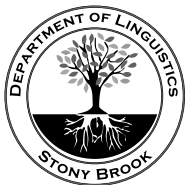


Syntactic tiers for movement and agreement

Day 1: Representations and the status of tiers

Thomas Graf



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

KU Leuven Lecture Series
December 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

- 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

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

*\$ 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 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 d \$



*z\$

*v\$

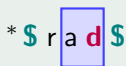
*d\$

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 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 d \$

*z\$

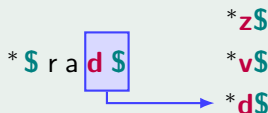
*v\$

*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



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 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: Intervocalic Voicing is SL-3

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

Example: Northern Italian

*\$ **a** **s** **o** | **a** \$

Example: Intervocalic Voicing is SL-3

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

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

Example: Northern Italian

*\$ **a s** o | a \$

Example: Intervocalic Voicing is SL-3

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

Example: Northern Italian

*\$ **a s o** | a \$

Example: Intervocalic Voicing is SL-3

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

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **aja**, **asi**, **aji**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

Example: Northern Italian

*\$ **a s o** | **a** \$

\$ **a z o** | **a** \$

Example: Intervocalic Voicing is SL-3

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

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **aja**, **asi**, **aji**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **aja**, **asi**, **aji**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **aja**, **asi**, **aji**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **aja**, **asi**, **aji**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **aja**, **asi**, **aji**, ...

Example: Northern Italian

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

\$ **a z o l a** \$

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

Example: Intervocalic Voicing is SL-3

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

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

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 $V[-\text{voice}]V$
- ▶ **Suppose:**
 - ▶ $[-\text{voice}] = \{s, f\}$
 - ▶ $V = \{a, i, o, u\}$
- ▶ **Compiled out:** don't have **asa**, **ajfa**, **asi**, **ajfi**, ...

Example: Northern Italian

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

\$ **a z o l a** \$

\$ **a + s o c i 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

- 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

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**.

***s**ʃ ***s**ʒ ***z**ʃ ***z**ʒ
 *ʃ**s** *ʒ**s** *ʃ**z** *ʒ**z**

- ▶ **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**.

***s**ʃ ***s**ʒ ***z**ʃ ***z**ʒ
 *ʃ**s** *ʒ**s** *ʃ**z** *ʒ**z**

- ▶ **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)

*\$ha **s**xintilawa **ʃ**\$

\$ha **ʃ**xintilawa **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 ʒ \$

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**.

***s**ʃ ***s**ʒ ***z**ʃ ***z**ʒ
 *ʃ**s** *ʒ**s** *ʃ**z** *ʒ**z**

- ▶ **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)

*\$ha **s**xintilawa **ʃ**\$

\$ha **ʃ**xintilawa **ʃ**\$

*\$ **s**tajanowonwa **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, *sɜ, *zʃ, *zɜ, ʃs, ɜs, ʃz, ɜz

*\$hasxintilawaʃ\$

\$haʃxintilawaʃ\$

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

*\$ha s xintilawa ʃ\$

\$ha ʃ xintilawa 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 ***s**ʃ, ***s**ʒ, ***z**ʃ, ***z**ʒ, *ʃ**s**, *ʒ**s**, *ʃ**z**, *ʒ**z**

\$	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



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 *s_J, *s₃, *z_J, *z₃, *_Js, *₃s, *_Jz, *₃z

\$	s	J	\$		\$	J	\$	J	\$
*\$	h	a	s	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	l
\$	h	a	J	x	i	n	t	i	

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 *s_J, *s₃, *z_J, *z₃, *_Js, *₃s, *_Jz, *₃z

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

\$	J		J	\$		J	\$							
\$	h	a	J	x	i	n	t	i	l	a	w	a	J	\$

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 j \$

\$ z i d a j \$

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

Another TSL Boon: Blocking

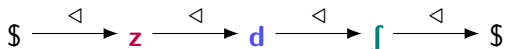
- ▶ TSL can also handle blocking effects.
- ▶ **Slovenian sibilant harmony with blocking**
 - 1 [-ant] ... [+ant] is illicit,
 - 2 unless t or d intervenes.
- ▶ **TSL-2 account**
 - 1 project all [-ant], [+ant], t, and d
 - 2 don't have [-ant] [+ant]

Example: Slovenian (Jurgec 2011; McMullin 2016)

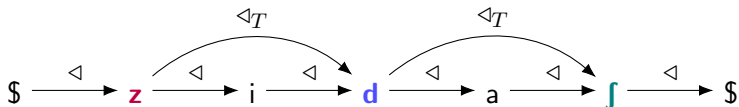
	\$	s		ɟ	\$					\$	z	d		ɟ	\$	
*	\$	s	p	i	ɟ	\$				\$	z	i	d	a	ɟ	\$

Tiers have many interpretations

1 Separate structure



2 Additional relation



3 Substructures

▶ without tier: $\$z, zi, id, da, aj, j\$$

▶ with tier: $\$z, zd, dj, j\$$

4 Memory configurations

	Input	\$	\mathbf{z}	\mathbf{i}	\mathbf{d}	\mathbf{a}	\mathbf{j}	\$
SL-2 memory	-	\$	\mathbf{z}	\mathbf{i}	\mathbf{d}	\mathbf{a}	\mathbf{j}	
TSL-2 memory	-	\mathbf{z}	\mathbf{z}	\mathbf{d}	\mathbf{d}	\mathbf{j}	\mathbf{j}	

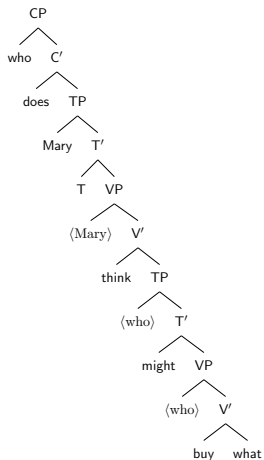
Epistemological modesty: what are tiers?

- ▶ 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!**

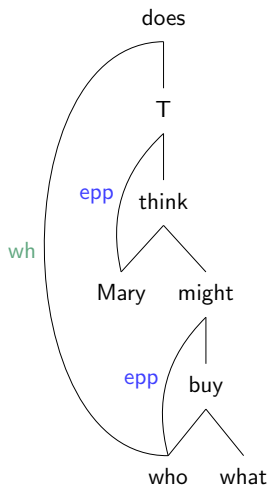
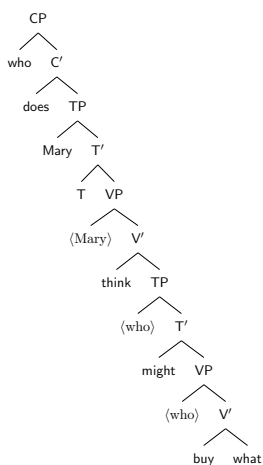
Outline

- 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

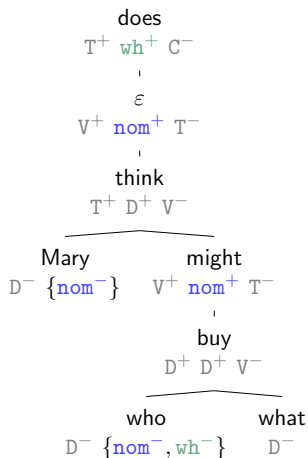
Freeing syntax from its output



Freeing syntax from its output



Using diacritics to arborize dependency graphs



► Terminology

- Category ~~feature~~ X^-
- Selector ~~feature~~ X^+
- Licensee ~~feature~~ x^-
- Licensor ~~feature~~ 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 [_{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 [_{VP} *t* laughed at Bill].

Sequence of Merge steps:

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

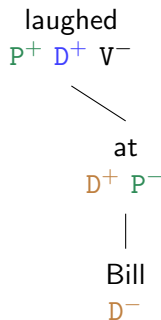
at
 $D^+ P^-$
 |
 Bill
 D^-

A Detailed Merge Example

(1) John [_{VP} *t* laughed at Bill].

Sequence of Merge steps:

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

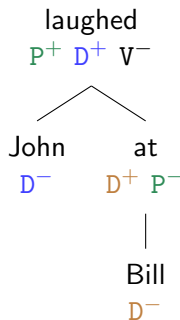


A Detailed Merge Example

(1) John [_{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

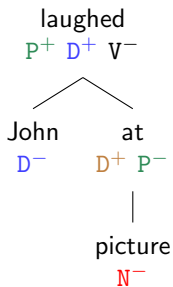
Merge is SL-2 over trees because we only need to look at

- 1 the mother and
- 2 its daughters

Merge is SL-2

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

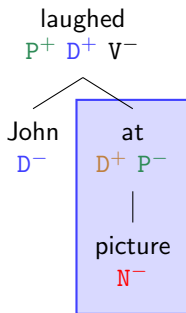
- 1 the mother and
- 2 its daughters



Merge is SL-2

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

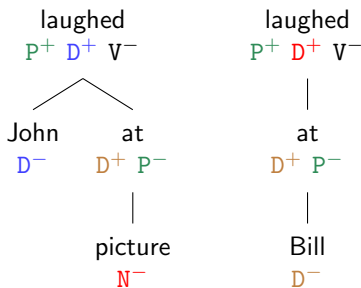
- 1 the mother and
- 2 its daughters



Merge is SL-2

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

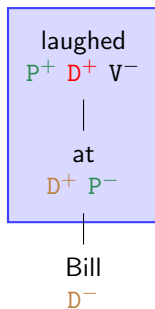
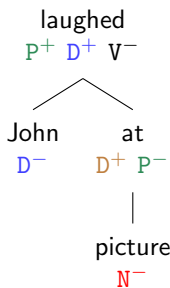
- 1 the mother and
- 2 its daughters



Merge is SL-2

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

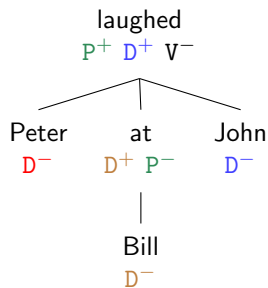
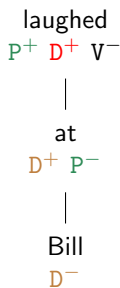
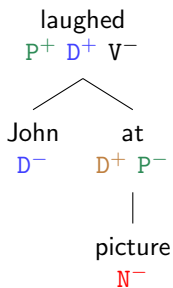
- 1 the mother and
- 2 its daughters



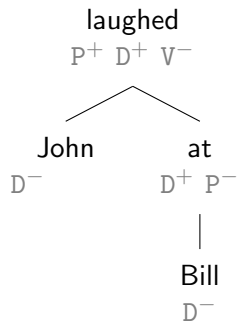
Merge is SL-2

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

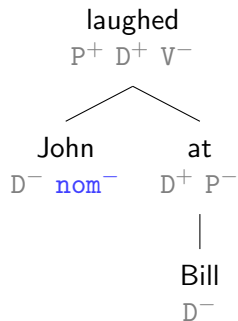
- 1 the mother and
- 2 its daughters



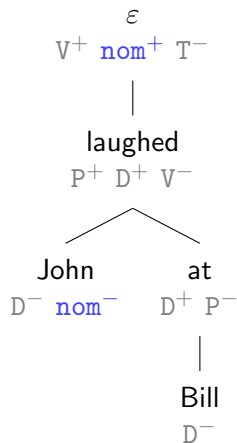
Movement is Local over Tree Tiers



Movement is Local over Tree Tiers

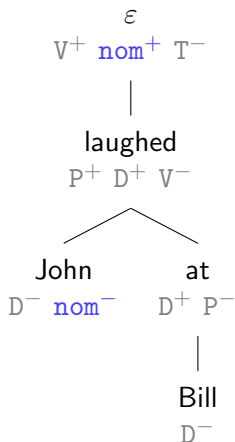


Movement is Local over Tree Tiers

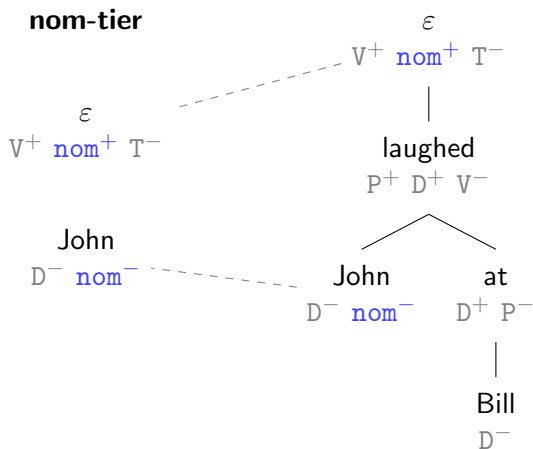


Movement is Local over Tree Tiers

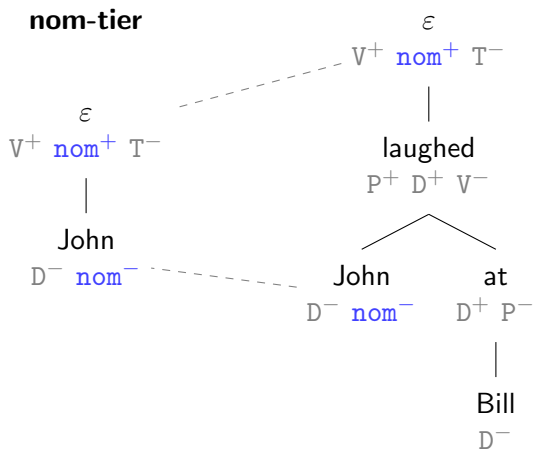
nom-tier



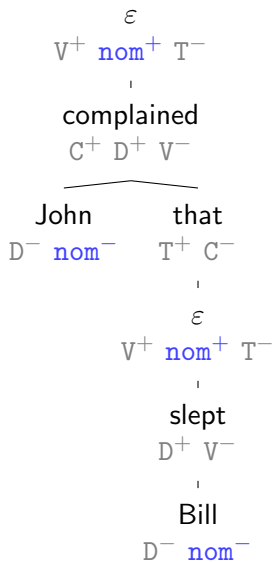
Movement is Local over Tree Tiers



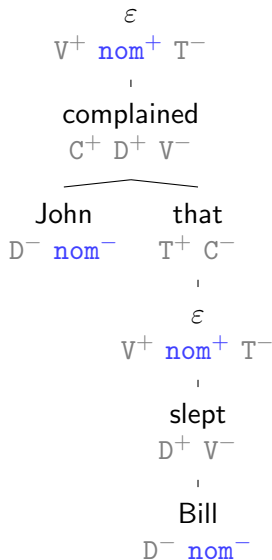
Movement is Local over Tree Tiers



Tier with Multiple Movers



Tier with Multiple Movers

nom-tier

Tier with Multiple Movers

nom-tier

ε
 $V^+ \text{ nom}^+ T^-$

ε
 $V^+ \text{ nom}^+ T^-$
 |
 complained
 $C^+ D^+ V^-$
 ───────────
 John that
 $D^- \text{ nom}^-$ $T^+ C^-$
 |
 ε
 $V^+ \text{ nom}^+ T^-$
 |
 slept
 $D^+ V^-$
 |
 Bill
 $D^- \text{ nom}^-$

Tier with Multiple Movers

nom-tier

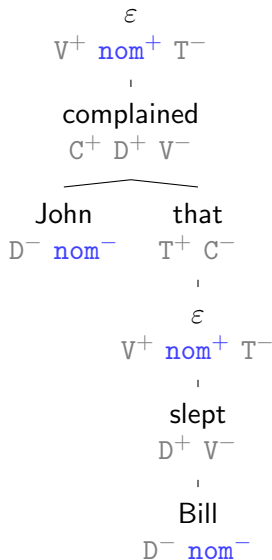
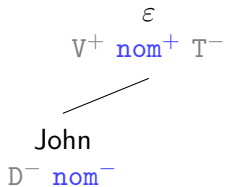
$$\begin{array}{c} \varepsilon \\ V^+ \text{ nom}^+ T^- \end{array}$$

$$\begin{array}{c} \text{John} \\ D^- \text{ nom}^- \end{array}$$

$$\begin{array}{c} \varepsilon \\ V^+ \text{ nom}^+ T^- \\ | \\ \text{complained} \\ C^+ D^+ V^- \\ \hline \text{John} \quad \text{that} \\ D^- \text{ nom}^- \quad T^+ C^- \\ | \\ \varepsilon \\ V^+ \text{ nom}^+ T^- \\ | \\ \text{slept} \\ D^+ V^- \\ | \\ \text{Bill} \\ D^- \text{ nom}^- \end{array}$$

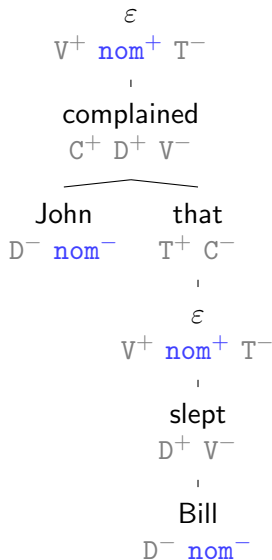
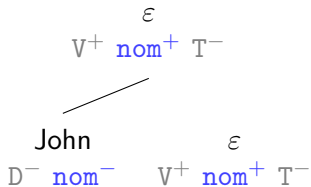
Tier with Multiple Movers

nom-tier



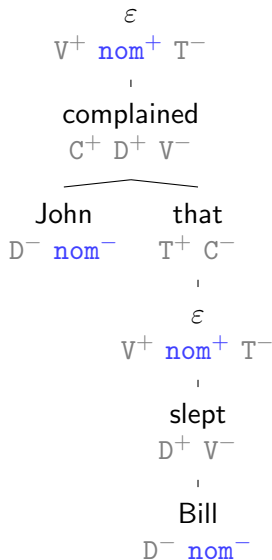
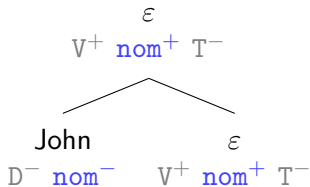
Tier with Multiple Movers

nom-tier



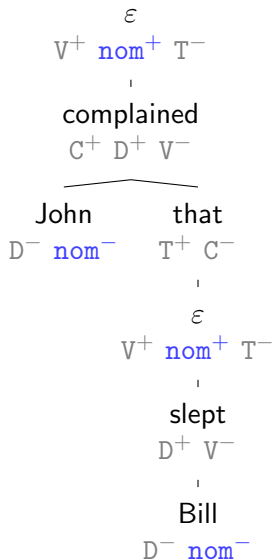
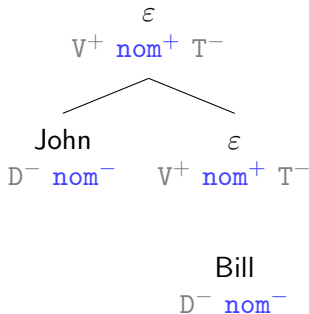
Tier with Multiple Movers

nom-tier



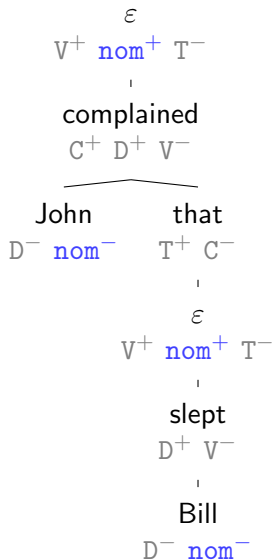
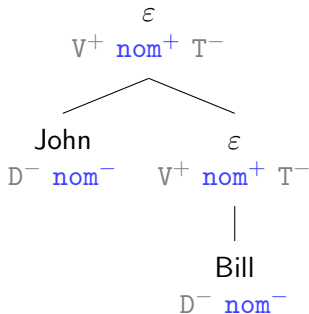
Tier with Multiple Movers

nom-tier

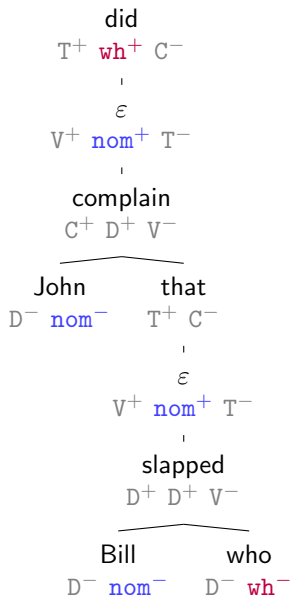


Tier with Multiple Movers

nom-tier

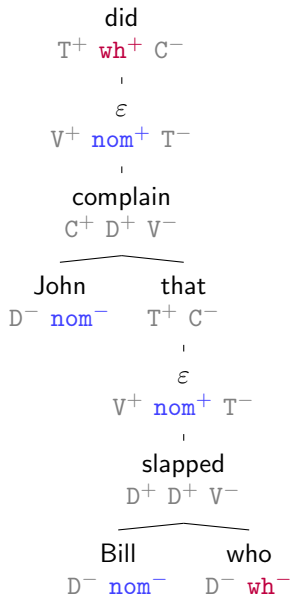
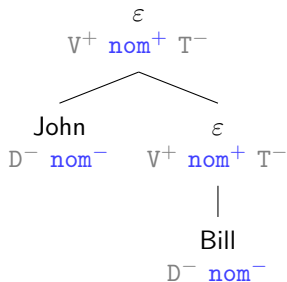


Separate Tier for Each Movement Type

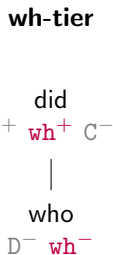
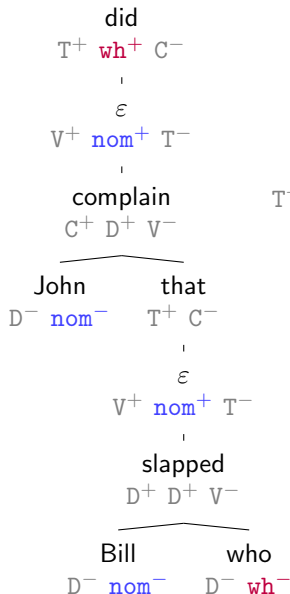
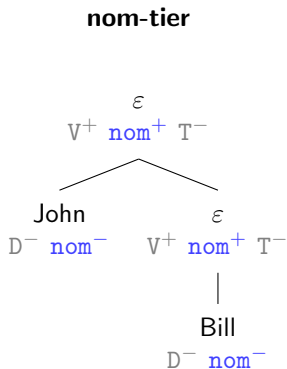


Separate Tier for Each Movement Type

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 f^- must have an f^+ mother.
 - 2 Every f^+ has exactly one 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 f^- must have an f^+ mother.
 - 2 Every f^+ has exactly one 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

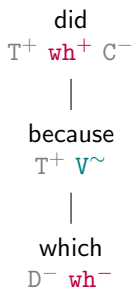
- 1** Movement requires the power of TSL-2.
- 2** TSL-2 can model islands as blocking effects.
- 3** The cognitive ability for movement entails the cognitive ability for islands.

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*].

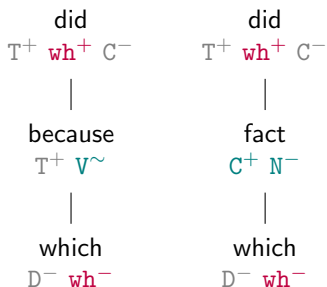
Islands Examples

- (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].



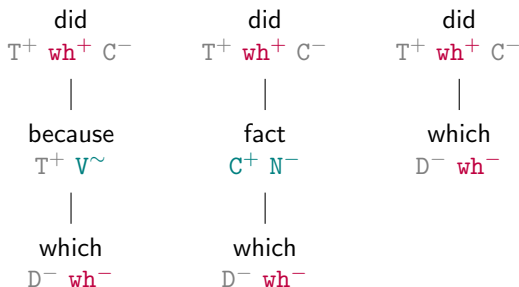
Islands Examples

- (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*].



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>.
- Jurģec, 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.