On the Order of Syntactic Operations
Lecture 3: A Local Optimality-Theoretic Perspective
Compact course, University of Cambridge, June 14-19, 2013
Gereon Müller (Universität Leipzig)

Note: This handout mainly reports joint work with Fabian Heck; see Heck & Müller (2013b).

1. Introduction

Four claims:

- There is a competition between Merge and Agree which can and must be resolved in one way or the other.
  (We already know that from lecture 2.)

- Resolving the competition by giving preference to (ranking) one or the other requirement accounts for (a) ergative vs. accusative patterns of argument encoding (on the vP cycle) and (b) a restriction on ergative movement (on the TP cycle). This can be modelled by invoking a (parametrized) preference requirement in minimalist syntax. (Again, we already know that from lecture 2.) Alternatively, it can be modelled in an optimality-theoretic way. (This will be new, but not very different.)

- Movement in accusative encoding systems is in some contexts accelerated, so that it applies before Agree; and movement in ergative encoding systems is in some contexts decelerated, so that it applies after Agree. This accounts for a priori unexpected mobility restrictions on dative arguments in German, and for a priori unexpected movement options for lexically case-marked (and some ergative) arguments in some languages. Such effects can straightforwardly be derived in an optimality-theoretic approach (they signal the presence of more specific, higher-ranked constraints); they are less straightforward in a minimalist approach.

- The reasoning presupposes an extremely local, derivational approach to optimization (because of the opacity (counter-bleeding) effects).

2. Background

Question: Does syntactic optimization apply once (harmonic parallelism: representational syntax) or more than once (harmonic serialism: derivational syntax)? If the latter holds: Is optimization global or local?

Viability of derivational OT:

- McCarthy (2008; 2010) and much related recent work (also cf. McCarthy (2000), and to some extent McCarthy (2007a)): A harmonic serialism (derivational OT) approach to phonology is viable (and empirically preferable).


Two kinds of derivational OT syntax:

While some see a major divide between the derivationally-oriented MP and OT, we do not. Of course, there are likely to be differences of empirical import between the non-derivational, chain-based theory of “Shortest Move” developed here and a particular derivational MP proposal, but such differences seem comparable to those between different approaches to syntax within OT, or to those between different proposals within MP; they do not seem to follow from some major divide between the OT and MP frameworks. In fact, derivational theories can be naturally formalized within OT. [2] “Harmonic serialism” is a derivational version of OT developed in Prince & Smolensky (1993) in which each step of the derivation produces the optimal next representation. [3] Another approach, seemingly needed to formalize MP within OT has Gen produce derivations; it is these that are evaluated by the constraints, the optimal derivation being determined via standard OT evaluation. Thus, on our view, while the issue of derivations is an important one, it is largely orthogonal to OT.

Legendre et al. (1998, 285-286)

Two ways to reconcile a derivational approach to syntax with OT:

- standard, parallel optimization of full derivations

- serial optimization
A relevant issue in the second case:

- In classical transformational grammar (e.g., Chomsky (1965)), syntactic transformations applying to the output of the base component effect derivational steps where the input and the output have roughly the same size, exactly as in phonology.
- In the minimalist program, operations of the "base" component and of the "transformational" component are systematically interspersed; syntactic structures start with two lexical items and grow throughout the derivation by iterated application of (external or internal) Merge. Here, iterated optimization cannot apply to objects of the same size—the optimal output of one optimization procedure is smaller than the optimal output of the next optimization procedure.

Consequence: two dichotomies

- Optimization may be parallel (i.e., apply once) or serial (i.e., apply more than once).
- Optimization may be global (applying to the full sentence) or local (applying also to smaller domains).

Serial optimization and locality

- Serial optimization in phonology must be global (phonology is not structure-building).
- Serial optimization in minimalist syntax must be local: The linguistic expressions created by Merge grow steadily throughout the derivation.

What are the local domains for optimization?

1. Classical assumption: The whole sentence is subject to a single, parallel optimization procedure (Grimshaw (1997), Pesetsky (1998), Legendre et al. (1998), etc.). The output candidates are usually taken to be representations (also in approaches that assume them to be generated by a minimalist GEN component; see Broekhuis & Dekkers (2000), Broekhuis (2000; 2006; 2008), Salzmann (2012)); but they can also be full derivations (as, e.g., in Müller (1997)).


3. Multiple optimization of smaller optimization domains: closely related to developments in the minimalist program.

Trade-offs: conceptual arguments

- The smaller the optimization domain is, the more the complexity of the overall system is reduced (reduction of the size of candidate sets).
- The larger the optimization domain is, the less often optimization procedures have to be carried out.

The view adopted here:

- Iterated optimization in small domains is more economical than single optimization of extremely large domains.
- Ultimately, empirical arguments should decide the issue: If the ranked constraints have access to more/less structure, a wrong winner is predicted.

2. Proposals for local domains for optimization:

a. Minimal clause:
   Ackema & Neeleman (1998) on wh-movement in Czech; Müller (2003) on extraction from verb-second clauses in German

b. Phase:
   Fanselow & Čavar (2001) on MeN-deletion in Malay; Müller (2000a; 2002) on R-pronouns in German

c. Phrase:

d. Derivational step:
   Heck & Müller (2007; 2013a) on gender agreement with dative possessors in German DPs. expletives in German verb-second clauses. and VP topicalization and do-support in English; Müller (2004; 2009) on ergative and accusative argument encoding patterns; Lahme (2008; 2009) on excluding SVO in ergative languages; Georgi (2012a) on global case splits in Tainyu; Georgi (2012b) on anti-agreement effects and defective intervention.

Extremely local optimization:
Let us pursue the consequences of the most radical position within a theory of local optimization, where the domain is the derivational step (also see McCarthy (2010)).

See also:
This is tantamount to the claim that each transformational rule application constitutes a ‘phase’ which we believe to be the null hypothesis.

Epstein & Seely (2002, 77)

Shape of the argument:

1. Sometimes, the order of applying Agree and Merge is under-determined. If there are no simultaneous rule applications in the grammar (see Epstein & Seely (2002); contra Pullum (1979), Chomsky (2008)), then a conflict arises: Only one of them can be executed at each step.

2. The conflict can be resolved by ranking the requirements: The highest-ranked requirement is satisfied immediately; lower-ranked ones must remain unsatisfied at the current derivational step. Such unsatisfiability does not lead to a crash of the derivation and thus suggests an analysis in terms of violable constraints.

3. If the optimization domain is larger than the step-level then, ceteris paribus, the order of elementary operations that is imposed by the ranking under step-level optimization cannot be preserved. Empirically, this is the wrong result.

3. Merge vs. Agree on the vP level: Accusative vs. Ergative Encoding

Refs.: Müller (2009); Heck & Müller (2013a)

3.1 Assumptions

(3) Clause structure:
[cr C [TP T [vP VP ext [\textit{v} [vP V DP int ]]]]]

(4) Two types of features that drive operations:

a. Structure-building features (edge features, subcategorization features) trigger Merge: [*F*].

b. Probe features trigger Agree: [*F*].

Three operations (Chomsky (2001)):

(5) Merge:
\(\alpha\) can be merged with \(\beta\), yielding \([\alpha, \alpha, \beta]\) if \(\alpha\) bears a structure-building feature [*F*] and \(F\) is the label of \(\beta\).

(6) Move:
Move is Merge. with \(\beta\) internal to \(\alpha\).

(7) Agree:
\(\alpha\) can agree with \(\beta\) with respect to a feature bundle \(\Gamma\) iff (a) and (b) hold:

a. \(\alpha\) bears a probe feature [*F*] in \(\Gamma\) and may thereby provide the \(\alpha\)-value for a matching goal feature [F] of \(\beta\) in \(\Gamma\).

b. \(\alpha\) m-commands \(\beta\).

(This permits an Agree relation between a head and its specifier.)

Constraints:

Designated constraints ensure that Merge (incl. Move) and Agree must take place as soon as their context of application is present. (This derives an earliness requirement for syntactic operations; see Pesetsky (1989).)

(8) Agree Condition (AC):
Probes ([*F*]) participate in Agree.

(9) Merge Condition (MC):
Structure-building features ([*F*]) participate in Merge.

(10) The role of \(T\) and \(v\) in argument encoding:

a. \(T\) bears [*CASE:ext*] that instantiates [CASE:ext] on DP.

b. \(v\) bears [*CASE:int*] that instantiates [CASE:int] on DP.

A conspicuous property:
The head \(v\) has a dual role: It participates in a Merge operation with a DP, and it also participates in an Agree relation with a DP. This dual role has far-reaching consequences for the nature of argument encoding.

A constraint conflict:

Consider a simple transitive context, with two arguments \(DP_{int}, DP_{ext}\). Suppose that the derivation has reached a stage \(\Sigma\) where \(v\) has been merged with a VP containing \(DP_{int}\), with \(DP_{ext}\) waiting to be merged with \(v\) in the workspace of the derivation. At this point, a conflict arises: AC demands that the next operation is Agree\((v,DP_{int})\) (see (a)). MC demands that it is Merge\((DP_{ext},v)\) (see (b)).

(11) Stage \(\Sigma\):

\[
\begin{tikzpicture}
\node (v) at (0,0) {$v$};
\node (dpint) at (-2,0) {$DP_{int}$};
\node (dpext) at (2,0) {$DP_{ext}$};
\node (vp) at (0,1) {$VP$};
\draw[->] (v) to (vp); \\
\end{tikzpicture}
\]
3.2 Analysis

(12) a. Agree before Merge: accusative

\[ T \rightarrow T' \]

\[ T' \rightarrow D_{[\text{int}]} \quad \text{DP}_{[\text{int}]} \]

\[ v \rightarrow v' \]

\[ \text{V} \rightarrow \text{DP}_{[\text{int}]} \]

\[ \text{TP} \rightarrow \text{TP} \]

\[ \text{T}_{[\text{sc-ext}]} \rightarrow \text{T}_{[\text{sc-ext}]} \]

\[ \text{vP} \rightarrow \text{vP} \]

\[ \text{DP}_{[\text{int}]} \rightarrow \text{DP}_{[\text{int}]} \]

\[ \text{V} \rightarrow \text{V} \]

\[ \text{DP}_{[\text{ext}]} \rightarrow \text{DP}_{[\text{ext}]} \]

b. Merge before Agree: ergative pattern

\[ T \rightarrow T' \]

\[ T' \rightarrow D_{[\text{int}]} \quad \text{DP}_{[\text{int}]} \]

\[ v \rightarrow v' \]

\[ \text{V} \rightarrow \text{DP}_{[\text{int}]} \]

\[ \text{TP} \rightarrow \text{TP} \]

\[ \text{T}_{[\text{sc-ext}]} \rightarrow \text{T}_{[\text{sc-ext}]} \]

\[ \text{vP} \rightarrow \text{vP} \]

\[ \text{DP}_{[\text{int}]} \rightarrow \text{DP}_{[\text{int}]} \]

\[ \text{V} \rightarrow \text{V} \]

\[ \text{DP}_{[\text{ext}]} \rightarrow \text{DP}_{[\text{ext}]} \]

\[ \text{Note:} \]

The derivation of the ergative pattern presupposes that a specifier is preferred with respect to Agree with its head to an item included in the complement of that head. This can be formulated as the Specifier-Head Bias (Chomsky 1986; 1995). Koopman (2006); see Béjar & Rezác (2009) for a similar idea with the bias inversed.

(13) SPECIFIER-HEAD BIAS (SHB):
Spec/head Agree is preferred to other instances of Agree.

This replaces standard minimality conditions (Relativized Minimality, MLC) (though with a somewhat different empirical coverage). The SHB is compatible with equidistance effects, which pose a problem for path-based definitions of minimality.

(14) Rankings:

a. (SHB) \(\Rightarrow\) AC \(\Rightarrow\) MC (accusative pattern)

b. (SHB) MC \(\Rightarrow\) AC (ergative pattern)

\(T_1\): Accusative pattern. step 1 (\(\Sigma\) as input): Agree

\[
\begin{array}{|c|c|c|}
\hline
\text{Input:} & \text{Workspace:} & \text{SHB}\ AC\ MC \\
\text{\([v\ [\text{case}:\text{int}\ ]/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]\)} & \{\text{DP}_{[\text{case}:\text{int}]/\ldots},\ \text{T}_{[\text{case}:\text{ext}/\ldots}]\} & \text{SHB}\ AC\ MC \\
\hline
\text{O}_1: & \text{[v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \#! \\
\hline
\text{O}_2: & \text{[v\ [\text{case}:\text{int}]/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \#! \\
\hline
\end{array}
\]

\(T_2\): Accusative pattern. step 2: Merge

\[
\begin{array}{|c|c|c|}
\hline
\text{Input:} & \text{Workspace:} & \text{SHB}\ AC\ MC \\
\text{\([v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]\)} & \{\text{DP}_{[\text{case}:\text{int}]/\ldots},\ \text{T}_{[\text{case}:\text{ext}/\ldots}]\} & \text{SHB}\ AC\ MC \\
\hline
\end{array}
\]

\(T_3\): Accusative pattern. step 3: Agree

\[
\begin{array}{|c|c|c|}
\hline
\text{Input:} & \text{Workspace:} & \text{SHB}\ AC\ MC \\
\text{\([v\ T_{[\text{case}:\text{ext}/\ldots}]/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \{\text{DP}_{[\text{case}:\text{int}]/\ldots},\ \text{T}_{[\text{case}:\text{ext}/\ldots}]\} & \text{SHB}\ AC\ MC \\
\hline
\end{array}
\]

\(T_4\): Ergative pattern step 1 (\(\Sigma\) as input): Merge

\[
\begin{array}{|c|c|c|}
\hline
\text{Input:} & \text{Workspace:} & \text{SHB}\ MC\ AC \\
\text{\([v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \{\text{DP}_{[\text{case}:\text{int}]/\ldots},\ \text{T}_{[\text{case}:\text{ext}/\ldots}]\} & \text{SHB}\ MC\ AC \\
\hline
\text{O}_1: & \text{[v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \#! \\
\hline
\text{O}_2: & \text{[v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \#! \\
\hline
\end{array}
\]

\(T_5\): Ergative pattern step 2: Agree (with \(\text{DP}_{\text{ext}}\))

\[
\begin{array}{|c|c|c|}
\hline
\text{Input:} & \text{Workspace:} & \text{SHB}\ MC\ AC \\
\text{\([v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \{\text{DP}_{[\text{case}:\text{int}]/\ldots},\ \text{T}_{[\text{case}:\text{ext}/\ldots}]\} & \text{SHB}\ MC\ AC \\
\hline
\text{O}_1: & \text{[v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \#! \\
\hline
\text{O}_2: & \text{[v\ [\text{case}:\text{int}/\omega/\ldots\text{DP}_{[\text{case}:\text{int}]/\ldots}]} & \#! \\
\hline
\end{array}
\]

\(\text{Note:} \)

In intransitive contexts, only \(T\) can still assign case via Agree. This derives the basic

ergative and accusative patterns of argument encoding (a version of Murasugi’s (1992) approach).

3.3 Extremely Local vs. Less Local Optimization

\(\text{Note:} \)

If phrases (or phrases, or clauses, or sentences) are the local domains for optimization, things do not change in ergative contexts (see \(T_5\)). But a wrong winner is predicted in accusative contexts (see \(T_5\)). Legitimate assignment of accusative case by \(v\) in the presence of \(\text{DP}_{\text{ext}}\) in Specv emerges as an instance of counter-bleeding opacity (Chomsky (1951; 1975), Kiparsky (1973)) – one that cannot be accounted for representationally by adding postulating abstract items (like traces).

\(\text{Note:} \)

In intransitive contexts, only \(T\) can still assign case via Agree. This derives the basic
\( T_b: \) **Ergative pattern, step 3: Agree**

\[
\begin{array}{c|c|c}
\text{Input:} & \text{Workspace} \rightarrow \{\ldots\} & \text{SHB, MC, AC} \\
[\text{\textit{DP}}\text{-\textit{case:ext}}, \text{\textit{DP}}\text{-\textit{case:inf}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots] & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} \\
\hline
\text{\textit{O}_1:} & \text{\textit{T}}^{\textit{vP}} \text{\textit{DP}}\text{-\textit{case:inf}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots & \text{SHB, MC, AC} \\
\end{array}
\]

\( T_v: \) *vP optimization under MC \( \gg \) AC (*“ergative”*) ranking: right result

\[
\begin{array}{c|c|c}
\text{Input:} & \text{Workspace} \rightarrow \{\text{T}\text{-\textit{case:ext}}, \ldots\} & \text{SHB, MC, AC} \\
[\text{\textit{DP}}\text{-\textit{case:ext}}, \text{\textit{vP}}\text{-\textit{case:inf}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots] & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} \\
\hline
\text{\textit{O}_1:} & \text{\textit{vP}} \text{\textit{DP}}\text{-\textit{case:inf}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots & \text{SHB, MC, AC} \\
\text{\textit{O}_2:} & \text{\textit{vP}} \text{\textit{DP}}\text{-\textit{case:ext}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots & \text{\textit{\textbf{!}}} \\
\end{array}
\]

\( T_k: \) *vP optimization under AC \( \gg \) MC (*“accusative”*) ranking: wrong result

\[
\begin{array}{c|c|c}
\text{Input:} & \text{Workspace} \rightarrow \{\text{T}\text{-\textit{case:ext}}, \ldots\} & \text{SHB, AC, MC} \\
[\text{\textit{DP}}\text{-\textit{case:ext}}, \text{\textit{vP}}\text{-\textit{case:inf}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots] & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} & \text{\textit{vP}} \\
\hline
\text{\textit{O}_1:} & \text{\textit{vP}} \text{\textit{DP}}\text{-\textit{case:inf}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots & \text{SHB, AC, MC} \\
\text{\textit{O}_2:} & \text{\textit{vP}} \text{\textit{DP}}\text{-\textit{case:ext}}, \text{\textit{vP}} \ldots \text{\textit{DP}}\text{-\textit{case:ext}} \ldots & \text{\textit{\textbf{!}}} \\
\end{array}
\]

4. **Interlude: Syntax vs. Phonology**

**Opacity in Syntax vs. Opacity in Phonology:**

- **Syntax:**
  1. Standard global parallel optimization cannot capture opacity effects.
  2. Extremely local serial optimization can capture opacity effects.
     (However, even slightly less local optimization with the phrase as the optimization domain, cannot account for the instance of opacity at hand.)

- **Phonology:**
  1. Standard parallel optimization cannot capture opacity effects.
     (Possible ways out: (i) enrichment of representations with abstract material encoding earlier movement steps: *turbid phonology* (Goldrick 2000), *virtual phonology* (Bye 2001), *coloured containment* (Oostendorp 2006, 2007), Trommer (2011)); (ii) *sympathy theory* (McCarty 1999): non-optimal candidates encoding earlier derivational stages whose properties the optimal candidate must be faithful to; (iii) *candidate chains* (McCarty 2007b): faithfulness violations incurred in the derivation are recorded on output candidates, and *precedence constraints* demand a certain order of faithfulness violations in the output candidate.)
  2. Serial optimization cannot account for opacity either.

**Observation** (McCarty (2007a, 37-38; 69-70)):

Serial optimization as such does not capture counter-bleeding and counter-feeding in phonology.

(15) **Counter-Bleeding in Bedouin Arabic**

- **UR**
  - /\textit{hakim-in}/
  - Palatalization: \textit{hak-im-in}
  - Vowel deletion: \textit{hak-m-in}
  - Output: \textit{hak-m-in}

See /\textit{t}-hakim-in/ \( \rightarrow \) [\textit{thakm-in}]

(16) **Counter-Feeding in Bedouin Arabic**

- **UR**
  - /\textit{da-fa}/
  - Vowel deletion: -
  - Vowel raising: \textit{di-fa’}
  - Output: \textit{di-fa’}

See /\textit{fari-b-at}/ \( \rightarrow \) [\textit{farbat}]

**Assumption** (McCarty (2000); also see Heck & Müller (2007)):

Optimization applies cyclically: The output of one competition forms the sole input for the next competition, and so on, until the derivation converges, i.e., until subsequent optimizations always return the same candidate as optimal.

**Additional assumption:**

The competing outputs generated by GEN on the basis of a given input differ from the input only by at most one application of an elementary operation.

**Problem:**

Opacity cannot be derived in this way.

(17) **Counter-Bleeding does not work**

a. **Constraints:**

   (i) *iCV*: triggers \textit{i}-deletion.
   (ii) *\textit{k}*: triggers palatalization of the velar before \textit{i}
   (iii) MAX: prohibits deletion of \textit{i}
   (iv) ID(back): prohibits palatalization of \textit{k}

b. **Cycle 1: wrong winner**

\[
\begin{array}{c|c|c|c|c}
\text{Input:} & \text{Output:} & \text{Constraint:} & \text{Result:} \\
\hline
/\textit{hakim-in}/ & \textit{*k}* & \textit{iCV} & \textit{MAX}\text{-\textit{ID}(back)} & \text{\textbf{!}} \\
\textit{hakim-in} & * & * & * & * \\
\textit{hak-im-in} & *! & * & * & * \\
\textit{hak-m-in} & *! & * & * & * \\
\end{array}
\]

- Validation
c. Cycle 2: fatal convergence

<table>
<thead>
<tr>
<th>/backimin/</th>
<th><em>/ks</em></th>
<th><em>/CV</em></th>
<th>MAX</th>
<th>Id(back)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
</tbody>
</table>

Underlying reason for the failure of serial optimization to capture counter-bleeding:

(i) In a rule-based approach, early application of the palatalization rule brings about palatalization.

(ii) In a serial OT approach, a high ranking of the constraint */ks*, which triggers palatalization, does in fact not trigger early palatalization here: The constraint triggers palatalization only indirectly and can in principle also be satisfied by other elementary operations that are also triggered in an indirect way by a markedness constraint.

(18) Another non-solution: reranking of constraints

a. Cycle 1: correct intermediate winner

<table>
<thead>
<tr>
<th>/backimin/</th>
<th><em>/ks</em></th>
<th><em>/CV</em></th>
<th>MAX</th>
<th>Id(back)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
</tbody>
</table>

b. Cycle 2: wrong final winner, fatal convergence

<table>
<thead>
<tr>
<th>/backimin/</th>
<th><em>/ks</em></th>
<th><em>/CV</em></th>
<th>MAX</th>
<th>Id(back)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
<tr>
<td>hackimin</td>
<td><em>!</em></td>
<td>*</td>
<td>!</td>
<td>*</td>
</tr>
</tbody>
</table>

Reason:
The high ranking of MAX vis-à-vis the counteracting markedness constraint */iCV* is needed in cycle 1 to generate the intended intermediate output, but afterwards it proves fatal since *i*-deletion is then not permitted anymore. (More generally, *i*-deletion should then be blocked.)

Conjecture:
Counter-bleeding can be derived by serial optimization in syntax, but not in phonology, because in syntax, it typically does not happen that the application of one elementary operation leads to a satisfaction of two markedness constraints (e.g., AC and MC). Strategy for phonology: Ensure (by adding an appropriate highly ranked, or inviolable, constraint) that only one new markedness constraint can be satisfied by an optimal candidate.

5. Merge vs. Agree on the TP level: A Constraint on Ergative Movement

Ref: Assmann, Georgi. Heck, Müller & Weisser (2012)

5.1 Data

In many morphologically ergative languages, ergative arguments (DP_{erg}) cannot undergo Á-movement (wh-movement, focussing, relativization).

(19) Wh-Movement of DP_{erg} vs. DP_{abs} in Kaqchikel (Mayan):

a. *achike n-Ô-n-lòq” jun sìk’i:w”
  Q INCOMPL-3SG.ABS-3SG.ERG-buy INDEF book
  ‘Who buys a book?’

b. atux n-Ô-n-lòq” a Carlos?
  Q INCOMPL-3SG.ABS-3SG.ERG-buy CL Carlos
  ‘What does Carlos buy?’

c. achike ri n-Ô-tze’en?
  Q DET INCOMPL-3SG.ABS-laugh
  ‘Who laughs?’

5.2 Assumptions

Locality of movement:
Movement takes place successive-cyclically, from one XP edge domain to the next one higher up. Given the Phase Impenetrability Condition (PIC; Chomsky (2001)), this follows automatically if every XP is a phase.

Edge features:
It must be ensured that intermediate steps of movement as required under the PIC are possible in the first place in a model of syntax where all operations are feature-drives. A standard assumption here is that edge features ([*X*]) that trigger intermediate movement steps can be inserted on all intervening phase heads.

Maraudage:
Certain goal features require checking in Spec/head configurations; this way, they may “maraud” a functional head and take away features that should normally be reserved for some other item. (See Georgi, Heck & Müller (2009); similar concepts are suggested in Chomsky (2001), Abels (2003), Anagnostopoulou (2005), Adger & Harbour (2007), Béjar & Rezáč (2009).

Case features and maraudage:
Structural case features trigger maraudage in Spec/head configurations even if they have already been checked (or valued). Independent motivation: the existence of case
stacking in the world’s languages (see Andrews (1996), Nordlinger (1998), Richards (2007)).

(20) **Activity of structural case features:**

Structural case features act as active goals.

Note:
Given the SHB, the configuration in (21-a) may involve checking of [case:int] by X or not (leading to a crash of the derivation or not because of an unchecked [case:C]), whereas the configuration in (21-b) must involve checking of [case:int] by X (which invariably leads to a crash).

(21)  
\[ \text{X: } X_{\text{case:ext*}} \{ZP \ldots \alpha_{\text{case:int}} \ldots \beta_{\text{case:C}} \ldots \} \]
\[ \text{X: } X_{\text{case:ext*}} \{ZP \ldots \alpha \ldots \beta_{\text{case:C}} \ldots \} \]

Recall:
There is no minimality condition on Agree or Merge; minimality effects are derivable from the PIC; see Müller (2011). Suppose that both \( \alpha \) and \( \beta \) are PIC-accessible to X in (21); this will imply that the PIC is slightly less restrictive, as eventually proposed in Chomsky (2001), or that Agree operations can escape the PIC, as suggested by Bošković (2007), among others.

Assumption:
Checking of [case:int] on \( \alpha \) with a conflicting [case:ext*] on X is harmless as such; \( \alpha \) will simply maintain its original feature value. However, [case:ext*] is then discharged, and not available for further operations anymore.

5.3 Analysis

5.3.1 Displacement in Languages with Ergative Encoding Patterns

(i) **DP_{erg} Movement**
Given the PIC, DP_{erg} needs to move from Specv to SpecT if it is to undergo subsequent movement to SpecC (wh-movement, relativization, focus movement). Given that the “ergative” ranking MC \( \gg \) AC is also maintained on the TP cycle, movement of DP_{erg} (as an instance of internal Merge) will have to precede Agree of T with the VP-internal DP that has not yet valued its case feature (as absolutive). Given the SHB, DP_{erg} will next maraud T’s case probe; the internal argument DP will consequently remain without a checked case feature. Assuming that all DPs must have their case features checked eventually (and assuming that there is no such thing as a default case), the derivation will crash. In a nutshell, ergative movement is impossible because the remaining argument cannot get absolutive case in this context.

(22) **Illegitimate movement of DP_{erg}**

a. Structure after T is merged

```
<table>
<thead>
<tr>
<th>TP</th>
</tr>
</thead>
</table>
| T'[
| V
| vP
| DP[case:int]
| \{\text{v} \ldots \text{DP[case:int]} \ldots \}] |
| V |
| DP[case:C] |
```

b. MC \( \gg \) AC triggers movement of DP_{erg} first

```
<table>
<thead>
<tr>
<th>TP</th>
</tr>
</thead>
</table>
| T'[
| V
| vP
| DP[case:int]
| \{\text{v} \ldots \text{DP[case:int]} \ldots \}] |
| V |
| DP[case:C] |
```

c. SHB triggers maraudage of T

```
<table>
<thead>
<tr>
<th>TP</th>
</tr>
</thead>
</table>
| T'[
| V
| vP
| DP[case:int]
| \{\text{v} \ldots \text{DP[case:int]} \ldots \}] |
| V |
| DP[case:C] |
```

T_{0}: Ergative movement. step 1: Move

```
| Input: \{\ldots \} |
| Workspace = \{\ldots \} |
| SHB, MC, AC |
| \| |
| **O1**: \{\ldots \} |
| **O2**: \{\ldots \} |
```

| **O1**: \{\ldots \} |
| **O2**: \{\ldots \} |

```
```
T10: Ergative movement, step 2: Agree (maraudage)

Assumption re inessentiality:
A DP with an unchecked case feature will eventually lead to a crash of the derivation; this is an instance of Grimshaw’s (1994) “no good output” approach to absolute ungrammaticality.

(ii) DPabs Movement
No such problem arises for movement of DPabs because DPerg has already been assigned case when DPabs moves to SpecT.

(23) Legitimate movement of DPabs

a. Structure after T is merged

b. MC ≫ AC triggers movement of DPabs first

c. Finally, Agree with T ensures external case of DPabs; no maraudage

Note:
On the vP cycle in (23-a), MC ≫ AC ensures that external Merge of DPext and (subsequent; Chomsky (2001; 2008)) internal Merge of DPint (both triggered by ([X•]) features on v) both precede Agree. Since there is no MLC-like constraint and both items occupy a Specv position (so the SHB does not discriminate the options), the derivation can now proceed in two ways: Agree(v, DPext) ultimately leads to a well-formed output, as indicated; in contrast, Agree(v, DPint) in (23-a) would lead to a crash because DPext would then never be assigned case.

5.3.2 Displacement in Languages with Accusative Encoding Patterns

(iii) DPacc Movement
The ranking Agree ≫ Merge that gives rise to an accusative pattern in the first place (on the vP cycle) is also active on the TP cycle. Here it ensures that Agree with the DPacc in Specv can be carried out before the DPacc undergoes successive-cyclic movement to SpecT (and then to a higher position).
T11: Absolutive movement, step 1: Move

\[ \text{Input: } \begin{array}{|c|c|c|} \hline \text{TP } T_{\text{case:ext}^c} [x^*] [VP DP_{\text{case}} [x^*] \langle \text{DP}_{\text{case}} \rangle ] \langle \text{DP}_{\text{case:inf}} \rangle \langle \text{DP}_{\text{case}} \rangle \langle \text{v} \ldots t \ldots \rangle ] & \text{SHB} & \text{MC} \\ \hline \text{Workspace} = \{ \ldots \} & \text{AC} \\ \hline \end{array} \]

\[ \begin{array}{c|c} \text{O1: TP DP_{\text{case:ext}^c} T_{\text{case:ext}^c} [x^*] [VP T' \langle \text{DP}_{\text{case}} \rangle ] \langle \text{DP}_{\text{case:inf}} \rangle \langle \text{DP}_{\text{case}} \rangle \langle \text{v} \ldots t \ldots \rangle ]] & \star \end{array} \]

\[ \begin{array}{c|c} \text{O2: TP T_{\text{case:ext}^c} [x^*] [VP T' \langle \text{DP}_{\text{case}} \rangle ] \langle \text{DP}_{\text{case:inf}} \rangle \langle \text{DP}_{\text{case}} \rangle \langle \text{v} \ldots t \ldots \rangle ]] & \star \\ \hline \end{array} \]

T12: Absolutive movement, step 2: Agree (with Spec T)

\[ \text{Input: } \begin{array}{|c|c|c|} \hline \text{TP DP_{\text{case:ext}^c} [T' T_{\text{case:ext}^c} [x^*] [VP T' \langle \text{DP}_{\text{case}} \rangle ] \langle \text{DP}_{\text{case:inf}} \rangle \langle \text{DP}_{\text{case}} \rangle \langle \text{v} \ldots t \ldots \rangle ]] & \text{SHB} & \text{MC} \\ \hline \text{Workspace} = \{ \ldots \} & \text{AC} \\ \hline \end{array} \]

\[ \begin{array}{c|c} \text{O1: TP DP_{\text{case:ext}^c} T_{\text{case:ext}^c} [T' T_{\text{case:ext}^c} [x^*] [VP T' \langle \text{DP}_{\text{case}} \rangle ] \langle \text{DP}_{\text{case:inf}} \rangle \langle \text{DP}_{\text{case}} \rangle \langle \text{v} \ldots t \ldots \rangle ]] & \star \end{array} \]

\[ \begin{array}{c|c} \text{O2: TP DP_{\text{case:ext}^c} [T' T_{\text{case:ext}^c} [x^*] [VP T' \langle \text{DP}_{\text{case}} \rangle ] \langle \text{DP}_{\text{case:inf}} \rangle \langle \text{DP}_{\text{case}} \rangle \langle \text{v} \ldots t \ldots \rangle ]] & \star \\ \hline \end{array} \]

(24) Legitimate movement of DP_{case}

a. Structure after T is merged
c. Finally, movement of DP_{acc} takes place to Spec{T

\[ \text{T}_{13}: \text{Accusative Movement, step 1: Agree} \]

\[
\text{Input:}\ [T \ T_{[\text{case:ext}], \text{xy}}] \ [\text{DP}_{\text{case}}] [v \ v \ v \ t \ \ldots] \]
\]

<table>
<thead>
<tr>
<th>SHB</th>
<th>AC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*O₂</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{T}_{14}: \text{Accusative Movement, step 2: Move} \]

\[
\text{Input:}\ [T \ T_{[\text{case:ext}], \text{xy}}] \ [\text{DP}_{\text{case:ext}}] [v \ v \ v \ t \ \ldots] \]
\]

<table>
<thead>
<tr>
<th>SHB</th>
<th>AC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\text{Note:}\nNothing so far rules out O₂ in T₁₄. However, because of the PIC, only DP_{ext} can move on. Eventually, this leads to unchecked operator features on the attracting head and DP_{int}, and thus to a crash of the derivation.

\text{iv) \ DP_{nom} Movement} \nSimilarly to the DP_{abs} case, there is no problem for movement of DP_{nom} because DP_{acc} has already been assigned case when DP_{nom} moves.

\[
\text{T}_{15}: \text{Nominitative movement, step 1: Agree} \]

\[
\text{Input:}\ [T \ T_{[\text{case:ext}], \text{xy}}] \ [\text{DP}_{\text{case}}] [v \ v \ v \ t \ \ldots] \]
\]

<table>
<thead>
<tr>
<th>SHB</th>
<th>AC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*O₂</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{b. AC} \gg \text{MC triggers valuation of DP_{nom} next} \]

\[
\text{c. Finally, movement of DP_{nom} takes place to Spec{T} }\]
T16: Nominal movement, step 2: Move

Input: \[ \text{TP} \left[ \text{T[case:ext] \[ \text{DP[case:ext]} \left[ \text{v} \ldots \text{DP[case:inv]} \ldots \right] \right] \right] \]

Workspace - \{ ... \}

\[ \text{O} : \left[ \text{TP} \left[ \text{DP[case:ext]} \left[ \text{v} \left[ \text{TP} \left[ \text{T[X]} \left[ \text{v} \left[ \text{TP} \left[ \text{T[X]} \left[ \text{v} \left[ ... \right] \right] \right] \right] \right] \right] \right] \right] \right] \right] \]

5.4 Extremely Local vs. Less Local Optimization

Note:
Again, if phrases (phrases, clauses, sentences) are optimization domains, a wrong prediction is made for accusative contexts: Maraudage would be expected to arise. As before, non-occurrence of Agree with the moved accusative DP, and thus maraudage of T's case features, in the presence of DP\textsubscript{inv} in Spec\textsubscript{TP} illustrates counter-bleeding opacity.

T17: TP optimization under AC \( \gg \) MC (“accusative”) ranking: wrong result

6. Accelerating Merge: A Constraint on Dative Movement in Accusative Systems

6.1 Data


A DP\textsubscript{dat} object cannot be extracted from an ECM complement in German.

(26) Scrambling and pronoun movement of a DP\textsubscript{dat} object from ECM complements

a. *dass keiner \[\text{DP dieser Frau} \] \[\text{XP den Jungen t1 helfen sah/ließ} \]
that no-one\textsubscript{nom} this woman\textsubscript{dat} the boy\textsubscript{acc} help saw/let
b. *dass er \[\text{DP ihm} \] \[\text{XP den Jungen t1 helfen sah/ließ} \]
that he\textsubscript{nom} him\textsubscript{dat} the boy\textsubscript{acc} help saw/let

c. *weil mir; niemand \[\text{XP Karl t1 helfen ließ} \]
because me\textsubscript{dat} no-one\textsubscript{nom} Karl\textsubscript{acc} help let

(27) Wh-movement and topicalization of a DP\textsubscript{dat} object from ECM complements

a. *Wem1 \[sah/ließ Karl \] \[\text{XP den Jungen t1 helfen} \]?
whom\textsubscript{nom} saw/let Karl\textsubscript{nom} the boy\textsubscript{acc} help
b. *Dem Lehrer1 \[sah/ließ Karl \] \[\text{XP den Jungen t1 helfen} \]
the teacher\textsubscript{dat} saw/let Karl\textsubscript{nom} the boy\textsubscript{acc} help

(28) Movement of DP\textsubscript{dat} from ECM complements with double object constructions:

a. *Wem1 \[ließ/sah Karl \] \[\text{XP den Jungen t1 das Buch geben} \]?
whom\textsubscript{dat} let/saw Karl the boy\textsubscript{acc} the book\textsubscript{acc} give
b. *dass keiner dieser Frau1 \[\text{XP den Jungen t1 das Buch geben} \]
that no-one\textsubscript{nom} this woman\textsubscript{dat} the boy\textsubscript{acc} the book\textsubscript{acc} give

(29) Legitimate movement of DP\textsubscript{dat} in other contexts:

a. *Wem1 meint sie \[\text{CP dass wir t3 das Buch geben sollten} \]?
whom\textsubscript{dat} thinks that we the book give should
b. Diesem Plan1 habe ich abgelehnt \[\text{CP PRO t1 meine Unterstützung zu} \]
this plan\textsubscript{dat} have I rejected my support to
give

c. dass ihm1 der Fritzi \[\text{XP t1 das Buch zu geben} \]
versuchte that him\textsubscript{dat} the Fritzi the book to give tried

Hypothesis:
This restriction has the same source as the ban on ergative movement in ergative systems: The dative arguments move too early, and thus maraud the matrix v's [+case:inv] feature, thereby precluding accusative case assignment to the ECM subject.

Complication:
German is an accusative language that has a ranking AC \( \gg \) MC, which would normally order case assignment of v to the embedded DP\textsubscript{dat} before an intermediate movement step of the dative DP to matrix Specv.

Proposal:
Movement of the dative DP is accelerated by a higher-ranked constraint in this particular context, one that regulates proper and improper movement.

(30) Long-Distance Scrambling in German:

a. dass das Buch1 keiner t1 ließ that the book\textsubscript{acc} no-one\textsubscript{nom} reads
b. *dass Karl das Buch1 glaubt \[\text{CP dass keiner t1 ließ} \]
that Karl\textsubscript{nom} the book\textsubscript{acc} thinks that no-one\textsubscript{nom} reads

(31) Raising vs. Super-Raising in English:

a. Mary1 seems \[\text{TP t1 to like John} \]
b. *Mary1 seems \[\text{CP t1 that t1 likes John} \]
A dilemma for improper movement, given that every XP is a phase:


6.2 Assumptions

6.2.1 Improper Movement

Ref.: Müller (2012)

Basic assumptions about improper movement:
(i) Edge features are defective copies of categorial features of phase heads.
(ii) Edge features successively value movement-related features of moved items, creating lists that record aspects of the derivational history of movement.
(iii) Such information is deleted when information of the same type is encountered.
(iv) When a critical landing site is reached, the functional sequence of categories (f-seq: C-T-v-V) must be respected on such lists; see Williams (1974; 2003). This constraint will be called the WILLIAMS CYCLE.

(33) Legitimate long-distance wh-movement:
What do you think [CP C [TP she 1 T [v₅ ᵐ p t₁ ᵐ v] [v₅ ᵐ p said t₂ ]]][CP what [sheetCP [TP she 1 T [v₅ ᵐ p you think she said ]]] ( pumped-f-seq)

(34) Illegitimate long-distance scrambling:
*dass Karl das Buch [v₅ ᵐ p dass keiner [tᵢ] liest ]
that Karlnom the booknom thinks that no-oneNom reads
[CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T )[CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ][CP [C dass ] [TP Karl ] [v₅ ᵐ p das Buch [CVP [v₅ ᵐ p dass keiner liest ]]] T ] ㎏ mishSeq → *WILLIAMS CYCLE → crash

6.2.2 Exceptional Case Marking in German

Observation:
German ECM complements lack typical properties associated with TP (or CP) (Ste- move & Sternewald (1988), Fasenlow (1991), Wurmbrand (2001)). There is no separate temporal specification, there is obligatory wide scope for negation, and a systematic absence of zu (‘to’): XP in the above examples is vP.

(35) German ECM complements are vPs:

a. *Wir sehen [v₅ ᵐ p ihn ᵐ den Diener ᵐ erschossen haben ]
we see *himAcc the servantAcc shot have

b. Wir lassen [v₅ ᵐ p den Diener den Mann nicht schlagen ]
we let the servantAcc the manAcc not hit

c. *Wir hören [v₅ ᵐ p ihn zu schnarchen ]
we hear *himAcc to snore

Note:
- With extraction from ECM complements, matrix v has a dual role: It assigns case to the ECM subject ([*case:mit*]), and it has an edge feature ([*edge*]) to effect the intermediate movement step.
- On the vP cycle, a moved dative DP originating in the embedded ECM complement (and having been assigned lexical case there by V) has a chance to immediately remedy temporary f-seq violations on the feature list of its movement-related feature, and thereby satisfy the WILLIAMS CYCLE quickly. Assuming that this constraint outranks AC in German, movement of the dative DP to Specv will have to precede case assignment by v to the embedded DPext in German.
- This gives rise to maraudage, and the derivation will ultimately crash because the embedded DPext’s case feature remains unvalued.
- The specific version of the WILLIAMS CYCLE (WC) that is required is given in (36). It is formulated such that a temporary violation can be initiated without violating WC, which must be possible given that WC is ranked high (above AC). (Also see Alderete (2001) on anti-faithfulness constraints in phonology; and Baković & Wilson (2000) on targeted constraints.)

(36) WILLIAMS CYCLE (WC):
If categorial information on a list of a movement-related feature does not conform to f-seq (C-T-v-V) in the input, it must conform to f-seq in the output.

23

24
6.3 Analysis

(37) **Ilegitimate movement of \( DP_{dat} \)** from ECM complements

a. Structure after matrix \( v \) is merged; \( DP_{dat} \) almost satisfies f-seq

b. \( WC \gg AC \gg MC \) triggers movement of \( DP_{dat} \) to Spec\( V \)

b. SHB triggers maraudage of \( v \)

\[
\begin{array}{c}
\text{Input: } DP_{ext} \oplus [v' [VP [DP_{CASE:dat}] [VP \ldots DP_{CASE:}] \ldots V] [v_{\text{int}} [X_{\bullet}][D_{\bullet}]]] \\
\text{SHB} & \text{WC} & \text{AC} & \text{MC} \\
\text{O1: } [v' [VP [DP_{CASE:dat}] [VP \ldots DP_{CASE:}] \ldots V] [v_{\text{int}} [X_{\bullet}][D_{\bullet}]]] & * & * \\
\text{O2: } [v' [VP [DP_{CASE:dat}] [VP \ldots DP_{CASE:}] \ldots V] [v_{\text{int}} [X_{\bullet}][D_{\bullet}]]] & * & * \\
\text{O3: } [v' [VP [DP_{CASE:dat}] [VP \ldots DP_{CASE:}] \ldots V] [v_{\text{int}} [X_{\bullet}][D_{\bullet}]]] & * & * \\
\end{array}
\]

\[
\begin{array}{c}
\text{Note:} \\
\text{The derivation continues by merging the matrix } DP_{ext}; \text{ however, it will ultimately crash because the case feature of the embedded } DP_{ext} \text{ can never be valued.}
\end{array}
\]

6.4 Extremely Local vs Less Local Optimization

\[
\begin{array}{c}
\text{Note:} \\
\text{This time, assuming larger optimization domains like the phrase, does not make a}
\end{array}
\]
wrong prediction: At the vP phrase level, WC, MC and AC are all satisfied, and SHB will continue to pick a maraudage output.

6.5 Consequences

(i) Predictions:

- If there is no embedded DP_{ext} in what is otherwise the same construction, movement of the dative DP should be fine (there can be no maraudage). This is borne out; see the so-called lassen-passive construction in (38).

- If the dative DP and the embedded DP_{ext} both undergo movement in ECM construction, the result should also be fine. As a tendency, this may also seem to be right; see (39).

38. \textit{DP_{dat} movement where an embedded DP_{ext} is not present}

a. dass keiner [\textit{DP} dieser Frau], gestern/gerne [\textit{vP} t1 helfen liess] that no-one\_nom this woman\_dat yesterday/glady help let

d. dass er [\textit{DP} ihm], gestern/germn [\textit{vP} t1 helfen liess] that he\_nom him\_dat yesterday reluctantly help let

e. Wem t1 liess Karl gestern/germ [\textit{vP} t1 helfen liess] ? whom\_dat let Karl\_nom yesterday reluctantly help

39. \textit{DP_{dat} movement where the embedded DP_{ext} also moves}

Diesen Jungen liess der Frau keiner [\textit{vP} t1 t2 helfen] this boy\_acc let the woman\_dat no-one\_nom help

(ii) Why can dative DPs undergo extraction from ECM constructions with an unaccusative embedded predicate?

The problem here is that the embedded DP_{dat} gets case from the matrix v and does not block dative movement; see (40) (from Fanselow (1990)). This problem is solved if there is no vP with unaccusative predicates, pace Legate (2003). Under this assumption, WC does not force early DP_{dat} movement because there is no improper f-seq when DP_{dat} enters the matrix vP domain; hence, there is no maraudage.

40. \textit{DP_{dat} movement is fine in unaccusative contexts}:

a. dass mir1 niemand [\textit{XP} Karl t1 helfen] lieb/saw that medat no-one\_nom Karl\_dat help let/saw

b. dass mir1 niemand [\textit{XP} t1 ein Unglueck zu stoessen] lieb/sah that medat no-one\_nom an accident happen to let/saw

(iii) What about extraction of other kinds of DPs from ECM contexts?

DP_{acc} can undergo such movement easily (see (41), (42)); DP_{gen} movement is arguably much more restricted. This would follow without further ado if maraudage is blocked if exactly the same case is involved; and genitive and accusative are sufficiently different.

41. \textit{Scrambling of a DP_{acc} object from ECM complements}

a. dass der Kollege [\textit{DP} den Antrag], [\textit{XP} seine Mitarbeiter t1 that the colleague\_nom the proposal\_acc his co-workers\_acc gerade schreiben lasst] currently write lets

b. dass [\textit{DP} den Antrag], der Kollege [\textit{XP} seine Mitarbeiter t1 that the proposal\_acc the colleague\_nom his co-workers\_acc gerade schreiben lasst] currently write lets

42. \textit{Pronoun movement of a DP_{acc} object from ECM complements}

a. dass er es1 [\textit{XP} den Jungen t1 lesen sah] that he\_nom he\_acc the boy\_acc read saw

b. dass er es1 [\textit{XP} den Jungen t1 machen liess] that he\_nom he\_acc the boy\_acc make let

43. \textit{Movement of a DP_{gen} object from ECM complements}

a. Karl sieht/lasst den Jungen der Toten gedenken Karl\_nom sees/lets the dead\_gen commemorate

b. ?*dass der/der Toten keiner den Jungen gedenken that they\_gen the dead\_gen no-one\_nom the boy\_acc commemorate sieht/lasst sees/lets

c. ?*Der Toten, sieht/lasst Karl den Jungen gedenken the dead\_gen sees/lets Karl\_nom the boy\_acc commemorate

7. Decelerating Merge: Mobility of Lexical/Oblique Arguments in Ergative Systems

Starting point:

Not all morphologically ergative languages ban extraction of the ergative. In some, the ergative extracts freely and without any special morphology (such as the agent focus morphology encountered in many Mayan languages).
a. Maxki1/2 tyi y-il-ā (t1) aj-Maria (t2)?
   who ASP A3-see-DTV DET-Maria
   ‘Who saw Maria?’ / ‘Who did Maria see?’
b. Maxki tyi y-il-ā t1 a-walak?
   who PRFV A 3-see-DTV A 2-cow
   ‘Who saw your cow?’

45. A-bar movement of DP_{erg} in Basque (Hualde & Ortiz de Urbina (2003))
a. Joneki omen daki t1 hori.
   JON.ERG apparently knows that
   ‘Apparently. JON knows that.’
b. Norki lagunduko die t1 sure lagunei?
   who.ERG help.FUT AUX your friends.DAT
   ‘Who will help your friends?’
c. [ Ō.ERG1 t1 Bonaparteren lanari ekin zioten ] langile
   Bonaparte.ERG work.DAT tackle AUX worker
   apalak
   humble.DET.PL
   ‘the humble workers who took on the work of Bonaparte’

46. Relativization of DP_{erg} in Avar (Nakh-Dagestanian, Polinsky et al. (2011))
[ Ō.ERG4 t1 ʻologana-y yas repetici-yal-de y-ač-ǐn
   unmarried-11 girl.ABS-rehearsal-OBL-LOC II-bring-GER
   y-ač-ăr-a-y ] artiskal1 bercina-y y-igo.
   II-come-PRTC-11 actress.ABS beautiful-11 II-AUX
   ‘The actress that brought the young girl to the rehearsal is pretty.’

47. Relativization of DP_{erg} in Pitjantjatjara (Pama-Nyungan. Bowe (1990. 101)):
Wati panyā | Ō.ERG4 t1 wany atu-nįtja-hu | ngayu-nyu u-ngu.
   man ANAPH wood chop-INF-ERG 1.sg-ACC give-PAST
   ‘The man who chops wood gave me some.’

Possible explanation:
Extraction of the ergative subject is an option in these languages because Agree applies before Merge on the TP-cycle despite the ‘ergative ranking’ MC ∋ AC.

Basic assumption:

a. [CASE:ānt] has two subfeatures [-OBL], [+GOV]; [CASE:+OBL,+GOV]
b. [CASEːext] has two subfeatures [−OBL], [−GOV]; [CASE:+OBL,−GOV]
c. [−GOV] maintains the external/internal distinction; [−OBL] indicates that the cases associated with T and v are structural (non-oblique).
d. DP arguments bear [OBL],[GOV] before valuation.

e. Structural ergative case: Both subfeatures are on v.
f. Lexical ergative case: [+CASE:+OBL] is on v; [+CASE:+GOV] is on V.

49. Structural vs lexical case assignment:
a. [+CASE:+OBL] is discharged by Agree under m-command (as before).
b. [+CASE:+GOV] is discharged under c-command.

Activity:
Arguments with partially valued case are inactive (cf. Richards (2008); also Chomsky’s (2001) ‘Activity Condition’).

50. Activity Condition (ActC):
Inactive elements cannot undergo internal Merge

51. Ranking:
ACTC ∋ MC ∋ AC

Conclusion:
Assignment of lexical ergative is only possible if v+V moves to T, either because this generally happens in the language, or as a repair strategy. Further proviso: Successive head movement maintains c-command (cf., e.g., Roberts (2010)).

52. Legitimate movement of DP_{erg} if the ergative is lexical
a. Structure after T is merged; partial case assignment to DP_{ext}
b. ActC >> MC >> AC blocks movement of DP_{ext}; permits movement of v-V

Note: The relevant competitions are given in tableaux T_{20}, T_{21}, and T_{22}. Note that there are four locally optimal continuations O_2–O_5 (that all carry out an Agree operation) in T_{20} (which illustrates the crucial step from (52-b) to (52-c)). In addition to O_1, which executes movement of DP_{ext} and thereby fatally violates ActC. However, of these four optimal outputs only O_2 (where T undergoes Agree with DP_{int}) will eventually lead to a well-formed output: In O_3 and O_5, DP_{ext} gets its case valued (by T and V, respectively), which means that it becomes active and will have to move in the next step, thereby marauding case features required for DP_{int}. Similarly, O_4 will invariably lead to a crash because DP_{int} undergoes Agree with V here, and will therefore never acquire a fully specified case feature (alternatively, DP_{ext} will fail to do so if DP_{int} marauds T's features as well).

T_{20}: Lexical ergative movement, step 1: Agree(T, DP_{int})

<table>
<thead>
<tr>
<th>Input: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</th>
<th>ActC</th>
<th>MC</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_1: [TP DP_{\text{c:oobl,,}} (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*!</td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>$^\ast$ O_2: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>$^\ast$ O_3: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>$^\ast$ O_4: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>$^\ast$ O_5: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Assuming a continuation with O_2, tableau T_{21} shows that the situation is still such that DP_{ext} cannot move without fatally violating ActC.

T_{21}: Lexical ergative movement, step 2: Agree(V, DP_{ext})

<table>
<thead>
<tr>
<th>Input: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</th>
<th>ActC</th>
<th>MC</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_1: [TP DP_{\text{c:oobl,,}} (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*!</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>$^\ast$ O_2: [TP (v-V_{\text{c:ogv}})–(T_{\text{c:oobl,-gov}})–(x)]</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Finally, tableau T_{22} illustrates the trivial final competition on the TP cycle: DP_{ext} is now active, and movement can finally be carried out.
\(T_{22}: \text{Lexical ergative movement. step 3: Move}(T, \text{DP}_\text{ext})\)

<table>
<thead>
<tr>
<th>Input: ([\text{TP} \cdot V\cdot T_{\text{ext}}] \vdash \left[ \text{DP}<em>\text{c} : \text{obl, gov} \ldots \text{DP}</em>\text{c} : \text{obl, gov} \right] )</th>
<th>ActC</th>
<th>MC</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{O}<em>1: \left[ \text{TP} \cdot \text{DP}</em>\text{c} : \text{obl, gov} \right] \cdot V\cdot T )</td>
<td>ActC</td>
<td>MC</td>
<td>AC</td>
</tr>
</tbody>
</table>

7.1 Extremely Local vs. Less Local Optimization

\(T_{23}: \text{TP optimization under ActC} \gg \text{MC} \gg \text{AC ranking: wrong result}\)

<table>
<thead>
<tr>
<th>Input: ([\text{TP} \cdot \text{DP}<em>\text{c} : \text{obl, gov} \cdot \text{sch}} \oplus \left[ \text{TP} \cdot \text{DP}</em>\text{c} : \text{obl, gov} \ldots \text{DP}_\text{c} : \text{obl, gov} \right] )</th>
<th>ActC</th>
<th>SHB</th>
<th>MC</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{O}<em>1: \left[ \text{TP} \cdot \text{DP}</em>\text{c} : \text{obl, gov} \right] \cdot V\cdot T )</td>
<td>ActC</td>
<td>*!</td>
<td>MC</td>
<td>AC</td>
</tr>
<tr>
<td>(\text{O}<em>2: \left[ \text{TP} \cdot \text{DP}</em>\text{c} : \text{obl, gov} \right] \cdot V\cdot T )</td>
<td>ActC</td>
<td>MC</td>
<td>AC</td>
<td></td>
</tr>
</tbody>
</table>

8. Ranked Constraints vs. Parametrized Preference Principles

**Question:**
How could acceleration and deceleration effects be addressed in a non-optimality-theoretic approach, such as the one laid out in lecture 2?

**Implausible preference principles:**
- Agree before Move, unless satisfaction of the Williams Cycle demands otherwise.
- Move before Agree unless satisfaction of the Activity Condition demands otherwise.

**Problem here:**
Constraint interaction is ignored as a phenomenon, and is simply built into the formulation of a single constraint. (See the remarks on Barss’s (1984) chain-binding approach to movement and reflexivization in lecture 1.)

**Possible solution:**
1. Only convergent steps are considered (Chomsky (1995)).
2. Violations of WC and ActC lead to non-convergence.
3. However: It is far from obvious that a simple notion of convergence can be devised that covers all relevant contexts in a natural way; see Collins (1994), Sternefeld (1996).

9. Conclusion

**Four claims:**
- (i) \textit{Competition of Merge and Agree}:
  1. vP cycle: \(v\cdot \text{case in } T \cdot \left[ \bullet \cdot D \right] \)
  2. TP cycle: \(T\cdot \text{case in } T \cdot \left[ \bullet \cdot X \right] \)
- (ii) \textit{Ranking vs. preference principle} (see Chomsky (1995; 2001) on Merge before Move):
  1. MC \gg AC on the vP cycle: ergative encoding system
  2. AC \gg MC on the vP cycle: accusative encoding system
  3. MC \gg AC on the TP cycle: *ergative movement (too early), \(\sqrt{}\)absolutive movement
  4. AC \gg MC on the TP cycle: \(\sqrt{}\)accusative movement, \(\sqrt{}\)nominative movement
- (iii) \textit{Acceleration and deceleration of movement: expected in OT, less so in Minimalism}:
  1. WC \gg AC \gg MC on the vP cycle: *ative movement in German ECM (too early)
  2. ActC \gg MC \gg AC on the TP cycle: \(\sqrt{}\)ergative movement (late)
- (iv) \textit{Extremely local vs. less local optimization} (opacity): If the domain is larger than the derivational step, then
  1. AC \gg MC on the vP cycle does not derive accusative encoding systems;
  2. AC \gg MC on the TP cycle wrongly blocks accusative movement;
  3. ActC \gg MC \gg AC on the TP cycle cannot circumvent the ban on ergative movement.

**Take-home message:**

Good things come to those who wait.

**Bibliography**


Goldrick, Matt (2000): Turbid Output Representations and the Unity of Opacity. In:


Institute. Amsterdam.
Oostendorp, Marc van (2007): Derived Environment Effects and Consistency of Exper-
P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis & D. Pesetsky, eds., Is the Best
Polinsky, María, Carlos Gallo, Peter Graff & Ekaterina Kravtchenko (2011): Subject
Pullum, Geoffrey (1979): Rule Interaction and the Organization of a Grammar. Gar-
Roberts, Ian (2010): Agreement and Head Movement. Clitics. Incorporation and De-
Salzmann, Martin (2012): On Three Types of Variation in Resumption: Evidence
in Favor of Violable and Ranked Constraints. In: H. Broekhuis & R. Vogel, eds.,
Sheehan, Michelle (2013): Towards a Parameter Hierarchy for Alignment. Ms., Uni-
Smolensky, Paul & Geraldine Legendre (2006): The Harmonic Mind. MIT Press. Cam-
Westdeutscher Verlag. Opladen.
tner & M. Bierwisch. eds., The Role of Economy Principles in Linguistic Theory.
Stiebels, Barbara (2006): Agent Focus in Mayan Languages. Natural Language and
Trommer, Jochen (2011): Phonological Aspects of Western Nilotic Mutation Morphol-
ogy. Habilitationsschrift, Universität Leipzig.

Wiese, Bernd (1999): Unterspezifizierte Paradigmen. Form und Funktion in der
pronominalen Deklation, Linguistik Online 4. (www.linguistik-online.de/3_99).
G. Legendre. J. Grimshaw & S. Vikner, eds., Optimality-Theoretic Syntax. MIT
Woolford, Ellen (1997): Four-Way Case Systems: Ergative, Nominative, Objective and
de Gruyter. Berlin.