Extremely Local Optimization

Fabian Heck & Gereon Müller
Universität Leipzig

1. Introduction

This article pursues the idea that properties of the Minimalist Program (MP, Chomsky 1995, Chomsky 2001) and Optimality Theory (OT, Prince & Smolensky 2004, McCarthy & Prince 2004) can be fruitfully combined. We concentrate on two properties. First, MP's tenet that syntactic structure is built up by repeated application of Merge, Move, and Agree.[1]Second, OT's assumption that the well-formedness of a syntactic object involves comparison with other objects, hence optimization.

Standardly, Merge, Move, etc. are assumed to apply first, thereby creating a set of candidates that is then subject to optimization. Optimization is *global* in the sense that it applies once to complete structures (see Grimshaw 1997, Pesetsky 1998, Legendre, Smolensky & Wilson 1998, among others). Alternatively, structure building operations and optimization apply in a cyclic interleaving fashion: Merge, Move, etc. create output candidates $\alpha_1 \ldots, \alpha_n$, which are subject to optimization. The optimal α_i serves as (part of) the input for the next cycle and so on, until the numeration is empty. Optimization is *local* in the sense that it applies iteratively to small portions of structure (see Ackema & Neeleman 1998, Müller 2000*c*, Heck & Müller 2000, Fanselow & Ćavar 2001, among others). This raises the natural question as to how large optimization domains are. Optimization has been suggested to apply to clauses (Ackema & Neeleman 1998), phases (Fanselow & Ćavar 2001, Müller 2000*a*), phrases (Müller 2000*c*, Heck & Müller 2000, Fischer 2004, Heck 2004), or after each step (Chomsky 2000, Epstein & Seely 2002). We pursue the consequences of the last, most radical position and refer to it as *extremely local optimization*.

We will provide empirical support for extremely local optimization, based on (a) argument encoding, (b) dative possessors, and (c) SpecC-expletives. The argument's structure is always the same: (i) Sometimes, the relative order of Agree and Merge is under-determined. Provided that they cannot apply simultaneously (see Epstein & Seely 2002; contra Pullum 1976, Chomsky 2005), this creates a conflict: Both are required to be executed immediately, but only one of them can. (ii) The conflict can be resolved by ranking the requirements: The higher ranked requirement is satisfied immediately; the lower ranked must remain unsatisfied, momentarily. As unsatisfiability does not lead to a crash of the derivation, this suggests an analysis in terms of violable constraints. (iii) Ceteris paribus, larger optimization domains cannot preserve the order of operations induced by extremely local optimization. Empirically, this yields the wrong result.

2. Constraints, Features, and Operations

We assume that the derivation is driven by two types of features: (i) Structure-building features, which trigger Merge[2]. And (ii) probe features, which trigger Agree. We write them as [•F•] and [$*$F$*$], respectively. Next, we assume that Move and Merge are dependent on the presence of [•F•]. This is ensured by the definitions in (1) and (2).

(1) *Merge*: α can be merged with β, yielding $\{\alpha,\{\alpha,\beta\}\}$, if α bears a structure-building feature [•F•] and F is the label of β.

(2) *Move*: Move is Merge, with β internal to α.

Moreover, we adopt Chomsky's (2001) operation Agree in (3).[3]

(3) *Agree*: α can agree with β with respect to a feature bundle Γ iff (a) and (b) hold:
   a. α bears a probe feature [$*$F$*$] in Γ and may thereby provide the α-value for a matching goal feature [F] of β in Γ.
   b. α m-commands β.

We also assume that [$*$F$*$] and [•F•] features are targeted once during the derivation by Merge or Agree respectively, see the constraints in (4), (5).

(4) AGREE CONDITION (AC):
Probes ([$*$F$*$]) participate in Agree.

(5) MERGE CONDITION (MC):
Structure-building features ([•F•]) participate in Merge.

Finally, we presuppose that the derivation unfolds in a cyclic manner and that movement is feature-driven, see the constraints (6) and (7).[4]

(6) STRICT CYCLE CONDITION (SCC, Chomsky (1973, 1993):
Merge of α and β is possible only if β has no active features. (A feature is active if it is a [•F•] or [$*$F$*$] feature that has not yet participated in Merge or Agree).

(7) LAST RESORT (LR, Chomsky 1995):
Move of α and β follows Agree of α and β.

3. Empirical Evidence

3.1. Argument encoding

3.1.1. Data

To simplify matters, there are two basic patterns of argument encoding. In accusative languages, the internal argument (DP$_{int}$) of a transitive verb (V$_t$) bears accusative case. In contrast, DP$_{int}$ of an intransitive verb (V$_i$) and external arguments (DP$_{ext}$) in general bear nominative case (see (8-a)). In ergative languages, DP$_{ext}$ of a V$_t$ bears ergative, whereas its internal co-argument and the only argument of a V$_{int}$ bear absolutive case (see (8-b)).

(8) *Basic patterns of argument encoding*:

a. Accusative marking		b. Ergative marking	
DP$_{ext}$-V$_i$	DP$_{int}$-V$_i$	DP$_{ext}$-V$_i$	DP$_{int}$-V$_i$
DP$_{ext}$-V$_t$	DP$_{int}$-V$_t$	DP$_{ext}$-V$_t$	DP$_{int}$-V$_t$
nom	acc	erg	abs

For lack of space, we dispense with giving examples from actual languages and rather leave it at presenting the abstract patterns in (8).

3.1.2. Analysis

Our analysis requires the following assumptions. (i) There is one structural argument encoding feature: [case]. (ii) [case] can have two values: ext(ernal) and int(ernal). (iii) The valued feature [case:ext] expresses nominative/absolutive case, the valued feature [case:int] expresses accusative/ergative case (see Murasugi 1992). (iv) [case] features figure in Agree relations involving T/v on the one hand and DP on the other hand, as in (9).

(9) *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.

Observe that v has a dual role: It participates in a Merge operation with DP$_{ext}$; but it also participates in an Agree relation. This dual role has far-reaching consequences for the nature of argument encoding.

To see why, consider a simple transitive context, with two arguments DP$_{int}$, DP$_{ext}$. Suppose that the derivation has reached a stage Σ, 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. Due to the dual role of v, a conflict arises at this point: AC demands that the next operation is Agree(v,DP$_{int}$), MC demands that it is Merge(DP$_{ext}$,v). Note that if constraints are evaluated after each derivational step, then this derives the effects of Pesetsky's (1989) Earliness Principle, see Chomsky (2001, 15). Evaluation at each step, in turn, will follow from the idea that constraint evaluation, hence optimization, is extremely local (see below).

Before we address the issue how the conflict is resolved, note that Agree is assumed to be subject to the MINIMAL LINK CONDITION, see (10).

(10) MINIMAL LINK CONDITION (MLC, Chomsky 1995, 2001):
An Agree operation involving α and β can only take place if there is no δ such that (i) and (ii) hold:

a. δ is closer to α than β.
b. δ bears a feature that has not yet participated in Agree.

(11) *Closeness*: δ is closer to α than β if the path from δ to α is shorter than the path from β to α.

(12) *Path* (Müller 1998, 130; cf. Pesetsky 1982, 289, Collins 1994, 56):
The path from X to Y is the set of categories Z such that (a) and (b) hold:

a. Z is reflexively dominated by the minimal XP that dominates both X and Y.
b. Z dominates X or Y.

The length of a path is determined by its cardinality.

The consequences of the MLC in (10), based on (11) and (12), are the following: (i) The specifier and the complement of a head qualify as equally close to the head. (ii) The specifier of a head is closer to the head than a category that is further embedded in the complement of the head. Applied to the derivational stage Σ above, it follows that once DP_{ext} indeed has undergone Merge with v in Σ, then DP_{ext} counts as closer to v than DP_{int}.

Returning to the main plot, we propose that the conflict is resolved by (language specific) ranking of the conflicting requirements AC and MC. The two possibilities of relative ranking yield the accusative and the ergative pattern of argument encoding. The relevant rankings are given in (13):

(13) a. (MLC $\gg$) AC $\gg$ MC (Accusative pattern)
b. (MLC $\gg$) MC $\gg$ AC (Ergative pattern)

Let us trace the relevant stages of the derivation, starting with the accusative pattern. We enter the derivation at stage Σ. Since AC $\gg$ MC, Agree takes priority over Merge. Agree targets the probe [$*$case:int$*$] of v and thus instatiates [case:int] on DP_{int}, see O_2 in tableau T_1.[5] O_2 from T_1 serves as the input for the next cycle. AC being satisfied, MC can now be fulfilled by applying Merge to DP_{ext}. This targets the [•D•]-feature on v, resulting in the optimal output O_1 in T_2. Again, O_1 will serve as input for the next derivational cycle.

T_1: Accusative pattern, step 1 (Σ as input): Agree

Input: $[_{v'}\ v_{[*case:int*],[\bullet D \bullet]}\ \dots\ DP_{[case:\square]}\ \dots\ ]$ Workspace = $\{DP_{[case:\square]},\ T_{[*case:ext*]},\ \dots\}$	MLC	AC	MC
O_1: $[_{v'}\ DP_{[case:\square]}\ [_{v'}\ v_{[*case:int*]}\ \dots\ DP_{[case:\square]}\ \dots\ ]]$		*!	
☞O_2: $[_{v'}\ v_{[\bullet D \bullet]}\ \dots\ DP_{[case:int]}\ \dots\ ]$			*

T_2: Accusative pattern, step 2: Merge

Input: $[_{v'}\ v_{[\bullet D \bullet]}\ \dots\ DP_{[case:int]}\ \dots\ ]$ Workspace = $\{DP_{[case:\square]},\ T_{[*case:ext*]},\ \dots\}$	MLC	AC	MC
☞O_1: $[_{vP}\ DP_{[case:\square]}\ [_{v'}\ v\ \dots\ DP_{[case:int]}\ \dots\ ]]$			

We skip step 3, which merges the T-head (bearing [$*$case:ext$*$]) with O_1 of T_2. Once T is merged, AC demands T's [$*$case:ext$*$] to establish Agree, which instantiates [case:ext] on DP$_{ext}$. The result is the accusative pattern.

T_3: Accusative pattern, step 4: Agree

Input: $[_{T'}\ T_{[*case:ext*]}\ [_{vP}\ DP_{[case:\square]}\ \dots\ DP_{[case:int]}\ \dots\ ]]]$ Workspace = { }	MLC	AC	MC
☞O_1: $[_{TP}\ T\ [_{vP}\ DP_{[case:ext]}\ [_{v'}\ v\ \dots\ DP_{[case:int]}\ \dots\ ]]]$			

Now reenter the derivation at stage Σ with the ranking MC $\gg$ AC. MC immediately forces Merge of DP$_{ext}$, consuming v's [$\bullet$D$\bullet$] feature at the cost of violating AC (v's [$*$case:int$*$] probe is still present), see O_1 in T_4. At the next step, v's [$*$case:int$*$] probe can and must undergo Agree. Crucially, as DP$_{ext}$ is present it counts as closer to v than DP$_{int}$. Therefore, v's probe instantiates [case:int] on DP$_{ext}$ (see O_1 in T_5). Valuation of DP$_{int}$'s goal fatally violates the MLC (see O_2). Once T with its [$*$case:ext$*$] probe has been merged, it instantiates [case:ext] on DP$_{int}$. This derives the ergative pattern. Although we have not depicted these last two competitions, note in passing that Agree(T,DP$_{int}$) is just local enough to be in accordance with the PIC in Chomsky (2001, 14). DP$_{ext}$ does not intervene between T and DP$_{int}$ because DP$_{ext}$'s case feature has already participated in Agree.

T_4: Ergative pattern, step 1 (Σ as input): Merge

Input: $[_{v'}\ v_{[*case:int*],[\bullet D \bullet]}\ \dots\ DP_{[case:\square]}\ \dots\ ]$ Workspace = $\{DP_{[case:\square]},\ T_{[*case:ext*]},\ \dots\}$	MLC	MC	AC
☞O_1: $[_{v'}\ DP_{[case:\square]}\ [_{v'}\ v_{[*case:int*]}\ \dots\ DP_{[case:\square]}\ \dots\ ]]$			*
O_2: $[_{v'}\ v_{[\bullet D \bullet]}\ \dots\ DP_{[case:int]}\ \dots\ ]$		*!	

T_5: Ergative pattern, step 2: Agree (with DP_{ext})

Input: $[_{v'}$ $DP_{[case:\square]}$ $[_{v'}$ $v_{[*case:int*]}$ ... $DP_{[case:\square]}$ ...]] Workspace = {$T_{[*case:ext*]}$, ...}	MLC	MC	AC
☞O_1: $[_{vP}$ $DP_{[case:int]}$ $[_{v'}$ v ... $DP_{[case:\square]}$ ...]]			
O_2: $[_{vP}$ $DP_{[case:\square]}$ $[_{v'}$ v ... $DP_{[case:int]}$ ...]]	*!		

3.1.3. Less local optimization

Suppose that optimization targets complete phrases. Then it will not apply unless v has been Merged with DP_{ext}, forming a complete vP. The ergative pattern is straightforwardly derived under the ranking MC $\gg$ AC, as before. We skip the relevant competition. Of more interest is the attempt to derive the accusative pattern under the ranking AC $\gg$ MC: In a fully-fledged vP DP_{ext} is always closer to [∗case:int∗] on v. But then, given the MLC, [∗case:int∗] cannot be instantiated on DP_{int}, but must be instantiated on DP_{ext}. This yields again the ergative but not the accusative pattern (see O_1 in T_6). Thus less local optimization undergenerates. The argument is independent of MLC's ranking relative to AC or MC.

T_6: vP optimization under AC $\gg$ MC ('accusative') ranking: wrong result

Input: $DP_{[case:\square]}$, $v_{[*case:int*],[\bullet D\bullet],...}$, $[_{VP}$ ... $DP_{[case:\square]}$ ...] Workspace = {$T_{[*case:ext*]}$, ...}	MLC	AC	MC
★O_1: $[_{vP}$ $DP_{[case:int]}$ $[_{v'}$ v ... $DP_{[case:\square]}$ ...]]			
O_2: $[_{vP}$ $DP_{[case:\square]}$ $[_{v'}$ v ... $DP_{[case:int]}$ ...]]	*!		

3.2. Prenominal dative possessors in German

3.2.1. Data

German DPs can contain a dative-marked possessor (DP_{dat}) in SpecD (see Haider 1988, Zifonun 2004). The head of such DPs is realized by a possessive pronoun, which exhibits a twofold agreement pattern. (i) The root of the pronoun agrees with DP_{dat} with respect to [num] and [gen]. (ii) The inflection of the pronoun agrees with its complement NP with respect to [num], [gen], and [case]. We focus here on agreement with respect to [gen] (see (14)), but everything can be transferred to the other features as well.

(14) *Gender agreement with dative possessors in German:*

a. $[_{DP}$ dem Fritz] sein -e Schwester
   the.masc Fritz his.masc -fem sister.fem
   "Fritz's sister"

b. *$[_{DP}$ dem Fritz] ihr -Ø Schwester
   the.masc Fritz her.fem -masc sister.fem

3.2.2. Analysis

Suppose the following. (i) DP$_{dat}$ is merged as a complement of the possessee (de Vries 2005) and undergoes [•EPP•]-driven movement to SpecD. (ii) Functional elements as pronouns are realized by post-syntactic morphology (see, e.g., Halle & Marantz 1993). (iii) The pronoun's inflectional features occupy a structurally higher position than its root ($\sqrt{\ }$) features.[6]

It follows that the pronoun has a dual role: It bears [$*$gen:$\Box *$] probes that trigger Agree and an [•EPP•]-feature that triggers (internal) Merge. This causes a conflict. Suppose the derivation has reached stage Σ, where the pronoun has been merged. Then AC demands Agree(D,DP$_{dat}$) or Agree(D, NP); and MC demands DP$_{dat}$ raising to SpecD. The conflict can be resolved by ranking AC over MC, yielding the correct agreement pattern.

(15) MLC $\gg$ AC $\gg$ MC $\gg$ LR (Ranking for German)

Suppose we want to derive (14-a). We enter the derivation at stage Σ. Due to AC $\gg$ MC, Agree must apply first. Since the pronoun's inflectional probes are structurally higher than its root probes, the former count as closer to both NP and DP$_{dat}$. Thus the MLC constrains Agree to the inflectional probes. Moreover, the NP counts as closer to the pronoun than DP$_{dat}$. Thus Agree(NP,infl) instantiates [gen$_{\mathrm{infl}}$:fem] on the prounoun (see O$_1$ in T$_7$). Having undergone Agree, the NP and the inflection are inactive. Hence, Agree can next affect the pronoun's root probes and DP$_{dat}$. This values [gen$_{\surd}$:masc] on the pronoun (see O$_1$ in T$_8$). Finally, MC can be satisfied by movement of the possessor DP to SpecD (this optimization is skipped).

T$_7$: Evaluation of gender inflection: Agree

Input: $[_{\mathrm{DP}_1}\ \mathrm{D}_{[*\mathrm{case:dat}*],[*\mathrm{gen}_{\surd}:\Box*],[*\mathrm{gen}_{\mathrm{infl}}:\Box*],[\bullet\mathrm{EPP}\bullet]}\ [_{\mathrm{NP}}\ \mathrm{N}_{[\mathrm{gen:fem}]}\ \mathrm{DP}_{2[\mathrm{case}:\Box],[\mathrm{gen:masc}]}\ ]]$	MLC	AC	MC	LR
☞O$_1$: $[_{\mathrm{DP}_1}\ \mathrm{D}_{[*\mathrm{case:dat}*],[*\mathrm{gen}_{\surd}:\Box*],[\mathrm{gen}_{\mathrm{infl}}:\mathrm{fem}],[\bullet\mathrm{EPP}\bullet]}\ [_{\mathrm{NP}}\ \mathrm{N}_{[\mathrm{gen:fem}]}\ \mathrm{DP}_{2[\mathrm{case}:\Box],[\mathrm{gen:masc}]}\ ]]$		**	*	
O$_2$: $[_{\mathrm{DP}_1}\ \mathrm{D}_{[*\mathrm{case:dat}*],[\mathrm{gen}_{\surd}:\mathrm{fem}],[*\mathrm{gen}_{\mathrm{infl}}:\Box*],[\bullet\mathrm{EPP}\bullet]}\ [_{\mathrm{NP}}\ \mathrm{N}_{[\mathrm{gen:fem}]}\ \mathrm{DP}_{2[\mathrm{case}:\Box],[\mathrm{gen:masc}]}\ ]]$	*!	**	*	

T$_8$: Evaluation of root's gender and possessor's case: Agree

Input: $[_{\mathrm{DP}_1}\ \mathrm{D}_{[*\mathrm{case:dat}*],[*\mathrm{gen}_{\surd}:\Box*],[\mathrm{gen}_{\mathrm{infl}}:\mathrm{fem}],[\bullet\mathrm{EPP}\bullet]}\ [_{\mathrm{NP}}\ \mathrm{N}_{[\mathrm{gen:fem}]}\ \mathrm{DP}_{2[\mathrm{case}:\Box],[\mathrm{gen:masc}]}\ ]]$	MLC	AC	MC	LR
☞O$_1$: $[_{\mathrm{DP}_1}\ \mathrm{D}_{[\mathrm{case:dat}],[\mathrm{gen}_{\surd}:\mathrm{masc}],[\mathrm{gen}_{\mathrm{infl}}:\mathrm{fem}],[\bullet\mathrm{EPP}\bullet]}\ [_{\mathrm{NP}}\ \mathrm{N}_{[\mathrm{gen:fem}]}\ \mathrm{DP}_{2[\mathrm{case:dat}],[\mathrm{gen:masc}]}\ ]]$			*	
O$_2$: $[_{\mathrm{DP}_1}\ \mathrm{DP}_{2[\mathrm{case}:\Box],[\mathrm{gen:masc}]}\ \mathrm{D}_{[*\mathrm{case:dat}*],[\mathrm{gen}_{\mathrm{infl}}:\mathrm{fem}],\ldots}\ [_{\mathrm{NP}}\ \mathrm{N}_{[\mathrm{gen:fem}]}\ t_2\ ]]$		*!*		*

3.2.3. Less local optimization

Suppose optimization applied to phrases. An optimal DP will always involve raising of DP_{dat}. But with DP_{dat} raised, both DP_{dat} and NP are equally close to the pronoun. Then the inflectional probe can recieve value [masc], deriving (14-b) (see O_2 in T_9): Thus the approach overgenerates.

T_9: Phrasal optimization: wrong result

Input: $D_{[*case:dat*],[*gen_{\surd}:\square*],[*gen_{infl}:\square*],[\bullet EPP\bullet],...}$ + $[_{NP}\ N_{[gen:fem]}\ DP_{2[case:\square],[gen:masc]}]$	MLC	AC	MC	LR
☞O_1: $[_{DP_1}\ DP_{2[case:dat],[gen:masc]}\ D_{[case:dat],[gen_{\surd}:masc],...}\ [_{NP}\ N_{[gen:fem]}\ t_2\ ]]$				
★O_2: $[_{DP_1}\ DP_{2[case:dat],[gen:masc]}\ D_{[case:dat],[gen_{infl}:masc],...}\ [_{NP}\ N_{[gen:fem]}\ t_2\ ]]$				
O_3: $[_{DP_1}\ D_{[case:dat],[gen_{\surd}:masc][gen_{infl}:fem],[\bullet EPP\bullet]}\ [_{NP}\ N_{[gen:fem]}\ DP_{2[case:dat],[gen:masc]}\ ]]$			*!	
O_4: $[_{DP_1}\ DP_{2[case:dat],[gen:masc]}\ D_{[case:dat],[*gen_{\surd}:\square*],...}\ [_{NP}\ N_{[gen:fem]}\ t_2\ ]]$		*!		

3.3. SpecC-expletives in German

3.3.1. Data

SpecC-expletive insertion in V/2 clauses in German looks like a repair phenomenon: The expletive *es* can only be inserted if no other element fills SpecC (see (16)) and if it is necessary at all to fill this position (see (17)).

(16) *Expletives in V/2 clauses in German* ('Vorfeld-*es*'):

a. Es haben viele Leute geschlafen
EXPL have many $people_{nom}$ slept

b. Viele Leute haben geschlafen
many $people_{nom}$ have slept

(17) *Blocked expletives in verb-final clauses in German*:

a. dass viele Leute geschlafen haben
that many $people_{nom}$ slept have

b. *es dass viele Leute geschlafen haben
EXPL that many $people_{nom}$ slept have

Es-insertion is optional. In Müller's (2000*b*, 48-49) analysis, this is traced back to a tie of the two crucial constraints (ECONOMY and FULL-INTERPRETATION are tied, and ranked below SPECV/2 ('The specifier of V/2 must be filled.')) Next, Bierwisch (1961, 111) observes that expletive insertion is incompatible with nominative pronouns (see also Erdmann 1886, §94): Another element moves to fill SpecC (see (18-a) vs. (18-b)).

(18) *Expletive/subject pronoun incompatibility*:

a. *Es habe ich geraucht
EXPL have$_{1.sg}$ I$_{1.sg.nom}$ smoked

b. Ich habe t geraucht
I$_{1.sg.nom}$ have$_{1.sg}$ smoked

3.3.2. Analysis

To begin with, if expletives never show up in numerations (Hornstein 2001), then their insertion will violate the constraint in (19).

(19) INCLUSIVENESS CONDITION (IC, Chomsky 2001):
Only material from the numeration can be used in a derivation.

Suppose now that AC and MC are actually relativized to phase domains. Thus there exist AC$_v$, AC$_D$, AC$_C$, MC$_v$, MC$_D$, and MC$_C$.[7] We assume here that MC$_c$ $\circ$ AC$_C$ (in contrast to the rankings AC$_v$ $\gg$ MC$_v$ and AC$_D$ $\gg$ MC$_D$ motivated above), where $\circ$ denotes a global constraint tie. The complete relevant ranking is shown in (20).

(20) MLC $\gg$ AC$_C$ $\circ$ MC$_C$ $\gg$ LR $\gg$ IC

Also, we follow Platzack (1987) (Holmberg & Platzack 1995; Chomsky 2005), assuming that C bears [Φ]/[case] relevant for subject agreement and nominative assignment (not T; cf. also Haider 1993).

It follows that a V/2 C has a dual role: It has an [•EPP•] feature that triggers Merge, and [Φ/case] features that trigger Agree. Consider a context with a V/2 C. Suppose that the derivation has reached a stage Σ where C has been merged with a TP containing DP$_{ext}$, with nothing waiting to be merged with C in the workspace. Then AC$_C$ demands application of Agree(C,DP$_{ext}$), and MC$_C$ demands insertion of an expletive pronoun in Spec (Merge(DP$_{expl}$,C)): a conflict.[8] The conflict is resolved by ranking AC$_C$ and MC$_C$, yielding expletive insertion and movement of DP$_{ext}$ in *one* language (because of the tie), in interaction with IC, LR, and MLC.

Suppose we enter the derivation at stage Σ. If AC$_C$ $\gg$ MC$_C$, then the probes of C enter into an Agree relation with DP$_{ext}$, thereby instantiating [Φ] and [case] on C (see O$_2$ in T$_{10}$). Expletive insertion is blocked. The second step involves movement of DP$_{ext}$ to SpecC, satisfying MC$_C$. Note that expletive insertion is blocked at this point by IC (see O$_2$ vs. O$_1$ in T$_{11}$).

If MC$_C$ $\gg$ AC$_C$, Merge applies first. As LR $\gg$ IC, expletive insertion is favored over movement of DP$_{ext}$ (see O$_1$ vs. O$_3$ in T$_{12}$). Once the expletive occupies SpecC, it counts as closer to the probes on C than DP$_{ext}$. Thus, the expletive values [pers:3] on the C-head (see O$_1$ in T$_{13}$). Agree(C,DP$_{ext}$) is blocked by the MLC (see O$_2$). Crucially, C can never check DP$_{ext}$'s [pers] feature. If features that are required for post-syntactic spell-out must be checked in syntax, this derives the incompatibility of expletive *es* and subject pronouns (see (18)): A subject pronoun cannot be spelled out in the context of an expletive because its [pers] feature has not been checked.[9] In the last step C's remaining probes establish Agree with DP_{ext}. The [pers] probe on the expletive is no obstacle, having participated in Agree at the preceding step. For reasons of space this competition is not illustrated here.

T_{10}: Subject movement, step 1 (Σ as input): Agree

Input: $[_{C'}\ C_{[*case:ext*],[*pers:\square*],...,[\bullet EPP\bullet]}$ ... $DP_{[case:\square],[pers:x],...}$ ...]	MLC	AC	MC	LR	IC
O_1: $[_{C'}\ DP_{[pers:3]}\ [_{C'}\ C_{[*case:ext*],[*pers:\square*],...}$ ... $DP_{[case:\square],[pers:x],[num:y],[gen:z]}$ ...]]		*!			*
☞O_2: $[_{C'}\ C_{[pers:x],...,[\bullet EPP\bullet]}$ ... $DP_{[case:ext],[pers:x],...}$ ...]			*		
O_3: $[_{C'}\ DP_{[case:\square],[pers:\square],...}$ $[_{C'}\ C_{[pers:x],...}$ ... t_{DP} ...]		*!		*	

T_{11}: Subject movement, step 2: Move

Input: $[_{C'}\ C_{[pers:x],...,[\bullet EPP\bullet]}$ ... $DP_{[case:ext],[pers:x],...}$ ...]	MLC	AC	MC	LR	IC
O_1: $[_{CP}\ DP_{[pers:3]}\ [_{C'}\ C_{[pers:x],...}$ ... $DP_{[case:ext],[pers:x],...}$ ...]]					*!
☞O_2: $[_{CP}\ DP_{[case:ext],[pers:x],...}$ $[_{C'}\ C_{[pers:x],...}$... t_{DP} ...]					

T_{12}: Expletive insertion, step 1 (Σ as input): Merge

Input: $[_{C'}\ C_{[*case:ext*],[*pers:\square*],...,[\bullet EPP\bullet]}$ ... $DP_{[case:\square],[pers:x],...}$ ...]	MLC	MC	AC	LR	IC
☞O_1: $[_{C'}\ DP_{[pers:3]}\ [_{C'}\ C_{[*case:ext*],[*pers:\square*],...}$ ... $DP_{[case:\square],[pers:x],...}$ ...]]			*		*
O_2: $[_{C'}\ C_{[pers:x],...,[\bullet EPP\bullet]}$ ... $DP_{[case:ext],[pers:x],...}$ ...]		*!			
O_3: $[_{C'}\ DP_{[case:\square],[pers:\square],...}\ [_{C'}\ C_{[pers:x],...}$ ... t_{DP} ...]			*	*!	

3.3.3. Less local optimization

Suppose that optimization affects the phrase. Note that a CP-candidate can perfectly satisfy AC, MC , MLC, and LR. Consequently, IC will be decisive, and will always block expletive insertion, irrespective of the ranking. Expletive insertion can be advantageous only at a certain specific stage in

T_{13}: Expletive insertion, step 2: (Partial) Agree with expletive

Input: O_1: $[_{C'}$ $DP_{[pers:3]}$ $[_{C'}$ $C_{[*case:ext*],[*pers:\square*],...}$ ... $DP_{[case:\square],[pers:x],...}$ ...]]	MLC	MC	AC	LR	IC
☞O_1: $[_{C'}$ $DP_{[pers:3]}$ $[_{C'}$ $C_{[*case:ext*],[pers:3],...}$ ... $DP_{[case:\square],[pers:x],...}$ ...]]			*		
O_2: $[_{CP}$ $DP_{[pers:3]}$ $[_{C'}$ $C_{[pers:x],...}$ ... $DP_{[case:ext],[pers:x],...}$ ...]]	*!				

the derivation (where moving DP_{ext} is blocked by LR); after that stage, the advantage is gone. Thus, less local optimization undergenerates, see T_{14}.

T_{14}: CP optimization under MC $\gg$ AC (expletive) ranking: wrong result

Input: $C_{[*case:ext*],[*pers:\square*],...,[\bullet EPP\bullet]}$ $[_{TP}$ ... $DP_{[case:\square],[pers:x],...}$ ...]	MLC	MC	AC	LR	IC
O_1: $[_{CP}$ $DP_{[pers:3]}$ $[_{C'}$ $C_{[pers:x],...}$ ... $DP_{[case:ext],[pers:x],...}$ ...]]					*!
★O_2: $[_{CP}$ $DP_{[case:ext],[pers:x],[num:y],[gen:z]}$ $[_{C'}$ $C_{[pers:x],...}$ ... t_{DP} ...]					

4. Conclusion

In this article, we have suggested that optimization in syntax is extremely local: The optimization domain is not the phrase, the phase, the clause, or the sentence; it is the syntactic operation. Syntax operates by constantly alternating between single applications of syntactic operations producing derivational steps and choosing the optimal derivational step.

We provided three empirical arguments for extremly local optimization of the following shape. (i) The order of operations in MP is sometimes underdetermined. (ii) Resulting conflicts can be resolved by assuming constraint violability and constraint ranking. (iii) The evidence suggests that optimization is extremely local, affecting the single operation: Less local optimization loses distinctions that extremly local optimization can make.

5. Notes

[1]But cf. Brody (1995), who takes a representational view.
[2]See Sternefeld (2006), Adger (2003).
[3]Unlike Chomsky, we permit an Agree relation between a head and its specifier.
[4]Move in (22) is a binary operation because Move is (binary) Merge with β internal to α.
[5]We introduce the convention that an unvalued feature [F] is written as [F:□].)
[6] For instance, assume the following structure for the pronoun: $[_{DP}$ $[_D$ $\sqrt{}$ D] inflection].

[7]We assume that, alongside vP and CP, DP consitutes a phase in the sense of Chomsky (2001).
[8]We focus on [case] and [pers] here; V/2 does not interact and is consequently ignored.
[9]A determiner, in contrast, can be spelled out because it does not bear a [pers] feature to begin with. Although pronouns are specified for [pers], other DPs are assumed not to be.

6. References

Ackema, Peter & Ad Neeleman. 1998. “Optimal Questions”, *Natural Language and Linguistic Theory* **16**. 443-490.

Adger, David. 2003. *Core Syntax*. Oxford, New York, Oxford University Press.

Bierwisch, Manfred. 1961. “Syntactic Features in Morphology: General Problems of So- Called Pronominal Inflection in German”. In: *To Honour Roman Jakobson*. The Hague/Paris, Mouton. 239-270.

Brody, Michael. 1995. *Lexico-Logical Form: A Radically Minimalist Theory*. Cambridge, Mass., MIT Press.

Chomsky, Noam. 1973. “Conditions on Transformations”. In: *A Festschrift for Morris Halle*. New York, Academic Press. 232-286.

Chomsky, Noam. 1993. “A Minimalist Program for Syntactic Theory”. In: *The View from Building 20*. Cambridge, Mass., MIT Press. 1-52.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, Mass., MIT Press.

Chomsky, Noam. 2000. “Minimalist Inquiries: The Framework”. In: *Step by Step*. Cambridge, Mass., MIT Press. 89-156

Chomsky, Noam. 2001. “Derivation by Phase”, In: *Ken Hale. A Life in Language*. Cambridge, Mass., MIT Press, 1-52.

Chomsky, Noam. 2005. “On Phases”. Ms., MIT, Cambridge, Mass.

Collins, Chris. 1994. “Economy of Derivation and the Generalized Proper Binding Condition”, *Linguistic Inquiry* **25**. 45-61.

de Vries, Mark. 2005. “Possessive Relatives and (Heavy) Pied Piping”, *Journal of Comparative Germanic Linguistics* **9**, 1-52.

Epstein, Sam & Daniel T. Seely. 2002. “Rule Applications as Cycles in a Level-Free Syntax”. In: *Derivation and Explanation in the Minimalist Program.* Oxford, Blackwell. 65-89.

Erdmann, Oskar. 1886. *Grundzüge der deutschen Syntax nach ihrer geschichtlichen Entwicklung*. Stuttgart, Cotta.

Fanselow, Gisbert & Damir Ćavar. 2001. “Remarks on the Economy of Pronunciation”. In: *Competition in Syntax*. Berlin, Mouton de Gruyter. 107-150

Grimshaw, Jane. 1997. “Projection, Heads, and Optimality”, *Linguistic Inquiry* **28**. 373-422.

Fischer, Silke. 2004. *Towards an Optimal Theory of Reflexivization*. PhD thesis, Universität Tübingen.

Haider, Hubert. 1988. “Zur Struktur der deutschen Nominalphrase”, Zeitschrift für Sprachwissenschaft 7, 32-59.

Haider, Hubert. 1993. Deutsche Syntax – generativ. Tübingen, Narr.

Halle, Morris & Alec Marantz. 1993. “Distributed Morphology and the Pieces of Inflection. In: *The View from Building 20*. Cambridge, Mass., MIT Press, 111-176.

Heck, Fabian. 2004. *A Theory of Pied Piping*. PhD thesis, Universität Tübingen.

Heck, Fabian & Gereon Müller. 2000. “Repair-Driven Movement and the Local Optimization of Derivations”. Ms., Universität Stuttgart and IDS Mannheim.

Holmberg, Anders & Christer Platzack. 1995. *The Role of Inflection in the Syntax of the Scandinavian Languages*. Oxford, Oxford University Press.
Hornstein, Norbert. 2001. *Move. A Minimalist Theory of Construal*. Oxford, Blackwell.
Legendre, Géraldine, Paul Smolensky & Colin Wilson. 1998. "When is Less More? Faithfulness and Minimal Links in Wh-Chains". In: *Is the Best Good Enough?* Cambridge, Mass., MIT Press. 249-289.
McCarthy, John & Alan Prince. 2004. "The Emergence of the Unmarked". In: *Optimality Theory in Phonology. A Reader*. Oxford, Blackwell . 483
Müller, Gereon. 1998. *Incomplete Category Fronting*. Dordrecht, Kluwer.
Müller, Gereon. 2000*a*. "Das Pronominaladverb als Reparaturphänomen", Linguistische Berichte **182**. 139-178.
Müller, Gereon. 2000*b*. *Elemente der optimalitätstheoretischen Syntax*. Tübingen, Stauffenburg.
Müller, Gereon. 2000*c*. "Shape Conservation and Remnant Movement". In: Proceedings of NELS 30. Amherst, Mass., GLSA, 525-539.
Murasugi, Kumiko. 1992. *Crossing and Nested Paths*. PhD thesis, MIT, Cambridge, Mass.
Pesetsky, David. 1982. *Paths and Categories*. PhD thesis, MIT, Cambridge, Mass.
Pesetsky, David. 1989. "Language-Particular Processes and the Earliness Principle". Ms., MIT, Cambridge, Mass.
Pesetsky, David. 1998. "Some Optimality Principles of Sentence Pronunciation". In: *Is the Best Good Enough?* Cambridge, Mass., MIT Press. 337-383.
Platzack, Christer. 1987. "The Scandinavian Languages and the Null Subject Parameter", *Natural Language and Linguistic Theory* **5**. 377-402.
Prince, Alan & Paul Smolensky. 2004. *Optimality Theory. Constraint Interaction in Generative Grammar*. Blackwell.
Pullum, Geoffrey. 1976. "Sequential and Simultaneous Rule Application in Spanish Phonology", *Lingua* **38**. 221-262.
Sternefeld, Wolfgang. 2006. *Syntax – Eine morphologisch motivierte generative Beschreibung des Deutschen*. Tübingen, Stauffenburg Verlag.
Zifonun, Gisela. 2004. "Dem Vater sein Hut – Der Charme des Substandards und wie wir ihm gerecht werden", *Deutsche Sprache* **03**. 97-126.