Title[1]

Nicholas Longenbaugh
nslonge@mit.edu

April 13, 2018

1 Introduction

- A conceptual question at the heart of modern syntactic theory:
  - *What is the correlation between φ-*Agree *and movement?*
- Early minimalism (e.g., **?**) postulated a strong correlation: *φ-Agree* is the result of movement through specific syntactic positions

(1) **Specifier-head agreement**:
If AgrX is an agreement head and DP a phrase bearing φ-features, morphological agreement obtains only if the following structural configuration obtains:

(2) $[_{\text{AgrXP}}$ DP $[_{\text{AgrXP}}$ AgrX $[\ldots$ ~~DP~~$\ldots]]]$

- This is especially successful for agreement phenomena in the *v*P domain, e.g., past-participle agreement in Romance and Scandinavian, which is (mostly) contingent on movement across the participle (**?**, **?**; **?**)

(3) *French*
  a. Jean n'a jamais fait(***es**) **ces sottises**
  Jean NEG.have.3SG never done.M.SG/*F.PL these stupid things.F.PL
  'Jean has never done these stupid things'
  b. Jean ne **les** a jamais fait**(es)**
  Jean NEG THEM.CL have.3SG never done-F.PL
  'John has never done them.'
  (adopted from **?**)

- Modern minimalist theories usually assume, however, that *φ-Agree* is formally dissociated from movement

(4) ***Agree*** (**? ?**, **?**):
An *Agree* relation obtains between a head H and a phrase XP, provided:
  (i) Matching: XP bears valued features that are a superset of the unvalued features on H
  (ii) Locality: There is no YP asymmetrically c-commanding XP that satisfies matching

- This formulation is based on a variety of cross-linguistic examples where *φ-Agree* obtains in the absence of overt movement (some examles may involve covert movement; see **?**)

[1]For helpful discussion and comments, I thank Kenyon Branan, Justin Colley, Colin Davis, Amy Rose Deal, Danny Fox, Heidi Harley, Sabine Iatridou, Daniel Margulis, David Pesetsky, Norvin Richards, Ian Roberts, Michelle Yuan, and audiences at DP-60. For patient help with judgements, I thank Paul Marty, Sophie Moracchinni, Keny Chatain, Ben Storme, Daniel Margulis, and Ezer Raisin.

– Tsez (**?**), English (**?**, **?**), Icelandic (**?**, **?**; **?**), Hindi-Urdu (**?**; **?**), Basque (**?**; **?**)

(5) Ram-ne [**rotii** khaa-nii] chaah-**ii**
Ram-ERG bread.F eat-inf.F want.PERF.FSG
'Ram wanted to eat bread.'
(**?**: 792)

- Cases where *φ-Agree* appears to trigger movement are captured by a stipulated feature, either on the head or on the *Agree*-probe itself

- This state of affairs leaves unanswered a number of fundamental questions, both theoretical and technical
  - How do we handle PPA and other apparent instances of Spec-Head agreement in a long-distance *φ-Agree* framework?
  - Can we predict the distribution of EPP features, i.e., which probes trigger movement, or must this be stipulated in an ad-hoc, language specific way?
  - Why should agreement and movement ever be correlated in the first place?
- Goals for today: Use PPA as a case study to probe the bigger questions surrounding *φ-Agree* and *Merge*/*Move*, in support of the following conclusion:

(6) ***φ-Agree*/*Merge* correlation**:
Every *φ*-probe is associated with an EPP feature that forces *Merge* to the triggering head

- Consequences:
  - ⇒ All else being equal, *φ-Agree* triggers movement
    - ∗ Spec-Head patterns result from interference effects on heads with a semantic requirement to introduce an argument: agreement trigger and argument compete for *Merge*
  - ⇒ (At least some) null subject languages have EPP and null expletives
  - ⇒ A new approach to expletive *there*
  - ⇒ Broad cross-linguistic empirical coverage of when *Agree* can be "long-distance"
- Outline
  - Past-participle agreement
    - ∗ Challenges to long-distance-*Agree* frameworks
    - ∗ A new empirical generalization
    - ∗ Capturing the data
  - A new treatment of expletive *there*
  - The *Agree*/*Merge* correlation
    - ∗ Proposal
    - ∗ Null subject languages have the EPP & null expletives
    - ∗ Predicting the cross-linguistic distribution of LDA

2 Proposal

2.1 Architectural Preliminaries

My proposal depends on some background conclusions concerning the nature and structure of the syntactic derivation, which I lay out here.

I will be taking for granted the main tenets of **?**'s (**?**) *Obligatory Operations* (ObOp) framework, which has the following central tenet. Each primitive syntactic object H is associated with a (potentially empty) set of operations $O = \{o_1, o_2, \cdots\}$, along with structural conditions $C = \{c_1, c_2, \cdots\}$ that govern when the operations can apply. If, in the course of the derivation, condition c_i on operation o_i on head H is met, then operation o_i must apply. If condition c_i is never met, however, o_i never applies and the derivation continues.

ObOp has important consequences for both of the main syntactic operations, *Agree* and *Merge*, that will be relevant to my proposal. The results concerning *Agree* are well known, so I'll just give a cursory overview here. In the terminology from above, we can define *Agree* as in (7).

(7) *X-Agree*
   a. o_i: copy the value of X on YP onto H
   b. c_i: apply o_i at H iff there is some YP with feature X such that:
      (i) Locality: H c-commands YP and there is no ZP c-commanded by H and asymmetrically c-commanding XP that bears feature X
      (ii) Accessibility: YP is accessible to H

The *accessibility* condition is added to reflect the independent hypothesis, which I will also adopt, that *φ-Agree* is case discriminating (**?**; **?**): case valuation determines whether or not a given DP is accessible to *φ-Agree* (see **??**), with accessibility parameterized across languages according to the *Moravcsik Hierarchy* (see (8b)). The most restrictive languages make only those DPs with unmarked case accessible for agreement, while some languages also make DPs with dependent case accessible, and some even tolerate agreement with DPs bearing lexical case.

(8) **Case Accessibility**:
Accessibility to *Agree* is determined according to the *Moravcsik Hierarchy*:
unmarked case » *dependent case* » *lexical/oblique* case

This notion in turn depends on the Dependent model of case (**?**; **?**), and in particular on the tenets (i) that case is valued configurationally in the syntax (Preminger 2014: Ch.9) according to the principles in (9), and (ii) that unvalued case features do not crash the derivation, in keeping with general ObOp logic. I adopt these in this paper.

(9) **Case valuation**:
   a. Lexical Case: Given the configuration [H DP], where H is a lexical case assigner, value the case feature on DP.
   b. Dependent Case: Given the configuration [DP_1 [...[...DP_2...]]], where DP_1 and DP_2 have unvalued case features,[2] value the case feature on DP_2.

[2]Note that DPs bearing lexical/oblique case do not trigger the case valuation algorithm. We can confirm this empirically by examining Icelandic quirky-subject constructions: in (ia), where neither of the subject or object bears lexical case, valuation proceeds as expected, with the object getting dependent (accusative) case and the subject unmarked (nominative) case; in (ib), however, where the subject gets lexical case, the object surfaces with unmarked

The ObOp logic then dictates that if a head H is associated with an agreement operation *X-Agree*, this operation must take place if the conditions on its application are met. If these conditions fail to be met, for example because the only possible target of *Agree* is not case-accessible, the derivation proceeds without crashing. I refer the reader to Bobaljik (2008) and Preminger's (2014) work for more information and simply accept these principles as given.

The second basic syntactic operation, *Merge*, receives comparatively less discussion in the development of the ObOp framework, although the basic logic can be readily extended to cover this operation as well. Because it will be crucially important to the eventual analysis, it's worth reviewing in slightly more detail how ObOp interacts with *Merge*. Following Preminger, I illustrate the interaction via the paradigmatic case of *wh*-movement, which I assume proceeds through all Spec(CP) positions along its path. The derivation of (10a) thus contains at least the two steps in (10b).

(10) a. What did John say that Sue bought?
b. [$_{\text{CP}}$ What [did John say [$_{\text{CP}}$ ~~what~~ [that Sue bought ~~what~~]]]]

Following Preminger (2014: 10.1.3), we can capture this behavior in a uniform and parsimonious way by assuming that all C heads, both interrogative and not, are associated with the operation *Merge-wh*, defined in general below.

(11) *Merge-X*
a. o_i: merge (a projection of) H with an YP bearing the feature X
b. c_i: apply o_i at H iff there is some YP with feature X such that:
(i) YP is present in the numeration/lexicon (hasn't been merged before), or
(ii) H c-commands XP in the structure and there is no YP c-commanded by H that both asymmetrically c-commands XP and bears the feature *wh*

In the case of the embedded clause in (10), ObOp therefore dictates that the non-interrogative C must merge with the *wh*-phrase in its scope. In examples where there is no *wh*-phrase present in the structure, however, the *Merge-wh* operation simply goes untriggered.[3] Because untriggered operations are unproblematic, the derivation converges, as desired. We can therefore safely assume that all non-interrogative Cs are alike in being associated with an obligatory *Merge-wh* operation which is triggered if and only if the proper conditions arise.

While Preminger does not discuss it, this basic system extends to other cases of *Merge* as well. For instance, we can capture the canonical EPP effect by positing that there is an obligatory *Merge*-D operation associated with T.[4] Granting that T is not capable of introducing new arguments (it is not a θ-position, nor is $[\![\text{TP}]\!]$ underlyingly type $\langle e, \tau \rangle$), there are therefore two cases in which *Merge*-D could apply, given the definition in (11): (i) there is an expletive, which does not require a θ-role and which I will take to be semantically vacuous, present in the lexicon/numeration, or (ii) there is an XP already present in the structure that can be moved to

(nominative) case, indicating that the case valuation algorithm did not take place.

(i) a. þeir seldu bókina.
they.PL.NOM sold.PL book.the.SG.ACC
'They sold the book'

b. Jóni líkuðu þessir sokkar
Jon.DAT liked.PL these socks.NOM.PL
'John liked these socks.'
(Preminger 2014: 145)

[3] The selection and introduction of a *wh*-phrase from the lexicon is ruled out by θ-theoretic concerns, I assume.

[4] As we will see later, the EPP on T is not limited to being satisfied by DPs, cross-linguistically. I set this aside for now.

Spec(TP).[5] Adopting the null hypothesis that *Merge* has access to both the lexicon and the outputs of all previous instances of *Merge*, one of these two conditions will always be met: either the complement to T contains an DP that can move to satisfy EPP, or there is no such DP, in which case an expletive can be selected from the lexicon. [6] In either case, there is always something available to be merged, so by the ObOp logic, Spec(TP) must always be occupied.

(12) a. Case (i): lower accessible DP
[EXPL/XP [T [... [... DP ...]]]] (*Merge*(T,DP) or *Merge*(T,EXPL) *obligatory*)
b. Case (ii): no lower accessible DP
[EXPL [T [... [... YP...]]]] (*Merge*(T,EXPL) *obligatory*)

An important question arises at this juncture. Once we admit that some heads are associated with obligatory *Merge* rules that operate according to the ObOp logic, we must ask if these cases are somehow special, or if all of the possible merger operations available at a given head are specified in the lexicon, like the types of available *Agree* operations. For the remainder of this paper, I will assume that the possible *Merge* operations available at *H* are indeed pre-specified, so that all cases of both *merge* and *agree* are governed by the same system and logic. It's important to point out that this does not replicating syntactic structure in the lexicon, a common criticism of other varieties of feature-driven *Merge*. Specifying the merge operations at head *H* merely serves to define the domain of operations available at that head, but gives no information about the order they apply in, which operations apply in which derivation, etc. In particular, I do not assume any predetermined ordering on operations, or any requirement that a given operation take place beyond those imposed by ObOp. The operations that take place in the derivation and their order are therefore governed purely by concerns of interpretability, e.g., as encoded in a type theory on semantic interpretation, and by the ObOp logic.

Before moving on, I illustrate a simplified derivation to highlight the key aspects of the system. For convenience, I will hereafter encode the operations available at a given head in terms of the features below, and say that a given feature is discharged by the associated operation.

(13) a. Agree features: [X:_], *Agree* with a YP bearing X
b. Merge features: [∘X∘], *Merge* with a YP bearing X

Limiting my attention to the heads V, *v*, T, C, and assume for simplicity that these are the only heads in the clausal spine. As above, I assume T has a canonical EPP feature, encoded here as [∘D∘], and a φ-probe, and that C has features for attracting various A′-elements, including *wh*, topic, etc. At *v*, I likewise assume there is a [∘D∘] feature for merging the external argument, as well as various merge features for A′-elements and a feature for merging with *v*'s complement. Transitive and unaccusative verb, I assume, are endowed at least with a feature selecting its complement.

[5]Note that those languages which disallow transitive expletives arguably impose a restriction on the first option, so that in general the EPP at T is always satisfied by movement. See fn.12, Section **??**.

[6] Languages like English, French, Mainland Scandinavian present a *prima facie* problem in that expletives are arguably restricted to being merged in Spec(*v*P) (see Section **??**; **?**; **?**; **?**; a.o.). This raises the possibility that such languages should allow an empty Spec(TP) in the case where the complement to T does not furnish an EPP-satisfying element, counter to fact. We can exclude this possibility if we assume that *v* also has an [∘X∘] feature. The logic above then proceeds identically, but at the *v*P level: either a lower XP must be merged or attracted, or an expletive merged. Spec(*v*P) will therefore always be occupied, which ensures that there will always be an XP that is accessible to satisfy the EPP at T. See Section **??** for more discussion.

(14) a. V: [[∘D∘],...]
b. *v*: [[∘V∘],[∘D∘],[∘wh∘],[∘Top∘],...]
c. T: [[∘*v*∘],[∘D∘],*φ*:_]
d. C: [[∘T∘],[∘wh∘],[∘Top∘],...]

The example in (15) is then derived as below, with the order of operations determined by concerns of interpretability. First, V merges with *what*, satisfying its [∘D∘] feature. Next *v* merges with VP, satisfying its [∘V∘] feature, then the external argument, satisfying its [∘D∘] feature, then finally with the internal argument *what*, satisfying its [∘wh∘] feature. T is then merged in the structure, satisfying its [∘*v*∘] feature, and attracts the external argument *Mary*, satisfying its [∘D∘] feature. Finally, C merges with TP, satisfying [∘T∘], and with *what*, satisfying [∘wh∘].

(15) a. What did Mary buy?
b. [$_{CP}$ what [C [Mary [T [what [Mary [*v* [buy what]]]]]]]]

2.2 The framework

The core hypothesis of the Obligatory Operations, hereafter ObOp, framework is that syntactic operations are *obligatory* but *fallible*: a given syntactic operation must apply if and only if the conditions on its application are met. The failure to apply a given operation is therefore tolerated if these conditions are never met. For our purposes, we will primarily be concerned with the primitive syntactic operations *Agree* and *Merge*, so I will enumerate my assumptions concerning these operations and illustrate how the ObOp system works here. Given the close link between *Agree* and (morphological) case, I will also discuss the version of case theory that I adopt, and how it interacts with the ObOp framework.

2.2.1 *Agree* & Case

To begin, I adopt the common position that *Agree* is a feature-driven operation, triggered by distinguished "probe" features (see (16)), and defined as in (17).

(16) **Probe features**: Given an unvalued probe feature [F:_] on head H, F can be discharged by an instance of *Agree* between H and some XP bearing a valued copy of feature F.

(17) **Agree**: Given an unvalued probe feature [F:_] on head H and a valued probe feature, [G:*i*], on phrase XP, specify the value of F on H as *i* iff:
a. Matching: G is a superset of F
b. Locality: H c-commands XP and there is no YP c-commanded by H and asymmetrically c-commanding XP that bears a valued feature satisfying matching
c. Accessibility: XP is case-accessible to H

With this in mind, the ObOp framework dictates that each unvalued probe feature must be discharged by *Agree*, if possible. If there is no XP that satisfies *Matching*, *Locality*, and *Accessibility* above, however, a probe feature can go undischarged/unvalued without crashing the derivation, unlike in the model of **?** (**?**; **?**). In such cases, I take it that unvalued/undischarged features are assigned a default value at spell out. I will presuppose a basic familiarity with the arguments in favor of this ObOp approach to *Agree*, and adopt it here without further discussion.

I will also adopt the Dependent Theory of case, which better compatible with the ObOp framework than the uninterpretable features theory (**?**, **?**). For our purposes, this model can be summarized in two core hypotheses. The first is that case is an essential reflection of argument structure rather than a licensing mechanism: the failure to value the case feature on DP thus does not result in a crash, and case valuation is determined according to the following two

rules: (i) certain lexical heads can value the case feature of their complement DPs, giving rise to so-called *lexical* or *oblique* case; (ii) when two DPs in the same local domain are such that one asymmetrically c-commands the other, the case feature on the lower DP gets valued.[7] Case valued in this way is said to be *dependent*, while the unvalued case feature on the higher DP is said to represent *unmarked* case. Unmarked case is taken to be spelled out as nominative at PF, and dependent case as accusative. Lexical case is spelled out according to the specific and possibly idiosyncratic specifications of the relevant case assigning head. I follow Preminger (2014: Ch.9) in assuming that case valuation is a syntactic operation, and moreover that it takes place as soon as the relevant structural configuration is achieved.

(18) **Case licensing & valuation**:
- a. There is no case licensing
- b. Case valuation is syntactic, and computed as follows:
  - (i) Lexical Case: Given the configuration [H DP], where H is a lexical case assigner, value the case feature on DP
  - (ii) Dependent Case: Given the configuration [DP_1 [... [... DP_2 ...]]], where DP_1 and DP_2 have unvalued case features,[8] value the case feature on DP_2

The second core hypothesis is that while *φ-Agree* is not responsible for case valuation, it is sensitive to it. Specifically, case valuation determines whether or not a given DP is accessible to *φ-Agree* (see **??**), with accessibility parameterized across languages according to the *Moravcsik Hierarchy* (see (19b)). The most restrictive languages make only those DPs with unmarked case accessible for agreement, while some languages also make DPs with dependent case accessible, and some even tolerate agreement with DPs bearing lexical case.

(19) **Case Accessibility**:
- a. *φ-Agree* is case discriminating (see **??**)
- b. Accessibility to *Agree* is determined according to the *Moravcsik Hierarchy*: *unmarked case* » *dependent case* » *lexical/oblique* case

2.2.2 *Merge*

With the basic framework surrounding agreement and case in place, I turn my attention now to the other primitive syntactic operation, *Merge*. Unlike with *Agree*, which is widely held to be a feature-driven operation, there is disagreement as to whether *Merge* should be treated similarly (yes: **??**; no: **??**). I will remain agnostic on this question in the general case, but I will adopt the better established position that there are designated "EPP" features capable of triggering *Merge* in some instances.

[7] In an ergative/absolutive case system, this is reversed, so that the higher of the two DPs' case feature is valued.

[8] Note that DPs bearing lexical/oblique case do not trigger the case valuation algorithm. We can confirm this empirically by examining Icelandic quirky-subject constructions: in (ia), where neither of the subject or object bears lexical case, valuation proceeds as expected, with the object getting dependent (accusative) case and the subject unmarked (nominative) case; in (ib), however, where the subject gets lexical case, the object surfaces with unmarked (nominative) case, indicating that the case valuation algorithm did not take place.

(i) a. Þeir seldu bókina.
they.PL.NOM sold.PL book.the.SG.ACC
'They sold the book'

b. Jóni líkuðu þessir sokkar
Jon.DAT liked.PL these socks.NOM.PL
'John liked these socks.'
(Preminger 2014: 145)

(20) **EPP features**: Given an EPP feature [○F○] on head H, [○F○] can be discharged by merging H (or a projection thereof) with an XP bearing feature F.

As with probing features, I will hold that EPP-feature discharge proceeds according to the basic ObOp logic: given an EPP feature [○F○] on head H, if there is an XP in the current workspace bearing feature F that can be felicitously merged with H, it must be; if there is no such XP, the derivation proceeds intact, without discharging the feature.

While I assume most readers are familiar with the ObOp treatment of *Agree*, the application to EPP features is somewhat more opaque, so I briefly consider two cases relevant to the ensuing analysis. The first is A′-EPP features, which are implicated in the ObOp analysis of long-distance A′-movement. I illustrate via the paradigmatic case of *wh*-movement, which I assume proceeds through all Spec(CP) positions along its path. The derivation of (21a) thus contains at least the two steps in (21b).

(21) a. What did John say that Sue bought?
b.

Following Preminger (2014: 10.1.3), we can capture this behavior in a uniform and parsimonious way by assuming that all C heads, both interrogative and not, have a [○*wh*○] feature. In the case of the embedded clause in (21), ObOp therefore dictates that the non-interrogative C must attract the *wh*-phrase, as it is capable of satisfying its [○*wh*○] feature. In examples where there is no *wh*-phrase present in the structure, [○*wh*○] is left undischarged on C.[9] Because undischarged features are unproblematic, the derivation converges, as desired.[10] We can therefore safely assume that all non-interrogative Cs are alike in bearing a [○*wh*○] feature. In contrast, if we assume, following Chomsky (2000, 2001), that uninterpretable features crash the derivation, the treatment of long-distance *wh*-movement requires that we posit two varieties of non-interrogative C: one that bears the movement-triggering feature, and which is used in exactly those cases where there is a higher interrogative C, and one that does not bear the movement-triggering feature, and which is used in all other cases. The ObOp framework thus provides a more parsimonious treatment of intermediate movement of this sort.

The second case to consider is the cross-linguistically widespread canonical EPP effect, whereby Spec(TP) must be occupied by some XP. Setting aside for now the precise identity of XP, which varies considerably across languages, we can encode this effect in the present system by positing an EPP feature, [○X○], at T. At first glance, the ObOp view of EPP would seem to be too lenient in this case: unlike with A′-EPP features, which can sometimes go undischarged, in languages exhibiting canonical EPP effects, Spec(TP) must be filled. The resolution to this apparent problem, I argue, is to recognize that the derivation *always* furnishes an XP capable of discharging the EPP feature at T. In ObOp terms, then, EPP satisfaction is obligatory because conditions always permit it to take place.

In particular, I adopt the null hypothesis that at any stage of the derivation, *Merge* has access to both the lexicon and the outputs of all previous instances of *Merge*. Granting that T is not capable of introducing new arguments (it is not a θ-position, nor is $[\![\text{TP}]\!]$ underlyingly type $\langle e, \tau \rangle$), there are therefore two options for satisfying EPP at T: (i) select an expletive, which does not require a θ-role and which I will take to be semantically vacuous, from the lexicon and merge it in Spec(TP), or (ii) select an XP already present in the structure and re-merge it in Spec(TP).[11]

[9] The selection and introduction of a *wh*-phrase from the lexicon is ruled out by θ-theoretic concerns, I assume.

[10] I set aside at present the question of how the above-mentioned "attraction" takes place, and simply assume that move/re-merge is constrained by familiar principles of locality. I return to this issue in Section **??**.

[11] My assumptions about expletives will be updated in the next section.

Taken together, these two cases suffice to capture all possible derivations: either the complement to T contains an XP that can move to satisfy EPP, in which case either XP or an expletive can be merged, or there is no such XP, in which case an expletive can still be merged. [12] In either case, there is always something available to satisfy the EPP, so by the ObOp logic, Spec(TP) must always be occupied.

(22) a. Case (i): lower accessible XP
[EXPL/XP [T [... [...XP ...]]]] (*Merge*(T,XP) or *Merge*(T,EXPL) *obligatory*)
b. Case (ii): no lower accessible XP
[EXPL [T [... [... YP...]]]] (*Merge*(T,EXPL) *obligatory*)

While this leaves unanalyzed several complications concerning the cross-linguistic behavior and distribution of expletives, which I return to in Section ?? (see also fns. 11, 12), the net result is that the canonical EPP is correctly captured under the ObOp logic.

2.2.3 Order of operations

I conclude with a final assumption concerning the derivational order of operations. I will take feature discharge to be cyclic and sequential, but otherwise unordered (see (23)). In particular, if head H has features F, G, there is no preference for which is discharged first, and if both orders yield licit results, the derivation continues in parallel, potentially producing two licit outputs.

(23) **Order of operations**:
Feature discharge is:
a. Cyclic: Only the features of the daughters of the root may be discharged
b. Sequential: Given head H with features F, G, either F must be discharged before G or G must be discharged before F.
c. Unordered: Given head H with features F, G, both orders of discharge F » G and G » F are, in principle, allowed.

2.3 Expletives and *v*/Voice

The eventual proposal will depend on the feature make up of the *v*/Voice head, as well as on the related issue of the basic merge position and case/agreement properties of expletives. I begin by considering the latter. Generally speaking, those languages with overt expletives can be broadly classified into two groups: those that allow expletives with transitive clauses, and those that restrict them to passive/unaccusative clauses. I illustrate the former category via Icelandic and the latter via English.

(24)

(25)

Our discussion will focus exclusively on the latter category of language, where I will follow **?**, **?** in assuming that the impossibility of transitive expletives is indicative that expletives may only be

[12] Languages like English, French, Mainland Scandinavian present a *prima facie* problem in that expletives are arguably restricted to being merged in Spec(*v*P) (see Section **??**; **?**; **?**; **?**; a.o.). This raises the possibility that such languages should allow an empty Spec(TP) in the case where the complement to T does not furnish an EPP-satisfying element, counter to fact. We can exclude this possibility if we assume that *v* also has an [∘X∘] feature. The logic above then proceeds identically, but at the *v*P level: either a lower XP must be merged or attracted, or an expletive merged. Spec(*v*P) will therefore always be occupied, which ensures that there will always be an XP that is accessible to satisfy the EPP at T. See Section **??** for more discussion.

merged in Spec(vP), not in Spec(TP) or Spec(CP). The absence of transitive-expletive constructions then follows if we assume that at most one DP can be introduced in Spec(vP).

(26) **Expletives are merged low**:
In languages without transitive expletives, expletives must be merged in Spec(vP).

I refer the reader to **?**, Deal 2009, and Wu 2017 for additional arguments in support of (26), and will for now adopt it wholesale.

As mentioned in fn.12, the low-merge theory of expletives raises non-trivial questions concerning the ObOp treatment of canonical EPP effects. If expletives cannot be merged at T, it follows that the only way to satisfy EPP in languages like English is to move an XP already present in the structure. Our theory, as stated, therefore predicts that if there is no such lower XP, the EPP feature on T should be allowed to go unsatisfied. Such a case is arguably attested with predicates like *seem*, whose CP complements cannot serve as Spec(TP) subjects. As (27) makes clear, however, the EPP must still be satisfied in such cases, namely by expletive *it*.

(27) a. *That John is upset seems to me.
b. *Seems to me that John is upset.
c. It seems to me that John is upset.

We can readily capture this behavior if we assume that those languages showing canonical EPP effects also have an EPP feature on v. Given that expletive merger is possible at v, it will always be possible to satisfy this EPP feature, so Spec(vP) will always be filled. This, in turn, ensures that T will always have a suitable target to attract to satisfy its EPP feature, capturing data like (27). If we also grant that Spec(vP) is an obligatory intermediate landing site for A′-movement (**?**; **?**; **?**; a.o.), so that it has an A′-EPP feature, then we can summarized the postulated feature make up on v in EPP-languages as below. I have include φ and A′-probe features in anticipation of the discussion to come, although nothing we have said so far requires they be present.[13]

(28) **Feature make up of** v:

$$v: \begin{bmatrix} \text{Canonical EPP:} & [\circ \text{D} \circ] \\ \text{A}'\text{-EPP:} & [\circ \text{A}' \circ] \\ \varphi\text{-probe:} & [\varphi : \_] \\ \text{A}'\text{-probe:} & [\text{A}' : \_] \end{bmatrix}$$

It's worth pointing out that (28) is intended to be constant across all varieties of v, transitive, unergative, passive, and unaccusative. I assume that θ-theory and/or basic concerns of interpretability rule out illicit such as expletive subjects with transitive and unergative v (see Section **??** for more discussion).

The final issue that will be relevant to the proposal is the case/agreement properties of expletives. We will be concerned in the remainder of the paper with two types of expletives: (i) the grammaticalization of the default 3rd person pronoun (*it* in English, *il* in French, *det* in MSc, etc.); (ii) the grammaticalization of the distal locative proform (*there* in English, *der* in MSc, etc.). In the case of the default 3.SG expletive, hereafter DefExp, I will adopt the widespread view that the expletive differs from the semantically contentful variety only in that it is semantically vacuous. Crucially, it maintains the formal syntactic properties of its non-expletive counterpart, and is therefore capable both of discharging probe features and triggering dependent case on a lower

[13] Note that I have made the simplifying assumption that the canonical EPP demands merger of category D. This is a simplification, even in the most rigid canonical-EPP languages like English, where locative PPs can satisfy EPP (**?**). I return to this issue in Section **??**.

DP. In the case of the locative expletive, hereafter LocExp, I take a similar stance: LocExp is a semantically vacuous but formally identical version of the corresponding locative proform. It therefore maintains all the formal properties of a locative, so it can satisfy the EPP (**?**), but it interacts for Case/Agree as an oblique. Most importantly, this means that LocExp (i) cannot trigger *φ-Agree* (locatives don't trigger *Agree*; but see Bantu **??**) and (ii) does not trigger dependent case on a lower DP (obliques don't trigger case valuation; see **?**; **?**; fn.**??**). I further justify this treatment of LocExp in the appendix, but for now I take it as given.

(29) **Expletive assumptions**:
- a. DefExp: the default 3.SG expletive is a semantically vacuous but formally identical version of the default 3.SG pronoun; it can discharge *φ*-probes and trigger case valuation on lower DPs
- b. LocExp: the locative expletive is a semantically vacuous but formally identical version of the locative proform; it cannot discharge *φ*-probes, and it does not trigger case valuation on lower DPs

3 Proposal

With these framework and structural/derivational assumptions in place, my proposal is that *Agree* and *Move/Merge* are related via the action of the following basic economy principle.

(30) **Feature Maximality** (FM):
Given head H with features [F$_1$] ... [F$_n$], if XP discharges [F$_i$], XP must also discharge each [F$_j$] that it is capable of.

The core idea is that once a phrase XP has been selected as the target for a syntactic operation originating at head H, the relationship between H and XP must maximize to include all possible additional operations originating at H capable of targeting XP. This principle subsumes and extends the "free rider" property of *Agree* (**?**, **?**; **?**; **?**), and is closely related to the notion of economy proposed by **?**.

For a brief demonstration, suppose H is a head bearing an valued probing feature, say [*φ*:_], and an EPP feature, say [∘D∘]. By (30), if [*φ*:_] on H is discharged via *Agree* with a *φ*-bearing target DP, then DP must also discharge [∘D∘], that is, DP must be merged with (a projection of) H. This has the effect that *Agree* obligatorily triggers *Move* if the head bearing the probe feature has an undischarged EPP feature. Alternatively, suppose that [∘D∘] on H is discharged by merging a new DP in the structure. Since *Agree* is conditioned on c-command by the head containing the probe feature, a first-merged DP is not eligible to discharge the [*φ*:_] on H, so only the EPP-feature is discharged. In such a scenario, a lower DP may then discharge the probing feature on H without undergoing obligatory movement, since the EPP feature has already been discharged.

This leads to the following prediction: on head H with both an EPP and a probe feature, *Agree* triggers obligatory movement unless an additional XP has been merged to satisfy H's EPP feature. I know argue that this basic state of affairs obtains with the core PPA data in Romance and Mainland Scandinavian, confirming the prediction and supporting the existence of a principle like FM in the grammar.

3.1 Capturing PPA: Core cases

I begin my discussion of PPA by considering the

- Italian & French transitives - French & Scandinavian passives/unaccusatives

3.2 Additional predictions

- Italian passives/unaccusatives/applicatives - French stylistic inversion - Scandinavian *there*-expletives - Clitic and *wh*-movement

4 Extending the account: the φ-*Agree*/EPP correlation

5 Conclusion