Synopsis  This paper proposes that morphological selection and subcategorization are derived from bare output conditions, specifically, conditions on the licensing of uninterpretable features. I show that, bundled with the appropriate feature assignments, the Merge Condition in (1) provides a deterministic and strictly local syntactic mechanism for structure building without recourse to ill-defined notions such as subcategorization and selection.

Agree as a licensor for Merge i) Verbal morphology  Following the traditional view, I assume that morphological dependency relations are top-down relations: the higher (typically, but not necessarily, the selecting) element licenses the lower element (e.g., an auxiliary licenses the participle). The dependency is instantiated syntactically as Reverse Agree in (2) (see Pesetsky and Torrego 2007, Bošković 2009, To appear for motivation that Agree is valuation driven). Specifically, following Wurmbrand (2010), I argue that verbs involve uninterpretable unvalued T-features \([uT: __]\), which must receive a value under Agree with a higher head (e.g., tense). Arguments for this view come, among many others, from VP-ellipsis in English, the IPP-construction and parasitic participle morphology in Germanic. To illustrate, the lack of morphological identity between overt and elided V(P)s poses a challenge for standard Agree à la Chomsky (2000, 2001), as well as the feature sharing version of Agree in Pesetsky and Torrego (2007). As shown in (3), if verbs are inserted fully specified (i.e., valued), the antecedent and elided VPs do not match and ellipsis would be unexpected. Under Reverse Agree, the two VPs start out identical, hence ellipsis is possible (before T is merged and valuation takes place). Assuming that all verbal elements involve \([uT: __]\), the upward ‘selection’ dependency follows from Agree.

ii) Movement/Re-Merge  Movement constitutes a second set of constructions in which Merge is licensed by Agree. Following standard assumptions, I propose that T (in EPP-languages such as English), as well as v involve uninterpretable unvalued qi-features \([uq: __]\). Merge (external or internal) is hence licensed, in fact necessary to value the unvalued features of T/v. As shown in (4), similar to Bošković (2007), this account derives movement without the stipulation of an EPP feature and allows us to eliminate the activation condition and reflex checking for Case. Similarly, wh-movement is the consequence of an unvalued feature \([Q: __]\) on C, and Merge of a wh-XP with C’ is licensed under Agree.

Matching of valued uninterpretable features as a licensor for Merge  Following Pesetsky and Torrego (2006), I propose that in the typical case Merge (complementation) is restricted to certain feature combinations. In contrast to their approach, I argue that this cannot be considered an Agree relation, since complementation Merge crucially must not involve valuation. Rather Merge involves the Matching relation in (1ii). I assume that the ‘selectional’ features of verbs are uninterpretable valued wh- and qi-features, (5). As shown in (6), a verb specified for \([uQ: wh]\) (e.g., ask) can then only Merge with a complement that is marked iq (typically, an interrogative CP); a verb specified for \([uq: val]\) must Merge with a complement marked iq (NPs or declarative CPs with that; that is assumed to involve iq:3sg).

Clausal complementation in English (E) and German (G) (Table 1)  E and G are identical, with the only difference that root clauses involve a verb second (V2) CP in G, but a bare TP in E. The obligatory occurrence of V/T-to-C in E & G matrix interrogatives is derived from the Root Generalization in (7): V/T-to-C creates a CP marked iT due to the presence of T in C (Pesetsky and Torrego 2001, 2004, 2007). The obligatory lack of V/T-to-C in embedded interrogatives follows from the Merge Condition and the restricted set of features in (5): a verb cannot combine with a complement marked iT: val on top, thus V/T-to-C is blocked. The difference boils down to ‘selection’ as defined in (7): embedded questions are always ‘selected’ (the matrix V involves uQ: wh), hence the CP must not be iT; matrix questions are not selected, hence the CPs must be iT. Lastly, I show that this system provides a new account of the distribution of embedded root clauses (that-less clauses in E, embedded V2 in G). I argue that embedded root clauses are V2 CPs in G, and bare TPs in E (Hegarty 1991, Webellhuth 1992, Doherty 1993, 1997, 2000, Bošković 1997, Svenonius 1994, Franks 2005)—that is, iT clauses in both languages. Given (1) and (5), embedded root clauses cannot Merge with V in syntax. I propose that Merge is only possible as part of Transfer in these cases (similar to late Merge of adjuncts). Merge at Transfer feeds into LF and PF, but is not part of syntax proper. This has the welcome result that embedded root clauses are syntactically frozen, which is documented by their inability to undergo topicalization, subject movement, or extraposition (Doherty 1997, Bošković and Lasnik 2003).
The Merge Condition: A syntactic approach to selection

(1) Merge Condition: Merge α and β if i. or ii.
   i. α can value (a sub-feature of) the head of β. [Agree/Valuation]
   ii. Every member of α matches a corresponding [af: val] on β. [Complementation]

(2) Reverse Agree: A feature F: __ on a head α is valued by a feature F: val on β, iff
   i. β asymmetrically e-commands α. AND
   ii. There is no γ with a valued interpretable feature F such that γ commands α and is e- commanded by β.

   a. John slept, and Mary will sleep/*sleep too.
   b. Standard Agree: T: __ » [vp V T: val (slept = T: past)] … and
      T: __ » [vp V T: val (sleep = T: inf)] No identity for ellipsis
   c. Reverse Agree: T: past » [vp V T: __ (V valued by T)] … and
      T: fut » [vp V T: __ (V valued by T)] Identity of VPs

(4) a. T [iT: past; uQ: __] » DP_SUBJ [uT: __; iQ: val]
   b. T Agrees with DP and values DP’s iT: __ (Case, see Pesetsky and Torrego 2007)
   c. DP_SUBJ Re-Merges with iT: __; DP_SUBJ values uQ: __

(5) ‘Selectional’ features of V: [uQ: wh], [uQ: val_DEFAULT]

(6) a. John ate <something> uQ: val an apple
   b. John told us <something> uQ: val, (uQ: val) a story, that/when she left
   c. John asked <something, a question> uQ: val, uQ: val a question, where Mary is
   d. John thinks <something, a proposition> uQ: val, *uQ: val that/*when she left

(7) Root generalization: A non-selected finite clause (i.e., a clause that does not Merge with a sister in syntax) is marked iT on its top projection.

<table>
<thead>
<tr>
<th>Table 1: Type of clause</th>
<th>(Not) Selected</th>
<th>Value</th>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix declarative: John has seen Peter.</td>
<td>not selected</td>
<td>iT</td>
<td>TP</td>
<td>CP (V2); V/T–to–C</td>
</tr>
<tr>
<td>Matrix interrogative: Who has John seen?</td>
<td>not selected</td>
<td>iT, iQ</td>
<td>CP (wh)</td>
<td>V/T–to–C</td>
</tr>
<tr>
<td>Embedded interrogative</td>
<td>selected</td>
<td>iT, iQ</td>
<td>CP (wh)</td>
<td>*V/T–to–C</td>
</tr>
<tr>
<td>I wonder who (*has) John has seen.</td>
<td></td>
<td>*iT</td>
<td>CP (wh)</td>
<td>V/T–to–C</td>
</tr>
<tr>
<td>Embedded declarative</td>
<td>not selected</td>
<td>iT</td>
<td>TP</td>
<td>CP (V2); V/T–to–C</td>
</tr>
<tr>
<td>John said (that) Mary left.</td>
<td>selected</td>
<td>iQ</td>
<td>TP</td>
<td>CP (that)</td>
</tr>
</tbody>
</table>

References