Interpreting movement across phase boundaries

Background: Moved elements exhibit reconstruction effects, i.e. behavior that we would expect if they had not moved. These effects are standardly considered to be the result of syntactic reconstruction, whereby the moved element is effectively “placed back” in its launching site at LF, e.g. via selective copy interpretation (1). If movement does not reconstruct, then it maps onto a λ-bound variable (2) (possibly with more internal structure, as with Trace Conversion), which we will refer to here as a trace.

(1) DP₁ ... Op ... DP₁ ... \((Op \gg DP)\) \hspace{1cm} (2) DP λx ... Op ... x ... \((DP \gg Op)\)

Problem: Under this standard analysis, the semantics needs access to both the higher and lower copies in order to determine how the lower copy is to be interpreted. Should it be interpreted as-is, because the movement reconstructs? Or should it be interpreted as a trace? If the latter, the variable in the lower copy must match the λ-operator associated with the higher copy—an inherently nonlocal relation. Moreover, should the lower copy be interpreted as part of a movement dependency in the first place? That is, is the lower DP a copy or not? The nonlocality inherent in these interpretative procedures is at odds with a phase model of syntax, wherein syntactic structure reaches the interfaces in small chunks: the complements of phase heads are shipped off to the interfaces cyclically (Chomsky 2000, 2001). The problem is that movement from within a phase complement to the phase edge (either criterial or intermediate) results in an interface chunk containing the lower copy, but not the higher copy. As such, movement (and in turn the structure at large) cannot be interpreted on a phase-by-phase basis under standard Minimalist assumptions, because the higher copy necessary for the interpretation is not part of the chunk that reaches LF. To illustrate, consider (3), where there is an intermediate step of movement, amplifying the problem.

(3) DP₁ ... [phase-comp ... DP₁ ... [phase-comp ... DP₁ ...] ...]

In (3), the lowest DP cannot be interpreted without the middle DP, which in turn cannot be interpreted without the highest DP. Furthermore, if the lower movement step is interpreted as reconstructing and the higher movement step as a trace, then the overall chain will not produce the intended interpretation (effectively interpreting DP₁ in full in two positions). Consequently, structures with movement cannot be interpreted until the entire structure has been constructed, undermining the purpose of phase theory.

Claim: The complication that phases pose for interpreting movement is overcome if we adopt a syntax where “movement” involves multidominant structures built using Parallel Merge, specifically the system developed in Johnson (2012, 2014) and extended in Fox & Johnson (2016), O’Brien (2017), and Poole (2017). We argue that this multidominant syntax for movement addresses the problem outlined above, and then show that it resolves other difficulties that the standard analysis for interpreting movement faces.

Proposal: We adopt Johnson’s multidominant syntax in which wh-movement and QR are representationally distinct. Wh-movement involves the wh-phrase parallel-merging with its base position and with a Q-particle (in the sense of Cable 2010, where Q merges with the wh-phrase and projects a QP). The resulting QP is then merged higher in the main structure, e.g. [Spec, CP], as schematized in (4). QR involves two DPs sharing an NP—the NP parallel-merging with the two D heads—where the lower DP is a bound definite description and the higher DP is the quantificational nominal, as schematized in (5). This is a syntactic implementation of Fox’s (2002) Trace Conversion, removing the need for Fox’s unrestricted operation that transforms DPs into bound definites at LF. An independent upshot of this system is that there is no Internal Merge—in the sense that there is no merging of an element with an element that dominates it. (Note that (6) is discussed shortly below.)

(4) QP
  \hspace{1cm} Q
  \hspace{1cm} XP

(5) DP₁
  \hspace{1cm} D
  \hspace{1cm} NP
  \hspace{1cm} THE₁

(6) QP
  \hspace{1cm} Q
  \hspace{1cm} DP
  \hspace{1cm} D
  \hspace{1cm} NP
  \hspace{1cm} THE₁
We propose that the structures in (4) and (5) are the general building blocks for displacement; we refer to this as the \emph{displacement effect} documented in the literature.

The spatial order of siblings in (4–5) does not represent a particular linearization. For reasons of space, the specifics of linearization are not discussed here, but the algorithm involves linearizing \emph{paths} (Johnson 2014), allows (4–5) to manifest as displacement (or not), and spells out the \emph{the} in (5) contingent on the higher D (not necessarily as \emph{the} or as covert). Important for present purposes is how (4–5) are interpreted. In (4), XP introduces focus alternatives, and Q catches those alternatives by returning the focus value of its argument as an ordinary value, yielding a question meaning (7) (Beck 2006; Cable 2010; Kotek 2014). Crucially, XP is only interpreted in its base position; Q does not semantically combine with XP. Thus, the interpretation of (4) corresponds to a reconstructed derivation, because XP is effectively interpreted as if it had not moved. (5) is interpreted like Trace Conversion: the lower DP is bound by a \emph{λ}-operator merged immediately below the landing site, as sketched in (8). The interpretation of (5) thus corresponds to a trace derivation. Note the structures are “flattened” for space, and strikethrough indicates not-interpreted.

\begin{align*}
(7) & \quad a. \quad [Q \alpha]^o = [\alpha]^f \\
& \quad b. \quad [\text{which cat}]^o \text{ is undefined}
\end{align*}

\begin{align*}
(8) & \quad a. \quad [\text{the}]^o = \lambda y f . \lambda x (f(x) \land x = y) \\
& \quad b. \quad [[D \text{ NP }] \lambda_1 \ldots [\text{the}] \text{ NP}]^o = [D \text{ NP}]^o [\lambda_1 \ldots \lambda x ([[\text{NP}^o](x) \land x = g(1)) \ldots ]]
\end{align*}

We propose that the structures in (4) and (5) are the general building blocks for displacement; we refer to them respectively as \emph{QP-mvt} and \emph{DP-mvt} for short. Movement dependencies are built out of sequences of QP-mvt and DP-mvt. Crucially, DP-mvt may precede QP-mvt, as schematized in (6), which has the effect of the displaced element taking scope in the position of the highest DP. In other words, the equivalent of QR (i.e. DP-mvt) is always involved in shifting the scope of a moved DP. The linearization algorithm assigns (6) a linearization equivalent to that of (4). The DP-mvt→QP-mvt sequence allows us to account for movement that optionally shifts scope (i.e. optionally reconstructs): the movement uniformly targets aQP, but that QP may have involved DP-mvt first. The licit and illicit sequences of DP-mvt and QP-mvt derive from the semantics in (7–8); the other combinations are not discussed here for reasons of space.

\begin{itemize}
\item \textbf{Consequences:} In both DP-mvt and QP-mvt, how the tail of the pair is interpreted is entirely independent of the head of the pair. The semantics can thus interpret the tail directly, based solely on the semantics of the elements involved. Assuming that what gets shipped to the interfaces is what the phase complement dominates (as with standard copy theory), when there is a phase boundary between the head and tail of a movement step, all of the information needed to interpret the tail will crucially be contained in the phase complement. Thus, under our proposal, movement dependencies can always be interpreted on a phase-by-phase basis. Our proposal has several other upshots: (i) Genuinely optional processes in the mapping from syntax to semantics are not possible because this mapping is a \emph{function} (following Montague). For any given syntactic representation, the denotation function \([\cdot] \) must always return the same semantic value. This is a problem under the standard analysis, where reconstruction is an optional LF process (see above). Under our proposal, because reconstructed and nonreconstructed movement correspond to distinct structures, no optional process is needed. \textit{Every syntactic structure has exactly one interpretation}. (ii) Reconstruction is not universally available for every movement type. For the standard analysis, this requires that LF be able to “see” syntactic movement types. Under our proposal, different movement types involve different syntactic structures, and LF can thus operate without direct reference to movement types. For example, \begin{itemize}
\item with scope reconstruction: movement that optionally reconstructs (e.g. \emph{wh}-movement) is ambiguous between just QP-mvt or DP-mvt followed by QP-mvt; movement that obligatorily reconstructs (e.g. Japanese long scrambling; Bošković & Takahashi 1998) involves only QP-mvt; and movement that never reconstructs (e.g. QR) involves only DP-mvt. (iii) Reconstruction effects do not always travel together, e.g. movement that reconstructs for scope does not necessarily reconstruct for Condition C (e.g. Sharvit 1998). Under our proposal, a limited form of scope reconstruction can be achieved by allowing the lower D in DP-mvt to be interpreted as a generalized-quantifier version of \textit{the}, rather than an entity version (so-called “semantic reconstruction”; e.g. Cresti 1995; Rullmann 1995). We show that this can capture the correlations between reconstruction effects documented in the literature.
\end{itemize}