Subregular Linguistics for Linguists

(100% Formula-Free)

Thomas Graf

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

You can get the slides here under "News"

Invited Talk UMass
November 8, 2019

Definition (Synchronous ISL transduction)

A node b in tree u can be **targeted** by an ISL-k context $\langle s, a, t \rangle$ iff there is some $p \in \mathbb{N}^*$ such that

node match $b = pa$, and

label match for all nodes g of s, $\ell_s(g) = \ell_u(pg)$,

full-width match for all nodes gi of s with $g \in \mathbb{N}^*$ and $i \in \mathbb{N}$, if pgj is a node of u ($j > i$), then gj is a node of s.

Now suppose furthermore that n in u has $d \geq 0$ daughters. Given an ISL-k tree transducer τ, we use $\overleftarrow{\tau}(u, b)$ to denote the set of all trees $t[\Box_1 \leftarrow \overleftarrow{\tau}(u, b1), \ldots, \Box_d \leftarrow \overleftarrow{\tau}(u, bd)]$ such that there is a rewrite rule $\langle s, a, t \rangle$ in τ that targets node b in u. If this set is empty, $\overleftarrow{\tau}(u, b)$ is undefined. For any Σ-tree t, we may simply write $\overleftarrow{\tau}(t)$ instead of $\overleftarrow{\tau}(t, \varepsilon)$.

For any tree language L, the transduction computed by τ in *synchronous mode* is $\overleftarrow{\tau}(L) := \{\langle i, o \rangle \mid i \in L, o \in \overleftarrow{\tau}(i)\}$. A transduction is *synchronous input strictly k-local* (sISL-k) iff it can be computed by some ISL-k transducer in synchronous mode.

Definition (Synchronous ISL transduction)

A node b in tree u can be **targeted** by an ISL-k context $\langle s, a, t \rangle$ iff there is some $p \in \mathbb{N}^*$ such that

node match $b = pa$, and

label match fo

full-width matc … and $i \in \mathbb{N}$, if

pg … a node of s.

Just kiddin'!

Now suppose fu … ughters. Given an ISL-k tree tr … te the set of all trees $t[\Box_1 \leftarrow \overleftarrow{\tau}$ … hat there is a rewrite rule $\langle s,$ … f this set is empty, $\overleftarrow{\tau}(u, b)$ is undefined. For any Σ-tree t, we may simply write $\overleftarrow{\tau}(t)$ instead of $\overleftarrow{\tau}(t, \varepsilon)$.

For any tree language L, the transduction computed by τ in *synchronous mode* is $\overleftarrow{\tau}(L) := \{\langle i, o \rangle \mid i \in L, o \in \overleftarrow{\tau}(i)\}$. A transduction is *synchronous input strictly k-local* (sISL-k) iff it can be computed by some ISL-k transducer in synchronous mode.

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 Opportunistic Program for Lazy Researchers Like Myself

- Stand on the shoulders of giants.
- Computer scientists have figured out a lot about complexity,
  so let's apply their ideas to language.

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 Opportunistic Program for Lazy Researchers Like Myself

- Stand on the shoulders of giants.
- Computer scientists have figured out a lot about complexity,
  so let's apply their ideas to language.

A Mathematical Distinctness Theorem

- From a computational perspective, there is a split between "P-side" and "S-side".

regular $<$ context-free $<$ mildly context-sensitive $< \cdots$

Phonology

Morphology

Syntax

- Matches linguistic practice (despite attempts at unification, e.g. Government Phonology, DM, OT syntax)
- A unified Theory of Everything is not on the linguistic horizon.

A Mathematical Distinctness Theorem

- From a computational perspective, there is a split between "P-side" and "S-side".

regular $<$ context-free $<$ mildly context-sensitive $< \cdots$

Kaplan and Kay (1994)

Phonology

Morphology

Syntax

- Matches linguistic practice (despite attempts at unification, e.g. Government Phonology, DM, OT syntax)
- A unified Theory of Everything is not on the linguistic horizon.

A Mathematical Distinctness Theorem

- From a computational perspective, there is a split between "P-side" and "S-side".

regular $<$ context-free $<$ mildly context-sensitive $< \cdots$

Kaplan and Kay (1994)

Phonology

Karttunen et al. (1992)

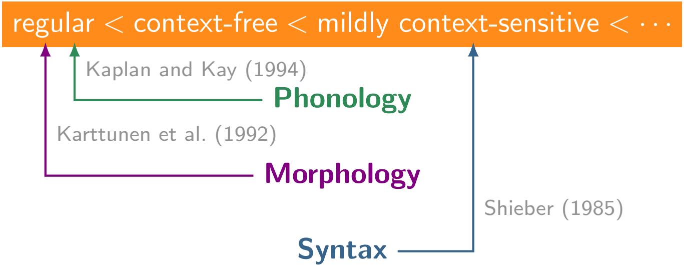
Morphology

Syntax

- Matches linguistic practice (despite attempts at unification, e.g. Government Phonology, DM, OT syntax)
- A unified Theory of Everything is not on the linguistic horizon.

A Mathematical Distinctness Theorem

- From a computational perspective, there is a split between "P-side" and "S-side".



- Matches linguistic practice (despite attempts at unification, e.g. Government Phonology, DM, OT syntax)
- A unified Theory of Everything is not on the linguistic horizon.

Cognitive Parallelism Hypothesis

- The postulated split is misleading.
- If we probe deeper, we find that
  - different modules are remarkably similar,
  - their dependencies are weaker than regular
    $\Rightarrow$ **subregular**
- Cognitive parallelism is **empirically fertile**.

Take-Home Messages

1. Phonology and syntax show surprising subregular parallels. (Morphology and morphosemantics, too...)
2. Like every good theory, subregularity yields new generalizations and data.

Outline

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

- There are a variety of subregular classes to choose from.
- SL and TSL are among the weaker ones.
- They 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)
- **Linguistic core idea:**
  Dependencies are local over the right structure.

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

- There are a variety of subregular classes to choose from.
- SL and TSL are among the weaker ones.
- They 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)
- **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

* $ 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 $\boxed{\textbf{d \$}}$

*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

	***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$

$ 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 $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **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 $\mathrm{V}[-\text{voice}]\mathrm{V}$
- **Suppose**:
  - $[-\text{voice}] = \{$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

* $ 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 $ $\boxed{\text{\$ a z}}$ 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 $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **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 $\mathrm{V}[-\mathrm{voice}]\mathrm{V}$
- **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 $ $ 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 + 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 + $\boxed{\text{s o c}}$ 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**:
  - $[-\mathrm{voice}] = \{\mathsf{s}, \mathsf{ʃ}\}$
  - $\mathrm{V} = \{\mathsf{a,i,o,u}\}$
- **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 $V[-\text{voice}]V$
- **Suppose**:
  - $[-\text{voice}] = \{$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 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 $

A 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 $

A 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** $

A 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 $

A 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 $\boxed{\text{ʃ x i n t i l a w a ʃ}}$ $

A 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 $\boxed{\text{ʃ x i n t i l a w a ʃ}}$ $

*$ s t a j a n o w o n w a ʃ $

A 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

Jeff Heinz

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

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

$ s ʃ $ $\boxed{\text{\$ ʃ}}$ 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 $        $  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 $       $   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 $
```

Example: 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** $

Example: 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** $

Example: 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	$

SL and TSL for Phonology

- Linguistically natural
- Correct and efficient learning algorithm
  (Jardine and McMullin 2017)
- Low resource demands $\Rightarrow$ cognitively plausible
- Captures wide range of phonotactic dependencies
- Cannot generate unattested patterns

Example: First-Last Harmony

- Harmony only holds between initial and final segments
- Linguistically plausible, yet unattested

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

SL and TSL for Phonology

- Linguistically natural
- Correct and efficient learning algorithm (Jardine and McMullin 2017)
- Low resource demands ⇒ cognitively plausible
- Captures wide range of phonotactic dependencies
- Cannot generate unattested patterns

Example: First-Last Harmony

- Harmony only holds between initial and final segments
- Linguistically plausible, yet unattested

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

SL and TSL for Phonology

- Linguistically natural
- Correct and efficient learning algorithm (Jardine and McMullin 2017)
- Low resource demands ⇒ cognitively plausible
- Captures wide range of phonotactic dependencies
- Cannot generate unattested patterns

Example: First-Last Harmony

- Harmony only holds between initial and final segments
- Linguistically plausible, yet unattested

$ s ʃ $

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

SL and TSL for Phonology

- Linguistically natural
- Correct and efficient learning algorithm (Jardine and McMullin 2017)
- Low resource demands $\Rightarrow$ cognitively plausible
- Captures wide range of phonotactic dependencies
- Cannot generate unattested patterns

Example: First-Last Harmony

- Harmony only holds between initial and final segments
- Linguistically plausible, yet unattested

$ s ʃ $

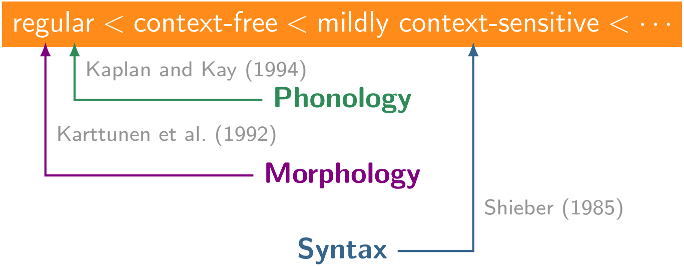
$ h a s x i n t i l a w a ʃ $

$ s ʃ $

*$ s t a j a n o w o n w a ʃ $

Outline

Against the Received View



- This is about strings.
- Syntax is about **trees**!

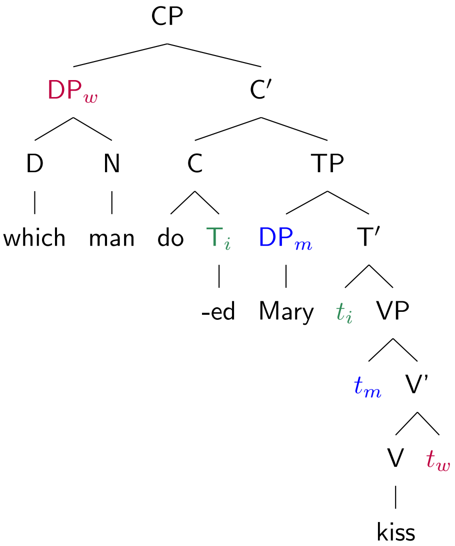
Minimalist Grammars as a Computational Model of Syntax

Ed Stabler

- Minimalist grammars (MGs) are a formalization of Minimalist syntax. (Stabler 1997, 2011)
- Operations: **Merge** and **Move**
- Adopt Chomsky-Borer hypothesis: Grammar is just a finite list of feature-annotated lexical items

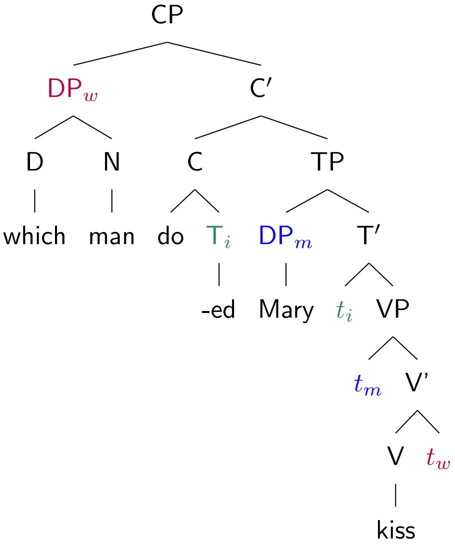
Chemistry	**Syntax**
atoms	words
electrons	features
molecules	sentences

Choice of Representation: Derivation trees

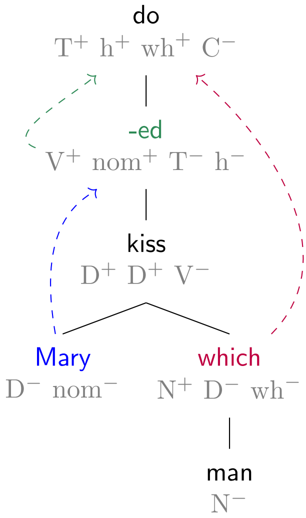


Phrase Structure Tree

Choice of Representation: Derivation trees



Phrase Structure Tree



Derivation Tree

A Detailed Merge Example

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

Sequence of Merge steps:

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

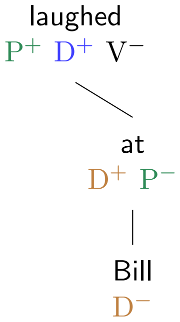
at
D^{+} P^{-}
|
Bill
D^{-}

A Detailed Merge Example

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

Sequence of Merge steps:

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

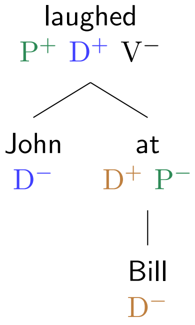


A Detailed Merge Example

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

Sequence of Merge steps:

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

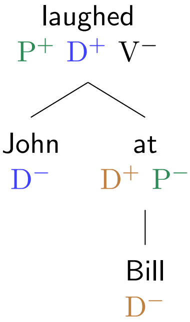


A Detailed Merge Example

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

Sequence of Merge steps:

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



Merge Features

Merge is controlled by

- selector features X^+
- category features X^-

Merge is SL-2

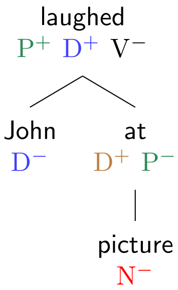
Merge is SL-2 over trees because we only need to look at

1. the mother and
2. its daughters

Merge is SL-2

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

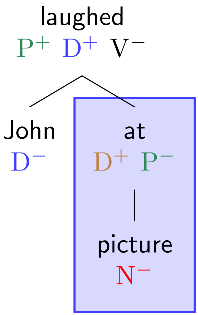
1. the mother and
2. its daughters



Merge is SL-2

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

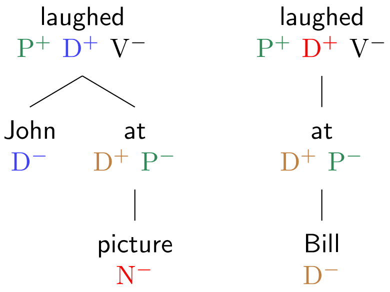
1. the mother and
2. its daughters



Merge is SL-2

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

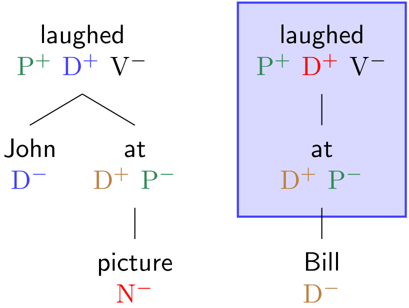
1. the mother and
2. its daughters



Merge is SL-2

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

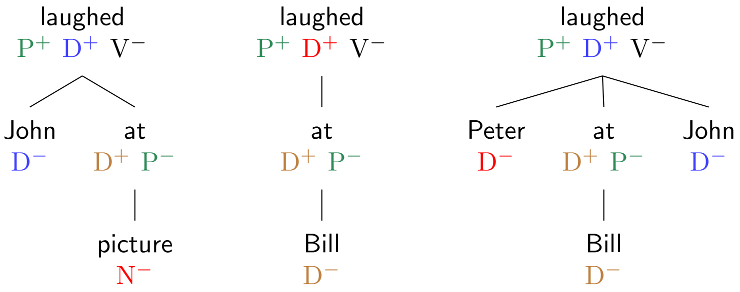
1. the mother and
2. its daughters



Merge is SL-2

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

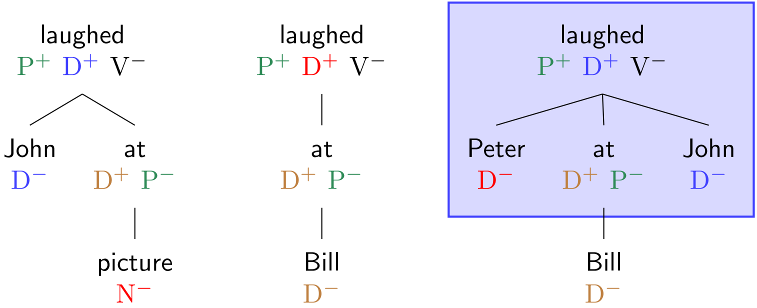
1. the mother and
2. its daughters



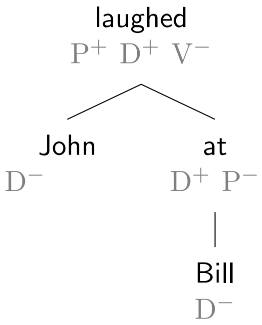
Merge is SL-2

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

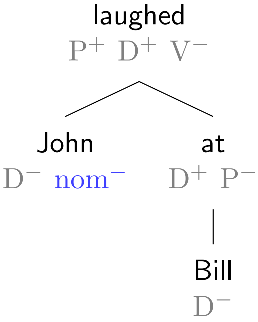
1. the mother and
2. its daughters



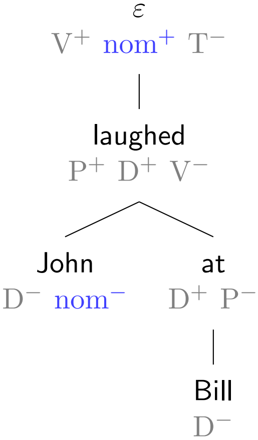
Movement is Local over Tree Tiers



Movement is Local over Tree Tiers

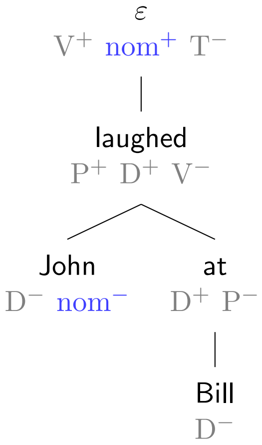


Movement is Local over Tree Tiers

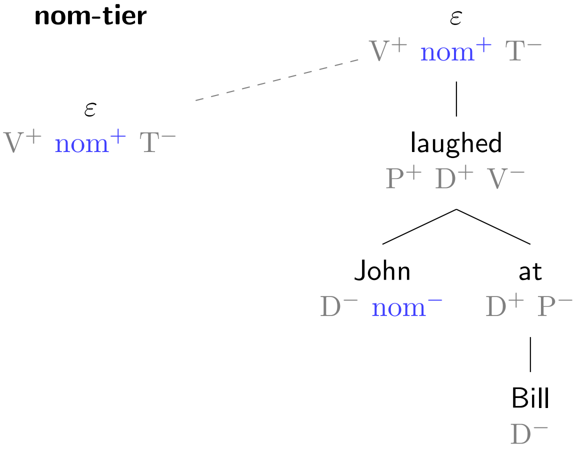


Movement is Local over Tree Tiers

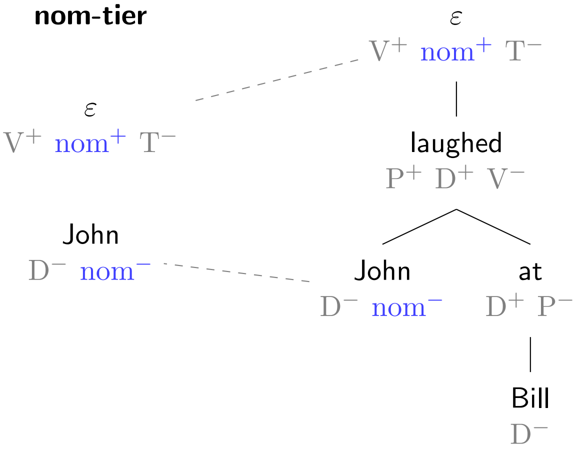
nom-tier



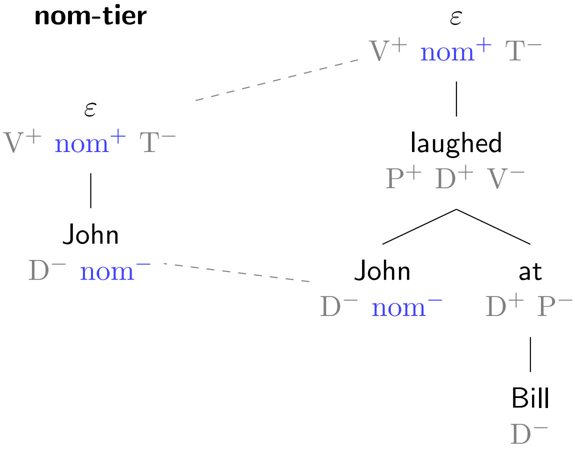
Movement is Local over Tree Tiers



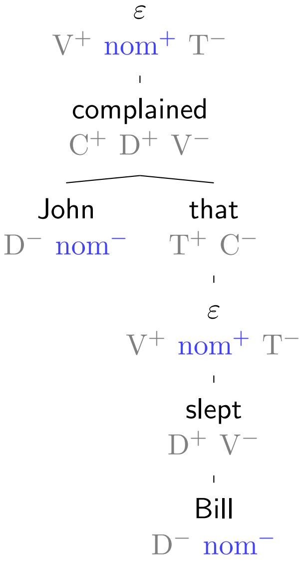
Movement is Local over Tree Tiers



Movement is Local over Tree Tiers

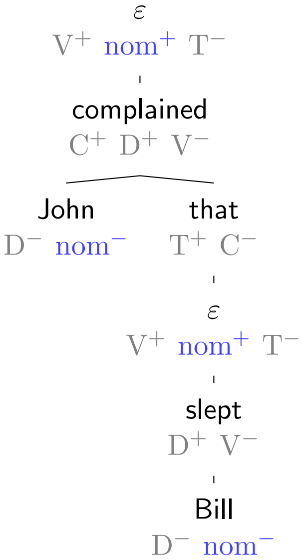


Tier with Multiple Movers

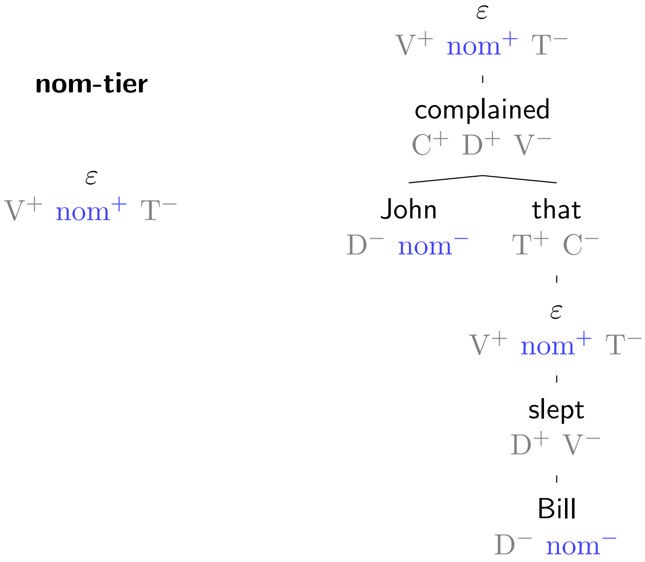


Tier with Multiple Movers

nom-tier

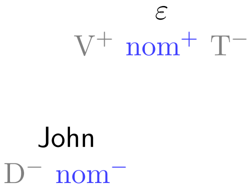


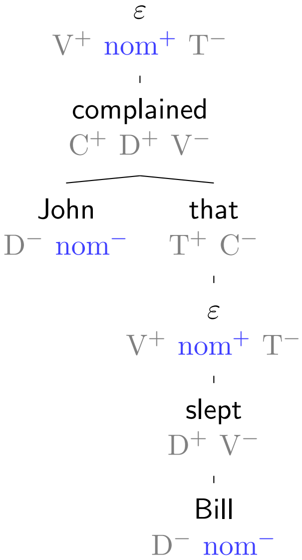
Tier with Multiple Movers



Tier with Multiple Movers

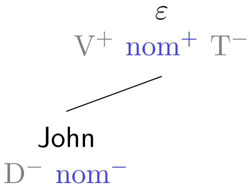
nom-tier

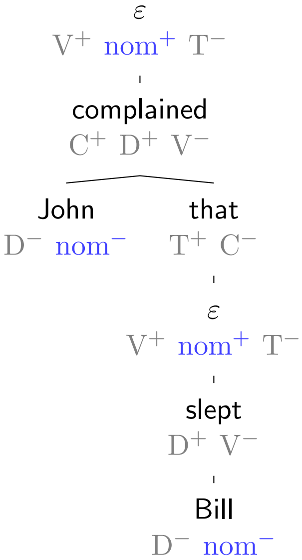




Tier with Multiple Movers

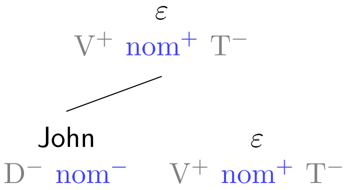
nom-tier

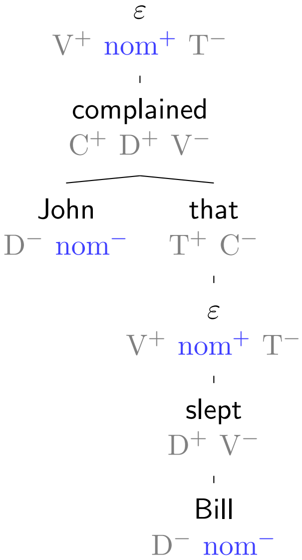




Tier with Multiple Movers

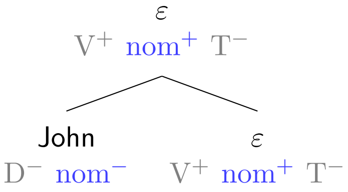
nom-tier

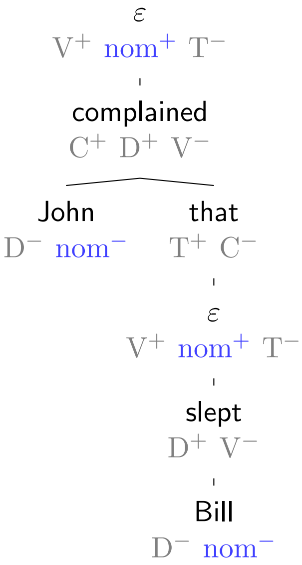




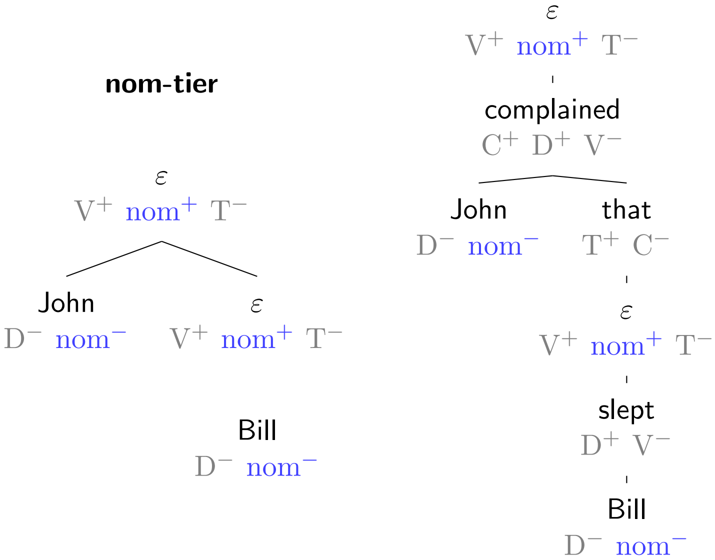
Tier with Multiple Movers

nom-tier



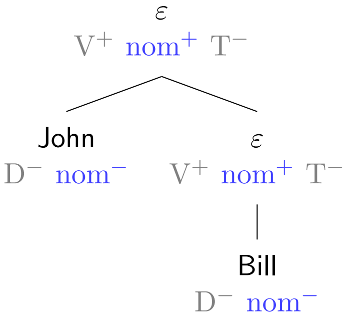


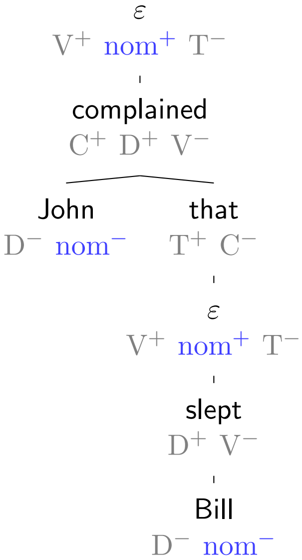
Tier with Multiple Movers



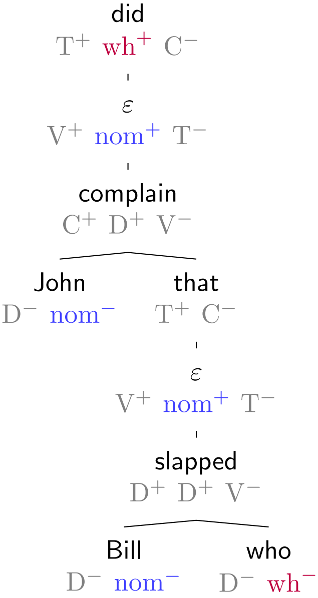
Tier with Multiple Movers

nom-tier

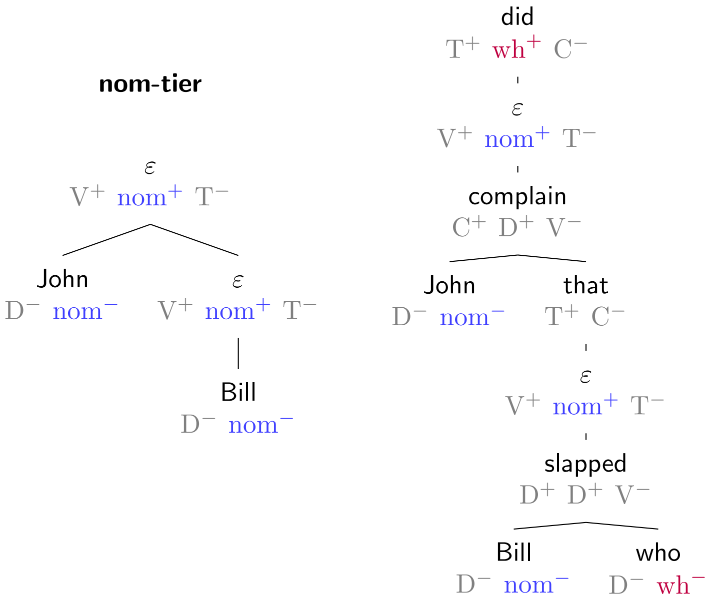




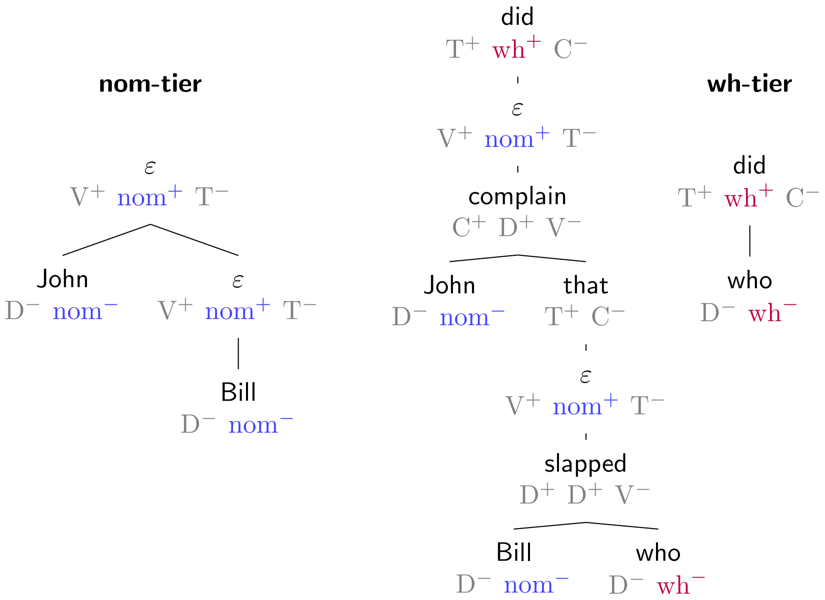
Separate Tier for Each Movement Type



Separate Tier for Each Movement Type



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

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
$\mathrm{T}^{+}\ \mathrm{wh}^{+}\ \mathrm{C}^{-}$
|
because
$\mathrm{T}^{+}\ \mathrm{V}^{\sim}$
|
which
$\mathrm{D}^{-}\ \mathrm{wh}^{-}$

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
T^{+} wh^{+} C^{-}

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

which
D^{-} wh^{-}

did
T^{+} wh^{+} C^{-}

fact
C^{+} N^{-}

which
D^{-} 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
T^{+} wh^{+} C^{-}
|
because
T^{+} $\mathrm{V}^{\sim}$
|
which
D^{-} wh^{-}

did
T^{+} wh^{+} C^{-}
|
fact
C^{+} N^{-}
|
which
D^{-} wh^{-}

did
T^{+} wh^{+} C^{-}
|
which
D^{-} 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."

Outline

Hidden Power of Merge Features

A Confession

- Subregularity does not limit anything!
- Merge can do pretty much anything you want.

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
Islands	all the ones mentioned before smuggle movers out of islands

Why?

- Complex constraints can be compiled into the features.
- Once compiled in, they are enforced via Merge.
- It's a generalized version of slash feature percolation.

Example: A grammar for modulo counting

$$\text{foo} :: \mathrm{O}^{-} \qquad \text{foo} :: \mathrm{E}^{+}\mathrm{O}^{-} \qquad \varepsilon :: \mathrm{O}^{+}\mathrm{C}^{-}$$

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

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

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

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

Local Feature Recoverability

- We need to restrict the power of Merge features, but how?
- Linguistic restrictions on categories don't work (morphology, distribution, semantics)

Feature Recoverability

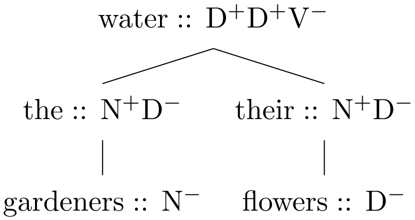
Merge features must be recoverable in an **SL** fashion.

Local Feature Recoverability

- We need to restrict the power of Merge features, but how?
- Linguistic restrictions on categories don't work (morphology, distribution, semantics)

Feature Recoverability

Merge features must be recoverable in an **SL** fashion.

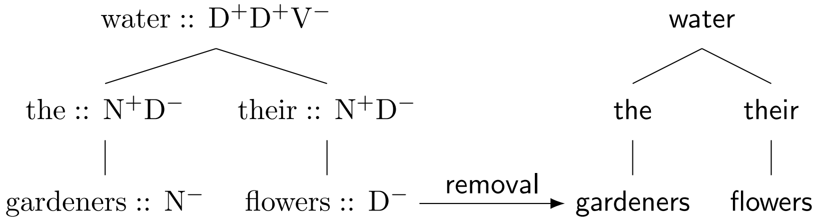


Local Feature Recoverability

- We need to restrict the power of Merge features, but how?
- Linguistic restrictions on categories don't work (morphology, distribution, semantics)

Feature Recoverability

Merge features must be recoverable in an **SL** fashion.

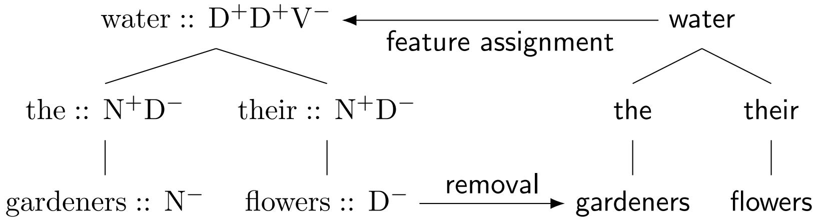


Local Feature Recoverability

- We need to restrict the power of Merge features, but how?
- Linguistic restrictions on categories don't work (morphology, distribution, semantics)

Feature Recoverability

Merge features must be recoverable in an **SL** fashion.



Modulo Counting is Not SL Recoverable

ε

bar

bar

foo

bar

bar

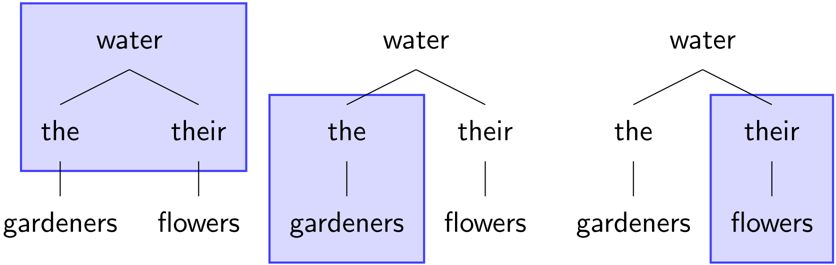
bar

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

Empirical Conjecture

SL-2 Recoverability Conjecture

For every lexical item l, the Merge features of l 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.
- Cognitive parallelism:
  Phonological feature inference equally complex (SL-1 or SL-2).

Open issues

- Needs to be tested across many languages
- Depends on theoretical assumptions
  - distribution of empty heads
  - lexical items fully inflected or bare roots?
- Move features cannot be inferred this way.

Towards a Learning Algorithm for Minimalism

- Categories are a major hurdle for syntactic learning algorithms.
- Feature recoverability opens up a new strategy.

A Learning Paradigm for Minimalist Syntax

1. **Input**
   - string (observed)
   - head-argument relations (basic semantic interpretation)
   - notion of feature recoverability (UG)
2. Construct feature-free dependency tree
3. Distributional learning of categories via recoverability (state merging)
4. Infer movement from string

Wrapping up: Concrete Results

- **Cognitive Parallelism**
  - Phonology and syntax are surprisingly similar.
  - SL and TSL play a central role in both.
  - Islands $\equiv$ Blocking
  - Both come for free
- **Specific Phenomenon: Subcategorization**
  - Linguists haven't paid enough attention to subcategorization.
  - Subregular complexity makes strong predictions about categories.

Subregular Linguistics as Linguistics

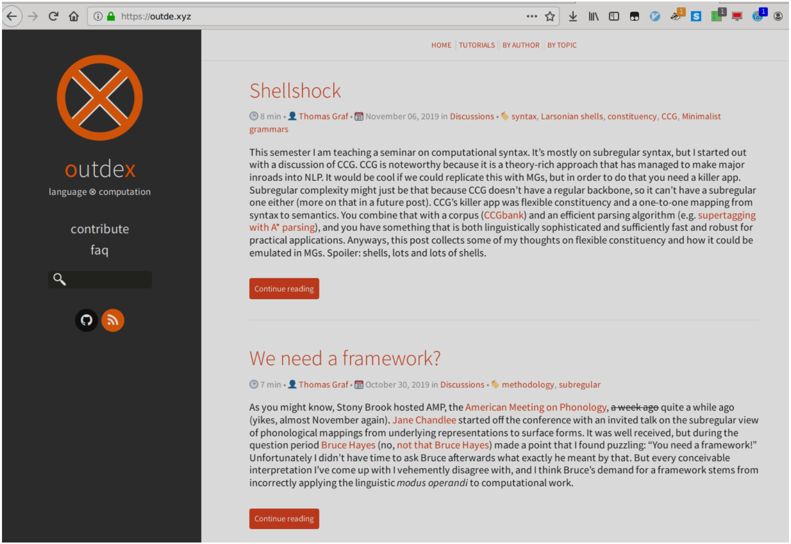
- Subregular **linguistics**
  - What are the laws of grammar?
  - How complex are they?
  - Why are those the laws?
  - Are some analyses simpler than others?
- Interplay of theory and data:
  - new typological claims
  - deeper understanding of formalism through data
  - new empirical questions
  - unification of diverse data points
  - learnability
  - direct ties to cognition
- It's just another tool. The more tools, the better!

Join the Program

Everybody has something to contribute!

- Do you have data that contradicts current predictions?
- Wanna add probabilities and gradience?
- In-depth analysis of specific phenomena
- grammar fragments
- artificial language learning experiments
- processing experiments

Follow Along (`https://outde.xyz`)



https://outde.xyz

outdex
language ⊗ computation

contribute
faq

HOME | TUTORIALS | BY AUTHOR | BY TOPIC

Shellshock

8 min • Thomas Graf • November 06, 2019 in Discussions • syntax, Larsonian shells, constituency, CCG, Minimalist grammars

This semester I am teaching a seminar on computational syntax. It's mostly on subregular syntax, but I started out with a discussion of CCG. CCG is noteworthy because it is a theory-rich approach that has managed to make major inroads into NLP. It would be cool if we could replicate this with MGs, but in order to do that you need a killer app. Subregular complexity might just be that because CCG doesn't have a regular backbone, so it can't have a subregular one either (more on that in a future post). CCG's killer app was flexible constituency and a one-to-one mapping from syntax to semantics. You combine that with a corpus (CCGbank) and an efficient parsing algorithm (e.g. supertagging with A* parsing), and you have something that is both linguistically sophisticated and sufficiently fast and robust for practical applications. Anyways, this post collects some of my thoughts on flexible constituency and how it could be emulated in MGs. Spoiler: shells, lots and lots of shells.

Continue reading

We need a framework?

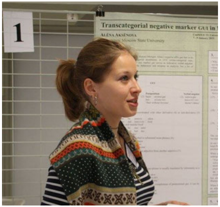
7 min • Thomas Graf • October 30, 2019 in Discussions • methodology, subregular

As you might know, Stony Brook hosted AMP, the American Meeting on Phonology, ~~a week ago~~ quite a while ago (yikes, almost November again). Jane Chandlee started off the conference with an invited talk on the subregular view of phonological mappings from underlying representations to surface forms. It was well received, but during the question period Bruce Hayes (no, not that Bruce Hayes) made a point that I found puzzling: "You need a framework!" Unfortunately I didn't have time to ask Bruce afterwards what exactly he meant by that. But every conceivable interpretation I've come up with I vehemently disagree with, and I think Bruce's demand for a framework stems from incorrectly applying the linguistic *modus operandi* to computational work.

Continue reading

Appendix

TSL Morphology



Alëna Aksënova

Sophie Moradi

- Joint work with Alëna Aksënova and Sophie Moradi.
- It seems that **morphotactics is also TSL**.
  (Aksënova et al. 2016)

Example: Unbounded *the day after*-Prefixation in German

- German has a prefix **über**.
- This prefix can be freely combined with *morgen* 'tomorrow'.

Example

morgen	tomorrow
über+*morgen*	the day after tomorrow
(**über**+)$^{\mathbf{n}}$*morgen*	(the day after)^n tomorrow

TSL Description

Tier: **über**, stem boundary +

Constraint	**Bigrams**
über must be prefix	*+ **über**

Example: Unbounded *the day after*-Prefixation in German

- German has a prefix **über**.
- This prefix can be freely combined with *morgen* 'tomorrow'.

Example

morgen	tomorrow
über+*morgen*	the day after tomorrow
(**über**+)$^{\mathbf{n}}$*morgen*	(the day after)^n tomorrow

TSL Description

Tier: **über**, stem boundary +

Constraint	**Bigrams**
über must be prefix	*+ **über**

Example: Unbounded *the day after*-Prefixation in German

- German has a prefix **über**.
- This prefix can be freely combined with *morgen* ‘tomorrow’.

Example

morgen	tomorrow
über+*morgen*	the day after tomorrow
(**über**+)$^{\mathbf{n}}$*morgen*	(the day after)^n tomorrow

TSL Description

Tier: **über**, stem boundary +

Constraint	**Bigrams**
über must be prefix	*+ **über**

$	**über**	**über**	+		+	**über**	$
$	**über**	**über**	+	*morgen*	+	**über**	$

Example: Bounded *the day after*-Circumfixation in Ilocano

- Ilocano has a circumfix **ka- -an**.
- This prefix can be combined once with *bigát* 'tomorrow'.

Example

bigát	tomorrow
ka+*bigát*+**an**	the day after tomorrow
*(**ka**)$^{\mathbf{n}}$+*bigát*+(**an**)$^{\mathbf{n}}$	(the day after)^n tomorrow

TSL Description

Tier: *ka*, *an*, stem boundary +

Constraint	Bigrams
ka must be prefix	*+ **ka**
an must be suffix	***an** +
ka before **an**	***an ka**
no iteration	***ka ka**, ***an an**
no lonely affix	***ka** ++ \$, *\$++ **an**

Example: Bounded *the day after*-Circumfixation in Ilocano

- Ilocano has a circumfix **ka- -an**.
- This prefix can be combined once with *bigát* ‘tomorrow’.

Example

bigát	tomorrow
ka+*bigát*+**an**	the day after tomorrow
*(**ka**)^{n}+*bigát*+(**an**)^{n}	(the day after)^{n} tomorrow

TSL Description

Tier: *ka*, *an*, stem boundary +

Constraint	**Bigrams**
ka must be prefix	*+ **ka**
an must be suffix	***an** +
ka before **an**	***an ka**
no iteration	***ka ka**, ***an an**
no lonely affix	***ka** ++ \$, *\$++ **an**

Example: Bounded *the day after*-Circumfixation in Ilocano

- Ilocano has a circumfix **ka- -an**.
- This prefix can be combined once with *bigát* ‘tomorrow’.

Example

bigát	tomorrow
ka+*bigát*+**an**	the day after tomorrow
*(**ka**)^{n}+*bigát*+(**an**)^{n}	(the day after)^{n} tomorrow

TSL Description

Tier: *ka*, *an*, stem boundary +

Constraint	**Bigrams**
ka must be prefix	*+ **ka**
an must be suffix	***an** +
ka before **an**	***an ka**
no iteration	***ka ka**, ***an an**
no lonely affix	***ka** ++ \$, *\$++ **an**

$ an ka ka +   + $
| | | | | | |
$ an ka ka + *bigát* + $

Typological Gap: No Unbounded Circumfixation

- There seems to be no language with an affix that is
  - freely iterable like German **über**, and
  - a circumfix like **ka- -an** in Ilocano.
- Why this gap? Because the **result would not be TSL**!

Explanation

- The pattern would be $\textbf{ka}^{n}+\textit{bigát}+\textbf{an}^{n}$.
- TSL cannot memorize exact numbers.
- All affixes would have to be visible in the same search window.
- But the window's size is bounded, while the pattern is not.

Typological Gap: No Unbounded Circumfixation

- There seems to be no language with an affix that is
  - freely iterable like German **über**, and
  - a circumfix like **ka- -an** in Ilocano.
- Why this gap? Because the **result would not be TSL**!

Explanation

- The pattern would be $\mathbf{ka}^{\mathrm{n}}+\textit{bigát}+\mathbf{an}^{\mathrm{n}}$.
- TSL cannot memorize exact numbers.
- All affixes would have to be visible in the same search window.
- But the window's size is bounded, while the pattern is not.

TSL Morpho-Semantics?

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

Case Study: Generalized Quantifiers

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

Quantifier Languages (van Benthem 1986)

(5) a. Every student cheated.
b. No student cheated.
c. Some student cheated.
d. Three students cheated.

students	John	Mary	Sue
cheated	yes	no	yes
string	Y	N	Y

- (5a): **False**, because the string contains a N
- (5b): **False**, because the string contains a Y
- (5c): **True**, because the string contains a Y
- (5d): **False**, because the string does not contain three Ys

Quantifier Languages (van Benthem 1986)

(5) a. Every student cheated.
b. No student cheated.
c. Some student cheated.
d. Three students cheated.

students	John	Mary	Sue
cheated	yes	no	yes
string	Y	N	Y

- (5a): **False**, because the string contains a N
- (5b): **False**, because the string contains a Y
- (5c): **True**, because the string contains a Y
- (5d): **False**, because the string does not contain three Ys

TSL Descriptions for Quantifier Languages

Quantifier	Constraint	n-grams	Tier
every	$\|N\| = 0$	*N	none
no	$\|Y\| = 0$	*Y	none
some	$\|Y\| \geq 1$	*$$	Y
at least **n**	$\|Y\| \geq \mathbf{n}$	$*\$1^{\mathbf{m}}\$$ ($\mathbf{m} < \mathbf{n}$)	Y
at most **n**	$\|Y\| \leq \mathbf{n}$	$*Y^{\mathbf{n}+1}$	Y

Example

$$\begin{array}{ccccc} \$ & Y & & Y & \$ \\ | & | & & | & | \\ \$ & Y & N & Y & \$ \end{array}$$

some	*$$	**True**
at least 2	*$$, *Y	**True**
at least 3	*$$, *Y, *YY	**False**
at most 2	*YYY	**True**

TSL Descriptions for Quantifier Languages

Quantifier	Constraint	n-grams	Tier
every	$\lvert \mathsf{N} \rvert = 0$	$^*\mathsf{N}$	none
no	$\lvert \mathsf{Y} \rvert = 0$	$^*\mathsf{Y}$	none
some	$\lvert \mathsf{Y} \rvert \geq 1$	$^*\$\$$	Y
at least **n**	$\lvert \mathsf{Y} \rvert \geq \mathbf{n}$	$^*\$1^{\mathbf{m}}\$$ ($\mathbf{m} < \mathbf{n}$)	Y
at most **n**	$\lvert \mathsf{Y} \rvert \leq \mathbf{n}$	$^*\mathsf{Y}^{\mathbf{n}+1}$	Y

Example

$$\begin{array}{ccccc} \$ & \mathsf{Y} & & \mathsf{Y} & \$ \\ | & | & & | & | \\ \$ & \mathsf{Y} & \mathsf{N} & \mathsf{Y} & \$ \end{array}$$

some	$^*\$\$$	**True**
at least 2	$^*\$\$$, $^*\$\mathsf{Y}\$$	**True**
at least 3	$^*\$\$$, $^*\$\mathsf{Y}\$$, $^*\$\mathsf{YY}\$$	**False**
at most 2	$^*\mathsf{YYY}$	**True**

TSL Descriptions for Quantifier Languages

Quantifier	**Constraint**	**n-grams**	**Tier**
every	$\vert\mathsf{N}\vert = 0$	*N	none
no	$\vert\mathsf{Y}\vert = 0$	*Y	none
some	$\vert\mathsf{Y}\vert \geq 1$	*$$	Y
at least **n**	$\vert\mathsf{Y}\vert \geq \mathbf{n}$	$*\$1^{\mathbf{m}}\$$ ($\mathbf{m} < \mathbf{n}$)	Y
at most **n**	$\vert\mathsf{Y}\vert \leq \mathbf{n}$	$*\mathsf{Y}^{\mathbf{n}+1}$	Y

Example

$ Y Y $

$ Y N Y $

some	*$$	**True**
at least 2	*$$, *Y	**True**
at least 3	*$$, *Y, *YY	**False**
at most 2	*YYY	**True**

Overview of Quantifier Languages

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

Quantifier	**TSL?**	**Tier**	**Mono.**
every	yes	none	yes
no	yes	none	yes
some	yes	Y	yes
(at least) two	yes	Y	yes
(at most) two	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
even number	no		no
prime number	no		no
infinitely many	no		no
most	no		???

(Paperno 2011)

Overview of Quantifier Languages

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

(Paperno 2011)

Quantifier	TSL?	Tier	Mono.
every	yes	none	yes
no	yes	none	yes
some	yes	Y	yes
(at least) two	yes	Y	yes
(at most) two	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
even number	no		no
prime number	no		no
infinitely many	no		no
most	no		???

Overview of Quantifier Languages

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

(Paperno 2011)

Quantifier	TSL?	Tier	Mono.
every	yes	none	yes
no	yes	none	yes
some	yes	Y	yes
(at least) two	yes	Y	yes
(at most) two	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
even number	no		no
prime number	no		no
infinitely many	no		no
most	no		???

Overview of Quantifier Languages

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

(Paperno 2011)

Quantifier	TSL?	Tier	Mono.
every	yes	none	yes
no	yes	none	yes
some	yes	Y	yes
(at least) two	yes	Y	yes
(at most) two	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
even number	no		no
prime number	no		no
infinitely many	no		no
most	no		???

The Case of *most*

There is good semantic evidence that "most" is internally complex and hence **not monomorphemic**. (Hackl 2009)

Quantifier	TSL?	Tier	Mono.
every	yes	none	yes
no	yes	none	yes
some	yes	Y	yes
(at least) two	yes	Y	yes
(at most) two	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
even number	no		no
prime number	no		no
infinitely many	no		no
most	no		no

References I

Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 121–130. URL https://www.aclweb.org/anthology/W/W16/W16-2019.pdf.

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

van Benthem, Johan. 1986. Semantic automata. In *Essays in logical semantics*, 151–176. Dordrecht: Springer.

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

Hackl, Martin. 2009. On the grammar and processing of proportional quantifiers: Most versus more than half. *Natural Language Semantics* 17:63–98.

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.

Jardine, Adam, and Kevin McMullin. 2017. Efficient learning of tier-based strictly k-local languages. In *Proceedings of Language and Automata Theory and Applications*, Lecture Notes in Computer Science, 64–76. Berlin: Springer. URL https://doi.org/10.1007/978-3-319-53733-7_4.

References II

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

Kaplan, Ronald M., and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378. URL http://www.aclweb.org/anthology/J94-3001.pdf.

Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. In *COLING'92*, 141–148. URL http://www.aclweb.org/anthology/C92-1025.

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

Paperno, Denis. 2011. Learnable classes of natural language quantifiers: Two perspectives. URL http://paperno.bol.ucla.edu/q_learning.pdf, ms., UCLA.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–345. URL http://dx.doi.org/10.1007/BF00630917.

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.

References III

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