Juan Uriagereka
juan@wam.umd.edu
Department of Linguistics, University of Maryland.

Reprojection: An Argument for Derivations

(Talk to be presented at GSAS lecture series, Department of Linguistics, Harvard University 03/24/00)

ABSTRACT

It is often hard to decide whether a given proposal must be expressed in derivational or representational terms, or rather either expression is possible. Arguably the best kind of argument for a derivational system is one involving representations that are destroyed as the derivation unfolds, and do not make it to the final representation in any direct form. This 'loss of information' situation is hard to obtain in models -such as the classical Principles and Parameters system- which make use of highly enriched representational codings (including traces, indices, and similar elements). Intuitively, whatever information may have been lost in the course of the derivation can be reconstructed through some abstract coding. Interestingly, the Minimalist Program drastically reduces this notational artifact, for intriguing methodological and even ontological reasons relating to grammatical 'good design'.

The present study can be seen in this light, as a minimalist exercise on whether one particular case study that we present is an instance of a 'loss of information' situation. We argue that it is, and rather than being of negative consequence, the lost information has good semantic and syntactic properties. So far as we can see our results can be replicated in representational terms only at the cost of much added, otherwise unnecessary notational machinery.

The basic idea concerns the labeling mechanism in (1):

(1) X <=> Y

When X merges to Y, obtaining the construct {X, Y}, what is the label of the expression? We must emphasize that by 'label' we simply understand the type of the expression in question, assuming this is important. Following Chomsky (1995) we may code that label as in:

(2) { L {X, Y}}

Where L is the label. This is the only substantive issue: is L identical to X or to Y? (Chomsky (1995) argues that among the simplest, set-theoretic relations between X and Y that one could consider to determine L's properties, identity is the only one that does not create formal problems.) The options are thus:

```
(3) a.   X   (Formally: {X, {X, Y}}   b.     Y (Formally: {Y, {X, Y}})
        /  \                               /  \
      X     Y                             X     Y
```

Since Chomsky (1998) it is assumed that what determines the choice in (3) are the properties of X and Y. For instance, if X is a verb and Y is a DP, then X will project. If in contrast X is a DP and Y is a v', then v will project. Collins (1999) has suggested that, if appropriately extended, this actually deduces the labeling operation proper. Suppose this is true in some form; labels are in no sense primitive objects, although of course they may still be real to the system (just as, say, molecules are not primitive, yet they are real at some level).

The question that intrigues us is this: could it be that during the course of the derivation we actually change from an object like (3a) to an object like (3b)? Call this a reprojection, which can be characterized as in (4):

(4) If X projects upon taking Y as a dependent (i.e., whenever X is a head), and in the course of the derivation Y projects (thus 'turning' X to a maximal projection and making X a head) we say that Y has been 'reprojected'.

If this is possible, and (3b) is the only object that reaches the interpretive component that LF bleeds into, then we would have a nice argument for a derivational system, since the only way a representational system could reproduce the fact that (3b) was 'once' (3a) is in terms of some notational trick that enriches the system in an undesirable way.

Chomsky (1995) presupposes two reasons that prevent a class of reprojections involving heads and their specifier (the only type we will be concerned with here). Assuming specifiers are obtained by movement, then reprojected specifiers will involve either improper or non-uniform chains:

```
(5) a.  Y                          b.        XPi
       /  \                                 /  \
    XPi    Y'    ====REPROJECTION====>      X'i   YP
          /  \                                   /   \
         Y ... ti ...                           Y ... ti ...
```

Clearly, the chain (XPi, ti) in (5a) is tampered with after the reprojection in (5b). If the intended chain is still (XPi, ti), then it will be improper (the first link doesn't c-command the last); if the chain in question is now (X'i, ti), then it will be non-uniform (the first link is not a maximal projection, while the last one is). At the same time, a second reason why the reprojection in (5) is undesirable is that it modifies checking configurations: whereas XP in (5a) is in the checking domain of Y, in (5b) YP is in the checking domain of X.

Those concerns are representational. It could be that the checking domain of Y (vis-a-vis XP) is used only at some derivational time Dt, while at some later derivational time Dt', instead, the checking domain of X (vis-a-vis YP) is what is relevant. Our work argues that if this is indeed the case, some peculiar syntactic and semantic properties of binary quantifiers can be naturally explained, thus arguing for reprojections and with them for a derivational system.

The gist of the idea comes from an observation in Larson and Segal (1995) that binary quantifiers like 'most' behave much like transitive verbs in that they take ordered arguments (thematic arguments in the verbal case, and a restriction and a scope in the case of a quantifer). If this is so, and provided that verbs 'take' their arguments in familiar syntactic ways (within verbal projections), it is reasonable to ask whether quantifiers too 'take' their arguments in familiar syntactic ways. The problem is easily seen in a sentence like (6):

(6) Most people love children.

Whereas the argument 'people' is in a standard syntactic relation with regards to the quantifier 'most' (as its complement), the argument 'love children' is in no serious sense a dependent of this quantifier; quite the opposite is the case: syntactically, 'most children' is the dependent (specifier) of the relevant IP, within customary assumptions.

But now suppose reprojections are possible. Then nothing prevents the quantifier 'most' from handling its argumental requirements after the Case and agreement requirements of IP have been met. At that stage in the derivation, (7b) is what we would have obtained from (7a):

```
(7) a.     IP                    b.          QP
          /  \                              /   \
       QPi     I'    ====REPROJECTION===>   Q'i     IP
      /   \  /  \                     /   \   /   \
   most ...  I ... ti ...             most ...   I ... ti ...
```

Aside from directly addressing a serious question about the syntax/semantics of binary quantification, the present proposal has a consequence that is really worth exploring. To put it basically: what was a 'right branch' in (7a) suddenly becomes a 'left branch' in (7b). This may relate to a very complex set of facts involving so-called 'quantifier-induced' islands, known at least since Linebarger (1980) (see

Den Diken and Szabolcsi (1999) for a recent summary of the literature). Basically put, whenever we try to establish an LF relation of grammar between positions X and Y in (8) across a binary QP, an ungrammaticality ensues, whereas the same type of relation of grammar is possible if it takes place in overt syntax:

(9) ... [X ... [QP ... [Y...]...]...] ...
Z

One can always code this fact as some kind of requirement on the relevant relations, but needless to say the issue is why such a requirement should hold. From the point of view of reprojections the expectation is that the process in question should in effect produce an island which was not there to begin with. It is hard to see how this could be replicated in representational terms.

The actual mechanism that we propose to induce the quantifier induced island has to do with the fact that the chain of QP in (7) must be uniform, and as we saw chains whose first link reproject are no longer uniform. In a system making use of levels of representation this is a reason to either reject uniformity or reprojection. However, in a radically derivational system of the sort argued for by Epstein and Seely (1999) or Uriagereka (1999), all this means is that chains must be 'cashed out' right before they become non-uniform. Suppose we do that, then, prior to reprojection. How does this affect the resulting structure?

It does rather drastically, assuming a chain is 'cashed out' by the system in an optimal fashion: by sending to the interpretive component the minimal amount of structure that contains all links of that chain. In the instances that are to reproject, that means that very structure. For instance, IP in (7a). The issue then is what happens to the constituent parts of IP. Inasmuch as IP is essentially gone from computation, those constituents should be inaccessible for further computation from higher positions. For instance, if IP is Z in (9), then a relation of grammar between X and Y would become impossible; the same type of relation would have been fine prior to the chain of QP being 'cashed out' prior to reprojection. As we implied, only binary quantifiers induce these LF islands. Nothing in the logic of our proposal forces a unary quantifier to reproject, hence its chain is never at risk of not being seen by the system as uniform, thus is in no need of an early 'cashing out'. If so no islands should emerge for them, as is the case.

The paper discusses other instances of possible reprojections, including those involving negation, and others that plausibly prevent them (because 'segments' of an adjunction are involved in the phrase-marker that undergoes reprojection). The latter provides an interesting account of seemingly unrelated facts: why binary quantifiers never incorporate and why there is a 'definiteness effect' in expletive-associate pairs (why associates cannot be binary quantifiers).

Reprojection is not an operation, just a property of derivations strictly considered as having different demands (theta and Case/agreement relations, quantificational relations) as the derivation unfolds. Although labels are real in reprojection instances (something that changes exists), they are crucially not primitive. It remains to be seen whether the analysis is replicable in representational terms, but prima facie this is very hard to imagine.