

1. The Status of the Standard Definition of Principle C

(1) An R-expression cannot be c-commanded by a coindexed category

Condition C as formulated in (1) is a primitive, for which neither Government-Binding Theory nor minimalist approaches provide intrinsic reasons. Furthermore, it violates the Inclusiveness Condition, which implies that indexes are not objects that narrow syntax deals with.

2. Probe-Goal Relation

A Probe is a lexical item with an *unvalued* feature that searches a Goal within its sister node. A Goal is a category whose intrinsic features can value those of the Probe. The prototypical case of Probe-Goal relation is ϕ -features in T that are valued (“on distance”) by the ϕ -features in D/N.

3. Labelling Algorithms (modified from “On Phases”)

(2) In $\{H, \alpha\}$, if H is a lexical item *and a Probe*, H is the label

(3) If α is internally merged to β , forming $\{\alpha, \beta\}$, then the label of β is the label of $\{\alpha, \beta\}$

(the italicized part in (2) is not present in the “On Phases” manuscript)

(2) should ensure that the head projects when it is merged with its complement, while (3) ensures that X' transmits its label to XP when it is merged with a specifier.

4. Some Consequences of the Labelling Algorithms

In a few cases (small clauses, Merge of the subject DP in Spec, ν and merge of two lexical items) algorithms (2) and (3) are not sufficient to capture the relevant facts (see “On Phases” for some speculations on these cases). A configuration that might not be captured due to the italicized part in (2) is merge of a lexical item with its XP complement, say, {read, {the, book}} in which “read” should project. This case is captured if the search within the *the remaining part of the* Numeration is a case of probing. An interesting case of interaction between (2) and (3) is free relatives (cf. Donati's 2006 analysis). Chomsky (2005) elliptically notes that another case of interaction between (2) and (3) might be Principle C. I will try to elaborate on this observation by Chomsky by making it fully explicit.

5. Referential Evaluation

(4) A referentially values B if the semantic component receives an instruction from narrow syntax which forces it to disregard all the assignment functions that do *not* assign the individual that is the intrinsic semantic value of A to B.

(A has an intrinsic semantic value if and only if its semantic value is independent from the function that assigns a value to free variables)

For example, “John” referentially values “he” if the semantic interpretation is restricted to those assignment functions that assign the individual that is the denotation of “John” to “he”.

6. Principle C Reduced to the Labelling Algorithms

I will assume that referential evaluation is a case of Probe-Goal matching, in which the Goal (a referential expression) values the Probe (a pronominal expression). So, the notion of 'Probe' is widened. Probe-Goal matching, in addition to phi-features (or *wh*-features) evaluation, also involves referential evaluation.

PROVISO: under the previously spelled out set of definitions, it is the c-commanded category (the Goal) that referentially values the c-commanding category (the Probe).

Take a standard case of Principle C violation like (5), in which the co-indexed categories have the same semantic value (co-indexing is a notational device for the reader's convenience with no theoretical import, since I will assume that narrow syntax does not contain referential indexes):

(5) *He_i likes John_i

If referential evaluation is a case of Probe-Goal match, the algorithm (2) determines that the pronominal SPEC “he” becomes a label when it is merged to T' because “he” is a lexical item and is a Probe. This makes (5) uninterpretable under the relevant reading, because it would receive a nominal label. Given this “wrong” label, (5) cannot be interpreted as a clausal constituent under the plausible assumption that CPs, and possibly TPs, but not DPs, have the semantic type of sentences.

PROVISO: in principle, the illicit Probe-Goal relation between “he” and “John” might take place also when “he” is merged with $\{v, \{\text{likes, John}\}\}$. If this happens, mislabelling takes place at this early stage of the derivation, because the category that should receive the vP label, receives the DP label instead.

If “he” and “John” are disjoint in interpretation (cf. 6), the algorithm in (2) does *not* require that the pronominal SPEC “he” become a label, because “he” is a lexical item but is *not* a Probe. The algorithm in (3) dictates that, since the pronominal SPEC is internally merged to T', the label of T' project.

(6) He_i likes John_j

The acceptability of (6) indicates that a pronoun is not forced to probe its sister node to get referentially valued. This introduces a difference with other cases of Probe-Goal relations but it is easy to see what is the basis for this difference: if a pronoun is not referentially valued in narrow syntax, the semantic component can still assign a value to it, via the assignment functions to free variables.

7. The Labelling Algorithms do not Extend Inappropriately

If “he” and “his” were in a Probe-Goal relation in (7), the reading in which “he” and “his” have the same semantic value should be ruled out by the same reasoning that rules (5) out:

(7) He likes his friends

However, “he” and “his” are not in such relation, because the Probe-Goal relation is asymmetric. In every case of Probe-Goal match, intrinsic features of the Goal value those of the Probe. A referential expression like a proper name has intrinsic referential features (it has an assignment independent semantic value), while a pronoun does not (it has an assignment dependent semantic value). So, “John” can be a Goal in (5) but “his” cannot be a Goal in (7). Therefore, mislabelling does not occur in (7).

8. When the Standard Definition of Principle C and the Definition in Terms of Mislabelling Diverge

- The idea that ultimately Principle C is a case of mislabelling and more canonical formulations of Principle C *might* make different predictions in cases like (8), depending on which analysis of DP structure in English is adopted:

(8) *His_i picture of John_j

- Cases like (9) are not excluded by the same explanation that excludes cases like (5).

(9) * [The president]_i likes [the president]_j

- The canonical formulation of Principle C makes an embarrassingly wrong prediction in identity sentences like (10). (10) should be a patent violation of Principle C. In fact, it is perfectly OK. In the next paragraph, I show that the approach to Principle C in terms of mislabelling fares better.

(10) He_i is John_i

9. When the Definition in Terms of Mislabelling Fares Better: Identity Sentences Avoid Mislabelling

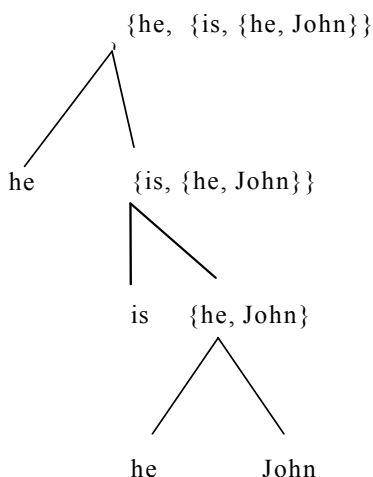
Sentences like (10) are conveniently ignored in many discussions about Principle C (but see Moro 1997 for discussion of a related case), probably because it is assumed that they are in the same boat with “accidental coreference” cases like (11). Heim and Kratzer (1998): 269-274 explicitly endorse this position.

(11) Everyone likes John. Bill likes John, Mary likes John, Robert likes John. He_i likes John_i, too.

However, it is dubious that (10) and (11) exemplify the same phenomenon. We have to set-up rather special discourse contexts to bring out the judgements that coreference is possible in (11) and similar sentences. No special discourse contexts is required to make the same reading perspicuous in (10).

Assuming the definition of Principle C in terms of mislabelling, the following derivation can explain the absence of Principle C effects in (10). When “he” and “John” are first merged, “he” can act as a Probe and search its sister node for a category that can value its *unvalued* feature. This category is the sister node “John”. Given (2), the label of {he, John} is contributed by “he”, namely, {he, John} is a DP. Assuming that “he” raises to Spec,T, (10) has the following structure:

(12)



- (i) Label of {he, John} = label of “he” (by algorithm (2)) = D
- (ii) Label of {is, {he, John}} = label of “is” (by algorithm (2)) = T
- (iii) Label of {he, {is, {he, John}}} = label of “is” (by algorithm (3)) = T

The crucial step is (iii). The algorithm (2) does not force “he” to project, unlike what happens in (5). The reason is that “he” cannot act as a Probe in Spec,T in (10), because its unvalued feature has already been valued in its base position. So, the copy of “he” in Spec,T is a lexical item but is not a Probe. The algorithm (2) does not apply, while the algorithm (1) correctly dictates that the root gets a T label. Assuming that the trace of “he” in its base position is not interpreted, the LF structure of identity sentences is [DP₁ is DP₂]. See Partee (1987) and Rothstein (2001) for a semantic analysis compatible with this LF structure.

The present approach has the potential to explain why “he” and “John” cannot have the same semantic value in (13). The semantics of copular sentences forces “he” to be a Probe and “he” finds its Probe in the category “the friend of John”.

(13) *He_i is [the friend of John_i]_j

The fact that “he” and “John” cannot have the same semantic value can be reduced to an intervention effect (Probing cannot take place inside a category that is a suitable Probe). Depending on the technical choices that are adopted, this can be either reduced to Relativized Minimality or to the *i*-within-*i* Filter.

We can also explain the pattern in (14). In order for “he” and “John” to have the same semantic value, “he” has to probe “John”. If this happens when “he” raises to the Spec,T, the familiar mislabeling problem, illustrated by sentence (5), will arise. If “he” probes when it is first merged, a different mislabeling problem will arise, because the category formed when “he” is merged with “envious of John”, namely {he, {envious, {of, John}}}} will get a D label, rather than being an adjectival category.

(14) *He_i is [envious of John_i]_j

10. Strong Crossover

The notion of referential evaluation does not overlap with the semantic notion of binding. A case of semantic binding is in (15):

(15) [Every boy]_i thinks that he_i can win the competition

In (15) the label of [every boy]_i in Spec,T is “every”. This label c-commands and binds a variable inside its sister node. However, this is not a Probe-Goal relation (“he” is *not* a Goal because it has an assignment dependent semantic value). So, the algorithm (2) is irrelevant in the case of (15). Since {every boy} is internally merged with {thinks that he_i can win the competition}, the algorithm (3) correctly dictates that the label of T' projects.

POTENTIAL PROBLEM: Couldn't the illicit reading of (5), which we have excluded as a case of mislabeling *when Probe-Goal matching takes place*, result in some other way? Free co-indexing is impossible, if indexes are not legitimate syntactic objects in compliance to the Inclusiveness Condition. But what about semantic binding?

The binder must c-command the bindee and in (5) this does not happen. However, the proper name in (5) in principle should be able to undergo QR and reach a position from which it c-commands (and binds) the pronoun. This QR application is independently required for cases like (17).

(17) He_i likes [every boy]_i

The impossibility of semantic binding in (5) and (17) is an example of the configuration traditionally called strong crossover. In the GB framework, strong crossover is reduced to Principle C as formulated in (1), but this is not possible anymore if Principle C does not have an independent status.

I propose that the strong crossover configuration is illicit because, after QR, the pronoun in the subject position creates an intervention effect for the chain that links the QR-ed expression and its trace:

(18) * [Every boy]_i he_i likes t_i
Binder intervener bindee

11. A Technical Note on Expletives

The present approach seems incompatible with analyses that take the expletive in Spec,T to be a Probe and T to be the associated Goal (cf. Chomsky 2001):

(19) There seem to be many linguists here

Either the expletives are not Probes (cf. Moro 1997) or they do not have a label. The latter assumption can also explain why they never surface in a position inside the vP shell, since, being label-less, they cannot satisfy thematic and selectional requirements of the verb.

12. Conclusion

Principle C can be reduced to the fundamental principles that govern phrase structure building, under the assumption that mislabelling is not tolerated by syntax-semantics rules that map syntactic categories to semantic types. The neglected case of absence of Principle C effects in identity sentences receives a principled explanation. Given that other authors have proposed similar reduction for Principles A and B (cf. Reuland 2001 and Hornstein 2006, among others), the project becomes feasible of maintaining the empirical coverage of Binding Theory without assuming binding principles as primitives.

REFERENCES

- Chomsky, N. (2001), “Derivation by Phase” in M. Kenstowicz (ed.), *Ken Hale: A Life in Language* 1-52, MIT Press
Chomsky, N. (2005), “On Phases, manuscript, MIT.
Donati, C. (2006), “On Wh-head movement”, in L. Cheng e N. Corver (eds.). *Wh-Movement: Moving on*, MIT Press
Heim, I and A. Kratzer (1998) *Semantics in Generative Grammar*, Oxford. Blackwell.
Hornstein, N. (2006). “Pronouns in a Minimalist Setting”, manuscript, University of Maryland.
Moro, A. (1997), *The Raising of Predicates* Oxford. Oxford University Press.
Partee, B. (1987), “Noun phrase interpretation and type-shifting principles” in J. Groenendijk, D. de Jong, and M. Stokhof, (eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, 115–143. Dordrecht: Foris.
Reuland, E. (2001) “Primitives of Binding”, *Linguistic Inquiry*, Volume 32, Number 3, 439–492.
Rothstein, S. (2001) *Predicates and their Subjects*. Dordrecht: Kluwer.