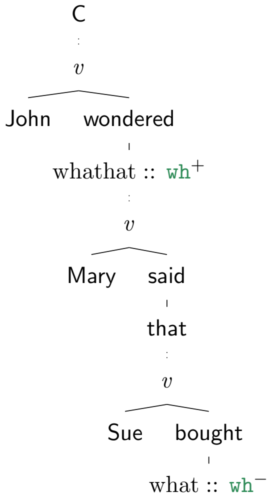
- treat it as a distribution problem (not mapping/spell-out)
- use tiers to ensure correct distribution of agreeing forms

Two simplifications due to data gaps

- exactly one A′-mover per sentence
- only one type of A′-movement per sentence
  (no *which car that John bought did we trash*)

Toy example 1: TF English *whathat*

- Suppose English had a special C-head *whathat*, which has to be the final landing site of some wh-mover.



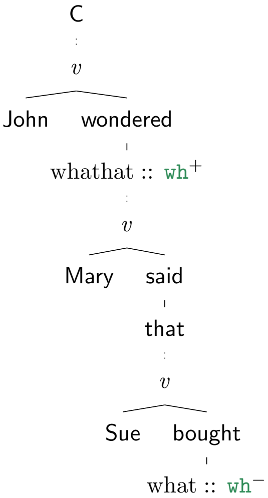
Analysis: Lexical accident

The lexicon happens to be such that

- *whathat* must carry $\mathtt{wh}^+$,
- *that* must not carry $\mathtt{wh}^+$,
- empty C must not carry $\mathtt{wh}^+$.

Toy example 2: TF English *whathat*/*whothat*

- Suppose English also had *whothat*,
  and targeted C must agree in animacy with the `wh`-mover.



Lexical accident + tier match

- `wh`$^+$ only on *whathat* and *whothat*
- on `wh`-tier, `wh`$^+$-head must match animacy of `wh`$^-$-daughter
  wh-tier: Xthat :: `wh`$^+$ X :: `wh`$^-$

Toy example 2: TF English *whathat*/*whothat*

- Suppose English also had *whothat*,
  and targeted C must agree in animacy with the wh-mover.

wh-tier

whathat :: $\mathtt{wh}^{+}$

what :: $\mathtt{wh}^{-}$

Lexical accident + tier match

- $\mathtt{wh}^{+}$ only on *whathat* and *whothat*
- on wh-tier, $\mathtt{wh}^{+}$-head must match animacy of $\mathtt{wh}^{-}$-daughter
  wh-tier: Xthat :: $\mathtt{wh}^{+}$ X :: $\mathtt{wh}^{-}$

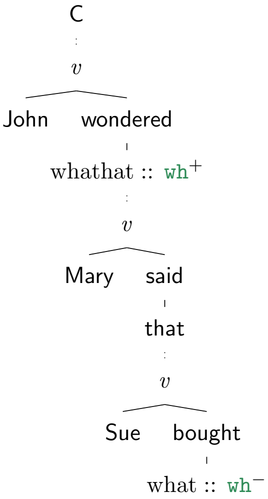
Toy example 3: TFN English *whathat*/*whothat*

- Suppose every C-head along a wh-movement path has to be *whathat*/*whothat*

Lexical accident + EM-tier

- $\mathtt{wh}^+$ only on *whathat* and *whothat*
- **EM-tier**
  - project all $\mathtt{wh}^+$, all $\mathtt{wh}^-$, and all C-heads
  - every *Xthat* must have daughter with *Xthat* (no $\mathtt{wh}^+$) or matching $\mathtt{wh}^-$
  - non-agreeing C above *Xthat*: *Xthat* must carry $\mathtt{wh}^+$

EM-tier: $\mathrm{Xthat} :: \mathtt{wh}^+\ \mathrm{Xthat}^*$
$\mathrm{X} :: \mathtt{wh}^-$



Toy example 3: TFN English *whathat*/*whothat*

- Suppose every C-head along a wh-movement path has to be *whathat*/*whothat*

Lexical accident + EM-tier

- $\mathtt{wh}^+$ only on *whathat* and *whothat*
- **EM-tier**
  - project all $\mathtt{wh}^+$, all $\mathtt{wh}^-$, and all C-heads
  - every *Xthat* must have daughter with *Xthat* (no $\mathtt{wh}^+$) or matching $\mathtt{wh}^-$
  - non-agreeing C above *Xthat*: *Xthat* must carry $\mathtt{wh}^+$

EM-tier: $\text{Xthat} :: \mathtt{wh}^+ \ \text{Xthat}^*$

$\text{X} :: \mathtt{wh}^-$

wh-tier

$\text{whathat} :: \mathtt{wh}^+$

$\text{what} :: \mathtt{wh}^-$

Toy example 3: TFN English *whathat*/*whothat*

- Suppose every C-head along a wh-movement path has to be *whathat*/*whothat*

N-tier

C

whathat :: $\mathtt{wh}^+$

that

what :: $\mathtt{wh}^-$

Lexical accident + EM-tier

- $\mathtt{wh}^+$ only on *whathat* and *whothat*
- **EM-tier**
  - project all $\mathtt{wh}^+$, all $\mathtt{wh}^-$, and all C-heads
  - every *Xthat* must have daughter with *Xthat* (no $\mathtt{wh}^+$) or matching $\mathtt{wh}^-$
  - non-agreeing C above *Xthat*: *Xthat* must carry $\mathtt{wh}^+$

EM-tier: $\mathrm{Xthat} :: \mathtt{wh}^+\ \mathrm{Xthat}^*$

$\mathrm{X} :: \mathtt{wh}^-$

Toy example 4: T(F)(N) English *whathat*/*whothat*

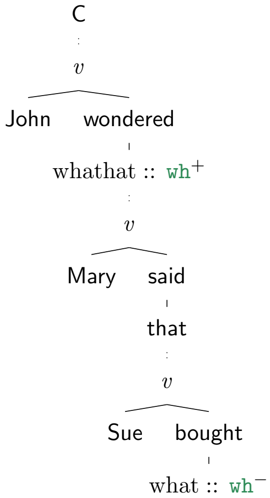
- Suppose as before, except that *Xthat* is optional

No accident + EM-tier with Xthat only

- $\texttt{wh}^+$ on *Xthat* **or that**
- **EM-tier**
  - project all $\texttt{wh}^+$, all $\texttt{wh}^-$, and all C-heads **that are Xthat**
  - every *Xthat* must have daughter with *Xthat* (no $\texttt{wh}^+$) or matching $\texttt{wh}^-$
  - non-agreeing C above *Xthat*: *Xthat* must carry $\texttt{wh}^+$

EM-tier: (X)that :: $\texttt{wh}^+$ Xthat*

X :: $\texttt{wh}^-$



Toy example 4: T(F)(N) English *whathat*/*whothat*

- Suppose as before, except that *Xthat* is optional

N-tier

whathat :: $\mathtt{wh}^+$

what :: $\mathtt{wh}^-$

No accident + EM-tier with Xthat only

- $\mathtt{wh}^+$ on *Xthat* **or that**
- **EM-tier**
  - project all $\mathtt{wh}^+$, all $\mathtt{wh}^-$, and all C-heads **that are Xthat**
  - every *Xthat* must have daughter with *Xthat* (no $\mathtt{wh}^+$) or matching $\mathtt{wh}^-$
  - non-agreeing C above *Xthat*: *Xthat* must carry $\mathtt{wh}^+$

EM-tier: (X)that :: $\mathtt{wh}^+$ Xthat*

X :: $\mathtt{wh}^-$

Summary of analytical tricks for T-patterns

- **F-patterns arise from lexicon**
  - no F: `wh`$^+$ only on default C
  - F: `wh`$^+$ only on agreeing C
  - (F): `wh`$^+$ on either one
- **N-patterns captured via tiers**
  - construct **daisy chain** of agreeing mother-daughter configurations
  - top of daisy chain: `f`$^+$ (don't agree beyond the landing site)
  - bottom of daisy chain: `f`$^-$ (don't agree below the base position)
  - optionality = not projecting uninflected heads
- If we need agreement with some feature of the mover, find a suitable tier to enforce the match condition on.

A brief look at an O pattern

- **Ewe (Niger-Congo)** is OF(N)
- 3rd person subject pronouns **é** and **wò** alternate based on wh-movement
- **é** is default
- if some XP wh-moves across Spec,TP, then
  - **é** **must** become **wò** in the clause with the final landing site,
  - **é** **may** become **wò** in a clause with a non-final landing site.

(5) $[_{CP}$ Meka-e$_k$ **wò**/***é** gblɔ $[_{CP}$ be **wò**/**é**-bu $[_{CP}$ be **wò**/**é**-fò t_k]]]?
who-Foc he say that he-think that he-hit
'Who did he$_i$ say that he$_j$ thinks that he$_m$ hit?'

Reanalyzing Ewe as a TF(N) pattern

- Successive-cyclic movement cannot explain Ewe alternations, but it's **very natural with tiers**
- Suppose Ewe has two T-heads: **default T** and **wh-agreeing $\mathbf{T}_{\texttt{wh}}$**
- Additional constraint on $\texttt{nom}$-tier: $\mathrm{T}_{\texttt{wh}} :: \texttt{nom}^+$ must not have $\acute{\mathrm{e}} :: \texttt{nom}^-$ as a daughter
- Distribution of $\mathrm{T}_{\texttt{wh}}$ almost the same as *Xthat* in toy example 4
  EM-tier: $\mathrm{C} :: \texttt{wh}^+ \ \mathrm{T}^*_{wh} \ \mathrm{X} :: \texttt{wh}^-$
- We need a second tier to enforce the F pattern:
  1. Project all $\texttt{wh}^+$, $\texttt{wh}^-$, and all T heads
  2. $\texttt{wh}^+$ must not have default T as daughter

  F-tier: $\mathrm{C} :: \texttt{wh}^+ \ \mathrm{T}_{wh} \ \{\mathrm{T}, \mathrm{T}_{wh}\}^* \ \mathrm{X} :: \texttt{wh}^-$

Another case: **no**-marking in Duala

- **Duala (Niger-Congo)** is OF
- A′-movement of object or adjunct triggers insertion of **no** after finite verb in T
- same analysis as Ewe, except that:
  - the mover must not carry $\mathtt{nom}^-$, and
  - we do not allow $\mathrm{T}_{\mathtt{wh}}$ as a daughter of $\mathrm{T}_{\mathtt{wh}}$

EM-tier: $\mathrm{C} :: \mathtt{wh}^+ \; \mathsf{T}_{wh} \; \mathrm{X} :: \mathtt{wh}^-$

F-tier: $\mathrm{C} :: \mathtt{wh}^+ \; \mathsf{T}_{wh} \; \mathsf{T}^* \; \mathrm{X} :: \mathtt{wh}^-$

An extraction morphology surprise: floating quantifiers!

(6) These kids may (all) have (all) been (all) believed to (all) t like candy.

- Languages vary greatly in
  - what quantifiers may float
  - what movements license quantifier float
  - in what positions quantifiers may float
  - whether floating quantifiers exhibit agreement
- **Tiers capture all that variation!**

Sketch of analysis

- Lexicon contains $\mathbf{QX}_{\mathtt{f}} :: \mathbf{P}^{+}\mathbf{P}^{-}$ for every
  - floating quantifier $\mathbf{Q}$ with agreement $\mathbf{X}$
  - movement type $\mathtt{f}$ that licenses quantifier float of $\mathbf{Q}$
  - position $\mathbf{P}$ where $\mathbf{Q}$ may float
- Distribution of floating quantifiers regulated via EM-tier

EM-tier: $\mathtt{f}^{+}\ \mathrm{QX}_{\mathtt{f}}^{*}\ \mathrm{X} :: \mathtt{f}^{-}$

An extraction morphology surprise: floating quantifiers!

(6) These kids may (all) have (all) been (all) believed to (all) t like candy.

- Languages vary greatly in
  - what quantifiers may float
  - what movements license quantifier float
  - in what positions quantifiers may float
  - whether floating quantifiers exhibit agreement
- **Tiers capture all that variation!**

Sketch of analysis

- Lexicon contains $\mathbf{QX}_{\mathtt{f}} :: \mathbf{P}^{+}\mathbf{P}^{-}$ for every
  - floating quantifier $\mathbf{Q}$ with agreement $\mathbf{X}$
  - movement type $\mathtt{f}$ that licenses quantifier float of $\mathbf{Q}$
  - position $\mathbf{P}$ where $\mathbf{Q}$ may float
- Distribution of floating quantifiers regulated via EM-tier

  EM-tier: $\mathtt{f}^{+}\ \mathrm{QX}_{\mathtt{f}}^{*}\ \mathrm{X} :: \mathtt{f}^{-}$

Acknowledgments

This work is supported by the National Science Foundation under Grant No. BCS-1845344.

References I

Georgi, Doreen. 2017. Patterns of movement reflexes as the result of the order of merge and agree. *Linguistic Inquiry* 48:585–626.

Keine, Stefan. 2016. *Probes and their horizons*. Doctoral Dissertation, University of Massachusetts, Amherst.

Keine, Stefan. 2019. Selective opacity. *Linguistic Inquiry* 50:13–62.

Mayer, Connor. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, 39–50.

McCloskey, James. 2001. The morphosyntax of wh-extraction in Irish. *Journal of Linguistics* 37:67–100.

Torrence, Harold. 2012. The morpho-syntax of silent *wh*-expressions in Wolof. *Natural Language and Linguistic Theory* 30:1147–1184.

Torres, Charles, Kenneth Hanson, Thomas Graf, and Connor Mayer. 2023. Modeling island effects with probabilistic tier-based strictly local grammars over trees. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*, pp–pp.

Syntactic tiers for movement and agreement

Day 3: Tree Rewriting & Externalization

Thomas Graf

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

KU Leuven Lecture Series
December 5–7, 2023

Outline

1. Subcategorization is too powerful
2. Feature recoverability as strictly local rewriting
3. Bare phrase structure: Local transductions for Merge and Move
4. The challenge of linearization

Take-home message

- **Overgeneration problem in syntax**
  Subcategorization can express very unnatural constraints, due to **category refinement**.
- **A linguistically fertile solution**
  Category features don't come for free.
  They must be inferable from the local context.

Hidden power of subcategorization

Every formalism with subcategorization can express **undesirable constraints**. (Graf 2017)

Counting	every DP contains at least five LIs
Symmetry closure	every reflexive c-commands its antecedent
Complement	sentence well-formed iff ill-formed in English
Boolean closure	sentence must obey either V2 or Principle A, unless there are less than 7 pronounced LIs
Domain blindness	a sentence is well-formed iff there are at least two words that display word-final devoicing
Is(n't)lands	an adjunct is an island iff it is inside an embedded clause or it contains no animate nouns

Why?

- Complex constraints can be lexicalized
  by decomposing them into refined categories.
- They are then enforced via subcategorization.
- It's a generalized version of slash feature percolation.
  (Gazdar et al. 1985; Graf 2011; Kobele 2011)

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$$
\begin{array}{rr}
\text{foo} :: \mathtt{X}^- & \text{foo} :: \mathtt{X}^+\mathtt{X}^- \\
\text{bar} :: \mathtt{X}^- & \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\
 & \varepsilon :: \mathtt{X}^+\mathtt{C}^-
\end{array}
$$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$$\begin{array}{rr} \text{foo} :: \mathtt{X}^- & \text{foo} :: \mathtt{X}^+\mathtt{X}^- \\ \text{bar} :: \mathtt{X}^- & \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ & \varepsilon :: \mathtt{X}^+\mathtt{C}^- \end{array}$$

$$\text{foo} :: \mathtt{X}^-$$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$$\text{foo} :: \mathtt{X}^- \qquad \text{foo} :: \mathtt{X}^+\mathtt{X}^-$$

$$\text{bar} :: \mathtt{X}^- \qquad \text{bar} :: \mathtt{X}^+\mathtt{X}^-$$

$$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$$

$$\text{bar} :: \mathtt{X}^+\mathtt{X}^-$$
$$|$$
$$\text{foo} :: \mathtt{X}^-$$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$\text{foo} :: \mathtt{X}^-$ $\quad$ $\text{foo} :: \mathtt{X}^+\mathtt{X}^-$

$\text{bar} :: \mathtt{X}^-$ $\quad$ $\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

|

$\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

|

$\text{foo} :: \mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$\text{foo} :: \mathtt{X}^-$

$\text{bar} :: \mathtt{X}^-$

$\text{foo} :: \mathtt{X}^+\mathtt{X}^-$

$\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

$$\begin{array}{c} \varepsilon :: \mathtt{X}^+\mathtt{C}^- \\ | \\ \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ | \\ \text{foo} :: \mathtt{X}^- \end{array}$$

$\text{foo} :: \mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$\text{foo} :: \mathtt{X}^-$

$\text{bar} :: \mathtt{X}^-$

$\text{foo} :: \mathtt{X}^+\mathtt{X}^-$

$\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

$$\begin{array}{c} \varepsilon :: \mathtt{X}^+\mathtt{C}^- \\ | \\ \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ | \\ \text{foo} :: \mathtt{X}^- \end{array} \qquad \begin{array}{c} \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ | \\ \text{foo} :: \mathtt{X}^- \end{array}$$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$$\begin{array}{l} \text{foo} :: \mathtt{X}^- \\ \text{bar} :: \mathtt{X}^- \end{array}$$

$$\begin{array}{r} \text{foo} :: \mathtt{X}^+\mathtt{X}^- \\ \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ \varepsilon :: \mathtt{X}^+\mathtt{C}^- \end{array}$$

$$\begin{array}{c} \varepsilon :: \mathtt{X}^+\mathtt{C}^- \\ | \\ \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ | \\ \text{foo} :: \mathtt{X}^- \end{array}$$

$$\begin{array}{c} \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ | \\ \text{bar} :: \mathtt{X}^+\mathtt{X}^- \\ | \\ \text{foo} :: \mathtt{X}^- \end{array}$$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

foo :: $\mathtt{X}^-$

bar :: $\mathtt{X}^-$

foo :: $\mathtt{X}^+\mathtt{X}^-$

bar :: $\mathtt{X}^+\mathtt{X}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$$
\begin{array}{ll}
\text{foo} :: \mathtt{O}^- & \text{foo} :: \mathtt{E}^+\mathtt{O}^- \\
 & \text{foo} :: \mathtt{O}^+\mathtt{E}^- \\
\text{bar} :: \mathtt{O}^- & \text{bar} :: \mathtt{E}^+\mathtt{O}^- \\
 & \text{bar} :: \mathtt{O}^+\mathtt{E}^- \\
 & \varepsilon :: \mathtt{O}^+\mathtt{C}^-
\end{array}
$$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$ $\quad$ $\text{foo} :: \mathtt{E}^+\mathtt{O}^-$

$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$ $\quad$ $\text{bar} :: \mathtt{E}^+\mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$ $\quad$ $\text{foo} :: \mathtt{O}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$

$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$$\begin{array}{c} \text{bar} :: \mathtt{O}^+\mathtt{E}^- \\ | \\ \text{foo} :: \mathtt{O}^- \end{array}$$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

foo :: $\mathtt{O}^-$	foo :: $\mathtt{E}^+\mathtt{O}^-$
	foo :: $\mathtt{O}^+\mathtt{E}^-$
bar :: $\mathtt{O}^-$	bar :: $\mathtt{E}^+\mathtt{O}^-$
	bar :: $\mathtt{O}^+\mathtt{E}^-$
	ε :: $\mathtt{O}^+\mathtt{C}^-$

$$
\begin{array}{c}
\varepsilon :: \mathtt{O}^+\mathtt{C}^- \\
| \\
\text{bar} :: \mathtt{O}^+\mathtt{E}^- \\
| \\
\text{foo} :: \mathtt{O}^-
\end{array}
$$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$

$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$	$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$
	$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$
$\text{bar} :: \mathtt{O}^-$	$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$
	$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
	$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$
|
$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
|
$\text{foo} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
|
$\text{foo} :: \mathtt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$

$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$$
\begin{array}{c}
\varepsilon :: \mathtt{O}^+\mathtt{C}^- \\
| \\
\text{bar} :: \mathtt{O}^+\mathtt{E}^- \\
| \\
\text{foo} :: \mathtt{O}^-
\end{array}
\qquad
\begin{array}{c}
\text{bar} :: \mathtt{E}^+\mathtt{O}^- \\
| \\
\text{bar} :: \mathtt{O}^+\mathtt{E}^- \\
| \\
\text{foo} :: \mathtt{O}^-
\end{array}
$$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$
$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$
$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$$\begin{array}{c} \varepsilon :: \mathtt{O}^+\mathtt{C}^- \\ | \\ \text{bar} :: \mathtt{O}^+\mathtt{E}^- \\ | \\ \text{foo} :: \mathtt{O}^- \end{array}$$

$$\begin{array}{c} \varepsilon :: \mathtt{O}^+\mathtt{C}^- \\ | \\ \text{bar} :: \mathtt{E}^+\mathtt{O}^- \\ | \\ \text{bar} :: \mathtt{O}^+\mathtt{E}^- \\ | \\ \text{foo} :: \mathtt{O}^- \end{array}$$

The problem with subcategorization

- Even very complex constraints can be
  1. compiled into the category system and
  2. enforced via subcategorization.
- works for all MSO constraints ⇒ massive overgeneration (Graf 2011; Kobele 2011)
- Linguistic criteria for determining categories are too weak to prevent this.
  - morphology
  - syntactic distribution
  - semantics

The central issue

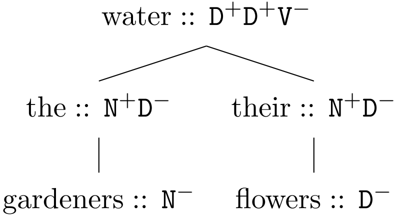
We need a more restrictive notion of category!

A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.

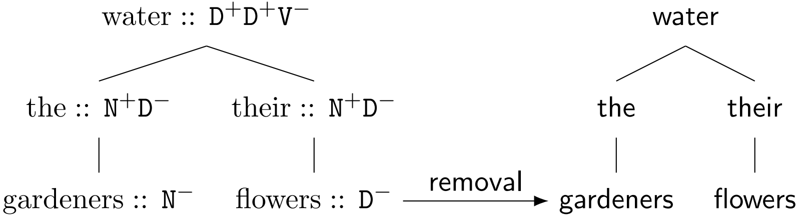
A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



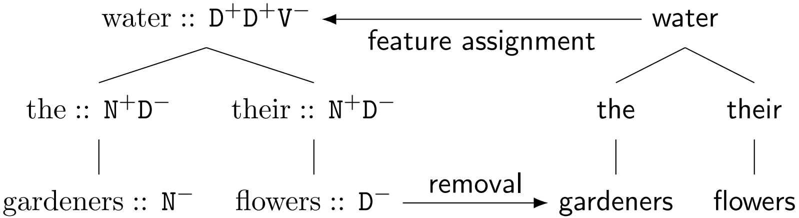
A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



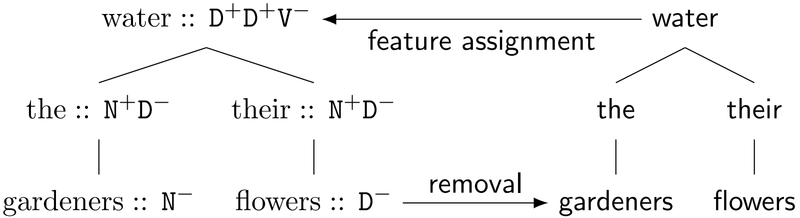
A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



Local feature recoverability

Features must be recoverable in an **ISL** fashion.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

$\boxed{a} \to b$

$b\ \boxed{b} \to a$

$a\ \boxed{c}\ a \to e$

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

b

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

b b

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

b b a

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

b b a b

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a $\rightarrow$ b

b b $\rightarrow$ a

a c a $\rightarrow$ e

a b b a c a

b b a b e

- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

$\boxed{a} \rightarrow b$

$b\,\boxed{b} \rightarrow a$

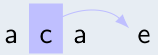
$a\,\boxed{c}\,a \rightarrow e$

a	b	b	a	c	a
b	b	a	b	e	b

- ISL is a very weak class,
  yet widely found in phonology and morphology.

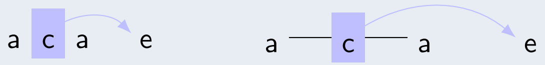
Lifting ISL relabelings to trees

String contexts as tree contexts



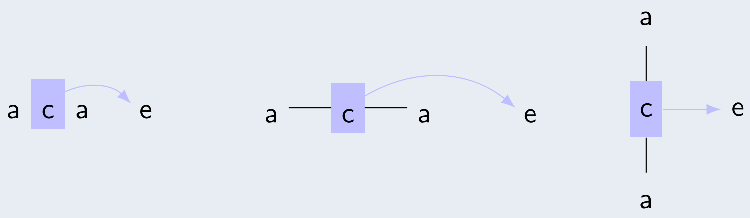
Lifting ISL relabelings to trees

String contexts as tree contexts



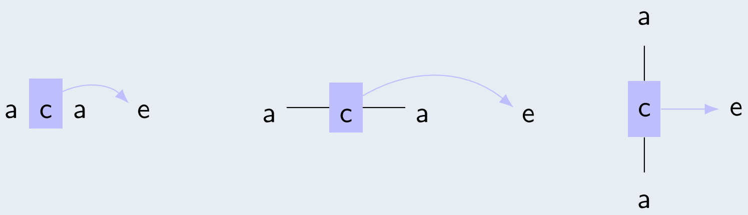
Lifting ISL relabelings to trees

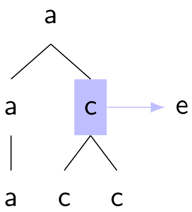
String contexts as tree contexts



Lifting ISL relabelings to trees

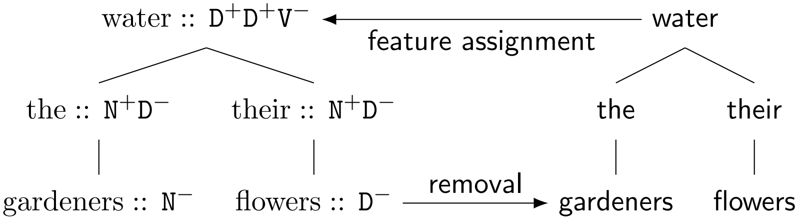
String contexts as tree contexts





Reminder: ISL for feature inference

- Feature cost ≈ how hard to assign by transduction?

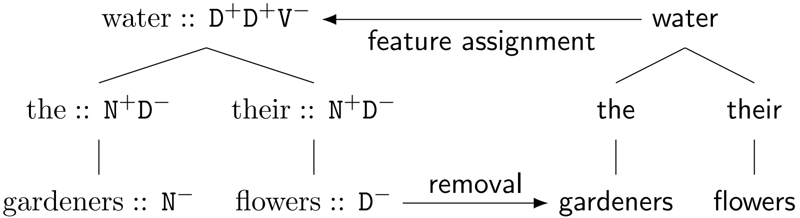


Local feature recoverability

Features must be recoverable in an **ISL** fashion.

Reminder: ISL for feature inference

- Feature cost ≈ how hard to assign by transduction?



Local feature recoverability

Features must be recoverable in an **ISL** fashion.

Intuition

Categorial ambiguity can be resolved within local context

Modulo counting is not ISL recoverable

$$\begin{array}{c} \varepsilon :: \mathtt{O}^{+}\mathtt{C}^{-} \\ | \\ \text{bar} :: \mathtt{E}^{+}\mathtt{O}^{-} \\ | \\ \text{foo} :: \mathtt{O}^{+}\mathtt{E}^{-} \\ | \\ \text{bar} :: \mathtt{O}^{-} \end{array}$$

Modulo counting is not ISL recoverable

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$
|
$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$
|
$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$
|
$\text{bar} :: \mathtt{O}^-$

ε
⋮
bar
|
bar
|
foo
|
bar
|
bar
⋮
bar

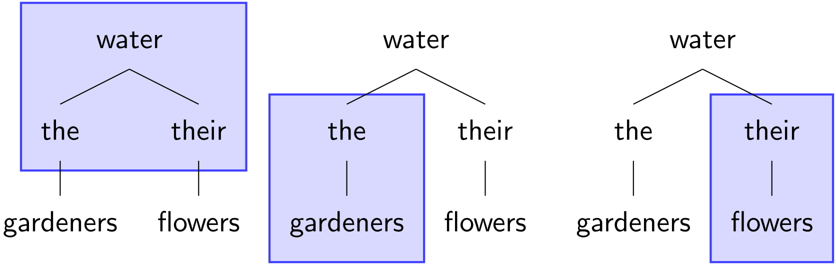
- Can you determine the features of **foo**?
  1. $\mathtt{O}^+\ \mathtt{E}^-$
  2. $\mathtt{E}^+\ \mathtt{O}^-$
- No, that's impossible.
- You need more than local information.
- Modulo counting is not ISL recoverable.

An empirical conjecture

SL-2 recoverability conjecture

The category and selector features of lexical items are

- recoverable from feature-less dependency trees
- using only a window of size 2.



Implications and open issues

Implications

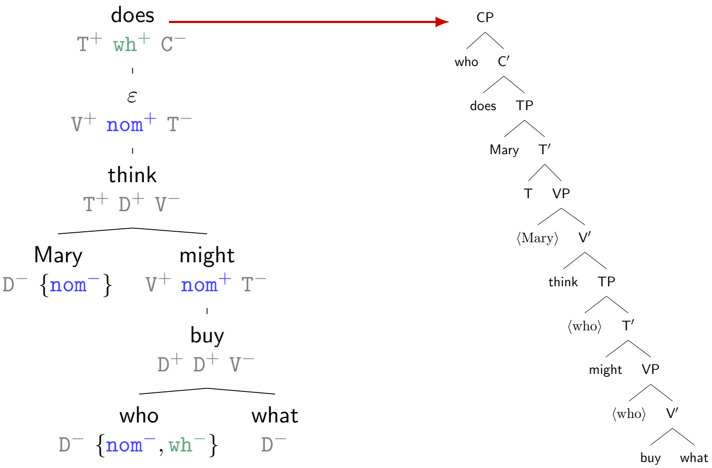
- We avoid tons of overgeneration.
- Heads only select for arguments, not arguments of arguments.

Open issues

- Needs to be tested across many languages
- Depends on theoretical assumptions
  - distribution of empty heads
  - lexical items fully inflected or bare roots? (Hale and Keyser 1993; Marantz 1997)
- SL-2 may be too tight, but SL-k recoverability seems safe
- Move features are not ISL recoverable!

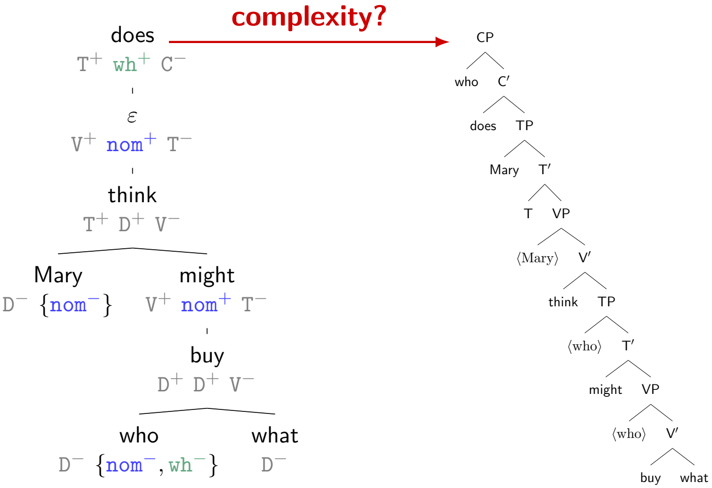
An incomplete picture

Movement isn't just a syntactic dependency, it **affects the output**.



An incomplete picture

Movement isn't just a syntactic dependency, it **affects the output**.

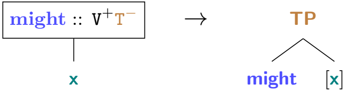


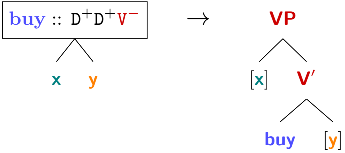
Phrase structure trees without movement

no movement $\Rightarrow$ easy translation to phrase structure trees

who :: $\mathtt{D}^-$ $\rightarrow$ who

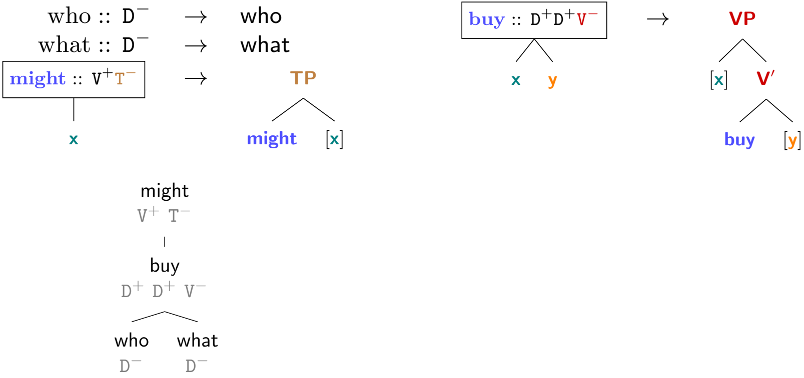
what :: $\mathtt{D}^-$ $\rightarrow$ what





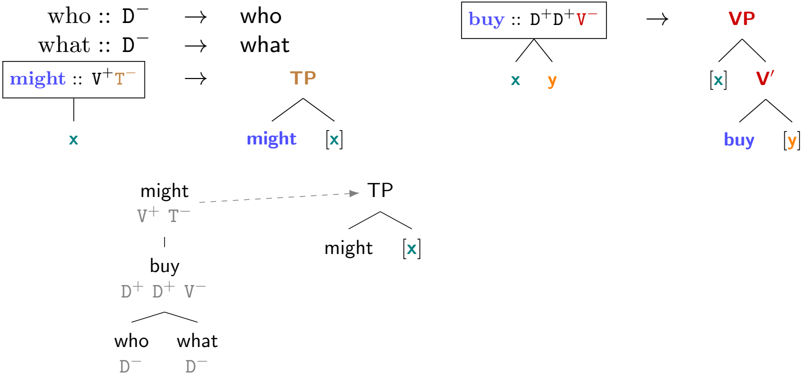
Phrase structure trees without movement

no movement $\Rightarrow$ easy translation to phrase structure trees



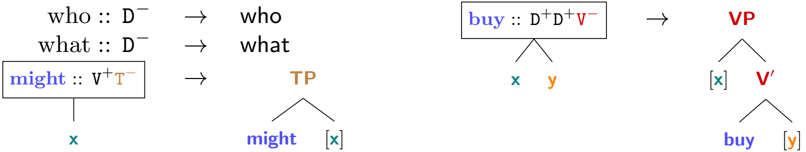
Phrase structure trees without movement

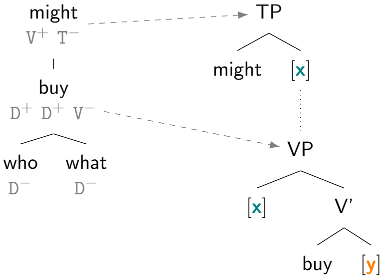
no movement $\Rightarrow$ easy translation to phrase structure trees



Phrase structure trees without movement

no movement ⇒ easy translation to phrase structure trees



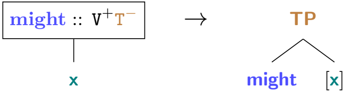


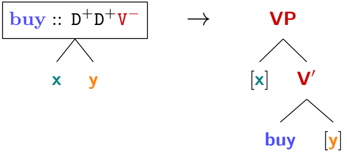
Phrase structure trees without movement

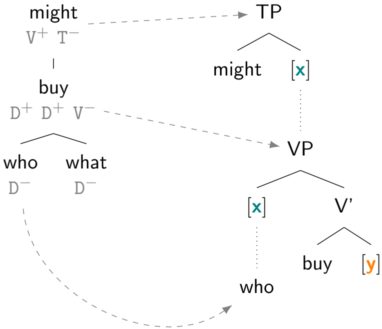
no movement $\Rightarrow$ easy translation to phrase structure trees

who :: $\mathtt{D}^-$ $\rightarrow$ who

what :: $\mathtt{D}^-$ $\rightarrow$ what





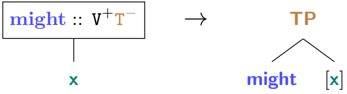


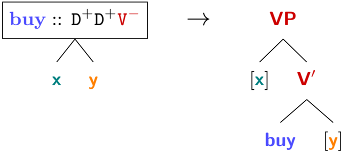
Phrase structure trees without movement

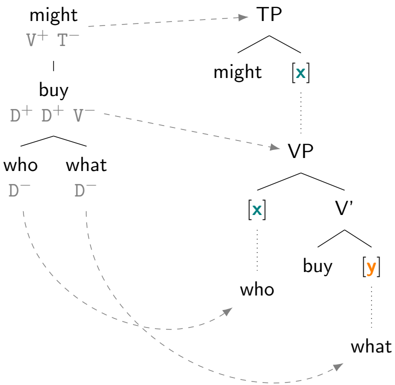
no movement $\Rightarrow$ easy translation to phrase structure trees

who :: $\mathtt{D}^-$ $\rightarrow$ who

what :: $\mathtt{D}^-$ $\rightarrow$ what

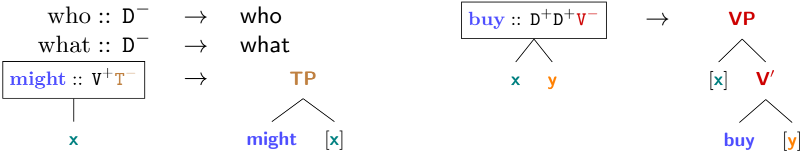


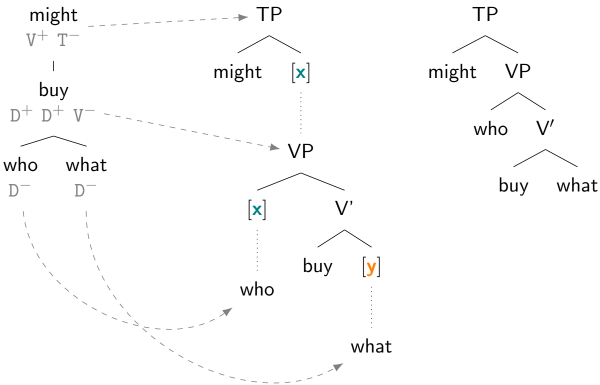




Phrase structure trees without movement

no movement ⇒ easy translation to phrase structure trees

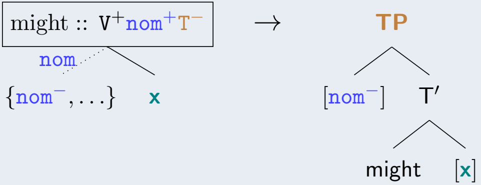




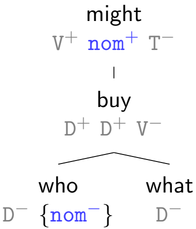
Adding tier relations

- ISL limited to local contexts $\Leftrightarrow$ unbounded movement
- **But:** movement is **local over tiers**
- suffices to enrich ISL rewrite rules with tier-daughter relation

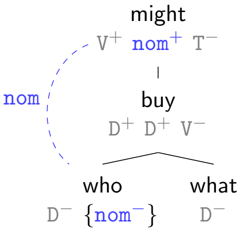
ISL rewrite rule with nom-daughter relation



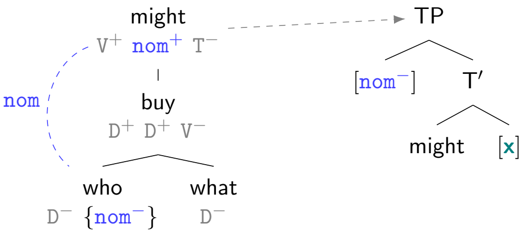
Example with subject movement



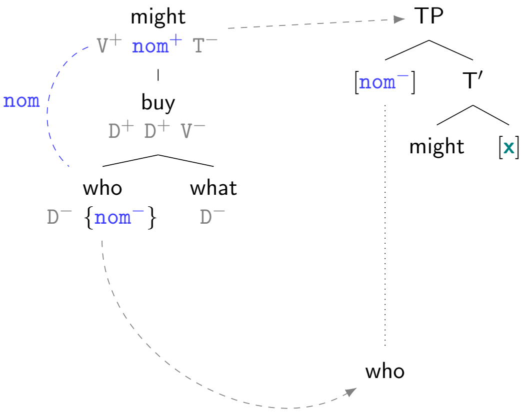
Example with subject movement



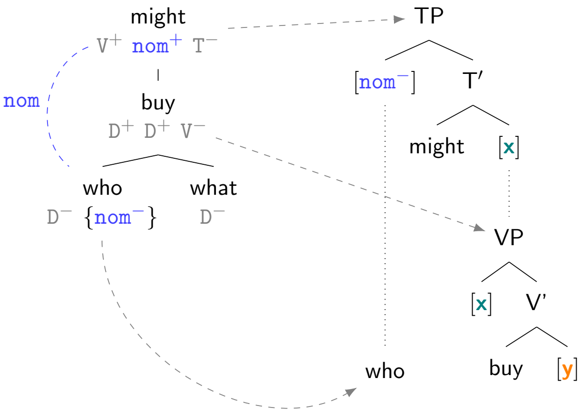
Example with subject movement



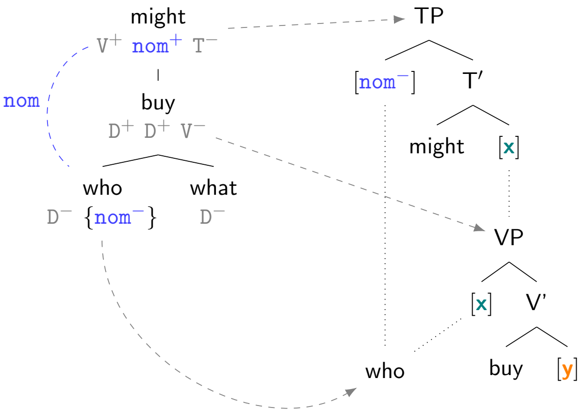
Example with subject movement



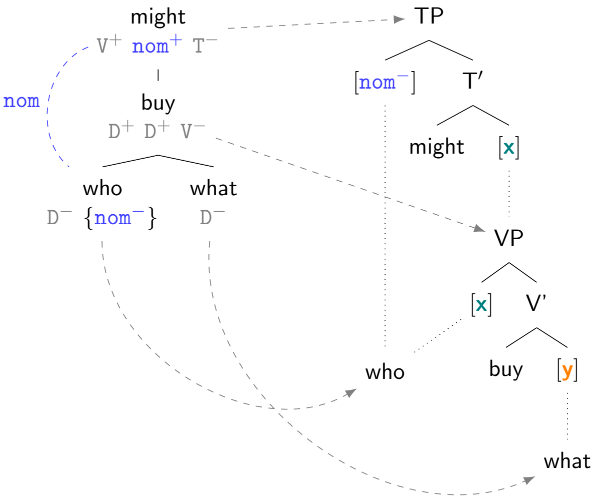
Example with subject movement



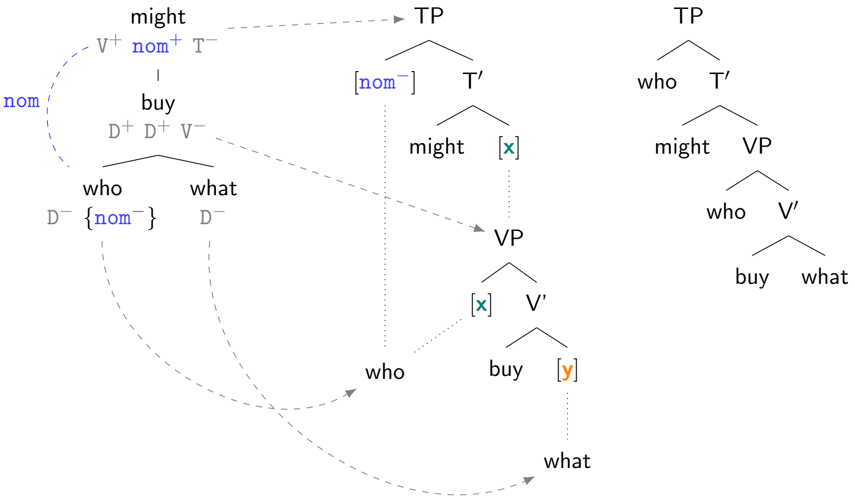
Example with subject movement



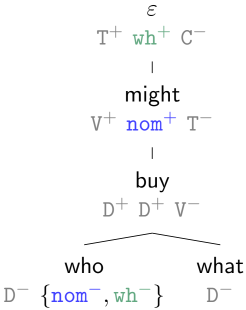
Example with subject movement



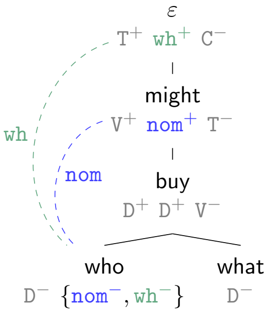
Example with subject movement



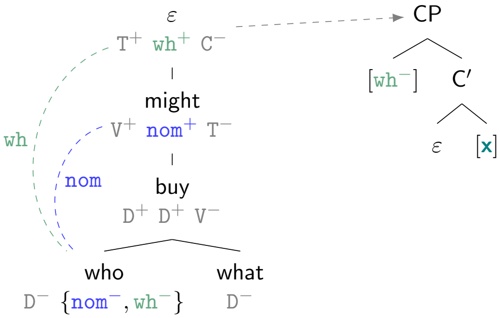
Multiple copies are straightforward



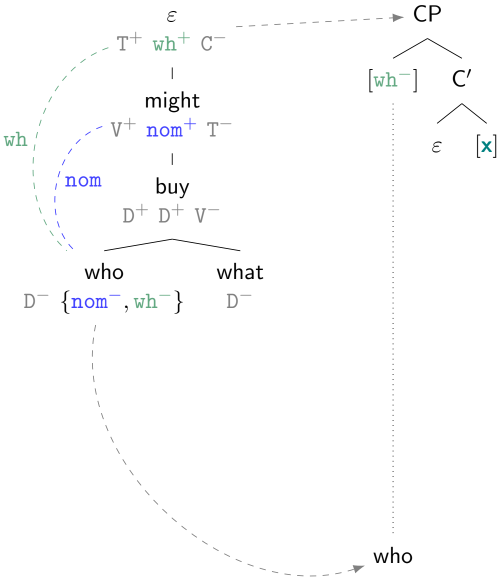
Multiple copies are straightforward



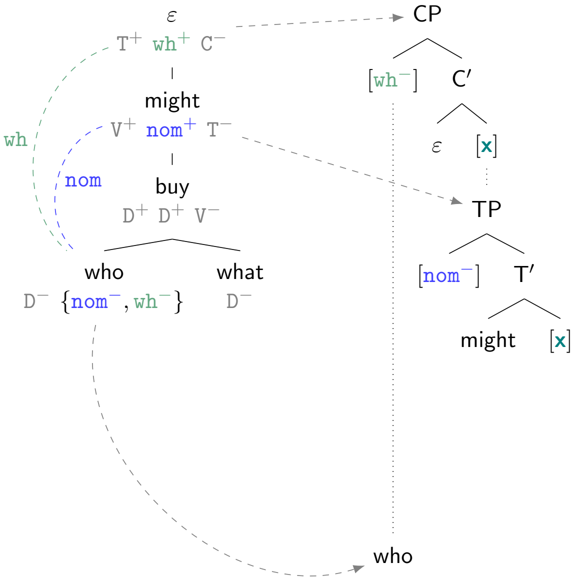
Multiple copies are straightforward



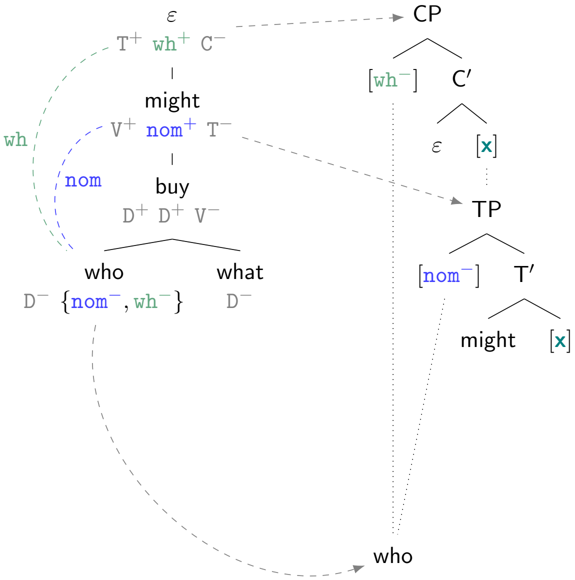
Multiple copies are straightforward



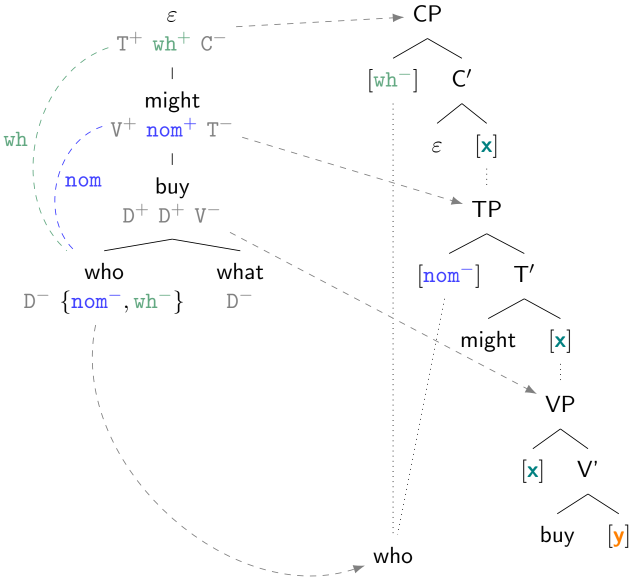
Multiple copies are straightforward



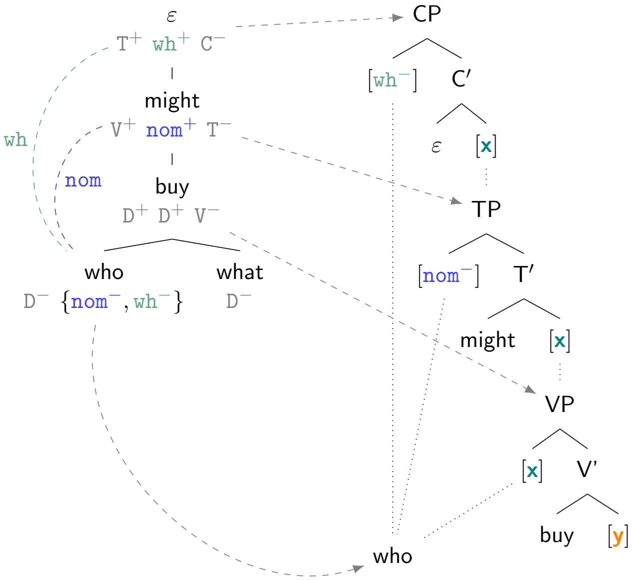
Multiple copies are straightforward



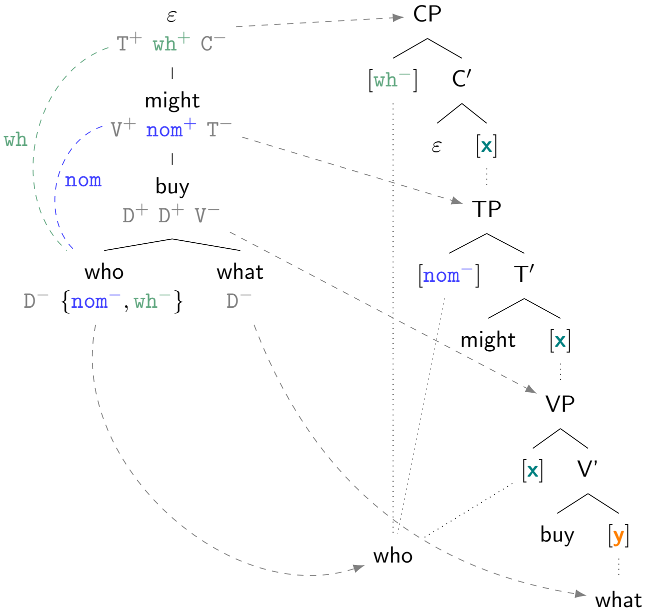
Multiple copies are straightforward



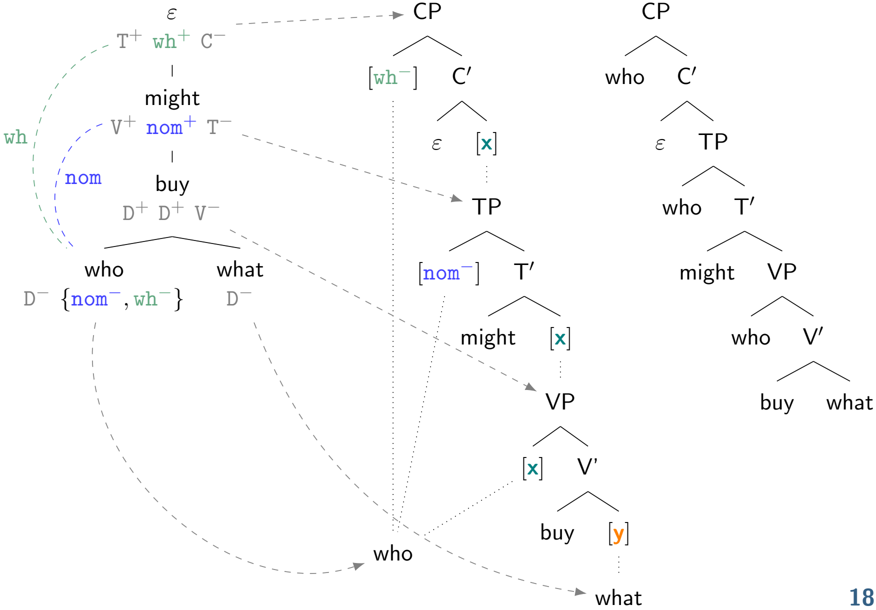
Multiple copies are straightforward



Multiple copies are straightforward



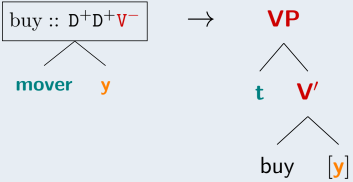
Multiple copies are straightforward



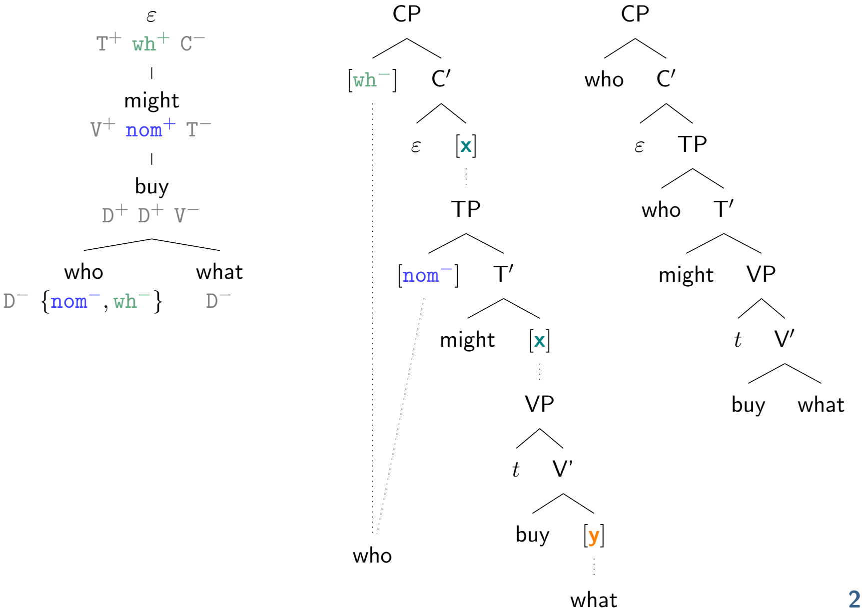
Linearization/traces: tricky

- Copies don't tell us how to pronounce the tree.
- **Traces**: unpronounced landing sites of movers

A delinking rule for base positions

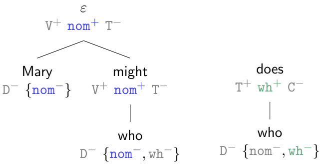


Base delinking example



Lexical predictability

- Given two landing sites **x** and **y** on different tiers, one cannot tell from the tiers whether **x** or **y** is higher.
- We **cannot distinguish final from non-final landing sites**.



Lexical predictability requirement of delinking

Delinking works only if one knows whether to

1. insert a copy, or
2. insert a trace.

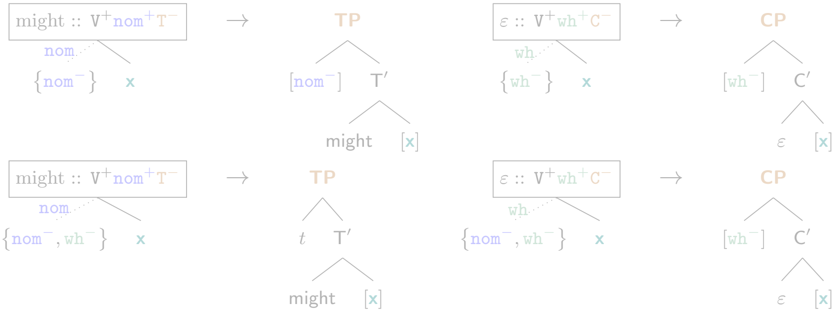
Due to the limitations of tiers,
this must be **inferable directly from the mover**.

Empirical support

- Lexical predictability holds for nom and wh movement.

Ban on Improper Movement (BoIM)

If a mover undergoes both nom and wh movement,
nom movement derivationally precedes wh movement.

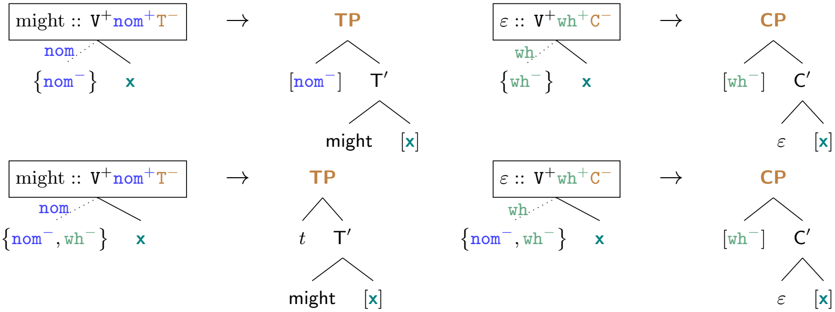


Empirical support

- Lexical predictability holds for nom and wh movement.

Ban on Improper Movement (BoIM)

If a mover undergoes both nom and wh movement,
nom movement derivationally precedes wh movement.



Output-oriented Ban on Improper Movement

- BoIM is a particular instance of a more general requirement.

Output-oriented Ban on Improper Movement (OOBoIM)

- Let l be an arbitrary lexical item with $\{\mathtt{f}^-, \mathtt{g}_0^-, \ldots, \mathtt{g}_n^-\}$.
- If l's final movement step is $\mathtt{f}$-movement in **some** derivation, then l's final movement step is $\mathtt{f}$-movement in **all** derivations.

- Kenneth Hanson's analysis of MG corpus supports **even stronger version**: if $\mathtt{f} \prec \mathtt{g}$ for l in some derivation, then $\mathtt{f} \prec \mathtt{g}$ for l in all derivations.
- OOBoIM permits BoIM violations

hyperraising

Kenneth Hanson

Conclusion

- Movement is both a syntactic dependency and an operation.
- In both cases the core of movement is **local over tiers**.
- Identifying mover with all landing sites
  (copies/multidominance)
  easier than identifying output-relevant landing sites (traces)

Outlook: Bringing it all together

- TSL perspectives of all movement types
  covert, successive-cyclic, sidewards, multiple wh
- Tiers for islands, extraction morphology and path conditions
  Wolof u-chains, floating quantifiers, Germanic wh-copying
- Learning
  SL learning of Merge features, ??? for Move features

Acknowledgments

This work is supported by the National Science Foundation under Grant No. BCS-1845344.

References I

Chandlee, Jane. 2014. *Strictly local phonological processes*. Doctoral Dissertation, University of Delaware. URL http://udspace.udel.edu/handle/19716/13374.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized phrase structure grammar*. Oxford: Blackwell.

Graf, Thomas. 2011. Closure properties of Minimalist derivation tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 96–111. Heidelberg: Springer. URL https://dx.doi.org/10.1007/978-3-642-22221-4_7.

Graf, Thomas. 2017. A computational guide to the dichotomy of features and constraints. *Glossa* 2:1–36. URL https://dx.doi.org/10.5334/gjgl.212.

Hale, Kenneth, and Samuel J. Keyser. 1993. On argument structure and the lexical expression of syntactic relations. In *The view from building 20: Essays in honor of sylvain bromberger*, ed. Kenneth Hale and Samuel J. Keyser. Cambridge, MA: MIT Press.

Kobele, Gregory M. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 129–144. URL https://doi.org/10.1007/978-3-642-22221-4_9.

Marantz, Alec. 1997. No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. *Penn Working Papers in Linguistics* 4:201–225.

References II

Stabler, Edward P. 1997. Derivational Minimalism. In *Logical aspects of computational linguistics*, ed. Christian Retoré, volume 1328 of *Lecture Notes in Computer Science*, 68–95. Berlin: Springer. URL `https://doi.org/10.1007/BFb0052152`.