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

1. The Received View: Highly Distinct Language Modules

2. Weak Parallelism: Syntax is Regular, Too

3. Strong Parallelism: Tier-Based Strict Locality
   - Regularity is Too Permissive
   - Tier-Based Strictly Local Phonology
   - Tier-Based Strictly Local Morphology
   - Tier-Based Strictly Local Syntax
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 ⇔ phonological OT
  - Government Phonology ⇔ GB
  - Dependency Phonology ⇔ Dependency grammar
  - Distributed Morphology ⇔ 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 < · · ·

Phonology

Morphology

Syntax
A Mathematical Distinctnessness Theorem

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

\[ \text{regular} < \text{context-free} < \text{mildly context-sensitive} < \cdots \]

Kaplan and Kay (1994)

Phonology

Morphology

Syntax
A Mathematical Distinctness Theorem

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

\[ \text{regular} < \text{context-free} < \text{mildly context-sensitive} < \cdots \]

- Kaplan and Kay (1994) - Phonology
- Karttunen et al. (1992) - Morphology
- Syntax
In formal language theory, stringsets are classified according to their formal complexity:

- **Regular**
- **Context-free**
- **Mildly context-sensitive**
- ...
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 string sets 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

<table>
<thead>
<tr>
<th>Chemistry</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>atoms</td>
<td>words</td>
</tr>
<tr>
<td>electrons</td>
<td>features</td>
</tr>
<tr>
<td>molecules</td>
<td>sentences</td>
</tr>
</tbody>
</table>
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
  ⇒ **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:

- sideways 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>MCS strings</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG trees</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG trees</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG trees</td>
</tr>
</tbody>
</table>

- 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)

```
  REG
   /\  \\
  SF/DBSP
   \  /
   LTT
   /  \
   LT
   /  \
   TSL
  /    \
SL/SP
```
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
  \[\Rightarrow \text{phonology } \equiv \text{phonotactics}\]
- 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

<table>
<thead>
<tr>
<th>Process</th>
<th>Constraint</th>
<th>Forbidden $n$-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word-final devoicing</td>
<td>$^{*}[+\text{voice}]$</td>
<td>$s$, $\theta$, $f$, ...</td>
</tr>
<tr>
<td>Intervocalic voicing</td>
<td>$^{*}\text{V[-voice]}\text{V}$</td>
<td>asa, asi, ..., isa, isi, ..., afa, afi, ..., ifa, ifi, ...,</td>
</tr>
<tr>
<td>CV template</td>
<td>$^{*}\text{V}$</td>
<td>$a$, $i$, ...</td>
</tr>
<tr>
<td></td>
<td>$^{*}\text{CC}$</td>
<td>pp, pb, ..., bp, bb, ...</td>
</tr>
<tr>
<td></td>
<td>$^{*}\text{VV}$</td>
<td>aa, ai, ..., ia, ii, ...</td>
</tr>
<tr>
<td></td>
<td>$^{*}\text{C}$</td>
<td>p$, b$, ...</td>
</tr>
</tbody>
</table>
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 \( n \) ⇒ 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**

$^*\int \ldots s$  $\int s$

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

Tier$_1$: $\dollar \int s \dollar$

Tier$_0$: $\dollar e \int i s i \dollar$

Tier$_1$: $\dollar \int \int \dollar$

Tier$_0$: $\dollar e \int i \int i \dollar$
Example: Stress Assignment

Culminativity  every word has exactly one primary stress

**Tier$_1$** contains segments with primary stress

**Tier$_0$** contains all segments

$n$-grams  só and $\$$ on **Tier$_1$**

```
$  $  $ á á á $  \\
|  |  |  |  |  |
$ a | a $  $ á l á $  \\

$ á $  $ $  $ á $  \\
|  |  |  |  |  |
$ á | l a $  $ a | l á $  
```
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
  
  \[
  \cdots \, V \, V \, V \, \cdots \, B \, \cdots
  \]
- B can be arbitrarily far away from VVVV \( \Rightarrow \) not a local relation
- But we need to know whether B is on the tier in order to determine the well-formedness of VVVV.
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

$ an \ 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<sub>1</sub> | contains all NMN affixes | $ an | ke | ke | $ |
| Tier<sub>0</sub> | contains all morphemes | | | | |
| n-grams | $an, ke$, keke, anan | $ an | m | s | ke | ke | $ |
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*
    SBJ:CL.1- NEG- REL:CL.8- OBJ:CL.8- read
    -vyo
    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**

<table>
<thead>
<tr>
<th>Tier</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Tier}_1 )</td>
<td>vyo, si, and stem edges $#$</td>
</tr>
<tr>
<td>( \text{Tier}_0 )</td>
<td>contains all morphemes</td>
</tr>
</tbody>
</table>

**\( n \)-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 -si that occurs at end of compounds.

\[(5)\quad \text{vod} \quad -o- \quad \text{voz} \quad -o- \quad \text{voz} \]
\[\text{water} \quad \text{-COMP-} \quad \text{carry} \quad \text{-COMP-} \quad \text{carry} \]
\[\text{‘carrier of water-carriers’}\]

\[(6)\quad \text{türk} \quad \text{bahçe} \quad \text{kapı} \quad -si \quad (\ast \text{-si}) \]
\[\text{turkish} \quad \text{garden} \quad \text{gate} \quad \text{-COMP} \quad (\ast \text{-COMP}) \]
\[\text{‘Turkish garden gate’}\]

- **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**.
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

<table>
<thead>
<tr>
<th></th>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG</td>
<td>strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG</td>
<td>strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>MCS</td>
<td>strings</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>TSL strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>TSL strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG trees</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>TSL</td>
</tr>
<tr>
<td>Morphology</td>
<td>TSL</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG</td>
</tr>
<tr>
<td></td>
<td>strings</td>
</tr>
<tr>
<td></td>
<td>strings</td>
</tr>
<tr>
<td></td>
<td>trees</td>
</tr>
</tbody>
</table>

- 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

Move
Move
Merge
do
Move
Merge
-ed
Merge
Mary
Merge
kiss
Merge
Mary
Merge
which
man

Phrase Structure Tree

Derivation Tree

Move
Move
Merge
do
Move
Merge
-ed
Merge
Mary
Merge
kiss
Merge
Mary
Merge
which
man
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

```
Merge
  /  \
which the
```

block Merger of which and the

```
Merge
  /  \
which Merge
     /  \
the Merge
```

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

```
Merge
  ∩
which  the
```
block Merger of `which` and `the`

```
Merge
  ∩
which  Merge
  ∩
  ∩
the  Merge
```
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

```
  Merge
    ├── Merge
      │   ├── Merge
      │   │   ├── Merge
      │   │   │   ├── Merge
      │   │   │   │   └── Merge
      │   │   │   └── Merge
      │   └── Merge
    └── 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)

<table>
<thead>
<tr>
<th>Constraints on Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move</strong></td>
</tr>
<tr>
<td><strong>SMC</strong></td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Constraints on Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move</strong></td>
</tr>
<tr>
<td><strong>SMC</strong></td>
</tr>
</tbody>
</table>
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
   |   Merge
   |   
   Move       Merge
    |  |    
  Merge      Move f
   |        |   
  Merge c   Merge 
   |   
   a b  d e
```
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
  |  Move
  |    Merge
  |     |
Merge
  |  Move
  |    |
  |     |
Merge
  |  |
  |  |
Merge
  |  |
  |  |
Merge
  |  a
  |  b
```
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?

---

**Diagram:***

```
  Move
    |   Merge
  Move   Move
      |       |
  Merge  Merge
    |       |
  Merge  Merge
      |       |
  a   b   c   d   e
```

---
Move Constraints over Tiers

**Original**

- **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.

**Tier**

- Every lexical item has a **mother** labeled Move.
- Exactly one of a Move node’s **daughters** is a lexical item.

---

**Tree $n$-gram Templates**

<table>
<thead>
<tr>
<th></th>
<th>Move</th>
<th>SMC1</th>
<th>SMC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Move}$</td>
<td>$\geq 1 \text{ LI}$</td>
<td>$\text{Move}$</td>
<td>$\geq 2 \text{ LIs}$</td>
</tr>
<tr>
<td>$\text{SMC1}$</td>
<td>$\text{no LI}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{SMC2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

39
Example of Ill-Formed Derivation
Example of Ill-Formed Derivation

```
$                  Move
   |               |
  a      f        |
  |        |       |
  Move   Move    |
     |        |
    Merge  Merge
       |       |
      Merge  Merge
         |       |
        Merge  Merge
           |       |
          Merge  Merge
             |       |
              Merge  Merge
                 |       |
                  Merge  Merge
                     |   a b c e f
```

```
≥ 1 LI
SMC1

≥ 2 LI
SMC2

≥ 1 LI
no LI
```
Example of Ill-Formed Derivation
Example of Ill-Formed Derivation
Example of Ill-Formed Derivation

$\text{Merge}$

$g \quad \text{Move}$

$\text{Merge}$

$\text{Merge} \quad \text{Merge}$

$\text{Merge} \quad \text{Move} \quad f$

$a \quad b$

$\text{Merge}$

$d \quad e$

$\text{Move}$

$a \quad f$

$\geq 1 \text{ LI}$

$\text{Move}$

$\text{Move}$

$\geq 2 \text{ LI}$

$\text{SMC1}$

$\text{SMC2}$
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
  ⇒ 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
  ⇒ island constraints as a means to limit lexical blow-up?
Non-local dependencies are not particularly complex.

They are local over a very simple relativization domain.

Unified perspective on

- cognition
- acquisition
- typology
Conclusion

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Data Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Morphology</td>
<td>REG strings</td>
</tr>
<tr>
<td>Syntax</td>
<td>REG trees</td>
</tr>
</tbody>
</table>

**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 IV


References VI


