Computational Parallels Across Language Modules

Thomas Graf

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

Yale Colloquium
Sep 12, 2016

The Talk in a Nutshell

A Humble Goal

A unified perspective on

- phonology,
- morphology,
- syntax,

that ties together

- typology,
- cognition,
- learning.

But beware. . .

The ground of science [is] littered with the corpses of dead unified theories.

Freeman Dyson
Disturbing the Universe, p62

Outline

The Big Divide

- Phonology, morphology, and syntax are **highly distinct**.
  - different empirical phenomena
  - different cognitive properties
  - different theories
- There have been attempts at unification,
  but the resulting theories still look very different:
  - OT-syntax $\Leftrightarrow$ phonological OT
  - Government Phonology $\Leftrightarrow$ GB
  - Dependency Phonology $\Leftrightarrow$ Dependency grammar
  - Distributed Morphology $\Leftrightarrow$ standard Minimalism
- The standard view is still that there is little common ground.

A Mathematical Distinctness Theorem

In formal language theory, stringsets are classified according to their formal complexity

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

Phonology

Morphology

Syntax

A Mathematical Distinctness Theorem

In formal language theory, stringsets are classified according to their formal complexity

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

Kaplan and Kay (1994)

Phonology

Morphology

Syntax

A Mathematical Distinctness Theorem

In formal language theory, stringsets are classified according to their formal complexity

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

Kaplan and Kay (1994)

Phonology

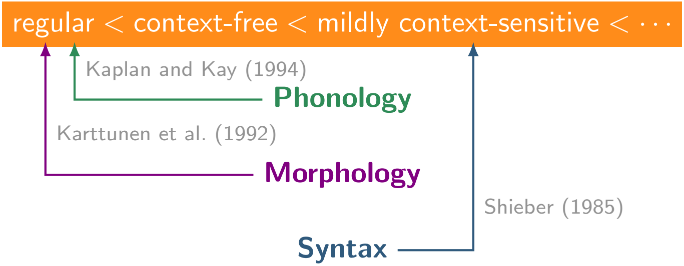
Karttunen et al. (1992)

Morphology

Syntax

A Mathematical Distinctness Theorem

In formal language theory, stringsets are classified according to their formal complexity



Implications of Mathematical Distinctness

Heinz and Idsardi (2011, 2013) highlight the implications:

- different typology
  center embedding, crossing dependencies
- different memory architecture
  flat & finite VS unbounded nested stacks
- different learning algorithms
  much harder for syntax

An Incomplete Picture

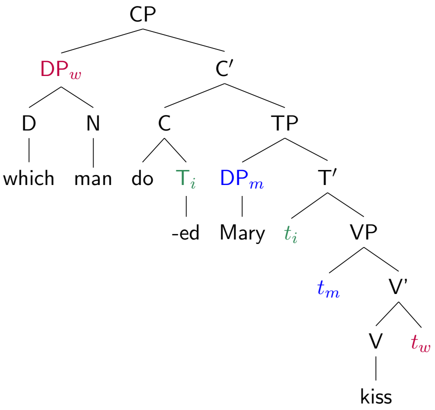
- The argument that phonology, morphology generate simpler stringsets than syntax is mathematically correct.
- But syntax is not about strings!
- What happens if we think of **syntax as generating trees?**

Minimalist Grammars

- 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

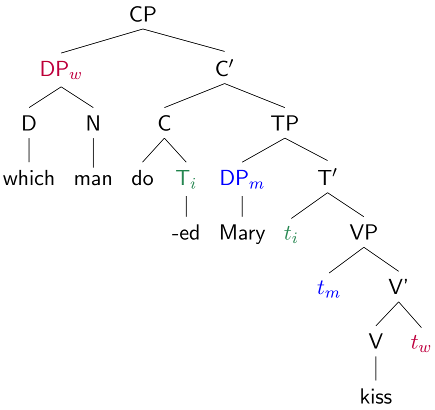
Chemistry	**Syntax**
atoms	words
electrons	features
molecules	sentences

MG Syntax in Action

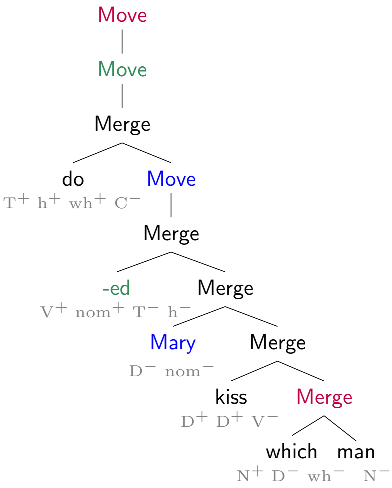


Phrase Structure Tree

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

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 and Marcinek 2014)
  - SRC < ORC in English (Graf and Marcinek 2014)
  - SRC < ORC in East-Asian (Graf et al. 2015)
  - quantifier scope preferences (Pasternak 2016)

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, 2016a)
- transderivational constraints (Graf 2010c, 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, 2016a)
- transderivational constraints (Graf 2010c, 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.

The Complexity of Minimalist Tree Languages

Another surprise: derivation trees are also **crucially simpler!**
This holds even with all the extensions listed before.

- The set of derivation trees is a regular tree language.
  (Michaelis 2001; Kobele et al. 2007; Graf 2012a)
- The set of phrase structure trees is not.
  (Doner 1970; Thatcher 1967; Michaelis 2001)

Computational Parallelism Hypothesis (Weak)

- Phonology and morphology are regular over strings.
- Syntax is regular over derivation trees.

The Complexity of Minimalist Tree Languages

Another surprise: derivation trees are also **crucially simpler!**
This holds even with all the extensions listed before.

- The set of derivation trees is a regular tree language.
  (Michaelis 2001; Kobele et al. 2007; Graf 2012a)
- The set of phrase structure trees is not.
  (Doner 1970; Thatcher 1967; Michaelis 2001)

Computational Parallelism Hypothesis (Weak)

- Phonology and morphology are regular over strings.
- Syntax is regular over derivation trees.

Implications and New Questions

- Different cognitive picture

	Complexity	Data Structure
Phonology	REG	strings
Morphology	REG	strings
Syntax	MCS	strings

- Similar memory architecture (flat, finite) and structural inference mechanisms for all three modules

Two Pressing Questions

1. Can we leverage this for typology, language acquisition?
2. Is this just an accident?

Implications and New Questions

- Different cognitive picture

	Complexity	**Data Structure**
Phonology	REG	strings
Morphology	REG	strings
Syntax	REG	**trees**

- Similar memory architecture (flat, finite) and structural inference mechanisms for all three modules

Two Pressing Questions

1. Can we leverage this for typology, language acquisition?
2. Is this just an accident?

Implications and New Questions

- Different cognitive picture

	Complexity	**Data Structure**
Phonology	REG	strings
Morphology	REG	strings
Syntax	REG	**trees**

- Similar memory architecture (flat, finite) and structural inference mechanisms for all three modules

Two Pressing Questions

1. Can we leverage this for typology, language acquisition?
2. Is this just an accident?

Implications and New Questions

- Different cognitive picture

	Complexity	**Data Structure**
Phonology	REG	strings
Morphology	REG	strings
Syntax	REG	**trees**

- Similar memory architecture (flat, finite) and structural inference mechanisms for all three modules

Two Pressing Questions

1. Can we leverage this for typology, language acquisition?
2. Is this just an accident?

Too Many Patterns are Regular

- Reminder: regular patterns at bottom of complexity hierarchy
- **Problem**
  - all phon/morph/syn patterns are regular,
  - not all regular patterns occur in natural languages
- Regularity is **too loose an upper bound**.

Example

- First-last consonant harmony
- Word with at least 3 suffixes must have exactly 5 prefixes
- Derivation contains even number of Move steps
- Principle A applies only if no wh-movement takes place

Too Many Patterns are Regular

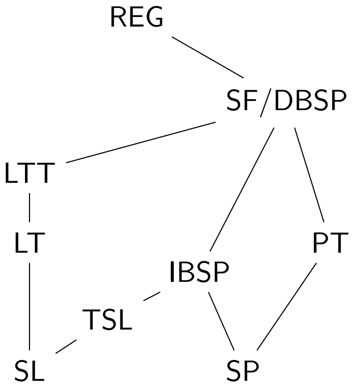
- Reminder: regular patterns at bottom of complexity hierarchy
- **Problem**
  - all phon/morph/syn patterns are regular,
  - not all regular patterns occur in natural languages
- Regularity is **too loose an upper bound**.

Example

- First-last consonant harmony
- Word with at least 3 suffixes must have exactly 5 prefixes
- Derivation contains even number of Move steps
- Principle A applies only if no wh-movement takes place

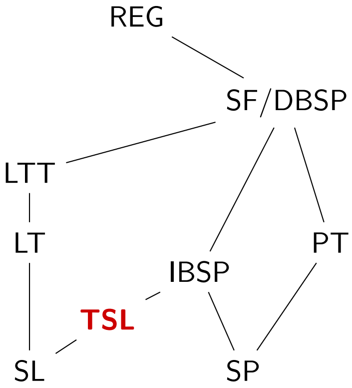
Subregular Languages

Often forgotten: hierarchy of **subregular languages**
(McNaughton and Papert 1971; Rogers et al. 2010; Rogers and Pullum 2011; Heinz et al. 2011; Graf 2016b)



Subregular Languages

Often forgotten: hierarchy of **subregular languages** (McNaughton and Papert 1971; Rogers et al. 2010; Rogers and Pullum 2011; Heinz et al. 2011; Graf 2016b)



TSL: Tier-Based Strictly Local

- There are a variety of subregular classes to choose from.
- But recent research suggests that TSL is the right fit.

Tier-Based Strictly Local Languages

- All patterns described by locally bounded constraints.
- Non-local dependencies are **local over tiers**.
- Much weaker than regular but still powerful enough.

Phonology as a TSL System

Tons of recent work by Jeffrey Heinz, Jane Chandlee, Adam Jardine, and others.

- Phonology as set of well-formed strings $\Rightarrow$ **phonology** $\equiv$ **phonotactics**
- Grammar is a finite set of hard, non-violable markedness constraints.
- Each constraint is represented by a finite collection of forbidden n-grams.

Example: Local Constraints

Process	Constraint	Forbidden n-grams
Word-final devoicing	*[+voice]$	s$, θ$, f$, ...
Intervocalic voicing	*V[-voice]V	asa, asi, ..., isa, isi, ..., afa, afi, ..., ifa, ifi, ...,
CV template	*$V	$a, $i, ...
	*CC	pp, pb, ... bp, bb, ...
	*VV	aa, ai, ..., ia, ii, ...
	*C$	p$, b$, ...

Tiers for Long-Distance Dependencies

- We can move to 3-grams, 4-grams, . . . n-grams in order to regulate less local processes (e.g. umlaut, vowel harmony).
- **Problem:** Still limited to locality domain of size n
$\Rightarrow$ unbounded processes cannot be captured
- **Solution:** selected segments project dedicated tier

Tier-Based n-gram Grammar

- Tier-projection is determined by the shape of the segment, not by structural properties (e.g. feet).
- A string is well-formed iff no tier T contains an illicit T-n-gram.

Tiers for Long-Distance Dependencies

- We can move to 3-grams, 4-grams, $\ldots n$-grams in order to regulate less local processes (e.g. umlaut, vowel harmony).
- **Problem:** Still limited to locality domain of size n
  $\Rightarrow$ unbounded processes cannot be captured
- **Solution:** selected segments project dedicated tier

Tier-Based n-gram Grammar

- Tier-projection is determined by the shape of the segment, not by structural properties (e.g. feet).
- A string is well-formed iff no tier T contains an illicit T-n-gram.

Example: Sibilant Harmony

Constraint	**Forbidden tier$_1$-n-grams**
*ʃ $\cdots$ s	ʃs

Tier$_1$	contains all sibilants
Tier$_0$	contains all segments

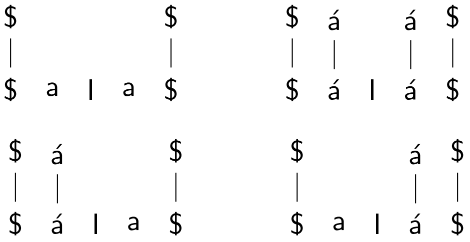
Tier$_1$:	\$		ʃ		s		\$
Tier$_0$:	\$	e	ʃ	i	s	i	\$

Tier$_1$:	\$		ʃ		ʃ		\$
Tier$_0$:	\$	e	ʃ	i	ʃ	i	\$

Example: Stress Assignment

Culminativity every word has exactly one primary stress

$\textbf{Tier}_1$	contains segments with primary stress
$\textbf{Tier}_0$	contains all segments
n-**grams**	śś and $$ on $\textbf{Tier}_1$



A Non-TSL Pattern: Sour Grapes Harmony

Sour Grapes vowel harmony applies only if it can apply to the whole word (i.e. there is no blocker)

Why Sour Grapes isn't TSL

- All vowels **V** must be on the vowel harmony tier.
- The blocker **B** must be on the same tier in order to block it.
- But there is no bound on the number of vowels per tier.
- The tier thus may have the shape

$$\cdots\ \mathbf{V}\ \mathbf{V}\ \mathbf{V}\ \cdots\ \mathbf{B}\ \cdots$$

- **B** can be arbitrarily far away from **VVV** $\Rightarrow$ not a local relation
- But we need to know whether **B** is on the tier in order to determine the well-formedness of **VVV**.

Complexity of Phonology

- All local phonological constraints are TSL.
- All segmental long-distance constraints are TSL.
  But my student Alëna Aksënova may have found a counterexample.
- Tone and stress constraints may go beyond TSL.
  (Graf 2010a,b; Jardine 2015)
- TSL avoids instances of OT overgeneration:
  - cannot generate *sour-grapes* or *majority rules* patterns
  - does not allow *agreement by proxy*
  - explains why consonant harmony is unbounded or transvocalic, but never transconsonantal
    (McMullin and Hansson 2015)

Cognitive Implications

- TSL languages **learnable** from positive data (Heinz et al. 2012; Jardine and Heinz 2016)
  - UG: specifies upper bound on size of n-grams
  - memorize which sequences have not been seen so far
  - induce tier (more complex)
  - still, learning input can be relatively small
- What cognitive resources are required?
  - Only memorization of the last n segments of a specific type
  - For most processes $n \leq 3$, and for all $n \leq 7$
  - Fits **within bounds of human working memory**

Interim Summary: Phonology

- **Phonology is TSL** (possibly with a few outliers).
- gives tighter bound on typology
- solves poverty of stimulus by greatly simplifying learning
- reduces cognitive resource requirements

A Tantalizing Possibility

- TSL is an appealing class.
- And it seems it isn't limited to phonology...

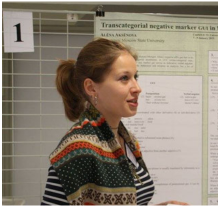
Interim Summary: Phonology

- **Phonology is TSL** (possibly with a few outliers).
- gives tighter bound on typology
- solves poverty of stimulus by greatly simplifying learning
- reduces cognitive resource requirements

A Tantalizing Possibility

- TSL is an appealing class.
- And it seems it isn't limited to phonology. . .

Tier-Based Strictly Local Morphology



- Join work with Alëna Aksënova and Sophie Moradi.
- It seems that **morphology is also TSL**.
  (Aksënova et al. 2016)
- Morphology ≡ Morphotactics of underlying forms
- We are unaware of any non-TSL patterns in this realm.
- Tight typology, explains gaps

Example: Circumfixation in Indonesian

- Indonesian has circumfixation with no upper bound on the distance between the two parts of the circumfix.

(1)	maha siswa big pupil 'student'	(2)	*(ke-) maha siswa *(-an) NMN- big pupil -NMN 'student affairs'

- Requirements: exactly one *ke-* and exactly one *-an*

Tier$_1$	contains all NMN affixes	\$	an			ke	ke	\$
Tier$_0$	contains all morphemes	|	|			|	|	|
n-**grams**	**\$an**, **ke\$**, **keke**, **anan**	\$	an	m	s	ke	ke	\$

Example: Circumfixation in Indonesian

- Indonesian has circumfixation with no upper bound on the distance between the two parts of the circumfix.

(1) maha siswa
big pupil
'student'

(2) *(ke-) maha siswa *(-an)
NMN- big pupil -NMN
'student affairs'

- Requirements: exactly one *ke-* and exactly one *-an*

Tier$_1$	contains all NMN affixes	\$ an ke ke \$
Tier$_0$	contains all morphemes	\$ an m s ke ke \$
n**-grams**	**\$an**, **ke\$**, **keke**, **anan**	

Example: Swahili *vyo*

Swahili *vyo* is **either a prefix or a suffix**,
depending on presence of negation. (Stump 2016)

(3) a.

a-	vi-	**soma**	**-vyo**
SBJ:CL.1-	OBJ:CL.8-	read	-REL:CL.8

‘reads’

b.

a-	si-	**vyo-**	vi-	**soma**
SBJ:CL.1-	NEG-	REL:CL.8-	read	-OBJ:CL.8

‘doesn’t read’

Example: Swahili *vyo* [cont.]

(4)

a. * a- **vyo-** vi- **soma**
SBJ:CL.1- REL:CL.8- OBJ:CL.8- read

b. * a- **vyo-** vi- **soma** **-vyo**
SBJ:CL.1- REL:CL.8- OBJ:CL.8- read -REL:CL.8

c. * a- si- **vyo-** vi- **soma** **-vyo**
SBJ:CL.1- NEG- REL:CL.8- OBJ:CL.8- read REL:CL.8-

d. * a- si- vi- **soma** **-vyo**
SBJ:CL.1- NEG- OBJ:CL.8- read REL:CL.8-

Example: Swahili *vyo* [cont.]

Generalizations About *vyo*

- may occur at most once
- must follow negation prefix *si-* if present
- is a prefix iff *si-* is present

Tier_1	contains *vyo*, *si*, and stem edges #	
Tier_0	contains all morphemes	
n-**grams**	**vyovyo**, **vyo##vyo**	"at most one *vyo*"
	vyosi, **vyo##si**	"*vyo* follows *si*"
	si##vyo, **$vyo##**	"*vyo* is prefix iff *si* present"

Explaining Typological Gaps

Restriction to TSL can also explain some typological gaps.

General Strategy

- Attested patterns **A** and **B** are TSL.
- But combined pattern **A**+**B** is not attested.
- Show that **A**+**B** is not TSL.

Example: Compounding Markers

- Russian has an infix *-o-* that may occur between parts of compounds.
- Turkish has a single suffix *-sı* that occurs at end of compounds.

(5) vod **-o-** voz **-o-** voz
water -COMP- carry -COMP- carry
'carrier of water-carriers'

(6) türk bahçe kapı **-sı** (***-sı**)
turkish garden gate -COMP (*-COMP)
'Turkish garden gate'

- **New Universal**
  If a language allows unboundedly many compound affixes, they are **infixes**.

Example: Compounding Markers [cont.]

- Russian and Turkish are TSL.

Tier$_1$	COMP affix and stem edges #
Russian n-grams	**oo**, **$o**, **o$**
Turkish n-grams	**sisi**, **$si**, **si#**

- The combined pattern would yield Ruskish: stem^{n+1}-si^{n}
- This pattern is not regular and hence **not TSL either**.

Interim Summary: Morphology

- While we know less about morphology than phonology at this point, it also seems to be TSL.
- Even complex patterns like Swahili *vyo* can be captured.
- At the same time, we get **new universals**:

Bounded Circumfixation
: No recursive process can be realized via circumfixation.

Non-proportionality
: The number of prefixes cannot be proportional to the number of stems or suffixes, *et vice versa*.

- We can reuse tools and techniques from TSL phonology, including learning algorithms.
- The cognitive resource requirements are also comparable.

Tier-Based Strictly Local Syntax

	Complexity	Data Structure
Phonology	REG	strings
Morphology	REG	strings
Syntax	**MCS**	strings

- Is syntax the odd one out after all?
- Hard to say in full generality, but **Merge and Move are TSL**. (Graf and Heinz 2016)

Tier-Based Strictly Local Syntax

	Complexity	**Data Structure**
Phonology	REG	strings
Morphology	REG	strings
Syntax	REG	**trees**

- Is syntax the odd one out after all?
- Hard to say in full generality, but **Merge and Move are TSL**. (Graf and Heinz 2016)

Tier-Based Strictly Local Syntax

	Complexity	Data Structure
Phonology	**TSL**	strings
Morphology	**TSL**	strings
Syntax	REG	trees

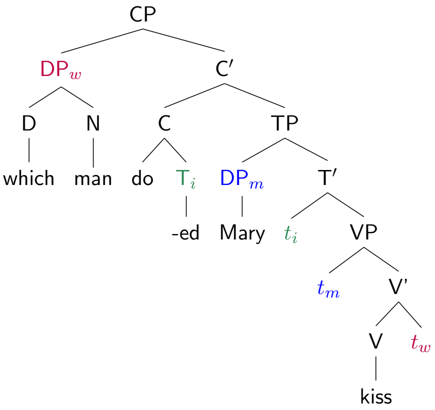
- Is syntax the odd one out after all?
- Hard to say in full generality, but **Merge and Move are TSL**. (Graf and Heinz 2016)

Tier-Based Strictly Local Syntax

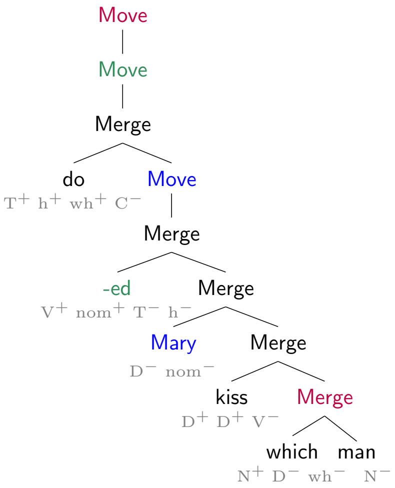
	Complexity	**Data Structure**
Phonology	**TSL**	strings
Morphology	**TSL**	strings
Syntax	REG	trees

- Is syntax the odd one out after all?
- Hard to say in full generality, but **Merge and Move are TSL**. (Graf and Heinz 2016)

Reminder: MG Derivation Trees



Phrase Structure Tree

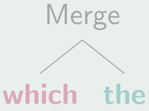


Derivation Tree

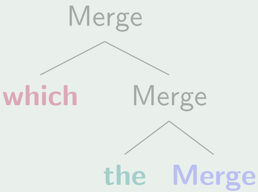
Tree n-gram Grammars

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

Example: Regulating Illicit Merge



block Merger of **which** and **the**

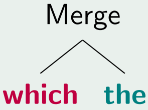


block Merger of **which** and **the XP**

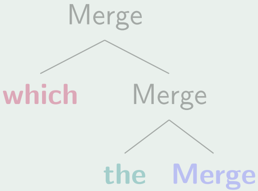
Tree n-gram Grammars

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

Example: Regulating Illicit Merge



block Merger of **which** and **the**

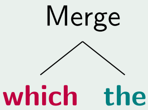


block Merger of **which** and **the XP**

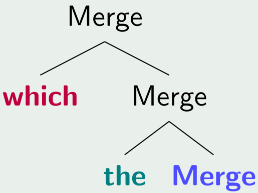
Tree n-gram Grammars

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

Example: Regulating Illicit Merge



block Merger of **which** and **the**



block Merger of **which** and **the XP**

Constraints on Move

Merge is a local process, regulated by tree n-grams.
But what about Move?

Suppose our MG is in **single movement normal form**,
i.e. every phrase moves at most once.
Then movement is regulated by two constraints. (Graf 2012a)

Constraints on Movement

Move	Every head with a negative Move feature is dominated by a matching Move node.
SMC	Every Move node is a closest dominating match for exactly one head.

Constraints on Move

Merge is a local process, regulated by tree n-grams.
But what about Move?

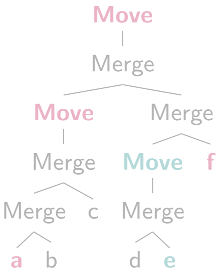
Suppose our MG is in **single movement normal form**,
i.e. every phrase moves at most once.
Then movement is regulated by two constraints. (Graf 2012a)

Constraints on Movement

Move	Every head with a negative Move feature is dominated by a matching Move node.
SMC	Every Move node is a closest dominating match for exactly one head.

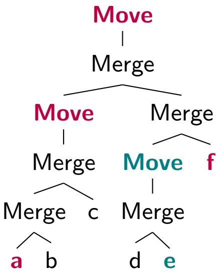
Tiers for Movement

- There is no upper bound on the distance between a lexical item and its matching Move node.
- Consequently, **Move dependencies are not local**.
- What if every movement type (wh, topic, ...) induces its own tier? Would that make Move dependencies local?



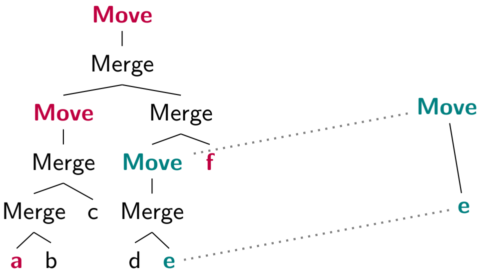
Tiers for Movement

- There is no upper bound on the distance between a lexical item and its matching Move node.
- Consequently, **Move dependencies are not local**.
- What if every movement type (wh, topic, ...) induces its own tier? Would that make Move dependencies local?



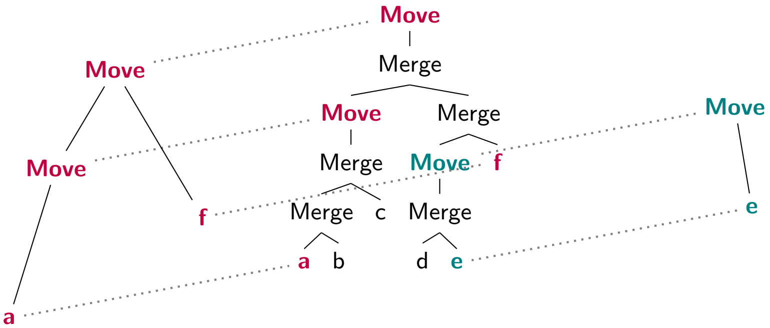
Tiers for Movement

- There is no upper bound on the distance between a lexical item and its matching Move node.
- Consequently, **Move dependencies are not local**.
- What if every movement type (wh, topic, . . .) induces its own tier? Would that make Move dependencies local?



Tiers for Movement

- There is no upper bound on the distance between a lexical item and its matching Move node.
- Consequently, **Move dependencies are not local**.
- What if every movement type (wh, topic, ...) induces its own tier? Would that make Move dependencies local?



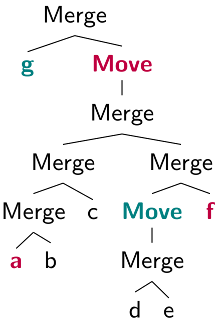
Move Constraints over Tiers

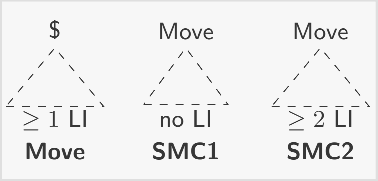
	Original	**Tier**
Move	Every head with a negative Move feature is dominated by a matching Move node.	Every lexical item has a **mother** labeled Move.
SMC	Every Move node is a closest dominating match for exactly one head.	Exactly one of a Move node's **daughters** is a lexical item.

Tree n-gram Templates

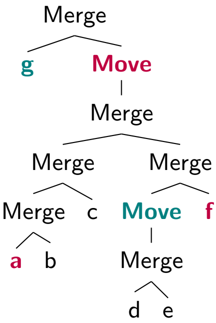
Move	**SMC1**	**SMC2**
\$	Move	Move
≥ 1 LI	no LI	≥ 2 LIs

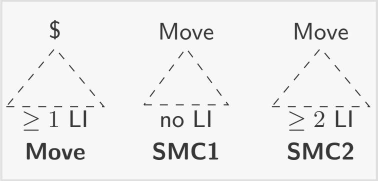
Example of Ill-Formed Derivation



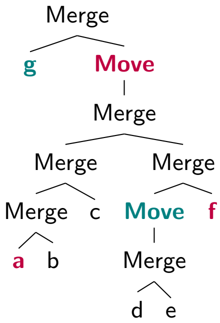


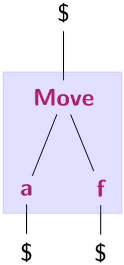
Example of Ill-Formed Derivation

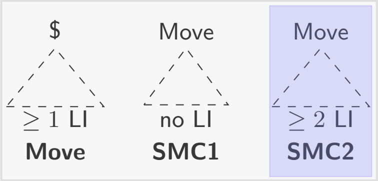




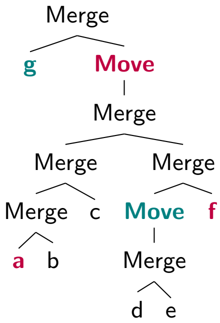
Example of Ill-Formed Derivation

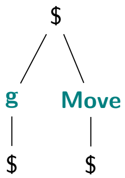


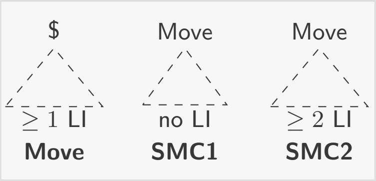




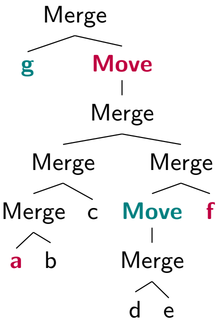
Example of Ill-Formed Derivation

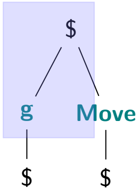


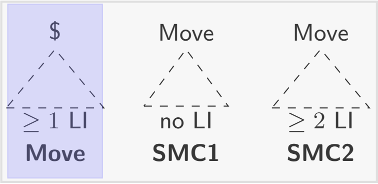




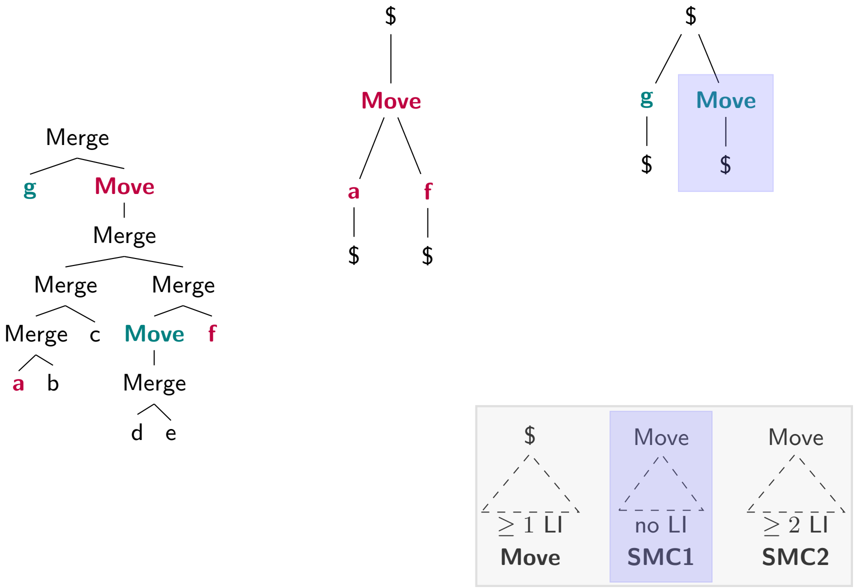
Example of Ill-Formed Derivation



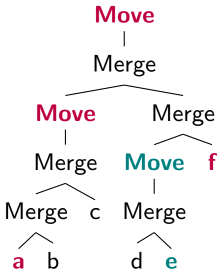


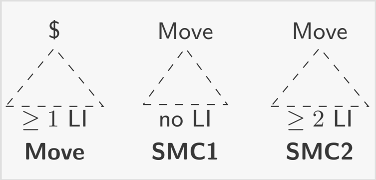


Example of Ill-Formed Derivation

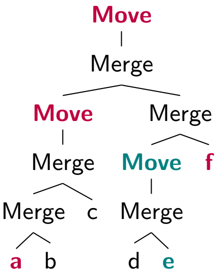


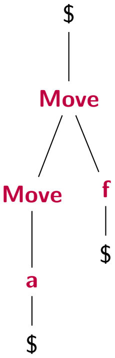
Example of Well-Formed Derivation

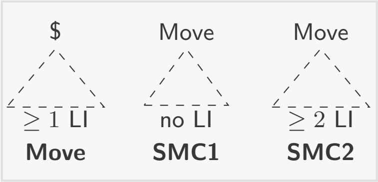




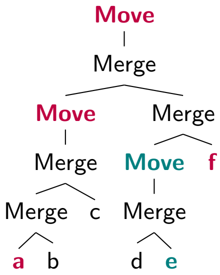
Example of Well-Formed Derivation

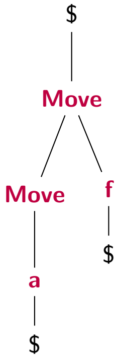


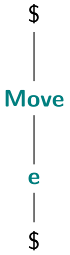


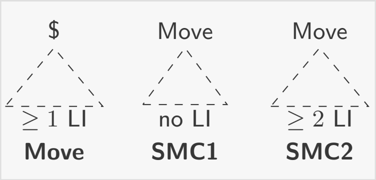


Example of Well-Formed Derivation









Syntax is TSL

- Generalizing tiers from strings to derivations shows Merge and Move to be TSL.
  caveat: single movement normal form
- It is unclear whether the MG extensions also fit into TSL.
- But we take an important step towards acquisition, **solving poverty-of-stimulus!**

Towards a New Learning Algorithm for Syntax

- Derivation trees without Move ≈ dependency graphs
- **Input:** dependency graph, surface string
- Child must infer movement relations:
  - Look at string order to compute possible derivations with Move nodes (in single-movement normal form).
  - Keep track of well-formed tier n-grams seen so far.
  - Favor derivations that do not introduce new tier n-grams.

Syntax is TSL

- Generalizing tiers from strings to derivations shows Merge and Move to be TSL.
  caveat: single movement normal form
- It is unclear whether the MG extensions also fit into TSL.
- But we take an important step towards acquisition, **solving poverty-of-stimulus!**

Towards a New Learning Algorithm for Syntax

- Derivation trees without Move $\approx$ dependency graphs
- **Input:** dependency graph, surface string
- Child must infer movement relations:
  - Look at string order to compute possible derivations with Move nodes (in single-movement normal form).
  - Keep track of well-formed tier n-grams seen so far.
  - Favor derivations that do not introduce new tier n-grams.

Remarks on Single Movement Normal Form

- Single Movement Normal Form seems unrealistic.
- **But:** does not rule out multiple movement steps, only says there is **single feature trigger in derivation**
- Intermediate landing sites can be part of structure built from the derivation tree.

A Conjecture on Movement Restrictions

- Conversion of an MG into single movement normal form causes large blow-up in size of lexicon.
- Blow-up varies a lot: from 0 to hundred times the original size
- The more fixed the position of movers, the smaller the blow-up $\Rightarrow$ island constraints as a means to limit lexical blow-up?

Remarks on Single Movement Normal Form

- Single Movement Normal Form seems unrealistic.
- **But:** does not rule out multiple movement steps, only says there is **single feature trigger in derivation**
- Intermediate landing sites can be part of structure built from the derivation tree.

A Conjecture on Movement Restrictions

- Conversion of an MG into single movement normal form causes large blow-up in size of lexicon.
- Blow-up varies a lot: from 0 to hundred times the original size
- The more fixed the position of movers, the smaller the blow-up $\Rightarrow$ island constraints as a means to limit lexical blow-up?

Conclusion

	Complexity	**Data Structure**
Phonology	REG	strings
Morphology	REG	strings
Syntax	**MCS**	strings

TSL Intuition

- Non-local dependencies are not particularly complex.
- They are **local over a very simple relativization domain**.

Unified perspective on

- cognition
- acquisition
- typology

Conclusion

	Complexity	**Data Structure**
Phonology	REG	strings
Morphology	REG	strings
Syntax	REG	**trees**

TSL Intuition

- Non-local dependencies are not particularly complex.
- They are **local over a very simple relativization domain**.

Unified perspective on

- cognition
- acquisition
- typology

Conclusion

	Complexity	**Data Structure**
Phonology	**TSL**	strings
Morphology	**TSL**	strings
Syntax	**TSL**	trees

TSL Intuition

- Non-local dependencies are not particularly complex.
- They are **local over a very simple relativization domain**.

Unified perspective on

- cognition
- acquisition
- typology

Future Work

There's tons of work that still needs to be done.
Most pressing:

- fully autosegmental structures for phonology
- derivations for morphology
- mappings
- look beyond Move in syntax

Join the Enterprise!

- typological universals/gaps
- grammar fragments and careful descriptions of phenomena
- TSL-analyses of phenomena
- counterexamples
- artificial language learning experiments
- processing experiments
- and of course: theorems and proofs

I wanna learn more...

- Computational Linguistics II lecture notes:
  `lin637.thomasgraf.net`
- Computational phonology seminar:
  `lin626.thomasgraf.net`
- Computational syntax seminar:
  next semester
- Check the references

References I

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

Doner, John. 1970. Tree acceptors and some of their applications. *Journal of Computer and System Sciences* 4:406–451.

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.

Graf, Thomas. 2010a. Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16:Article 10. URL `http://repository.upenn.edu/pwpl/vol16/iss1/10`.

Graf, Thomas. 2010b. *Logics of phonological reasoning*. Master's thesis, University of California, Los Angeles. URL `http://thomasgraf.net/doc/papers/LogicsOfPhonologicalReasoning.pdf`.

References II

Graf, Thomas. 2010c. 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.

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.

References III

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. 2016a. A computational guide to the dichotomy of features and constraints To appear in *Glossa*.

Graf, Thomas. 2016b. The power of locality domains in phonology. 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, Brigitta Fodor, James Monette, Gianpaul Rachiele, Aunika Warren, and Chong Zhang. 2015. A refined notion of memory usage for minimalist parsing. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, 1–14. Chicago, USA: Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W15-2301`.

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

References IV

Graf, Thomas, and Bradley Marcinek. 2014. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the 2014 ACL Workshop on Cognitive Modeling and Computational Linguistics*, 28–36.

Heinz, Jeffrey, and William Idsardi. 2011. Sentence and word complexity. *Science* 333:295–297.

Heinz, Jeffrey, and William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in Cognitive Science* 5:111–131.

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. 2015. Computationally, tone is different. *Phonology* URL `http://udel.edu/~ajardine/files/jardinemscomputationallytoneisdifferent.pdf`, to appear.

References V

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`.

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`.

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.

References VI

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.

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

Kobele, Gregory M., Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to Minimalism. In *Model Theoretic Syntax at 10*, ed. James Rogers and Stephan Kepser, 71–80.

McMullin, Kevin, and Gunnar Ólafur Hansson. 2015. Long-distance phonotactics as tier-based strictly 2-local languages. In *Proceedings of AMP 2014*.

McNaughton, Robert, and Seymour Papert. 1971. *Counter-free automata*. Cambridge, MA: MIT Press.

Michaelis, Jens. 2001. Transforming linear context-free rewriting systems into Minimalist grammars. *Lecture Notes in Artificial Intelligence* 2099:228–244.

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

References VII

Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Vischer, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In *The mathematics of language*, ed. Christan Ebert, Gerhard Jäger, and Jens Michaelis, volume 6149 of *Lecture Notes in Artificial Intelligence*, 255–265. Heidelberg: Springer. URL `http://dx.doi.org/10.1007/978-3-642-14322-9_19`.

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.

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.

References VIII

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

Stump, Greg. 2016. Rule composition in an adequate theory of morphotactics. Ms., University of Kentucky.

Thatcher, James W. 1967. Characterizing derivation trees for context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences* 1:317–322.