II: FEATURES

1. MOVEMENT BY FEATURE CHECKING

• The ill-formedness of all the examples in (1) can be traced to a single source: A category has been displaced without there being a trigger for movement.

(1) a. *Sleeps, John \(i\)  
   b. *I know that is, John \(i\) here  
   c. *The book, did she read \(i\)  
      (cf. Which book did she read)  
   d. *Sally, (it) seems that \(i\) is sick

Thus, constituents move for a reason. This can be expressed by adopting the principle in (2):

(2) **LAST RESORT:**  Movement has to be motivated.  
    ["Move only if necessary!"]

• Conversely, in certain contexts, movement is obligatory, indicating that if the representation contains a trigger for movement, the instruction given by this trigger cannot be ignored:

(3) a. He wondered who, to call \(i\)  
   b. *He wondered to call who

(4) a. John, seemed \(i\) to be sick  
   b. *Seemed John to be sick

In MP, the obligatory nature of movement is assumed to be linked to the requirement that the derivation eliminates unnecessary (redundant) morphological features, as stated by (5):

(5) a. Certain (yet to be specified) features must not be passed on to the interfaces.  
   b. These features can be **CHECKED** (i.e. erased) by movement.

As a consequence of (5), movement saves the derivation from crashing at one of the interfaces.

○ Note that by assumption (5)a, neither PF nor LF tolerates unchecked features. Thus, for a derivation to crash, it is sufficient that a feature survives at one of the two interfaces. However, given the architecture of the grammar, a feature that has been eliminated before Spell-Out is automatically also gone by LF (\(\Rightarrow\) in (6)). By contrast, a feature that has *not* been erased prior to Spell-Out, triggering a PF-crash, can still be checked prior to LF, ensuring LF-convergence (\(\Leftarrow\); for empirical consequences of these options see Handout IV on Head Movement):

(6) \[
\begin{array}{|c|c|c|}
\hline
\text{Feature } \alpha \text{ is} & \text{unchecked at LF} & \text{checked prior to LF} \\
\hline
\text{unchecked at PF} & \text{Crash at PF and LF} & \Rightarrow \text{Crash at PF, converges at LF} \\
\hline
\text{checked prior to PF} & \text{impossible} & \Leftarrow \text{Derivation converges} \\
\hline
\end{array}
\]
**REDUNDANCY**

Feature checking in the MP is used to cancel *morphological redundancies*: languages overtly mark many properties that are strictly speaking not necessary in order to identify the pertinent relations inside the clause or to interpret the sentence. Subject and verb agree e.g. in many languages, providing two times number and person information:

(7) a. Ich\textsubscript{1st. sg.} kenne\textsubscript{1st. sg.} die Antwort
b. Du\textsubscript{2nd. sg.} kennst\textsubscript{2nd. sg.} die Antwort
c. Sie\textsubscript{3rd. sg.} knnnt\textsubscript{3rd. sg.} die Antwort

Cancelling the redundant features leads to representations that are more parsimonious and resemble to a certain degree formal languages (the languages of logic and mathematics such as Calculus, Predicate Logic, etc ...).

**WHERE DO FEATURES COME FROM?**

Features such as the EPP-feature are part of the numeration. More precisely, functional heads (T, D, C,...) come in varieties that potentially differ in their feature specification. The numeration for *John seemed to win* could e.g. contain, among others, two T-nodes that bear the EPP-feature (note that by assumption, all T’s are specified [EPP]).

(8) \( N = \{ \text{John}_i, T^\circ_{[\text{EPP}], 2}, \text{seemed}_i, to_i, \text{win}_i \} \)

Movement can now be motivated by demanding that the EPP-features be eliminated (checked) before the derivation reaches the interfaces PF and LF. Since PF is the first interface, checking has to occur before Spell-Out, and will have an audible effect. In the sample derivation (9), the two EPP-features are checked by moving the DP *John* into the two respective SpecTP positions (other checking relations, e.g. for subject-verb agreement, omitted):

(9) \( \text{John seemed to win} \)
   a. \( [\text{TP}, T^\circ_{[\text{EPP}], } [\text{vp, John win}]] \quad \text{Merge T} \)
   b. \( [\text{TP, John, T}^\circ_{[\text{EPP}], } \text{to, [vp, t, win]}] \quad \text{Move John} \)
   c. \( [\text{TP, John, T}^\circ_{[\text{EPP}], } \text{to, [vp, t, win]}] \quad \text{Check EPP-feature} \)
   d. \( [\text{vp, seemed, [TP, John, T}^\circ_{[\text{EPP}], } \text{to, [vp, t, win]}]] \quad \text{Merge seemed} \)
   e. \( [\text{TP, T}^\circ_{[\text{EPP}], } [\text{vp, seemed, [TP, John, T}^\circ_{[\text{EPP}], } \text{to, [vp, t, win]}]]] \quad \text{Merge T} \)
   f. \( [\text{TP, John, T}^\circ_{[\text{EPP}], } [\text{vp, seemed, [tp, t, T}^\circ_{[\text{EPP}], } \text{to, [vp, t, win]}]]] \quad \text{Move John} \)
   g. \( [\text{TP, John, T}^\circ_{[\text{EPP}], } [\text{vp, seemed, [tp, t, T}^\circ_{[\text{EPP}], } \text{to, [vp, t, win]}]]] \quad \text{Check EPP-feature} \)

\* If features are not checked, the derivation crashes, accounting for the ill-formedness of the b-examples below (for more on the wh-feature see section 3):

(10) a. \( \text{John, T}^\circ_{[\text{EPP}], } \text{seemed, to be sick} \)
    b. \*\( T^\circ_{[\text{EPP}], } \text{seemed, John, to be sick} \quad \text{x unchecked [EPP]} \)

(11) a. \( \text{He wondered, [C}^\circ_{[\text{wh}], } \text{who, to call, t}_j] \)
    b. \*\( \text{He wondered, [C}^\circ_{[\text{wh}], } \text{to call, who} \quad \text{x unchecked [wh]} \)
The checking mechanism is defined by the following assumptions (essentially Chomsky 1991, 1995; for other views see Adger 2003; Hornstein et. al 2005; Sternefeld 2006):

(12) **Feature Checking (1st version)**

a. Feature checking involves a **Probe** and a **Goal**.

b. Prior to movement, the probe c-commands the goal.

c. Features can only be checked in a **Checking Relation** (see (13)).

   [Roughly, these configurations are:
   - Specifier-head relation (resulting in XP-movement)
   - Head-adjunction (resulting in head movement)]

d. Checking **Deletes** the checked feature.

Technically, the structural configurations can be described more precisely as in (13):

(13) Two nodes $\alpha$ and $\beta$ can enter into a checking relation if and only if a. and b. hold.

a. **C-command Clause**

   $\alpha$ c-commands $\beta$

b. **Intervention Clause**

   There is no other node $\gamma$ that c-commands one member of the set \{$\alpha$, $\beta$\} and is c-commanded by the other member of this set.

- On the one hand, (13)a covers cases in which some specifier YP c-commands a head X (see (14)a). On the other hand, (13)a is also satisfied if some head Y adjoins to another head X as in (14)b (NB: Y c-commands X, but not vice versa, because a segment of X contain Y).

- Finally, there is a third possible checking relation that is admitted by (13)a and which involves mutual c-command, i.e. the head-complement relation in (14)c. (For manifestations of this relation see discussion of *wh*-interrogatives in section 3).

- (13)b is crucial to block contexts such as (14)d in which $\alpha$ and $\beta$ are not ‘close’ enough.

(14) a. $\checkmark$ XP

   $YP_{[F]} (= \alpha)$

   $X'$

   $X^*_{{[F]}} (= \beta)$

   ....

b. $\checkmark$ X'

   $X^{*}_{[F]} (= \beta)$

   $Y^{*}_{[F]} (= \alpha)$

   ....

c. $\checkmark$ XP

   $YP_{[F]} (= \alpha)$

   $X'$

   $X^{*}_{[F]} (= \beta)$

   YP$^{*}_{[F]} (= \alpha)$

d. $\times$ XP

   $YP_{[F]} (= \alpha)$

   $X'$

   Intervener $\equiv$ X$^* (= \gamma)$

   ZP

   Z$^*_{[F]} (= \beta)$

   ...
2. INTERPRETABILITY AND VALUATION

Features come in two flavors, interpretable and uninterpretable. Uninterpretable features (a.k.a. formal features) are properties that are only readable by C_{HL}, i.e. they can be processed during the syntactic derivation, but do not have any purpose to fulfill at the interfaces PF and LF. Now, the principle of Full Interpretation demands that at any point of the deprivation, the syntactic representations only contain elements that can be interpreted, i.e. used at this point. From this, it follows that uninterpretable features have to be checked and eliminated by PF/LF. (For a further distinction, which divides features in weak and strong ones, see Handout IV.)

- The plural feature on the auxiliary in (15) is uninterpretable, as it does not allow inferences as to the cardinality of the subject denotation. (15) actually means John is asleep and Bill is asleep and Sally is asleep, with subjects that denote singular individuals in SpecTP. Thus, the plural feature on the auxiliary does not contribute to the interpretation but merely serves as a formal feature marking subject-verb agreement.

(15) a. [John, Bill and Sally]Plural arePlural asleep.
   b. [John, Bill and Sally]Plural arePlural asleep.

Other uninterpretable features include EPP, phi-features on verbs and (structural) Case on DPs.

(16) [This group of people]Singular isSingular asleep.

- In contrast, the plural feature on the pronoun their bears an interpretable plural feature. Otherwise, the meaning contrast between (17)a and (17)b could not be captured.

(17) a. Sally, and Mary sold her car
   b. Sally, and Mary sold their car

In order to incorporate interpretability, it is necessary to supplement the definitions for checking theory in (12) by the following set of assumptions (a version of Chomsky 1999, 2000):

(18) FEATURE CHECKING (2ND VERSION)
   a. Uninterpretable features do not have a value. (to be revised)
   b. The probe bears UNINTERPRETABLE features. (to be revised)
   c. Checking VALUES uninterpretable features.
   d. Checking deletes uninterpretable features.

(19) Notational convention: Uninterpretable features are prefixed by u-

Example: [u-wh] signifies that [wh] is uninterpretable

- To see how the system works, consider the sample derivation for She left in (20):

(20) a. [TP T° [u-a person, u-β number, u-EPP, NOM] [vP she[3rd sg. fem, u-γ Case] left]]
   b. [TP she[3rd sg. fem, u-γ Case] T° [u-a person, u-β number, u-EPP, NOM] [vP t left]] Move she
   c. [TP she[3rd sg. fem, u-NOM] T° [u-3rd sg., u-EPP, NOM] [vP t left]] Valuation of features on DP and T
(20)   d. \[TP \text{she}^\gamma_3 [\text{[3rd sg. fem, } \underline{\text{NOM}}] T^\circ [\text{[3rd sg., } \underline{\text{EPP, NOM]} [\text{vP t, left}}]]] \]

Elimination of uninterpretable features
on DP and T

- T in (20) bears the following features:
  - Uninterpretable Phi-features: person, number and gender (latter omitted for space reasons)
  - Uninterpretable EPP
  - Nominative Case feature (it is not clear whether this feature is uninterpretable or not - for the moment, it will be taken to be interpretable. This assumption will be revised below, though!)

- DP in (20) bears the following features:
  - Interpretable Phi-features
  - Uninterpretable Case feature

Note that the last step of (20) - elimination of uninterpretable features - is required by *Full Interpretation*. At LF, the information whether the verb is singular or plural does (arguably) not matter, it can be read off the subject DP due to obligatory subject-verb agreement. Thus, the Phi-features of T\(^\circ\) are uninterpretable at the interfaces and need to be erased prior to Spell-Out. This is achieved by the checking algorithm in (12).

- Movement eliminates uninterpretable features before the derivation reaches the interfaces.

**Greed and Case Freezing**

- (21) illustrates the **Case Freezing Effect**. Raising out of finite clauses is blocked because a DP (*John*) is ‘frozen in place’ once the DP has checked its uninterpretable features:

(21)   *John seems that t left

- One way to analyze (21) consists in adopting a economy principle generally referred to as **Greed**.

(22)   **Greed**

A category \(\alpha\) can move only if movement leads to elimination of features on \(\alpha\).

Greed has the consequence that checking Case features of *John* in the lower clause (see (21)b) eliminates further motivation for raising *John* into the higher clause:

(21)   *John seems that t left

a. \[TP \text{John} [\underline{\text{γ Case}}] T [\underline{\text{EPP, NOM]} [\text{left}}]] \]
b. \[TP \text{John} [\underline{\text{γ Case}}] T [\underline{\text{EPP, NOM]} [\text{left}}]] \]
c. *\[TP T [\underline{\text{EPP, NOM]} \text{seems John} [\underline{\text{γ Case}}] T [\underline{\text{EPP, NOM]} [\text{left}}]] \]

Greed prevents *John* from checking EPP-feature of higher T
3. Feature interaction in interrogatives

In English (unlike Chinese, Japanese,...), question formation always involves overt movement to the C-domain:

(3)  a. He wondered who to call
     b. *He wondered to call who

(23) a. What did you read?
     b. *You read what? (in direct question interpretation - ok as an echo question)

(24) a. Did somebody read the book?
     b. *Somebody did read the book?

3.1. Feature checking analysis of matrix questions

- Each question bears an uninterpretable [u-wh]-feature in C° which needs to be checked by overt movement. Thus, the [u-wh]-feature in C° works just like an [u-EPP]-feature in that it gives the derivation the instruction to fill a specific position in the tree (here SpecCP or C°) with overt material. (In the literature it is sometimes assumed that the [u-wh]-feature on C° is an EPP-feature - for reasons of perspicuity, I will keep the terminological distinction, without any ontological commitment, though [i.e. maybe the [u-wh]-feature on C° is an EPP-feature, maybe not].)

- Each wh-phrase bears an interpretable [wh]-feature that can be used to check off [u-wh] on C°.

- In matrix questions, checking can be achieved in one of two ways:
  - a wh-phrase moves to SpecCP, as in (25)a, or
  - C° is filled by a head, such as the finite auxiliary in (25)b (alternatively, one might assume that SpecCP in (25)b is filled by an empty question operator - nothing bears on the choice here).

     b. CP[Q] C’ C°[u-wh, Q] did somebody VP read the book

- An interrogative C° is interpretable in semantics, it is responsible for turning a clause into a question. Thus, C° also bears an interpretable feature [Q] in addition to the uninterpretable [u-wh]-feature triggering movement (see (25)). Non-interrogative C° lacks both of these features.

- The Q-feature in (25) is not eliminated in course of the derivation, but projected to the CP level, passing on the information that the clause is interpreted as an interrogative.
For an earlier, pre-minimalist account see Rizzi’s analysis in terms of the Wh-Criterion.

If the wh-phrase does not move, as in (23)b, the ill-formedness of the string can be traced to two possible sources.

- Either the derivation contains an unchecked wh-feature, as detailed by (26)a:

\[
\begin{align*}
(26) & \quad \ast \text{You read what?} \\
& \quad a. \quad [\text{CP, } [Q] C^o_{[u\text{-wh, } Q]} \text{ You read what}_{[wh]} ]? \quad \times \text{unchecked } [u\text{-wh}] \\
& \quad b. \quad [\text{CP } C^o \text{ You read what}_{[wh]} ]? \quad \times \text{not interpretable as question}
\end{align*}
\]

- Alternatively, the derivation converges at the interface, but cannot be interpreted as a question in semantics (see (26)b). More precisely, if the numeration selects a non-interrogative \( C^o \) from the lexicon, the clause simply lacks an obligatory ingredient for assigning to (26) a question meaning in the semantic component. Thus, the derivation fails to be interpretable.

**Exercise:** The system correctly leads one to expect that the following clauses cannot be interpreted as questions ((27)a), or are ungrammatical ((27)b). Why is this so?

\[
(27) \quad a. \quad \text{This boy read this book} \\
\quad b. \quad \ast \text{This book did this boy read} \\
\quad \text{(cf. Which book did this boy read?)}
\]

### 3.2. Feature Checking Analysis of Embedded Questions

Embedded questions, as in (28), are structured around verbs which require an interrogative complement (*ask, wonder, inquire, question, [one interpretation of] know,...*).

\[
(28) \quad a. \quad \text{She asked/wondered who read the book} \\
\quad b. \quad \text{She asked/wondered if Bill read the book}
\]

One way to implement this requirement is to assume that question embedding verbs bear an uninterpretable \([u\text{-Q]}\)-feature (*عاش*), and that this feature needs to be checked by a [Q]-feature on its sister node, the embedded sentence. (Recall from section 2 that on current assumptions, head complement relations also qualify as checking relations (14)c)). As the [Q]-feature on the embedded \( C^o \) projects up to the CP-level (\( \Rightarrow \)), this condition is fulfilled in (28), as shown in detail by the trees in (29).

---

\[1\]For an earlier, pre-minimalist account see Rizzi’s analysis in terms of the Wh-Criterion.

Embedded questions provide a good illustration for how different choices of C° interact with the feature checking mechanism.

3.2.1. S-SELECTIONAL MISMATCHES I: PREDICATES THAT EMBED INTERROGATIVES

The feature checking mechanism requires that each question embedding verb, which by assumption bears the [u-Q] feature, is supplied with a question in its complement position, blocking structures such as in (30).

(30) *She wondered Bill read the book

In order to prove that the system actually excludes (30), it is necessary to demonstrate that all possible combinations of C°s with the numeration lead to non-convergent outputs:

**OPTION A:** If C° bears the specification [u-wh, Q], the [u-Q]-feature of the embedding verb wonder can be eliminated, but the [u-wh]-feature on C° remains unchecked, causing the derivation to crash:

(31) a. She wondered\(_{[u-Q]}\) [CP C°\(_{[u-wh, Q]}\) Bill read the book]
    b. She wondered\(_{[u-Q]}\) [CP C°\(_{[u-wh, Q]}\) Bill read the book] \(\times\) unchecked [u-wh]

**OPTION B:** If a non-interrogative C° is selected in the numeration, the [u-Q]-feature of the matrix predicate cannot be checked, leading to non-convergence:

(32) She wondered\(_{[u-Q]}\) [CP C° Bill read the book] \(\times\) unchecked [u-Q]

Given that there are only two types of C°s in English (an interrogative C°\(_{[u-wh, Q]}\) and a bare declarative C° that lacks features), no other combinations for assembling the sentence are available. Thus, it can be concluded that every possible derivation of (30) crashes.

QED
Finally, sentences such as (33) are ill-formed because either the wh-feature on C° ((33)a) or the Q-feature on the matrix verb remains unchecked:

(33) *She wondered Bill bought what

a. She wondered[u-Q] [CP C°[u-wh,Q] Bill bought what] X [u-wh] unchecked
b. She wondered[u-Q] [CP C° Bill bought what] X [u-Q] unchecked

3.2.2. S-SELECTIONAL MISMATCHES II: PREDICATES THAT EMBED DECLARATIVES

Selectional mismatches can also go in the other direction. In (34), the embedding verb selects for an interrogative, while the complement contains a wh-phrase:

(34) a. *She claimed that Bill bought what
b. *She claimed what Bill bought

○ The sentences in (34) are not excluded by the feature checking mechanism, as the respective representations do not contain unchecked uninterpretable features. As revealed by (35), both the wh-feature on the wh-phrase in (35)a and the Q-feature on C° in (35)b are interpretable. (Other possible combinations for deriving (34)b such as the one in which a non-interrogative C° is combined with wh-phrase in SpecCP are excluded by Last Resort).

(35) a. She claimed [CP that Bill bought what[wh]] all featural requirements fulfilled
b. She claimed [CP, [Q] C°[u-wh,Q] what Bill bought] all featural requirements fulfilled

Rather, (34) illustrates the requirement already mentioned in the previous section that a derivation has to yield an interpretable output representation. More precisely,

○ (34)a is uninterpretable for the same reason that (26) was, the clause lacks an interrogative C° that would supply it with a question meaning.

○ (34)b is blocked because the embedded clause is assigned a question meaning, but the embedding predicate cannot combine with a question meaning in the semantic component. Thus, once again, the derivation is syntactically well-formed, but fails to deliver a meaningful interpretation.

3.2.3. AN ALTERNATIVE/EXTENSION

Is it also possible to define a feature checking analysis for (34)? The answer is at least partially positive: yes for (34)b, no for (34)a. The required ingredients are specified below:

(36) ASSUMPTIONS

a. Non-declarative verbs bear an uninterpretable [u-D] (for declarative) feature
b. C° comes in two varieties: C°[u-wh,Q] and C°[D]

As a consequence, the numerations underlying both sentences in (34) must contain C°[D] for the embedded C° in order to eliminate [u-D] on claim.
The new assumptions trace the ungrammaticality of (34)b now to a syntactic reason in terms of an economy condition. As shown by (37)a, the wh-phrase has moved without there being a trigger for movement, in violation of Last Resort. The alternative derivation (37)b, which includes an interrogative C°, provides a trigger for movement but fails to check the [u-D]-feature:

\[(37)\]
\[
\begin{align*}
\text{a. She claimed}_{[\text{CP, [D]}]} & \text{ C°}_{[\text{D}]} \text{ What Bill bought} & \mathcal{X} \text{ Last Resort} \\
\text{b. She claimed}_{[\text{CP, [Q]}]} & \text{ C°}_{[\text{u-\text{wh}, Q}]} \text{ what Bill bought} & \mathcal{X} \text{ [u-D] unchecked}
\end{align*}
\]

For (34)a, nothing changes and the analysis in terms of interpretability remains unaffected.

4. REVISING THE ANALYSIS

● Assumptions adopted up to now:
  o Uninterpretable features are illegitimate objects at the interfaces (PF and LF).
  o Uninterpretable features can be eliminated by checking with a matching feature in designated syntactic relations (spec-head, head-complement, head adjunction), triggering overt movement.
  o Greed: categories move only if movement satisfies needs of the moved element.

  Motivation: Case Freezing effect in (21). Greed blocks raising of John because John has already checked all its uninterpretable features in lower, finite clause.

\[(21)\]
\[
\begin{align*}
\text{a. TP John}_{[\text{EPP, NOM}]} \text{ seems John}_{[\text{EPP, NOM}]} \text{ left} \\
\text{b. TP}_{[\text{EPP, NOM}]} \text{ seems John}_{[\text{EPP, NOM}]} \text{ left} & \text{ Greed prevents John from checking EPP-feature of higher T}
\end{align*}
\]

o Nominative Case on T is interpretable.

  Motivation: none so far

● A problem for the analysis: Why do wh-phrases move in (English) questions, given that their wh-feature is interpretable, hence does not have to be checked? This type of movement violates Greed!

\[(38)\]
\[
\begin{align*}
\text{a. CP, [Q]} & \text{ C°}_{[\text{u-wh, Q}]} \text{ you read what}_{[\text{wh}]} \\
\text{b. CP, [Q]} & \text{ what}_{[\text{wh}]} \text{ C°}_{[\text{u-wh, Q}]} \text{ you read t}_i \\
\end{align*}
\]

Violation of Greed
Revised assumptions:

- Nominative Case on T is *uninterpretable* (not interpretable, as was assumed so far)
- A category α moves to establish a checking relation with a probe β only if movement leads to checking of features on α or on β → **ENLIGHTENED SELF-INTEREST**.
- Two of the explicit instructions of how checking and valuation work in (18), listed below, can be eliminated. Checking now simply values features, and renders them invisible, irrespective whether they are on the Goal or on the Probe.

\[\begin{align*}
\text{(18) } & \quad \text{a. Uninterpretable features do not have a value} \\
& \quad \quad (\Rightarrow \text{invalidated by assumption that } T^\circ \text{ bears uninterpretable but valued NOM}) \\
& \quad \text{b. The probe bears uninterpretable features} \\
& \quad \quad (\Rightarrow \text{possibly true, but no longer required due to Enlightened Self-Interest})
\end{align*}\]

**ANALYSIS OF WH-MOVEMENT**

The wh-phrase moves to satisfy featural requirements of C°, which is now permitted by Enlightened Self-Interest.

**ANALYSIS OF CASE FREEZING**

- Observe to begin with that a DP can check more than a single EPP-feature, as shown by the option of successive cyclic raising:

\[\begin{align*}
\text{(39) } & \quad [\text{TP } \text{John} \left[ u-\text{EPP} \right] \text{ happened } [\text{TP } \text{t} \left[ u-\text{EPP} \right] \text{ to seem } [\text{TP } \text{t} \left[ u-\text{EPP} \right] \text{ to } [\_ \text{t}, \text{be sick}]]]]
\end{align*}\]

- However, in such contexts, Case is only checked in a single position, the highest one. More precisely, the DP checks the *uninterpretable* NOM feature on T, and NOM values the *uninterpretable* Case feature of the DP.

\[\begin{align*}
\text{(40) } & \quad a. \quad [\text{TP } \text{John} \left[ u-\gamma \text{Case} \right] \left[ T \left[ u-\text{EPP}, u-\text{NOM} \right] \right] \text{ happened } [\text{TP } \text{t} \left[ T \left[ u-\text{EPP} \right] \right] \text{ to seem } [\text{TP } \text{t} \left[ T \left[ u-\text{EPP} \right] \right] \text{ to } [\_ \text{t}, \text{be sick}]]]] \\
& \quad b. \quad [\text{TP } \text{John} \left[ u-\gamma \text{Case} \right] \left[ T \left[ u-\text{EPP-NOM} \right] \right] \text{ happened } [\text{TP } \text{t} \left[ T \left[ u-\text{EPP-NOM} \right] \right] \text{ to seem } [\text{TP } \text{t} \left[ T \left[ u-\text{EPP-NOM} \right] \right] \text{ to } [\_ \text{t}, \text{be sick}]]]]
\end{align*}\]

- Checking in (40) proceeds symmetrically, α checks features of β and β checks features of α. Such symmetric relations are expected to exist, because nothing that has been said to far prevents them. It just happened to be the case that so far, all the instances of checking only went into one direction.

- Recall that only finite T has a nominative case feature. Thus, there is no Case checking relation in lower domains of (40).

- While a single DP can check EPP more than once, each DP can only check a *single* Case feature (this is what Case Freezing is about):

\[\begin{align*}
\text{(21) } & \quad *\text{John seems } [\_ \text{CP that } \text{t left}]
\end{align*}\]
Interestingly, it does not seem to be possible to reduce this at first sight innocent assumption to independent principles. In particular, it does not seem to follow from economy. Applying valuation once is - just as any operation - more costly than foregoing valuation, but required by convergence. Thus, even though double valuation appears to be more costly than applying valuation only once, double valuation should be licit if it renders a derivation convergent.

The restriction to uninterpretable features is necessary in order to permit an interpretable [wh]-features to check more than one [u]-wh]-feature on C° in contexts of successive cyclic movement.

On the revised set of assumptions, this follows because once an uninterpretable feature has been checked and assigned a value, it cannot be checked and valued a second time. Thus, the [u-NOM] on the higher T-node remains unchecked (see (41)d), causing a violation of Full Interpretation at the interfaces (for discussion of a simplified version without valuation see Hornstein et. al 2005: 296f):

(41) a. [TP John, [u-γ Case] T1[u-EPP, u-NOM] t, left] Move John  
   b. [TP Johni [u-NOM] T1[u-EPP, u-NOM] t, left] Checking EPP; checking NOM on T1 and checking and valuation of Case on John  
   c. [TP Johni [u-NOM] T2[u-EPP, u-NOM] seems that [TP T1[u-EPP, u-NOM] t, left] Move John to check EPP on T2 (possible due to Enlightened Self Interest)  
   d. [TP Johni [u-NOM] T2[u-EPP, u-NOM] seems that [TP T1[u-EPP, u-NOM] t, left] ✗ [u-NOM] on T2 unchecked

Note that the analysis contains two hidden assumptions that cannot be reduced to other principles. (Thus, the present account will presumably be superseded by a more profound analysis some time in the future.)

○ First, it must be assumed that a feature cannot be valued twice.  

(42) ASSUMPTION I: A feature F can be valued only once in course of a derivation.  
   ⇔ Consequence for (41): [u-NOM] on John cannot be valued by [u-NOM] on T2

○ Second, a valued uninterpretable feature cannot enter into further checking relations. This intuition is made explicit in (43):

(43) ASSUMPTION II: An uninterpretable feature F₁ can check a feature F₂ only if F₁ has been valued by F₂.  
   ⇔ Consequence for (41): [u-NOM] on John cannot check [u-NOM] on T2 because the two features do not enter into a valuation relation.

---

2Interestingly, it does not seem to be possible to reduce this at first sight innocent assumption to independent principles. In particular, it does not seem to follow from economy. Applying valuation once is - just as any operation - more costly than foregoing valuation, but required by convergence. Thus, even though double valuation appears to be more costly than applying valuation only once, double valuation should be licit if it renders a derivation convergent.

3The restriction to uninterpretable features is necessary in order to permit an interpretable [wh]-features to check more than one [u-wh]-feature on C° in contexts of successive cyclic movement.
A FURTHER ISSUE RELATED TO UNINTERPRETABLE FEATURES

How can uninterpretable features have any effect on the derivation after they have been deleted? Shouldn’t the information stored in the features (e.g. 3rd sg. NOM for *she*) be preserved so that it can be used by morphology?

→ Checked and valued uninterpretable features become invisible to the LF/PF, but they are not gone (Metaphor: in a room with the lights switched off (= LF), an offensive picture will be invisible, but it will not have been ceased to exist.)

○ The values of these checked uninterpretable features must be accessible to morphology.
 ○ These features are still in the syntactic component, where their presence can be detected in locality effects (they trigger minimality violations; see Handout VI on XP-movement)

5. SUMMARY OF ASSUMPTIONS ABOUT FEATURE CHECKING

(44) **ENLIGHTENED SELF-INTEREST**
A category α moves to establish a checking relation with a probe β only if movement leads to checking of features on α or on β

(45) **FEATURE CHECKING AND VALUATION**

a. Feature checking involves a **PROBE** and a **GOAL**.
b. Prior to movement, the probe c-commands the goal.
c. Features can only be checked in a **CHECKING RELATION** (see (13)).
d. Checking **VALUES** uninterpretable features.
e. Checking **RENDERS** the checked feature **INVISIBLE** for the interfaces.

(13) Two nodes α and β can enter into a checking relation if and only if a. and b. hold.

a. **C-COMMAND CLAUSE**
   α c-commands β

b. **INTERVENTION CLAUSE**
   There is no other node γ that c-commands one member of the set {α, β} and is c-commanded by the other member of this set.
(46) **FEATURE SPECIFICATION OF SOME CATEGORIES:**

a. \( T^\circ: \)
   - Uninterpretable EPP
   - Uninterpretable \( \Phi \) features
   - Uninterpretable Nominative Case (NOM) only if \( T^\circ \) is finite

b. \( DP: \)
   - Interpretable \( \Phi \) features
   - Uninterpretable Case-features (NOM or ACC)

c. \( \text{wh-phrase (for argument wh-phrases: add DP-features):} \)
   - Interpretable wh-feature

d. Interrogative \( C^\circ: \)
   - Uninterpretable wh-feature
   - Interpretable Q-feature

e. Declarative \( C^\circ: \)
   - Interpretable D-feature

f. Interrogative embedding predicate:
   - Uninterpretable Q-feature

g. Declarative embedding predicate:
   - Uninterpretable D-feature

**REFERENCES**


