Phases, strong islands, and computational nesting

In this paper we revisit the connectedness effect (Kayne 1983), whereby an illegitimate gap in a strong island (1a) can be rescued if its dependency path forms a connected tree with that of a legitimate gap, as in (1b).

(1) a. * [Which famous playwright]$_i$ did [close friends of e_i] become famous ?
b. ? [Which famous playwright]$_i$ did [close friends of e_i] admire e_i ?

On our view, Kayne's notion of "g-projection" incorporates the essential insight that legitimate gaps lie on the main recursive branch of the tree, whereas illegitimate gaps lie on "secondary" branches, which do not allow for unlimited recursion. We propose an implementation of this insight within a revised and explicitly formalized minimalist grammar (based on Stabler 1997), where the structure building operations *Merge* and *Move* and the notion *phase* are redefined so as to meet precise computational desiderata (tractability, determinism) and be directly usable by parsing and generation algorithms (see Chesi 2004).

A considerable body of psycholinguistic evidence suggests that the structure building operations work in a top-down fashion (cf. a.o. Phillips 1996). Accordingly, we redefine *Move* as a top-down oriented operation which stores the moved element in a memory buffer and re-merges it at the point of the computation where the element is s-selected. We redefine the *phase* as the minimal part of a top-down computational process in which all the functional and s-selectional specifications associated to a given lexical head are satisfied.

Crucial to our argument is the distinction between *sequential* and *nested* phases. When a phase reaches the lowest position s-selected by its lexical head, it is closed off, and the expansion of the complement constitutes the next sequential phase. Any DP or CP that does not occur in the lowest position s-selected by a lexical head instead initializes a "nested" phase, which must be processed while another phase is still incomplete. The distinction between sequential and nested phases is independently justified by their different effects on the computational complexity function (see Chesi 2004 for thorough discussion).

Each phase has its own local memory buffer for Move. A *success condition* requires that at the end of each phase the local buffer be empty, or else its content be discharged into the memory buffer of the next sequential phase (via re-merge of an intermediate copy into the edge of this phase). Obviously, at the end of the last phase of all the local buffer will have to be empty.

Consider now a computation for (1a). The algorithm postulates a root node (CP), initializing phase 1; then it computes the wh-phrase in a separate nominal phase 2, and stores it in the local memory buffer of phase 1. The computation of phase 1 proceeds, inserting *did* in C. At the next step, a distinct nominal phase 3 for the subject DP must be opened, while the clausal phase 1 is still incomplete. The DP phase 3 is thus a nested phase, and its local memory buffer does not contain the wh-phrase which was stored in the memory buffer of phase 1: hence, the wh-phrase cannot be discharged in the s-selected gap position within the subject DP. The wh-phrase also remains undischarged at the end of the computation; this accounts for the strong island effect. Crucially, the buffer of phase 1 cannot be discharged into that of the subject phase 3, because this is not a sequential, but a nested phase. Suppose however that we optionally allow the memory buffer of the nested DP phase to "copy" (though without discharging it) the buffer of the immediately superordinate phase 1, which contains the wh-phrase: then, the wh-phrase can be discharged in the gap position within the DP phase 3. Note that this step will only empty the local memory buffer of phase 3, but not the buffer of the yet incomplete clausal phase 1. Therefore, phase 1 must contain another gap position in which the wh-phrase can be discharged from its local memory buffer, so as to meet the success condition at the end of the computation. This is indeed the case in (1b), as opposed to (1a): this accounts for the connectedness effect.

In the final part of the paper, we turn to adjunct islands. We assume that all adjuncts are licensed in dedicated positions, which are however not s-selected by the lexical head of the modified phase (cf. Cinque 2002). As for left-hand adjuncts, they are processed before the lowest s-selected position of the modified phase has been reached; hence they constitute nested phases, like preverbal subjects. As for right-hand adjuncts, they are processed after the lowest s-selected position of the modified phase, but since they are not s-selected, the system is forced to backtrack and re-open the modified phase. As a result, the adjunct phase is nested in the re-opened phase**,** and it can at best "copy", but not discharge the buffer of the latter: this gives rise to a strong island effect for extraction (3). Time permitting, we will also discuss double embedding in two nested islands (Kayne 1983), which suspends the connectedness effect (4).

(3) * [$_{CP}$ [$_{DP}$Which famous playwright]$_i$ is Mary upset because [everybody criticized e_i]?

(4) * a person who$_i$ you admire e_i because [[close friends of e_i] became famous]

References

Chesi, C. 2004. *Complexity and determinism in linguistic computation: phases and cartography in phrase structure building*. Ms., University of Siena.

Cinque, G. 1999. *Adverbs and Functional Heads*. New York, Oxford University Press.

Cinque, G. 2002 *Complement and adverbial PPs: implications for clause structure*. Abstract, University of Venice.

Kayne, R. S. 1983. *Connectedness*. Linguistic Inquiry 14.

Phillips, C. 1996. *Order and Structure*. PhD thesis, MIT.

Stabler, E. 1997. *Derivational minimalism*. Ms., UCLA.