Grammar Size and Quantitative Restrictions on Movement

Thomas Graf

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

SCiL 2018
Jan 4–7, 2018
Outline

1. Movement in Minimalist Grammars
   - Merge and Move
   - Intermediate Movement

2. Single Movement Normal Form

3. Movement Constraints and Grammar Size: A Curious Conspiracy
Minimalist Grammars (MGs)

- Minimalist grammars (MGs) are a formalization of Chomskyian syntax (Stabler 1997, 2011)
- Succinct formalism for defining MCFGs
- Operations: Merge and Move
- Grammar is just a finite list of feature-annotated lexical items (LIs)

<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>
Merge

Merge combines subtrees to encode head-argument dependencies.

- **Category feature**: $N^-$, $V^-$, …
- **Selector feature**: $N^+$, $V^+$, …

---

<table>
<thead>
<tr>
<th>the</th>
<th>men</th>
<th>like</th>
<th>which</th>
<th>men</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N^+$</td>
<td>$N^-$</td>
<td>$D^+$ $D^+$ $V^-$</td>
<td>$N^+$ $D^-$</td>
<td>$N^-$</td>
</tr>
</tbody>
</table>

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
Merge

Merge combines subtrees to encode **head-argument dependencies**.

category feature  $N^-, V^-$, \ldots

selector feature  $N^+, V^+, \ldots$

<table>
<thead>
<tr>
<th>the</th>
<th>men</th>
<th>like</th>
<th>which</th>
<th>men</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N^+$</td>
<td>$N^-$</td>
<td>$D^+$</td>
<td>$D^+$</td>
<td>$V^-$</td>
</tr>
</tbody>
</table>

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
Merge combines subtrees to encode **head-argument dependencies**.

**Category feature** \( N^- , V^- , \ldots \)

**Selector feature** \( N^+ , V^+ , \ldots \)

- **the** and **men** have matching features, triggering Merge
- **same steps for which men**
- **like** merged with **which men**
- **like** merged with **the men**
Merge

Merge combines subtrees to encode **head-argument dependencies**.

category feature $N^−$, $V^−$, …

selector feature $N^+$, $V^+$, …

- the and men have matching features, triggering Merge
- same steps for which men
- like merged with which men
- like merged with the men
Merge

Merge combines subtrees to encode **head-argument dependencies**.

- **category feature**: $N^-, V^-, \ldots$
- **selector feature**: $N^+, V^+, \ldots$

---

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
Merge

Merge combines subtrees to encode **head-argument dependencies**.

- **Category feature**: $N^-, V^-, \ldots$
- **Selector feature**: $N^+, V^+, \ldots$

```
the                men
N^+    D^-

like
D^+    D^+    V^-

which        men
N^+    D^-
```

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
Merge

Merge combines subtrees to encode **head-argument dependencies**.

- **category feature**: \( N^-, V^- \), \( \ldots \)
- **selector feature**: \( N^+, V^+ \), \( \ldots \)

\[ \begin{array}{c}
\text{the} \\
\text{men}
\end{array} \quad \begin{array}{c}
\text{like} \\
\text{which} \\
\text{men}
\end{array} \]

- \( N^+ \) \( D^- \) \( N^- \) \( D^+ \) \( V^- \) \( N^+ \) \( D^- \) \( N^- \)

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
Merge

Merge combines subtrees to encode **head-argument dependencies**.

category feature $N^-, V^-$, …

selector feature $N^+, V^+$, …

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
**Merge**

Merge combines subtrees to encode **head-argument dependencies**.

- **category feature** $N^-, V^-, \ldots$
- **selector feature** $N^+, V^+, \ldots$

Diagram:

```
  VP
 /   \
DP    V'
  \   /  \\
   men like which men
   N^- D^- V^- N^+ D^- N^- D^+ V^- N^+ D^- N^-
  the
   N^+ D^-
```

- *the* and *men* have matching features, triggering Merge
- same steps for *which men*
- *like* merged with *which men*
- *like* merged with *the men*
Merge in Derivation Trees

```
[DP
  the [N+ D-]
  men [N-]]

VP
  like [D+ D+ V-]

V'
  which [N+ D-]
  men [N-]]

Derived Tree
```

```
[DP
  the [N+ D-]
  men [N-]]

[DP
  which [N+ D-]
  men [N-]]

V'
  like [D+ D+ V-]

Derived Tree
```
Move

Move displaces subtrees to derive the correct **linear order**.

**licensee feature**  $\text{wh}^-, \text{top}^-, \ldots$

**licensor feature**  $\text{wh}^+, \text{top}^+, \ldots$

\[
\begin{array}{c}
\text{do} \\
V^+ \quad \text{wh}^+ \quad C^-
\end{array}
\quad
\begin{array}{c}
\text{DP} \\
\text{the} \\
N^+ \quad D^-
\end{array}
\quad
\begin{array}{c}
\text{men} \\
N^-
\end{array}
\quad
\begin{array}{c}
\text{like} \\
D^+ \quad D^+
\end{array}
\quad
\begin{array}{c}
V^- \\
\text{which} \\
N^+ \quad D^-
\end{array}
\quad
\begin{array}{c}
\text{men} \\
\text{wh}^- \\
N^-
\end{array}
\]

- **Merge** *do*
- **Move** triggered by features of opposite polarity
Move

Move displaces subtrees to derive the correct **linear order**.

- Licensee feature: \( \text{wh}^-, \text{top}^- \), ...
- Licensor feature: \( \text{wh}^+, \text{top}^+ \), ...

- **Merge** `do`

- **Move** triggered by features of opposite polarity
Move

Move displaces subtrees to derive the correct **linear order**.

- Licensee feature $wh^-, top^-, \ldots$
- Licensor feature $wh^+, top^+, \ldots$

▶ Merge *do*

▶ Move triggered by features of opposite polarity
Move

Move displaces subtrees to derive the correct **linear order**.

licensee feature  $wh^-, top^-, \ldots$

licensor feature  $wh^+, top^+, \ldots$

▶ Merge *do*

▶ Move triggered by features of opposite polarity
Move

Move displaces subtrees to derive the correct **linear order**.

**licensee feature**  \( wh^-, \) top\(^-\), . . .

**licensor feature**  \( wh^+, \) top\(^+\), . . .

\[
\begin{align*}
CP & \rightarrow DP \rightarrow C' \\
C' & \rightarrow VP \\
VP & \rightarrow V' \\
V' & \rightarrow t \\
\end{align*}
\]

- Merge **do**
- Move triggered by features of opposite polarity
Move in Derivation Trees
Move in Derivation Trees
Intermediate Movement

Intermediate movement is possible, but has no effect on output.
An Issue with Intermediate Movement

Minimalist analyses posit an unbounded number of intermediate landing sites.

(1) Who does John think that Mary believes that Sue said that . . . that Bill hates t?

But every LI can only carry finitely many features!
Derivational Solution (Kobele 2006)

- Only final landing site has feature.
- Intermediate movement is inserted by mapping to phrase structure trees

Recipe for Successive Cyclic Movement

Add trace in every crossed Spec,CP.
Generalization: No Intermediate Movement (Graf et al. 2016)

- Kobele’s solution can be generalized.
- Intermediate movement never needs to be feature triggered.
- It is derivationally redundant.

Definition (Single Movement Normal Form)

An MG is in single movement normal form (SMNF) iff every LI has at most one licensee feature.

Theorem

For every MG there is a strongly equivalent MG that is in SMNF.
An Annoying Complication

MGs have one central locality restriction on Move.

Shortest Move Constraint (SMC)

If two LIs in a tree both have a licensee feature as their first currently unchecked feature, then these features must be distinct.
An Annoying Complication

MGs have one central locality restriction on Move.

Shortest Move Constraint (SMC)

If two LIs in a tree both have a licensee feature as their first currently unchecked feature, then these features must be distinct.
An Annoying Complication

MGs have one central locality restriction on Move.

**Shortest Move Constraint (SMC)**

If two LIs in a tree both have a licensee feature as their first currently unchecked feature, then these features must be distinct.
An Annoying Complication

MGs have one central locality restriction on Move.

Shortest Move Constraint (SMC)

If two LIs in a tree both have a licensee feature as their first currently unchecked feature, then these features must be distinct.
An Annoying Complication

MGs have one central locality restriction on Move.

Shortest Move Constraint (SMC)

If two LIs in a tree both have a licensee feature as their first currently unchecked feature, then these features must be distinct.

“Don’t cross the streams!”

N+  D−  wh−  N−  D+  D+  V−  N+  D−  wh−  N−
A Work-Around with Subscripts

Feature Subscripting

- For every LI $l$, only keep its last licensee feature.
- Add subscripts to licensee features to avoid SMC violations.
A Work-Around with Subscripts

Feature Subscripting

- For every LI \( l \), only keep its last licensee feature.
- Add subscripts to licensee features to avoid SMC violations.
Lexical Blow-Up

- SMNF translation induces **linear lexical blow-up**
- Effect varies a lot depending on movement configurations:
  - lower bound  linear size reduction(!),
    1:1 for non-redundant grammars
  - upper bound  large linear blow-up

\[
\sum_{l \in \text{Lex}} \mu \gamma(l) + \delta(l)
\]

- \(\mu\) . . . maximum number of required indices
- \(\gamma(l)\) . . . number of licensor features of LI \(l\) in original grammar
- \(\delta(l)\) . . . 1 if \(l\) has licensee features, 0 otherwise
Abstract Example (Sketch)
Abstract Example (Sketch)
Abstract Example (Sketch)
Abstract Example (Sketch)
Abstract Example (Sketch)
Abstract Example (Sketch)
Abstract Example (Sketch)
Abstract Example (the Math)

- The original grammar contains a single LI.
  \[ c ::= C^+ f^+ C^+ g^+ f^+ C^- g^- f^- \]

- The SMNF grammar contains 8 variants.
  \[
  \begin{align*}
  c &:: C^+ f_0^+ C^+ f_0^+ C^- f_0^- \\
  c &:: C^+ f_0^+ C^+ f_1^+ C^- f_0^- \\
  c &:: C^+ f_1^+ C^+ f_0^+ C^- f_0^- \\
  c &:: C^+ f_1^+ C^+ f_1^+ C^- f_0^- \\
  c &:: C^+ f_1^+ C^+ f_0^+ C^- f_1^- \\
  c &:: C^+ f_1^+ C^+ f_1^+ C^- f_1^- \\
  c &:: C^+ f_1^+ C^+ f_0^+ C^- f_1^- \\
  c &:: C^+ f_1^+ C^+ f_1^+ C^- f_1^- 
  \end{align*}
  \]

- We can get \( n \) variants of the LI by changing the phonetic exponent, so the grammar size increases at least by \( 8n \).

- But we can keep increasing number \( m \) of arguments:
  \[ 2^{m+1} \times n \]
Interim Summary

- Every MG is a finite set of LIs.
- The more LIs, the larger the grammar.
- Derivation trees are the primary data structure.
- Intermediate movement is derivationally redundant and costly:
  - complicates proofs
  - increases computational complexity (Graf and Heinz 2015)
  - at odds with MG processing models (Graf et al. 2017)
- But SMNF MGs may be much larger, which is bad for
  - parsing
  - learning
  - explanatory adequacy

A New Empirical Puzzle

Are the movement configurations we find in natural language exactly those that induce little lexical blow-up?
Interim Summary

- Every MG is a finite set of LIs.
- The more LIs, the larger the grammar.
- Derivation trees are the primary data structure.
- Intermediate movement is derivationally redundant and costly:
  - complicates proofs
  - increases computational complexity (Graf and Heinz 2015)
  - at odds with MG processing models (Graf et al. 2017)
- But SMNF MGs may be much larger, which is bad for
  - parsing
  - learning
  - explanatory adequacy

A New Empirical Puzzle

Are the movement configurations we find in natural language exactly those that induce little lexical blow-up?
What Produces a Large Blow-Up in Grammar Size?

- Large blow-up occurs whenever there are multiple LIs s.t.
  1. they all have the same final movement feature, and
  2. they have overlapping movement paths, and
  3. their relative configuration is not fixed across derivations.

- That’s easy to do with abstract examples, but natural examples are tough.

The Constraint-Grammar-Size Conspiracy

Patterns that would induce a large blow-up are independently forbidden.
What Produces a Large Blow-Up in Grammar Size?

- Large blow-up occurs whenever there are multiple LIs s.t.
  1. they all have the same final movement feature, and
  2. they have overlapping movement paths, and
  3. their relative configuration is not fixed across derivations.

- That’s easy to do with abstract examples, but **natural examples are tough**.

The Constraint-Grammar-Size Conspiracy

Patterns that would induce a large blow-up are independently forbidden.
Improper Movement

(2) John wonders who Bill saw.

(3) Who wonders Bill saw?
Improper Movement

(2) John wonders who Bill saw.
Improper Movement

(2) John wonders who Bill saw.
Improper Movement

(2) John wonders who Bill saw.

(3) * Who wonders Bill saw?
Superraising

John seems Bill believes Mary likes John.
Superraising

Merge

C

Move

T

Move

seems

Move

Bill

believes

Move

John

likes

Mary
Superraising

(4) * John seems Bill believes likes Mary
Freezing Effects

- One could also get overlapping paths by extracting an $f$-mover from within an $f$-mover.
- But this produces freezing effects.

(5) It seems your comment about John annoys Sue.
(6) * John seems your comment about $t$ annoys Sue.
(7) * Who don’t you know [which pictures of $t$] Mary bought.
Wh-Islands

▶ Multiple A′-movers of the same type would also be an option.
▶ But here the wh-island constraint intervenes.

(8) a.  \textbf{What}_{wh} did John say Mary gave \textit{t}_{wh} to Bill?
     
b.  * \textbf{What}_{wh0} did John say \textbf{who}_{wh1} Mary gave \textit{t}_{wh0} to \textit{t}_{wh1} ?
     (Wh-island violation)

c.  * \textbf{What}_{wh0} did Bill think \textbf{which}_{wh1} man \textit{t}_{wh1} says \textbf{who}_{wh1} Mary gave \textit{t}_{wh0} to \textit{t}_{wh1} ?

d.  * \textbf{What}_{wh0} did Sue claim \textbf{who}_{wh1} Bill thinks \textbf{which}_{wh2} man 
     \textit{t}_{wh2} says Mary gave \textit{t}_{wh0} to \textit{t}_{wh1} ?
Taking Stock

- MGs are all about two structure-building operations: **Merge** and **Move**.
- Intermediate movement complicates formalism.
- SMNF simplifies MGs, but at the risk of larger lexicons.
- Realistic grammars block the truly dangerous configurations.
- Unclear whether this is **coincidence or conspiracy**
References


