It's a (Sub-)Regular Conspiracy

Locality and Computation in Phonology Morphology, Syntax, and Semantics

Thomas Graf

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

You can get the slides here under "News"

CLS
May 26, 2017

The Big Linguistic Questions

- What are the laws that govern each structural level?
- 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 Me

- 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?
- 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 Me

- 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. DM)
- 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

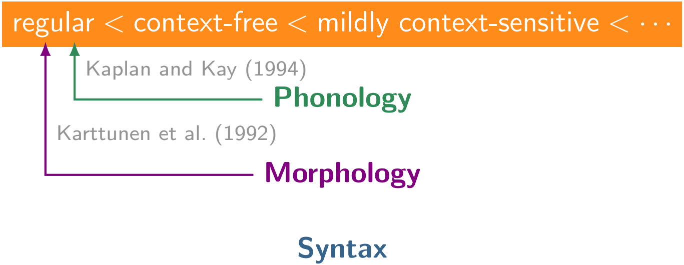
Morphology

Syntax

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

A Mathematical Distinctness Theorem

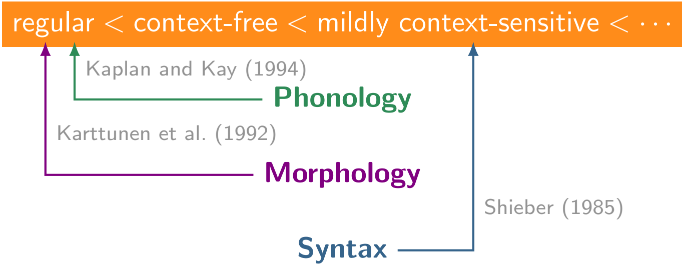
- From a computational perspective, there is a split between "P-side" and "S-side".



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

The Subregular Conspiracy...

- 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**
  - relativized locality plays a major role,
  - and is approximated by the formal class **TSL**.

Subregular Conspiracy

- TSL crops up everywhere.
- TSL is shockingly useful.

Outline

TSL: Tier-Based Strictly Local

- There are a variety of subregular classes to choose from.
- TSL is among the weaker ones.
- TSL works 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.

TSL: Tier-Based Strictly Local

- There are a variety of subregular classes to choose from.
- TSL is among the weaker ones.
- TSL works 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

- 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

- 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

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

Example: German

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

***z$**

***v$**

***d$**

Example: Word-Final Devoicing

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

Example: German

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

***z$**

***v$**

***d$**

Example: Word-Final Devoicing

- 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

- 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

- 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 $\boxed{\text{d \$}}$ *v$

*d$

Example: Word-Final Devoicing

- 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

- 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

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

Example: German

* $ r a d $

*z$
*v$
*d$

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

Example: Word-Final Devoicing

- 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 $\boxed{\text{a t}}$ **$**
	***d$**	

Example: Word-Final Devoicing

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

Example: German

* $ r a d $

*z$
*v$
*d$

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

Example: Intervocalic Voicing

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

Example

* \$ a z u s a \$

Example: Intervocalic Voicing

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

Example

* $ **a** z **u** **s** **a** $

Example: Intervocalic Voicing

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

Example

$* \boxed{\$\ \text{a}\ \text{z}}\ \text{u}\ \text{s}\ \text{a}\ \$$

Example: Intervocalic Voicing

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

Example

* \$ a z u s a \$

Example: Intervocalic Voicing

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

Example

$*$ \$ **a** $\boxed{\text{z } \textbf{u} \textbf{ s}}$ **a** \$

Example: Intervocalic Voicing

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

Example

$$* \$ \; a \; z \; \boxed{u \; s \; a} \; \$$$

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

*$ 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

*$ 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

*$ 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

* $ 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

* $ 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

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

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

*$ $\boxed{\text{s t a j a n o w o n w a ʃ}}$ $

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**.
- Tier projection is determined by the segments, not their environment. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

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

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

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**.
- Tier projection is determined by the segments, not their environment. (Heinz et al. 2011)

Jeff Heinz

Example: Samala Revisited

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

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

Making Long-Distance Dependencies Local

- Let's take a clue from phonology: create locality with **tiers**.
- Tier projection is determined by the segments, not their environment. (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**.
- Tier projection is determined by the segments, not their environment. (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**.
- Tier projection is determined by the segments, not their environment. (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**.
- Tier projection is determined by the segments, not their environment. (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**.
- Tier projection is determined by the segments, not their environment. (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**.
- Tier projection is determined by the segments, not their environment. (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**.
- Tier projection is determined by the segments, not their environment. (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 $
```

Why is TSL Interesting?

- Linguistically natural
- Correct and very 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 \$

Why is TSL Interesting?

- Linguistically natural
- Correct and very 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 $

Why is TSL Interesting?

- Linguistically natural
- Correct and very 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 $

Why is TSL Interesting?

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

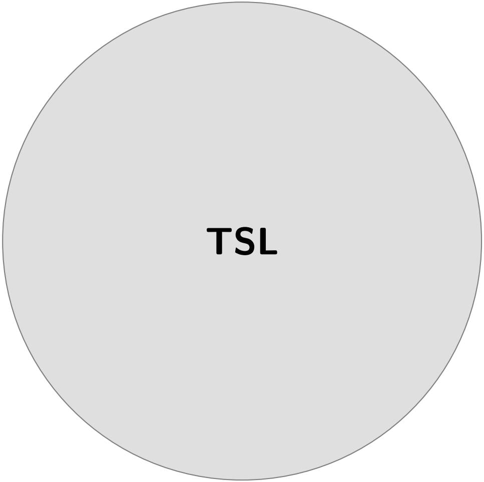
Example: First-Last Harmony

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

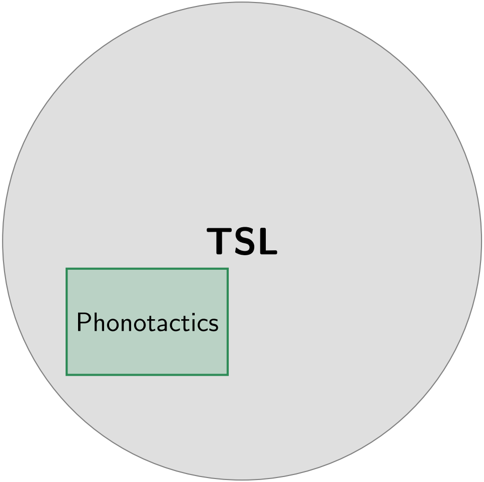
$ s ʃ $ $ s ʃ $

$ 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 $

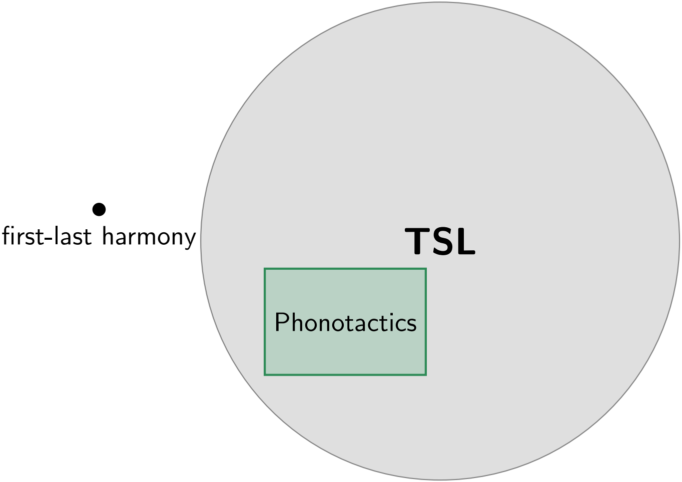
Place of Phonotactics



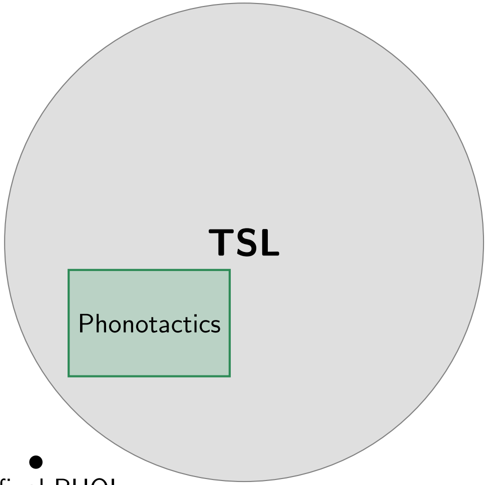
Place of Phonotactics



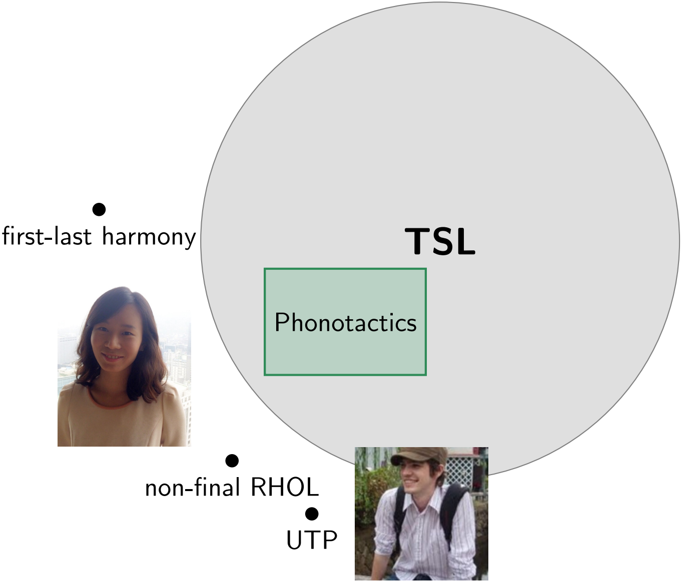
Place of Phonotactics



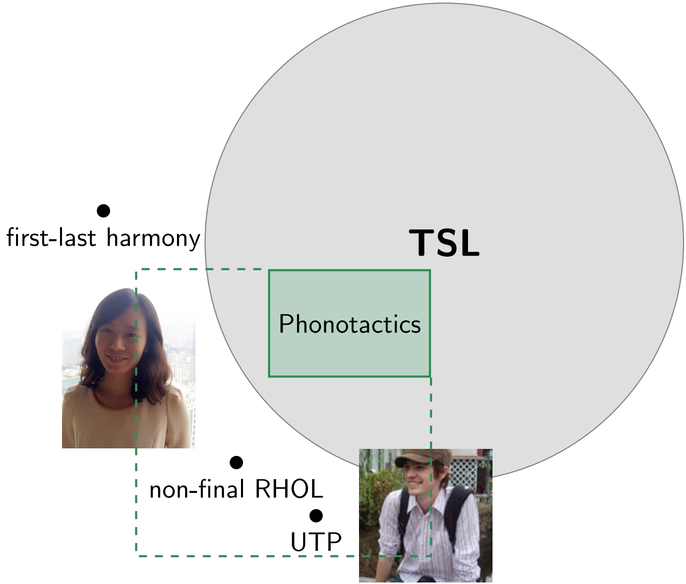
Place of Phonotactics



Place of Phonotactics



Place of Phonotactics



Going Beyond Phonology

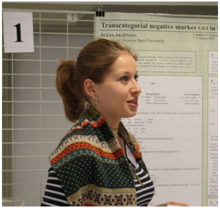
TSL provides a good fit for phonological dependencies.

The $\$10^6$ Question

Is TSL also a good fit for other linguistic structures?

- Morphology?
- (Morpho-)Semantics?
- Syntax?

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**

$$
\begin{array}{cccccccc}
\$ & \textbf{über} & \textbf{über} & + & & + & \textbf{über} & \$ \\
| & | & | & | & & | & | & | \\
\$ & \textbf{über} & \textbf{über} & + & \textit{morgen} & + & \textbf{über} & \$
\end{array}
$$

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.

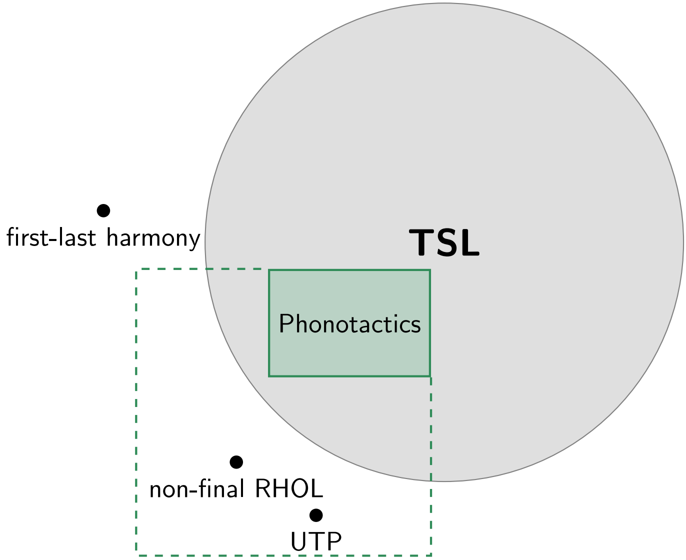
Typological Gap: No Unbounded Circumfixation

- 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**!

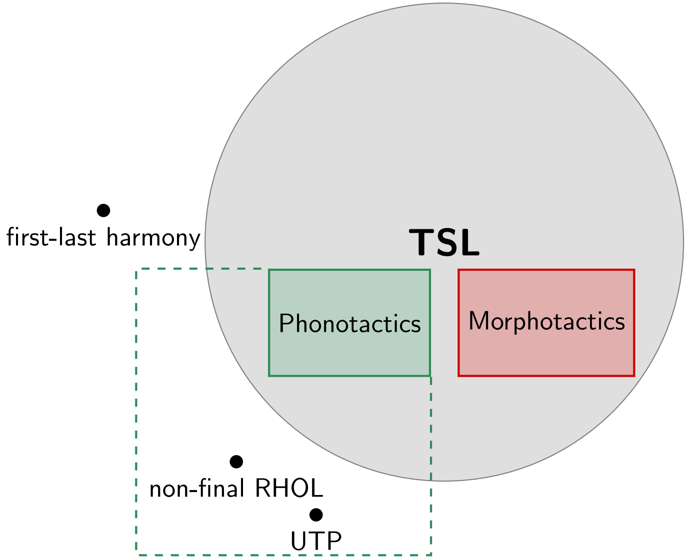
Explanation

- The pattern would be $\textbf{ka}^{\mathbf{n}} + \textit{bigát} + \textbf{an}^{\mathbf{n}}$.
- 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.

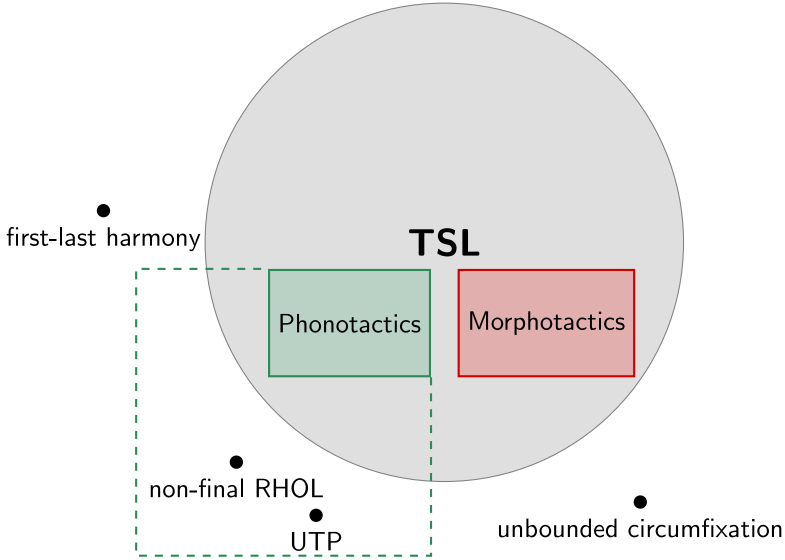
Place of Morphotactics



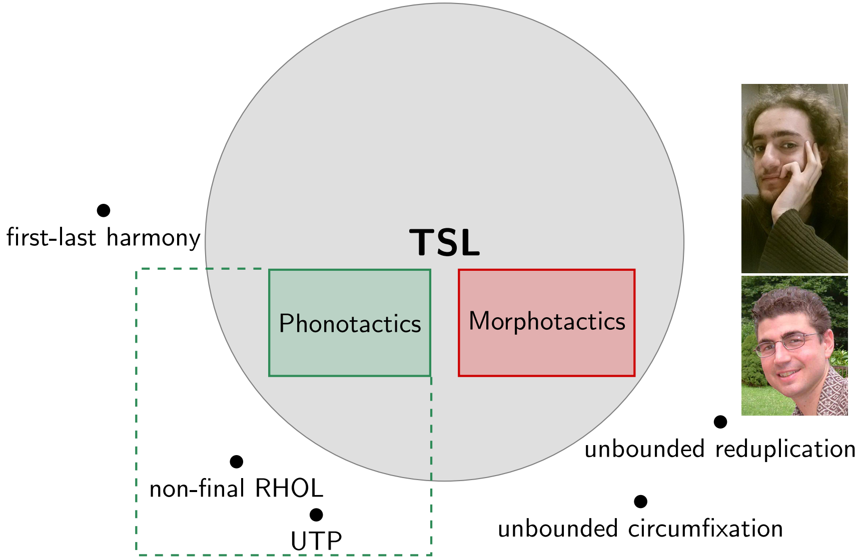
Place of Morphotactics



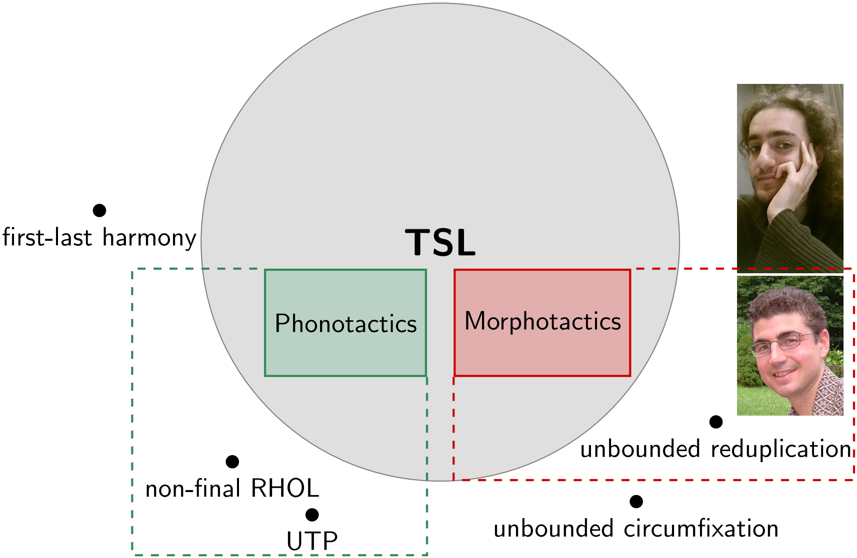
Place of Morphotactics



Place of Morphotactics



Place of Morphotactics



TSL Morpho-Semantics?

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

Case Study: Generalized Quantifiers (Graf 2017d)

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

Quantifier Languages (van Benthem 1986)

(1) 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

- (1a): **False**, because the string contains a N
- (1b): **False**, because the string contains a Y
- (1c): **True**, because the string contains a Y
- (1d): **False**, because the string does not contain three Ys

Quantifier Languages (van Benthem 1986)

(1) 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

- (1a): **False**, because the string contains a N
- (1b): **False**, because the string contains a Y
- (1c): **True**, because the string contains a Y
- (1d): **False**, because the string does not contain three Ys

TSL Descriptions for Quantifier Languages

Quantifier	Constraint	n-grams	Tier
every	$\lvert\mathsf{N}\rvert = 0$	*N	none
no	$\lvert\mathsf{Y}\rvert = 0$	*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

```
$  Y     Y  $
|  |     |  |
$  Y  N  Y  $
```

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	$\lvert\text{N}\rvert = 0$	*N	none
no	$\lvert\text{Y}\rvert = 0$	*Y	none
some	$\lvert\text{Y}\rvert \geq 1$	*$$	Y
at least **n**	$\lvert\text{Y}\rvert \geq \mathbf{n}$	$*\$1^{\mathbf{m}}\$\ (\mathbf{m} < \mathbf{n})$	Y
at most **n**	$\lvert\text{Y}\rvert \leq \mathbf{n}$	$*\text{Y}^{\mathbf{n}+1}$	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**

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		???

Overview of Quantifier Languages

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

(Paperno 2011)

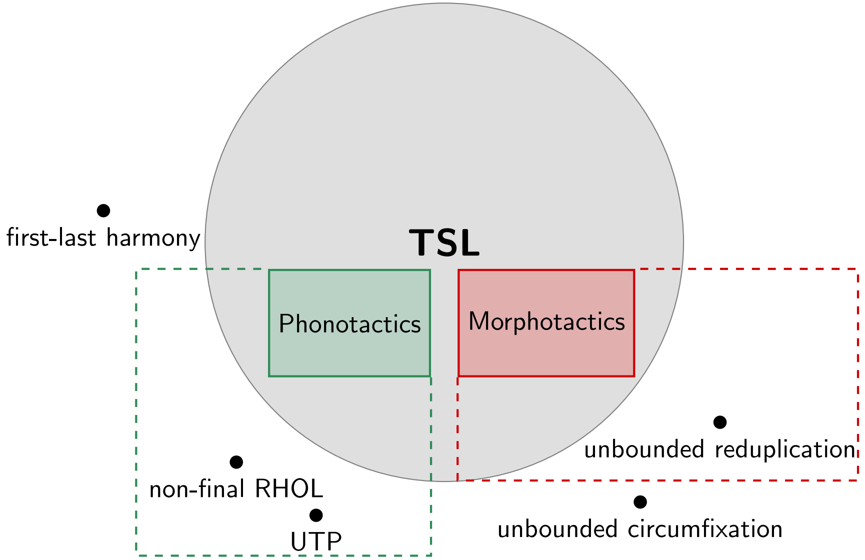
Quantifier	TSL?	Tier	Mono.
every	yes	none	yes
no	yes	none	yes
some	yes	Y	yes
(at least) two	yes	Y	yes
(at most) two	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
even number	no		no
prime number	no		no
infinitely many	no		no
most	no		???

The Case of *most*

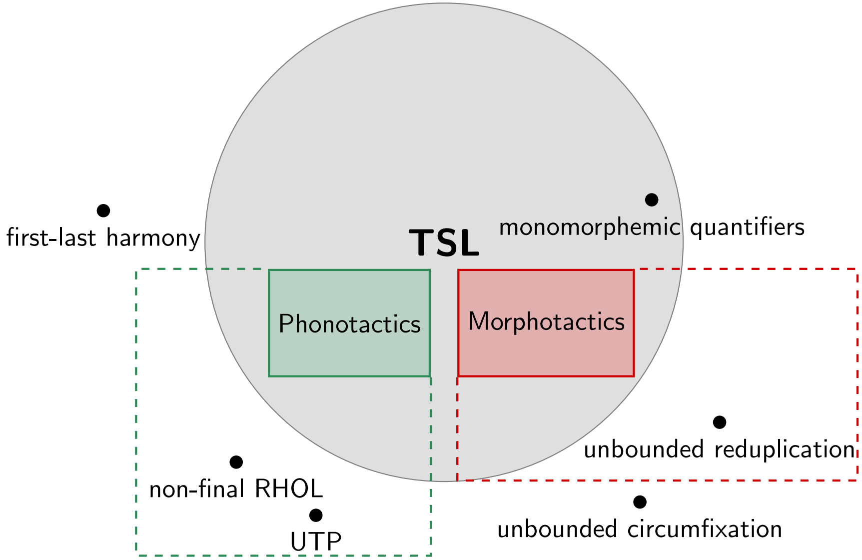
There is good semantic evidence that "most" is internally complex and hence **not monomorphemic**. (Hackl 2009)

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

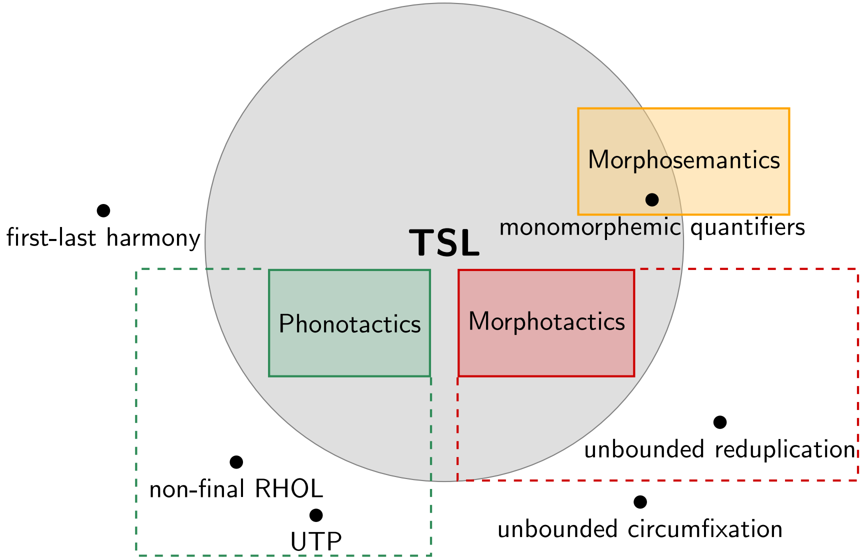
Place of Morphosemantics



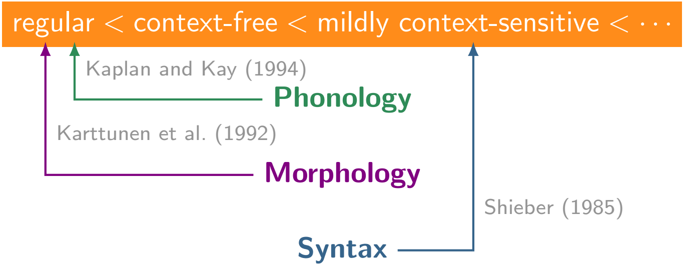
Place of Morphosemantics



Place of Morphosemantics



Against the Received View



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

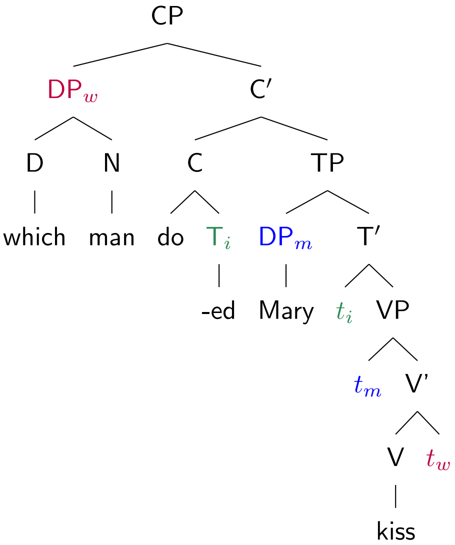
Minimalist Grammars

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

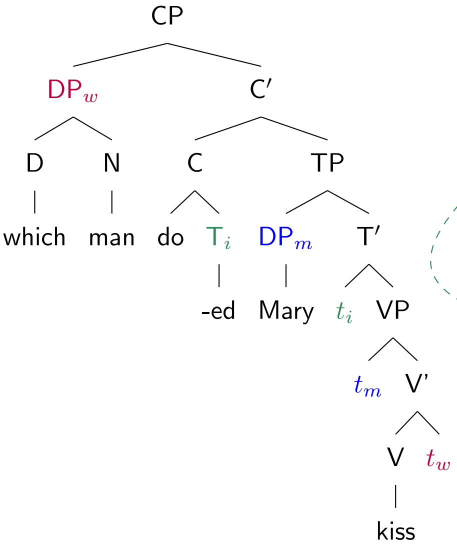
Chemistry	Syntax
atoms	words
electrons	features
molecules	sentences

MG Syntax in Action

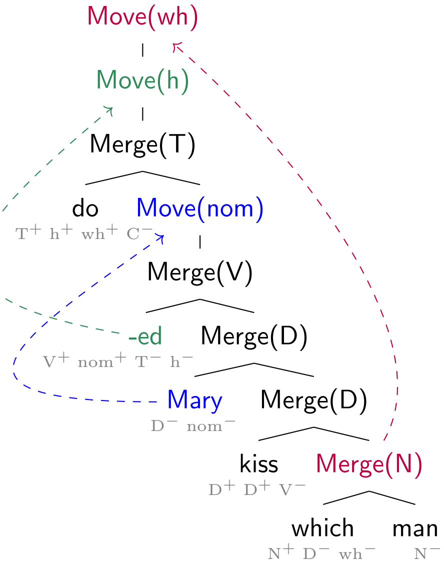


Phrase Structure Tree

MG Syntax in Action



Phrase Structure Tree

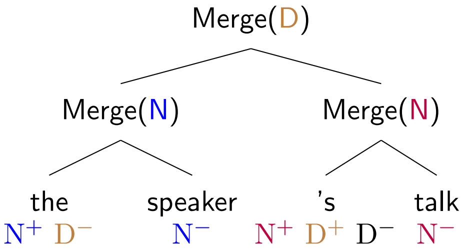


Derivation Tree

The Central Role of Derivation Trees

- Derivation trees are rarely considered in generative syntax.
  (but see Epstein et al. 1998)
- Satisfy Chomsky's structural desiderata:
  - no linear order
  - label-free
  - extension condition
  - inclusiveness condition
- Contain all information to produce phrase structure trees
  $\Rightarrow$ **central data structure** of Minimalist syntax

Merge is TSL

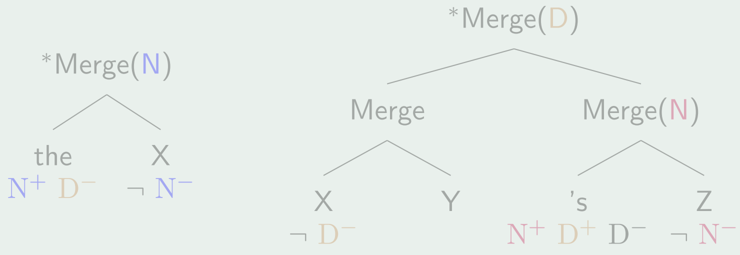


- The selector features of the head have to match the category features of the arguments.
- Since every head has a bounded number of arguments, the **distance between those features is bounded**.
- So Merge establishes only local dependencies.

Tier-Less Description for Merge

- We need to lift string n-grams to **tree n-grams**.
- Instead of strings of length n, use subtrees of depth n.
- Each subtree encodes a constraint on the derivation.

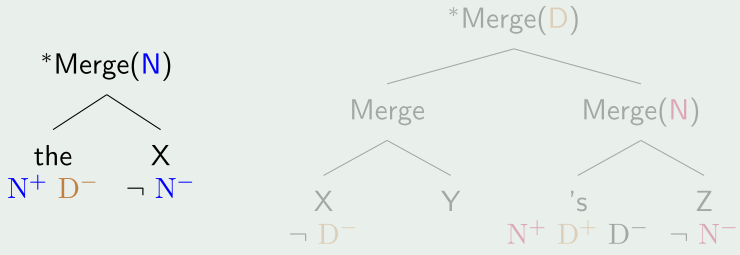
Example



Tier-Less Description for Merge

- We need to lift string n-grams to **tree n-grams**.
- Instead of strings of length n, use subtrees of depth n.
- Each subtree encodes a constraint on the derivation.

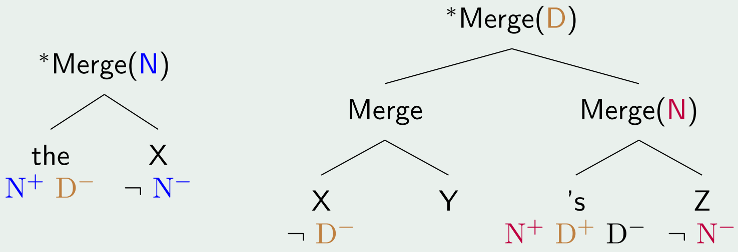
Example



Tier-Less Description for Merge

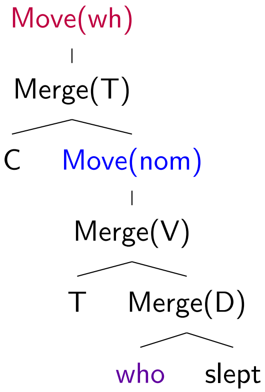
- We need to lift string n-grams to **tree n-grams**.
- Instead of strings of length n, use subtrees of depth n.
- Each subtree encodes a constraint on the derivation.

Example



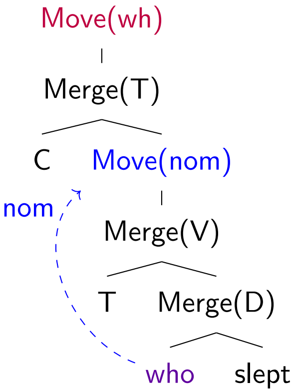
Move: Single Movement Normal Form

- **Assumption:** every phrase at most one movement feature
- Intermediate landing sites not feature-triggered (Graf et al. 2016)



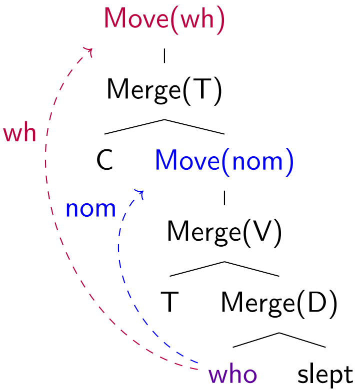
Move: Single Movement Normal Form

- **Assumption:** every phrase at most one movement feature
- Intermediate landing sites not feature-triggered
  (Graf et al. 2016)



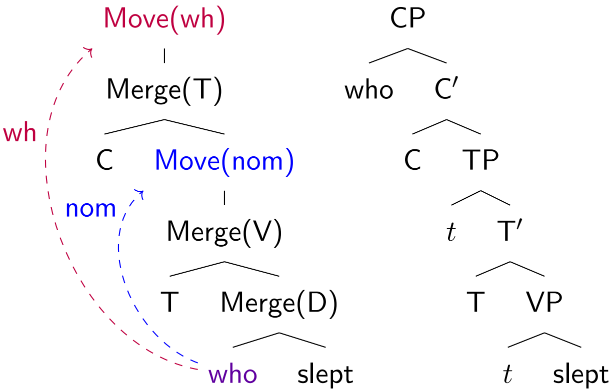
Move: Single Movement Normal Form

- **Assumption:** every phrase at most one movement feature
- Intermediate landing sites not feature-triggered
  (Graf et al. 2016)



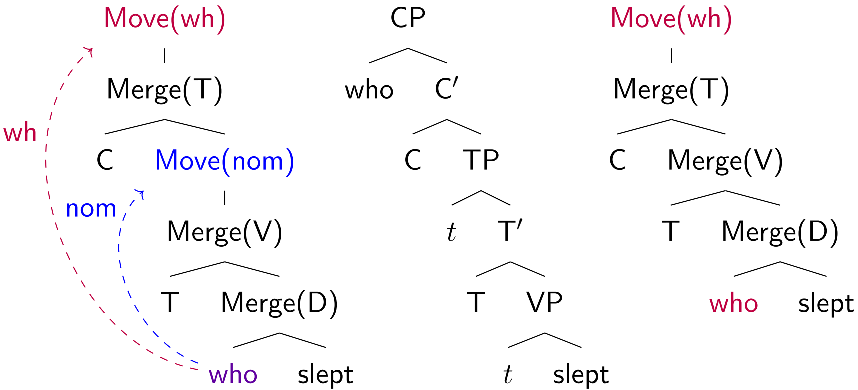
Move: Single Movement Normal Form

- **Assumption:** every phrase at most one movement feature
- Intermediate landing sites not feature-triggered
  (Graf et al. 2016)



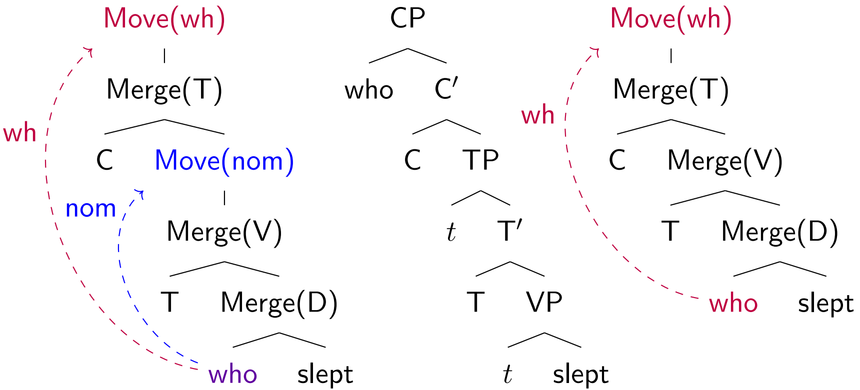
Move: Single Movement Normal Form

- **Assumption:** every phrase at most one movement feature
- Intermediate landing sites not feature-triggered (Graf et al. 2016)



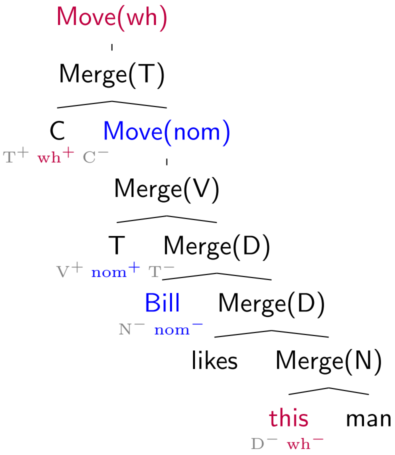
Move: Single Movement Normal Form

- **Assumption:** every phrase at most one movement feature
- Intermediate landing sites not feature-triggered (Graf et al. 2016)



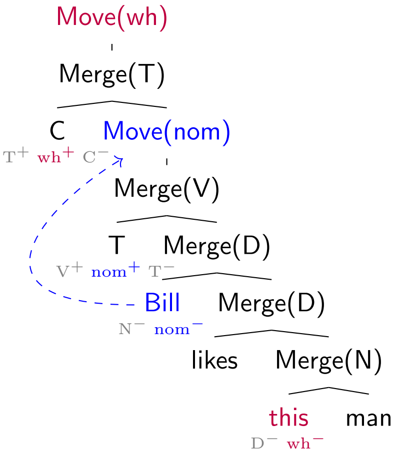
Move Tiers

- Movement is not unbounded.
- But maybe it is still TSL?



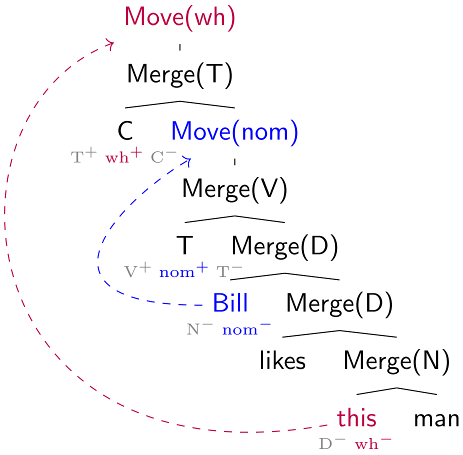
Move Tiers

- Movement is not unbounded.
- But maybe it is still TSL?



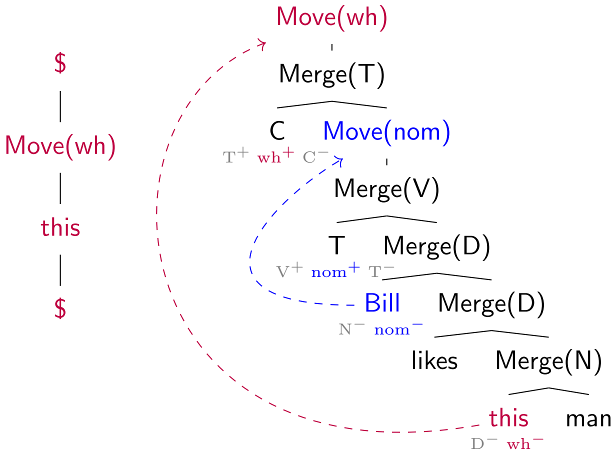
Move Tiers

- Movement is not unbounded.
- But maybe it is still TSL?



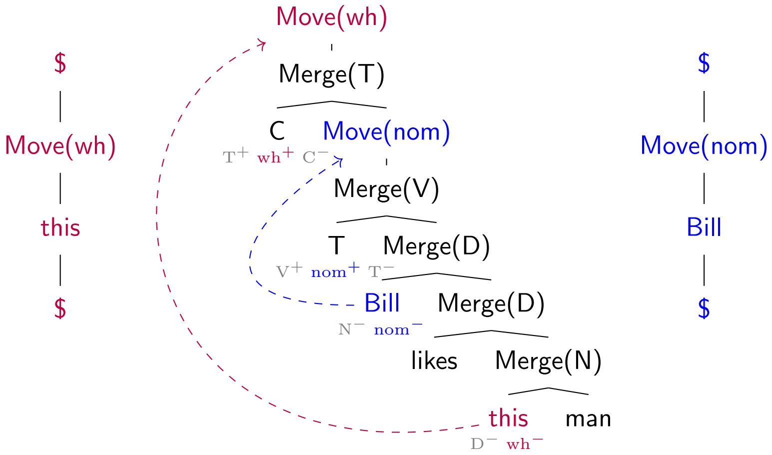
Move Tiers

- Movement is not unbounded.
- But maybe it is still TSL?

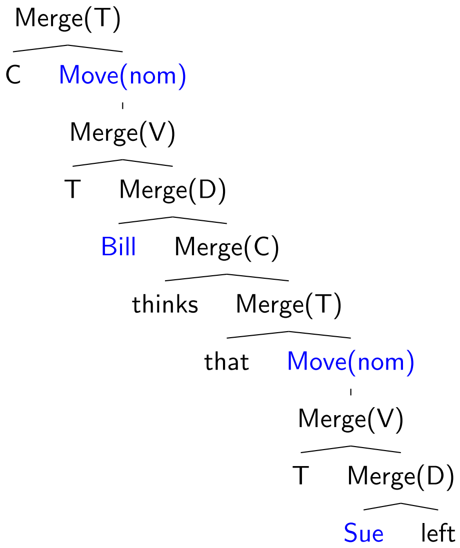


Move Tiers

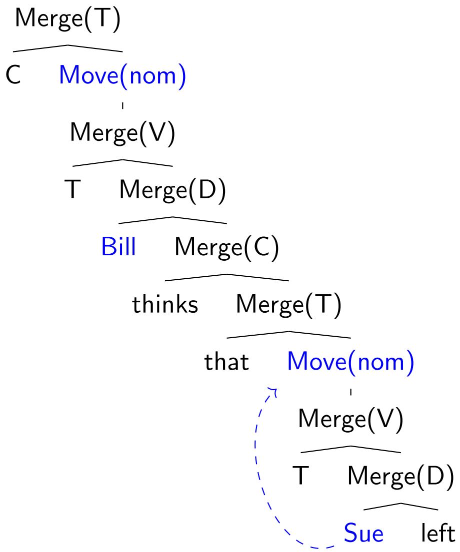
- Movement is not unbounded.
- But maybe it is still TSL?



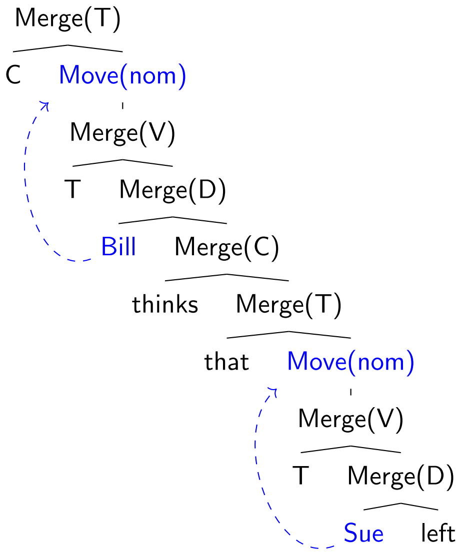
A Tier With Multiple Movers



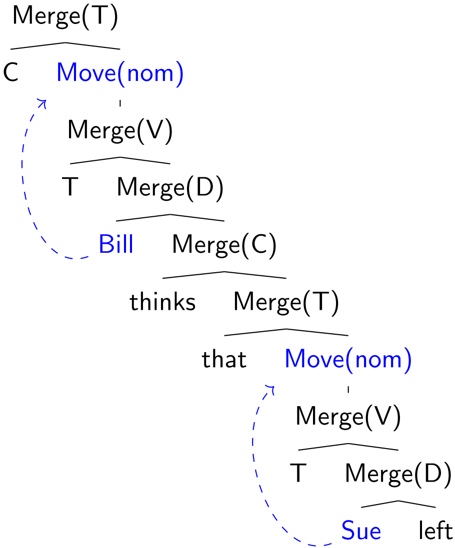
A Tier With Multiple Movers

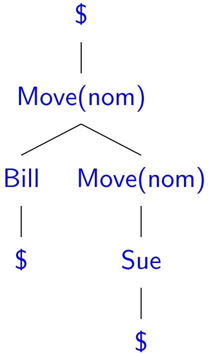


A Tier With Multiple Movers

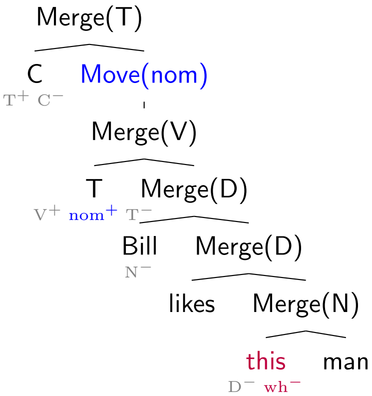


A Tier With Multiple Movers



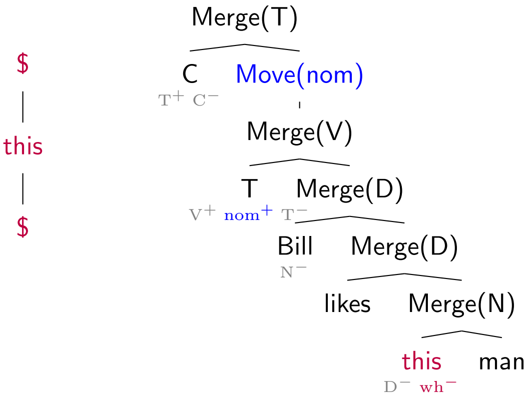


Blocking Simple Cases of Illicit Movement



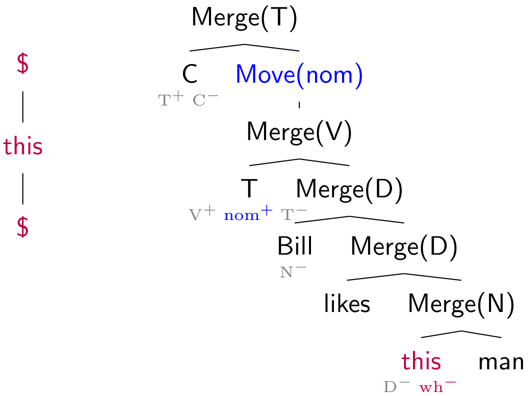
TSL Grammar for Move

Blocking Simple Cases of Illicit Movement

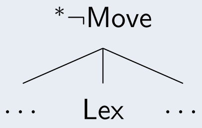


TSL Grammar for Move

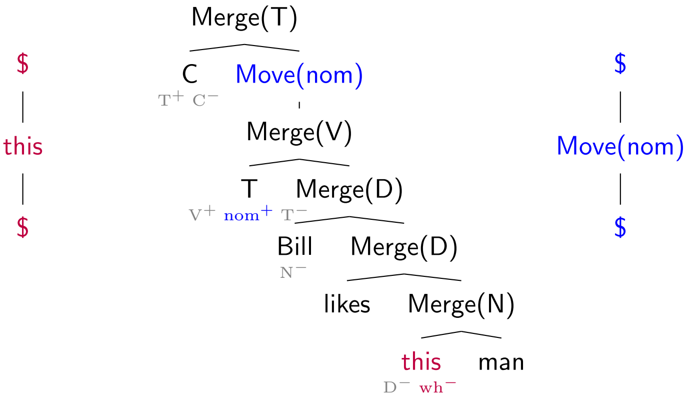
Blocking Simple Cases of Illicit Movement



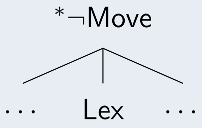
TSL Grammar for Move



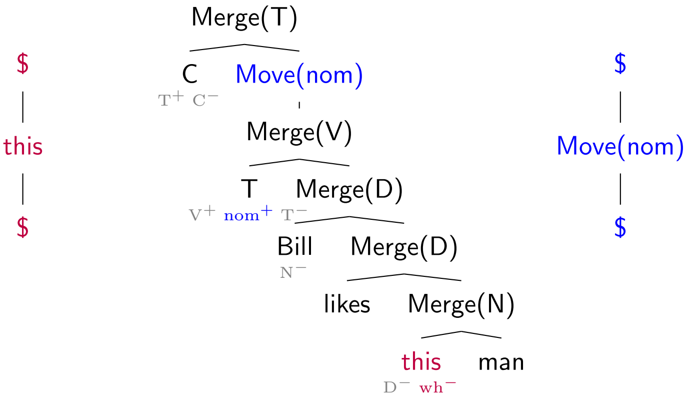
Blocking Simple Cases of Illicit Movement



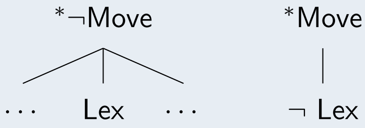
TSL Grammar for Move



Blocking Simple Cases of Illicit Movement

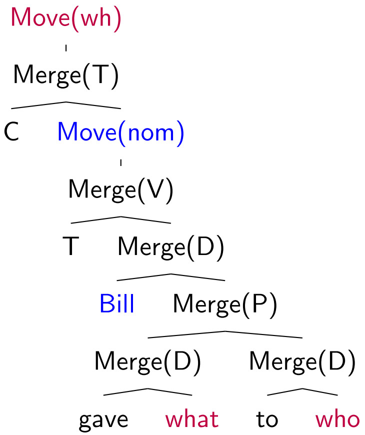


TSL Grammar for Move



Shortest Move Constraint

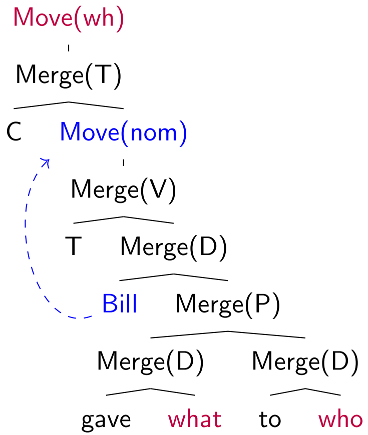
(2) * **What**$_i$ did John wonder **who**$_j$ Bill gave **t**$_i$ to **t**$_j$?



SMC Movers must not target the same position.

Shortest Move Constraint

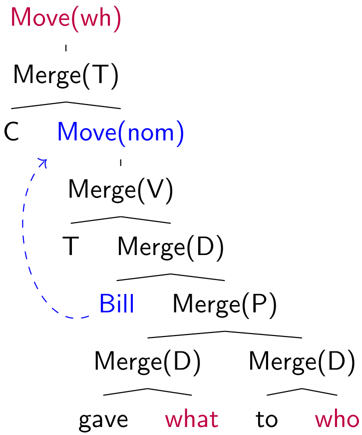
(2) * **What**$_i$ did John wonder **who**$_j$ Bill gave **t**$_i$ to **t**$_j$?



SMC Movers must not target the same position.

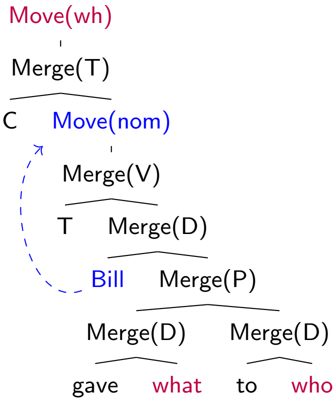
Shortest Move Constraint

(2) * **What**$_i$ did John wonder **who**$_j$ Bill gave **t**$_i$ to **t**$_j$?

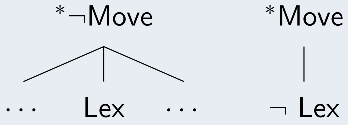


SMC Movers must not target the same position.

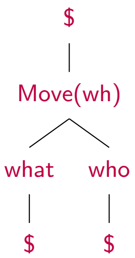
The Full TSL Description

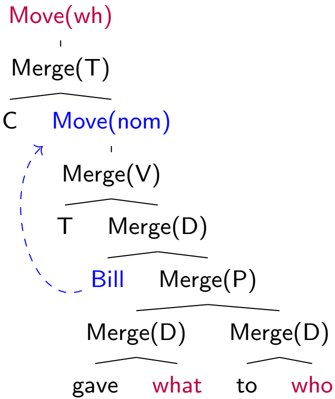


TSL Grammar for Move

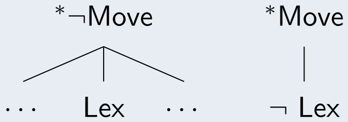


The Full TSL Description

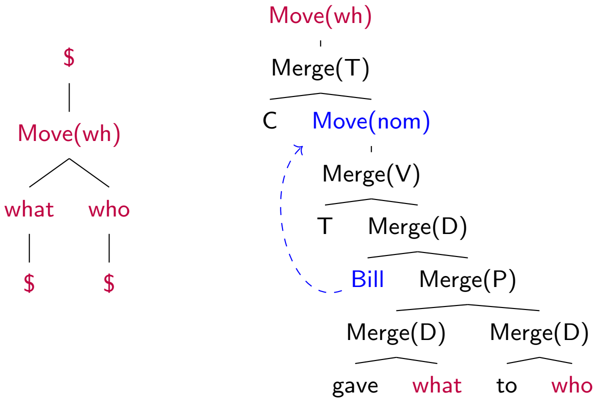




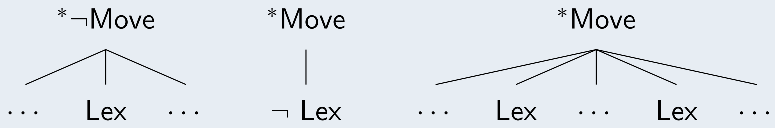
TSL Grammar for Move



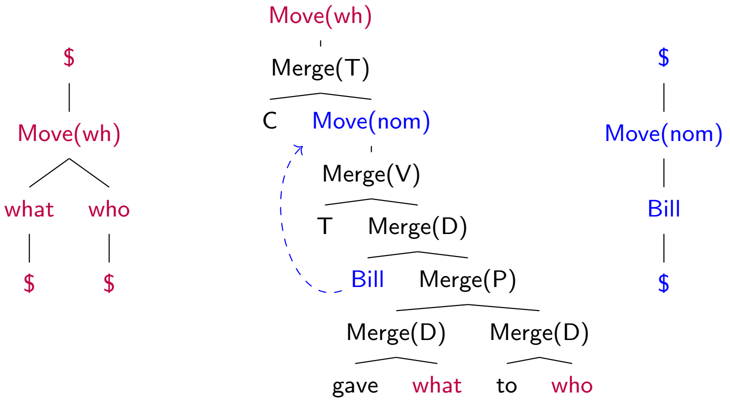
The Full TSL Description



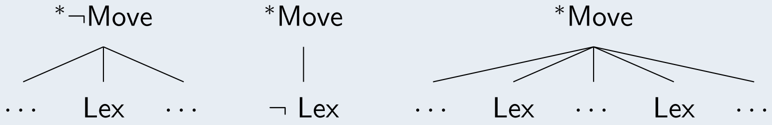
TSL Grammar for Move



The Full TSL Description



TSL Grammar for Move



Upward versus Downward Movement

- Without intermediate movement, **upward movement is TSL**.
- Nice and dandy, but what does it tell us about syntax?

Why is There No Downward Movement?

Downward = movement to c-commanded position

Usually ruled out by Extension Condition, but. . .

- Head movement
- Affix hopping
- Late adjunction
- Tucking in

Upward versus Downward Movement

- Without intermediate movement, **upward movement is TSL**.
- Nice and dandy, but what does it tell us about syntax?

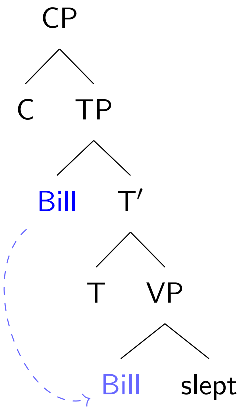
Why is There No Downward Movement?

Downward = movement to c-commanded position

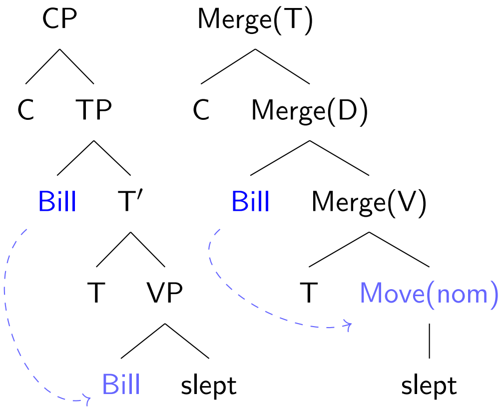
Usually ruled out by Extension Condition, but...

- Head movement
- Affix hopping
- Late adjunction
- Tucking in

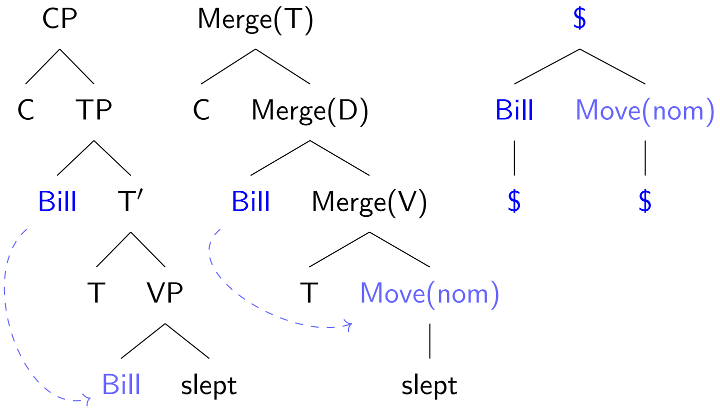
Downward Movement in MGs (Graf 2012b, 2014a)



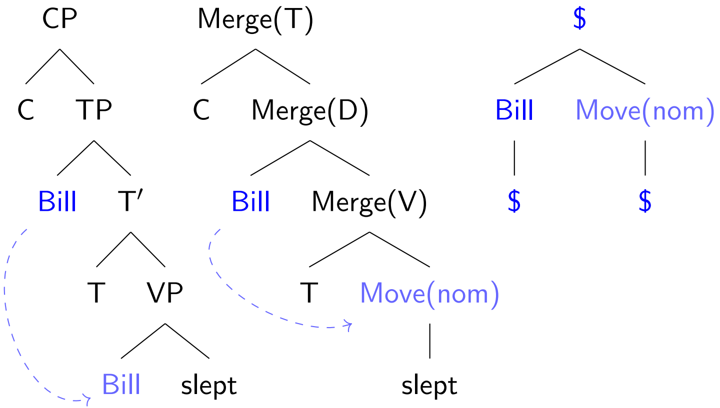
Downward Movement in MGs (Graf 2012b, 2014a)



Downward Movement in MGs (Graf 2012b, 2014a)

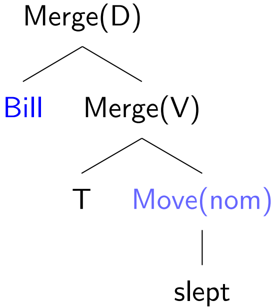


Downward Movement in MGs (Graf 2012b, 2014a)



Downward movement is **not TSL**, because . . .

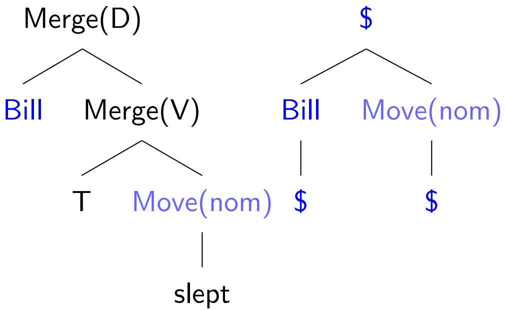
C-Command is not TSL



Important Questions

- Should c-command always be reanalyzed as movement?
- movement : constraints = segmental : suprasegmental?
- Phonological/Morphological c-command?

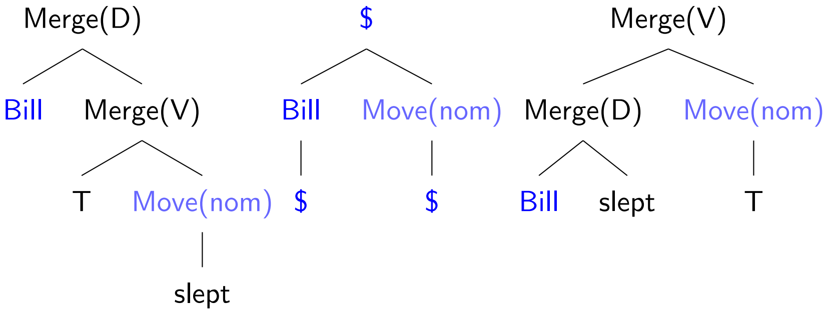
C-Command is not TSL



Important Questions

- Should c-command always be reanalyzed as movement?
- movement : constraints = segmental : suprasegmental?
- Phonological/Morphological c-command?

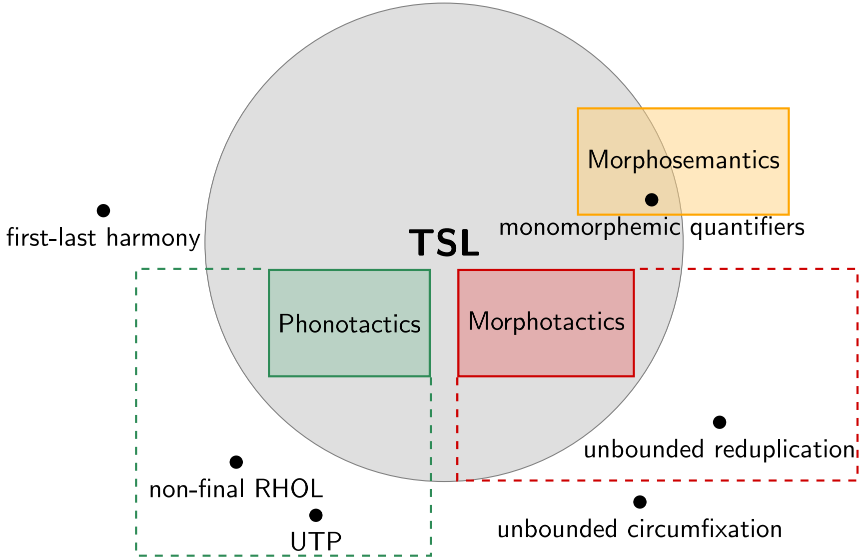
C-Command is not TSL



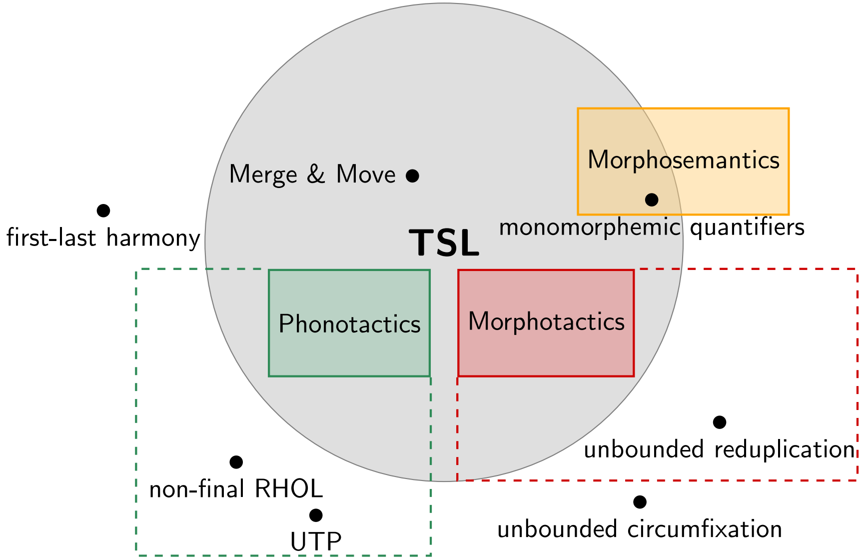
Important Questions

- Should c-command always be reanalyzed as movement?
- movement : constraints = segmental : suprasegmental?
- Phonological/Morphological c-command?

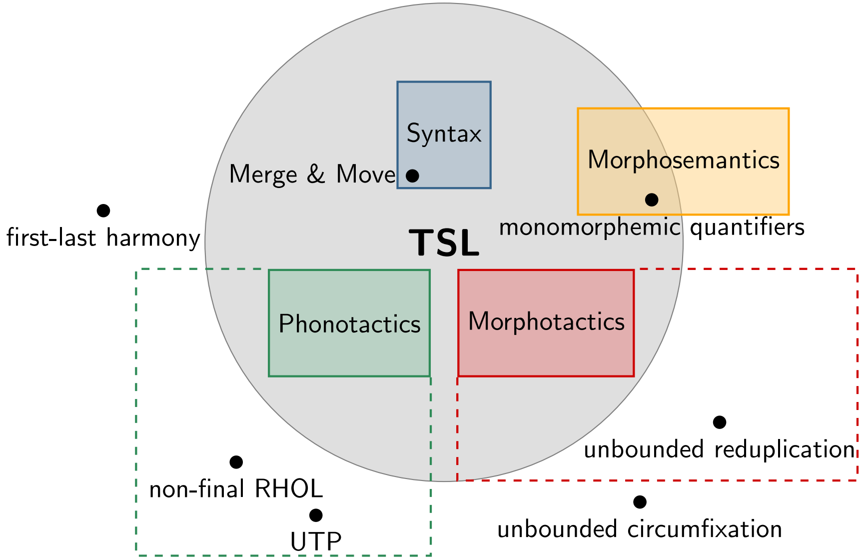
The Full TSL Picture



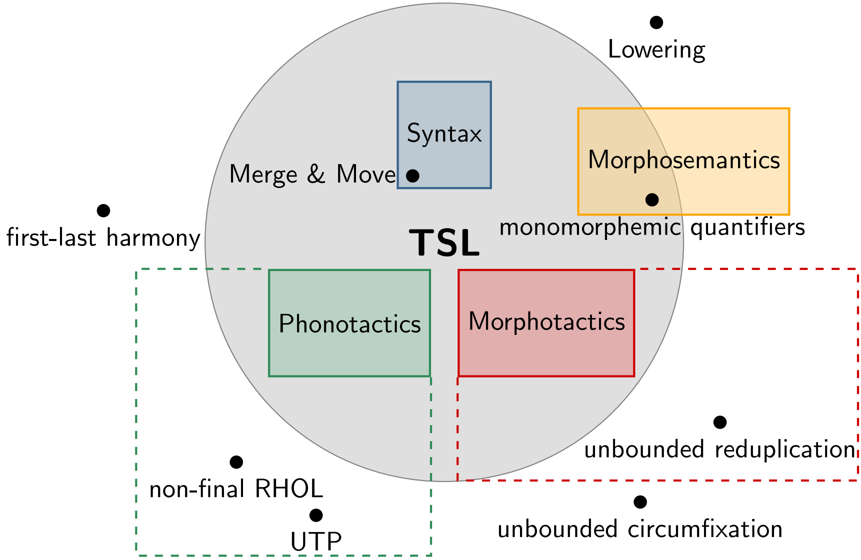
The Full TSL Picture



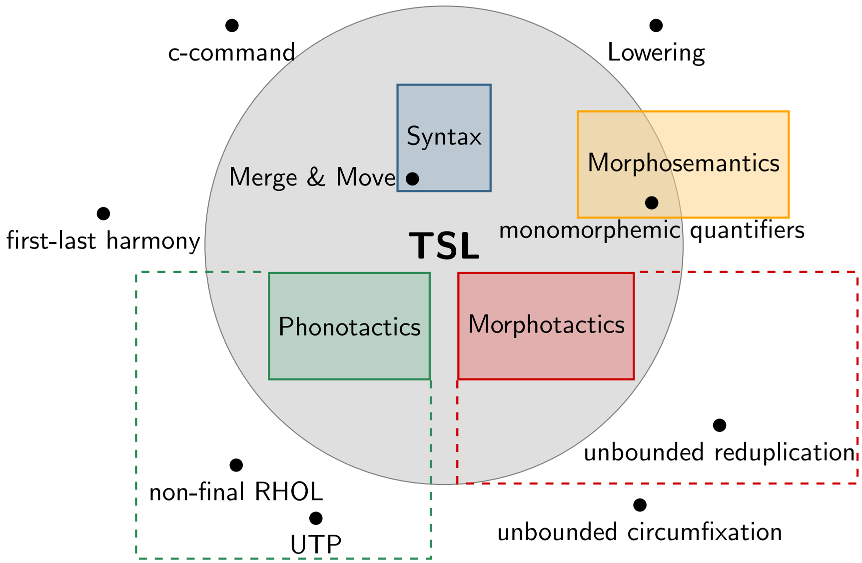
The Full TSL Picture



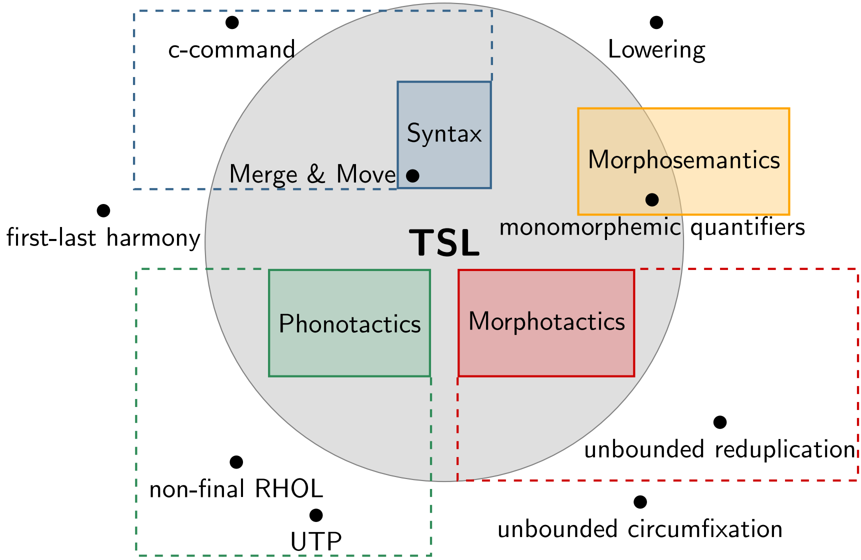
The Full TSL Picture



The Full TSL Picture



The Full TSL Picture



This is Just the Tip of the Iceberg

Mappings	Representations	Beyond TSL	Sign language
			
Jane Chandlee	**Adam Jardine** @3:15	**Aniello De Santo**	**Jon Rawski** @3:45

. . . and many open questions

What CompLing Can Do For You

- Computational linguistics is not a field, it is a **perspective**:
  - What patterns are truly complex?
  - How complex can dependencies be?
  - Are some analyses simpler than others?
- As in any formalism, 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!

What You Can Do For CompLing

Everybody can contribute!

- Do you have data that contradicts our predictions?
- probe the status of c-command in syntax
- grammar fragments
- artificial language learning experiments
- processing experiments

Resources and Readings

1. **Survey papers:** Pullum and Rogers (2006); Heinz (2011a,b, 2015); Rogers and Pullum (2011); Chandlee and Heinz (2016)
2. **TSL and its extensions:** Heinz et al. (2011); McMullin (2016); Baek (2017); De Santo (2017); De Santo and Graf (2017); Graf (2017c)
3. **TSL morphology:** Aksënova et al. (2016); Graf (2017b)
4. **TSL morpho-semantics:** Graf (2017d)
5. **TSL syntax:** Graf (2012a); Graf and Heinz (2016)
6. **Mappings:** Courcelle and Engelfriet (2012); Chandlee (2014, 2016); Jardine (2016)
7. **Learnability:** Heinz (2010); Kasprzik and Kötzing (2010); Heinz et al. (2012); Jardine et al. (2014); Lai (2015); Jardine and Heinz (2016); Jardine and McMullin (2017)

Appendix

Psychological Reality of Derivation Trees

Central role of derivation trees backed up by **processing data**:

- Derivation trees can be parsed top-down (Stabler 2013)
- Parsing models update Derivational Theory of Complexity, make correct processing predictions for
  - right $<$ center embedding (Kobele et al. 2012)
  - crossing $<$ nested dependencies (Kobele et al. 2012)
  - SC-RC $<$ RC-SC (Graf et al. 2017)
  - SRC $<$ ORC in English (Graf et al. 2017)
  - SRC $<$ ORC in East-Asian (Graf et al. 2017)
  - quantifier scope preferences (Pasternak 2016)
  - stacked relative clauses (Zhang 2017)
  - Korean attachment ambiguities

Technical Fertility of Derivation Trees

Derivation trees made it easy for MGs to accommodate the full syntactic toolbox:

- sidewards movement (Stabler 2006; Graf 2013)
- affix hopping (Graf 2012b, 2013)
- clustering movement (Gärtner and Michaelis 2010)
- tucking in (Graf 2013)
- ATB movement (Kobele 2008)
- copy movement (Kobele 2006)
- extraposition (Hunter and Frank 2014)
- Late Merge (Kobele 2010; Graf 2014a)
- Agree (Kobele 2011; Graf 2012a)
- adjunction (Fowlie 2013; Graf 2014b; Hunter 2015)
- TAG-style adjunction (Graf 2012c)

Even More MG Extensions

- local and global constraints (Kobele 2011; Graf 2012a, 2017a)
- transderivational constraints (Graf 2010, 2013)
- Principle A and B (Graf and Abner 2012)
- GPSG-style feature percolation (Kobele 2008)
- idioms (Kobele 2012)
- grafts (multi-rooted multi-dominance trees) (Graf in progress)

Long Story Short

Derivation trees are a more useful and fertile data structure than phrase structure trees.

Even More MG Extensions

- local and global constraints (Kobele 2011; Graf 2012a, 2017a)
- transderivational constraints (Graf 2010, 2013)
- Principle A and B (Graf and Abner 2012)
- GPSG-style feature percolation (Kobele 2008)
- idioms (Kobele 2012)
- grafts (multi-rooted multi-dominance trees) (Graf in progress)

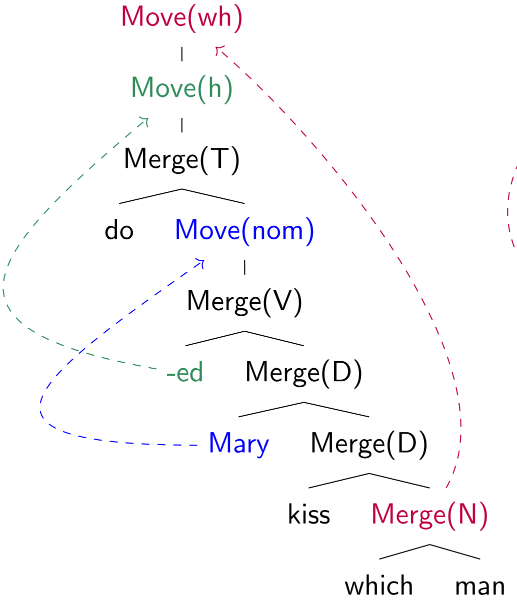
Long Story Short

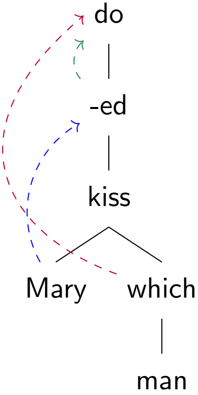
Derivation trees are a more useful and fertile data structure than phrase structure trees.

More on C-Command

- C-command-like relations can be added
- Useful for some phonological phenomena:
  - non-final RHOL
  - bounded harmony due to long-distance blocking in Copperbelt Bemba
  - long-distance blocking of local dissimilation in Samala

“Dependency” Derivation Trees





Sideward Movement

- Move anywhere except m-commanded positions

Relation	TSL?
move upward	yes
move anywhere	yes
m-command	no
sideward	no

- **But:** m-command is TSL over dependency graphs, because it reduces to dominance
  $\Rightarrow$ sideward movement can be TSL

References I

Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of SIGMorPhon 2016*. To appear.

Baek, Hyunah. 2017. Computational representation of unbounded stress: Tiers with structural features. Ms., Stony Brook University.

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

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

Chandlee, Jane. 2016. Computational locality in morphological maps. Ms., Haverford College.

Chandlee, Jane, and Jeffrey Heinz. 2016. Computational phonology. Ms., Haverford College and University of Delaware.

Courcelle, Bruno, and Joost Engelfriet. 2012. *Graph structure and monadic second-order logic: A language-theoretic approach*. Cambridge, UK: Cambridge University Press.

De Santo, Aniello. 2017. Extending TSL languages: Conjunction as multiple tier-projection. Ms., Stony Brook University.

References II

De Santo, Aniello, and Thomas Graf. 2017. Structure sensitive tier projection: Applications and formal properties. Ms., Stony Brook University.

Epstein, Samuel D., Erich M. Groat, Ruriko Kawashima, and Hisatsugu Kitahara. 1998. *A derivational approach to syntactic relations*. Oxford: Oxford University Press.

Fowlie, Meaghan. 2013. Order and optionality: Minimalist grammars with adjunction. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, ed. András Kornai and Marco Kuhlmann, 12–20.

Gärtner, Hans-Martin, and Jens Michaelis. 2010. On the treatment of multiple-wh-interrogatives in Minimalist grammars. In *Language and logos*, ed. Thomas Hanneforth and Gisbert Fanselow, 339–366. Berlin: Akademie Verlag.

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

Graf, Thomas. 2010. A tree transducer model of reference-set computation. *UCLA Working Papers in Linguistics* 15:1–53.

Graf, Thomas. 2012a. Locality and the complexity of Minimalist derivation tree languages. In *Formal Grammar 2010/2011*, ed. Philippe de Groot and Mark-Jan Nederhof, volume 7395 of *Lecture Notes in Computer Science*, 208–227. Heidelberg: Springer. URL `http://dx.doi.org/10.1007/978-3-642-32024-8_14`.

References III

Graf, Thomas. 2012b. Movement-generalized Minimalist grammars. In *LACL 2012*, ed. Denis Béchet and Alexander J. Dikovsky, volume 7351 of *Lecture Notes in Computer Science*, 58–73. URL http://dx.doi.org/10.1007/978-3-642-31262-5_4.

Graf, Thomas. 2012c. Tree adjunction as Minimalist lowering. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 19–27.

Graf, Thomas. 2013. *Local and transderivational constraints in syntax and semantics*. Doctoral Dissertation, UCLA. URL http://thomasgraf.net/doc/papers/PhDThesis_RollingRelease.pdf.

Graf, Thomas. 2014a. Late merge as lowering movement in Minimalist grammars. In *LACL 2014*, ed. Nicholas Asher and Sergei Soloviev, volume 8535 of *Lecture Notes in Computer Science*, 107–121. Heidelberg: Springer.

Graf, Thomas. 2014b. Models of adjunction in Minimalist grammars. In *Formal Grammar 2014*, ed. Glynn Morrill, Reinhard Muskens, Rainer Osswald, and Frank Richter, volume 8612 of *Lecture Notes in Computer Science*, 52–68. Heidelberg: Springer.

Graf, Thomas. 2017a. A computational guide to the dichotomy of features and constraints. *Glossa* 2.

References IV

Graf, Thomas. 2017b. Graph transductions and typological gaps in morphological paradigms. Ms., Stony Brook University.

Graf, Thomas. 2017c. The power of locality domains in phonology. *Phonology* In press.

Graf, Thomas. 2017d. The subregular complexity of monomorphemic quantifiers. Ms., Stony Brook University.

Graf, Thomas, and Natasha Abner. 2012. Is syntactic binding rational? In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 189–197.

Graf, Thomas, Alëna Aksënova, and Aniello De Santo. 2016. A single movement normal form for Minimalist grammars. In *Formal Grammar*. To appear.

Graf, Thomas, and Jeffrey Heinz. 2016. Tier-based strict locality in phonology and syntax. Ms., Stony Brook University and University of Delaware.

Graf, Thomas, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for Minimalist parsing. *Journal of Language Modelling* In press.

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

Heinz, Jeffrey. 2010. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 897–906. URL `http://www.aclweb.org/anthology/P10-1092.pdf`.

References V

Heinz, Jeffrey. 2011a. Computational phonology — part I: Foundations. *Language and Linguistics Compass* 5:140–152.

Heinz, Jeffrey. 2011b. Computational phonology — part II: Grammars, learning, and the future. *Language and Linguistics Compass* 5:153–168.

Heinz, Jeffrey. 2015. The computational nature of phonological generalizations. URL `http://www.socsci.uci.edu/~lpearl/colareadinggroup/readings/Heinz2015BC_Typology.pdf`, ms., University of Delaware.

Heinz, Jeffrey, Anna Kasprzik, and Timo Kötzing. 2012. Learning with lattice-structure hypothesis spaces. *Theoretical Computer Science* 457:111–127.

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`.

Hunter, Tim. 2015. Deconstructing merge and move to make room for adjunction. *Syntax* 18:266–319.

Hunter, Tim, and Robert Frank. 2014. Eliminating rightward movement: Extraposition as flexible linearization of adjuncts. *Linguistic Inquiry* 45:227–267.

Jardine, Adam. 2016. Computationally, tone is different. *Phonology* URL `http://udel.edu/~ajardine/files/jardinemscomputationallytoneisdifferent.pdf`, to appear.

References VI

Jardine, Adam, Jane Chandlee, Rémi Eryaud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, 94–108. URL http://www.jmlr.org/proceedings/papers/v34/jardine14a.html.

Jardine, Adam, and Jeffrey Heinz. 2016. Learning tier-based strictly 2-local languages. *Transactions of the ACL* 4:87–98. URL https://aclweb.org/anthology/Q/Q16/Q16-1007.pdf.

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. Springer.

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.

References VII

Kasprzik, Anna, and Timo Kötzing. 2010. String extension learning using lattices. In *Language and automata theory and applications: 4th international conference, lata 2010, trier, germany, may 24-28, 2010. proceedings*, ed. Adrian-Horia Dediu, Henning Fernau, and Carlos Martín-Vide, 380–391. Berlin, Heidelberg: Springer. URL `http://dx.doi.org/10.1007/978-3-642-13089-2_32`.

Kobele, Gregory M. 2006. *Generating copies: An investigation into structural identity in language and grammar*. Doctoral Dissertation, UCLA. URL `http://home.uchicago.edu/~gkobele/files/Kobele06GeneratingCopies.pdf`.

Kobele, Gregory M. 2008. Across-the-board extraction and Minimalist grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Frameworks*.

Kobele, Gregory M. 2010. On late adjunction in Minimalist grammars. Slides for a talk given at MCFG+ 2010.

Kobele, Gregory M. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 129–144.

Kobele, Gregory M. 2012. Idioms and extended transducers. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 153–161. Paris, France. URL `http://www.aclweb.org/anthology-new/W/W12/W12-4618`.

References VIII

Kobele, Gregory M., Sabrina Gerth, and John T. Hale. 2012. Memory resource allocation in top-down Minimalist parsing. In *Proceedings of Formal Grammar 2012*.

Lai, Regine. 2015. Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry* 46:425–451.

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

Pasternak, Robert. 2016. Memory usage and scope ambiguity resolution. Qualifying paper, Stony Brook University.

Pullum, Geoffrey K., and James Rogers. 2006. Animal pattern-learning experiments: Some mathematical background. Ms., Radcliffe Institute for Advanced Study, Harvard University.

Rogers, James, and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* 20:329–342.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–345.

References IX

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.

Stabler, Edward P. 2006. Sidewards without copying. In *Formal Grammar '06, Proceedings of the Conference*, ed. Gerald Penn, Giorgio Satta, and Shuly Wintner, 133–146. Stanford: CSLI.

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

Stabler, Edward P. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science* 5:611–633.

Zhang, Chong. 2017. *Stacked relatives: Their structure, processing, and computation*. Doctoral Dissertation, Stony Brook University.