EXPLAINING ELLIPSIS IDENTITY

GREGORY M. KOBELE
INRIA FUTURS, LABRI

A modern statement about the structural relationship between an ellipsis site and its antecedent runs as follows.

> an elided XP must have a syntactically identical antecedent XP', modulo traces

This is quite a bizzare condition. Why should some parts of structure count for being identical, and others not? Of all the possible relations between two structures, why does this one describe the ellipsis data so well, and not another, equally (or less!) complex one?

There are in fact *two* structures minimalist syntacticians associate with an expression: its derivation, and the familiar derived tree. Assuming that it is the derivation itself, and not the derived tree, which is the proper locus of the identity condition on ellipsis, we can simplify our situation considerably: ellipsis identity is *actual* identity. The condition given above was so complex because we were working with the wrong structure. Thus, the syntactic identity condition on ellipsis is now the following.

> A phrase XP may be elided iff it was derived in exactly the same way as its antecedent.

Unlike the derived tree, which can be deformed in arbitrary ways, the derivation can be changed only by changing the way we build up the expression itself! Thus, apparent 'mismatches' in structure between ellipsis site and antecedent put strong constraints on any theory of syntax, and therefore become particularly fruitful loci of investigation.

Here, I explore two such mismatches. Given the grammaticality of sentences like 1, we must assume that the two VPs are built up identically—a fact superficially at odds with basic ideas of passive structures. Behind this problem lies the A, A-bar distinction, and a solution to it extends immediately to the second mismatch, antecedent contained deletion, as exemplified by 2. Here we must assume that, contrary to appearances, the antecedent is not built atop the ellipsis site.

(1) Hans [$_{\text{VP}}$ wants to be loved], and Frans does [$_{\text{VP}}$ $\emptyset$], too.
(2) Ivan will cook every chicken that Katya will.