University of Pennsylvania Working Papers in Linguistics

Volume 12
Issue 1 *Proceedings of the 29th Annual Penn Linguistics Colloquium*

Article 3

1-1-2006

Phases, Left Branch Islands, and Computational Nesting

Valentina Bianchi
Università di Siena, bianchi10@unisi.it

Cristiano Chesi
cri@mit.edu

Phases, Left Branch Islands, and Computational Nesting

Phases, Left Branch Islands, and Computational Nesting

Valentina Bianchi and Cristiano Chesi*

1 Left Branch Islands and the Connectedness Effect

In this paper we reconsider the connectedness effect discussed by Kayne (1983) and illustrated in the examples (1)-(3). Kayne observed that in VO languages, left branch constituents are strong islands for extraction;[1] however, an illegitimate gap inside a left branch island can be rescued by another gap embedded in a lower right branch constituent. The examples in (1)-(2) illustrate preverbal subject islands, and (3) a small clause subject island: while in the (a) examples, extraction from the left-branch subject is impossible, in the (b) examples, the illegitimate gap is followed by a legitimate gap on a right branch and this creates a grammatical configuration.

(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 ?
(2) a. *Who did [my talking to e_i] bother Hilary?
  b. $^{\surd}$ Who did [my talking to e_i] bother e_i?
(3) a. *Who$_i$ did you consider [friends of e_i] angry at Sandy?
  b. $^{\surd}$ Who$_i$ did you consider[friends of e_i] angry at e_i ?

Kayne (1983) proposed a representational constraint to account for these data, the Connectedness Condition (henceforth CC). The central notion is that of a g-projection, which is defined in (4)-(5). In a VO language like English, every right branch is in a canonical government configuration, by definition (4); the recursive definition in (5) ensures that all the maximal projections dominating a structural governor X and lying on a right branch are g-projections of X.

*We wish to thank Adriana Belletti, Cedric Boekcx, Alec Marantz, Andrew Nevins, Luigi Rizzi, Ur Shlonsky, and Michal Starke for discussion of the material presented in this paper, as well as the audiences at the 29th Penn Linguistic Colloquium (February 2005), the Harvard Grammatical Locality reading group and at the 28th Glow Colloquium (Geneva, March-April 2005).

[1]Throughout the paper, by "strong islands" we mean *nonselective* islands, which do not give rise to argument/adjunct asymmetries in extraction, as opposed to weak (Relativized Minimality) islands (see Rizzi 1990:2002).

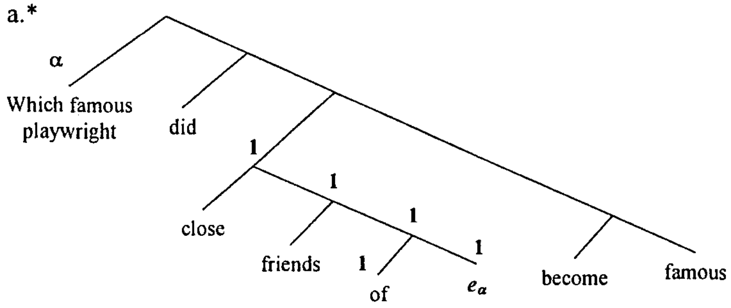
(4) W and Z (Z a maximal projection, and W and Z immediately dominated by some Y) are in a canonical government configuration iff
  a. V governs NP to its right in the grammar of the language and W precedes Z
  b. V governs NP to its left in the grammar of the language and Z precedes W.

(5) Y is a g-projection of X iff
  a. Y is an (X') projection of X or of a g-projection of X, or
  b. X is a structural governor and Y immediately dominates W and Z, where Z is a maximal projection of a g-projection of X, and W and Z are in a canonical government configuration.

Thus, the g-projections of X can extend upward as long as any dominating maximal projection is on a right branch. The CC requires that the set of the g-projections of (the governor(s) of) the empty category(ies) bound by a given binder and the binder itself form a connected subtree.

In case of a single gap, the CC requires that all the maximal projections in the path between the gap and its binder be on a right branch. Consider for instance the ungrammatical example in (1a): as the tree graph below makes clear, the g-projections of the gap stop at the level of the preverbal subject, which is a left branch and hence not in a canonical government configuration. Therefore, the g-projections cannot extend upward to reach the binder, and the CC is violated:[2]

(1) a.*

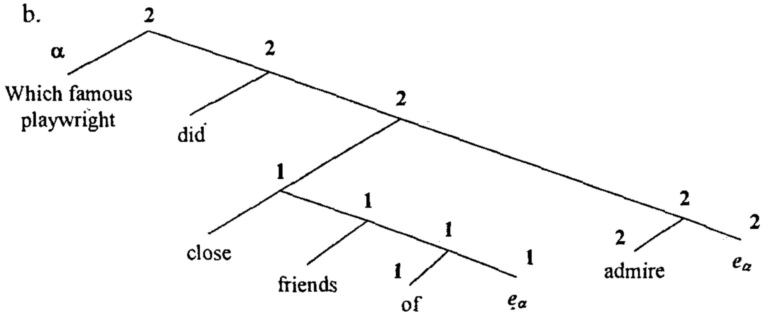


The rescuing effect in (1b) is due to the fact that the g-projections of the lower gap in the object position extend upward and connect to the g-

[2]As in Kayne (1983), the numerical indices are introduced for expository purposes to mark the g-projection paths of the empty categories, and have no theoretical significance.

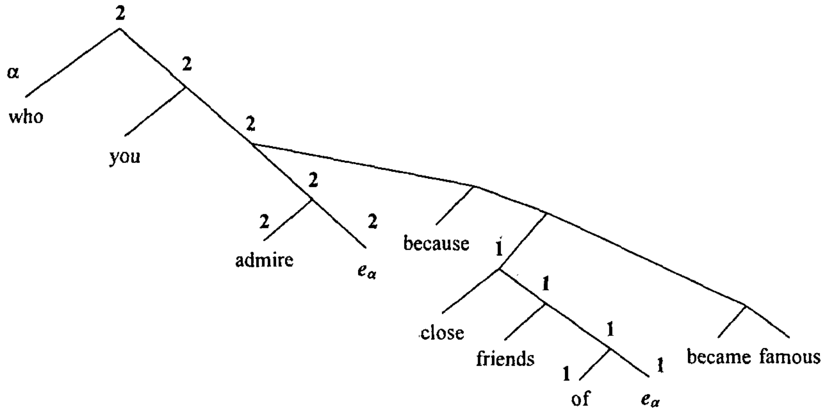
projections of the illegitimate gap embedded in the subject, as shown in the tree below. As a result, the two g-projections sets form a connected subtree including the binder, and the CC is satified:

(1) b.



On the contrary, no rescuing effect arises if the legitimate gap is too high in the tree for its g-projection set to connect to that of the illegitimate gap, as in the following example:

(1) c. *a person who you admire *e* because [close friends of *e*] became famous



The CC differs in various respects from other approaches to parasitic gaps in the GB framework. Firstly, even though, in the (b) examples of paradigms (1)–(3), there is a clear sense in which the gap inside the left branch is "illegitimate" or parasitic, and the other one is legitimate, there is no other assumed difference between them, either with respect to the nature of the empty category or of its relation to the binder. But the status of the parasitic gap and of its relation to the binder is actually debated, as can be seen in the collection of papers edited by Culicover & Postal (2001). Cinque (1990) and Postal (1994) have pointed out various types of evidence which suggest that the parasitic gap is a null resumptive pronoun rather than an ordinary extraction gap. The evidence comes from the lack of reconstruction effects in the parasitic gap position, the impossibility for parasitic gaps to occur in Postal's antipronominal contexts, and the restriction of parasitic gaps to the NP category. However, all these types of evidence have been called into question by other authors (cf. e.g. Levine et al. 2001, Levine & Sag 2003); it seems fair to say that the issue is still open. Secondly, note that the CC is designed to capture left branch islands only. Other strong island types, like e.g. right-hand adjuncts and relative clauses, are not subsumed under this condition (cf. Longobardi 1985). Again, it is an open question whether strong islands are a uniform class falling under a single principle (as proposed for instance in Cinque 1990). Despite these open problems, we believe that the CC incorporates an important insight, which we will formulate as follows:

(6) *Generalization on legitimate recursion and gap licensing*
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 (in that such a secondary branch cannot be the lowest one in a tree).

It is this insight that we will try to capture in our approach, though in an essentially derivational perspective. We will propose a derivational hypothesis that has the same empirical scope as Kayne's original CC, and only accounts for left branch islands (§3) (cf. Bianchi and Chesi 2005 for an attempt at subsuming right-hand adjuncts under this approach). As to the question of the (a)symmetry between "legitimate" and "parasitic" gaps, we will remain neutral. For the sake of simplicity, we will assimilate the parasitic gap-antecedent dependency to a standard antecedent-gap dependency, and treat both in terms of copy-remerging. However, we believe that the constraints on the structure of the computation that we are going to highlight are also consistent with an analysis in terms of a null resumptive pronoun. Our proposal will be implemented in the computational model of a top-to-bottom oriented Minimalist Grammar proposed in Chesi (2004). Although limitations of space prevent us from fully justifying the proposed model, we will now give a brief sketch, which will constitute the background of our proposal.

2 The Computational Model

2.1 The General Architecture

Chesi (2004) proposes a formalization of a minimalist grammar (adapting the formalism discussed in Stabler 1997) with two main components:

a. a lexicon consisting of feature structures (in the sense of unification grammars, e.g. HPSG) composed of semantic, syntactic and phonetic features;
b. three structure building operations (Merge, Move, Phase Projection).

Chesi argues that for reasons of computational efficiency and cognitive plausibility,[3] the grammar should have the property of *flexibility*: namely, it should be directly usable both in a parsing and in a generation context. The flexibility requirement leads Chesi to abandon the bottom-to-top orientation of the standard minimalist derivation, and to assume instead a top-to-bottom orientation (as in Phillips 1996).

Assume a Structural Description (SD) to be definable simply in terms of immediate relations (immediate dominance[4] and immediate precedence); assume, moreover, that any item is licensed within a SD (leading then to grammaticality) if and only if it is *selected*[5] or it is a possible *functional specification* of a lexical head. Accordingly, a lexical head is specified for two types of features: the SELECT features specify its argumental valency, and license the head's arguments (they correspond to the standard theta-grid or argument structure); the LICENSOR features instead specify the possible functional specifications that can be associated with the head: these correspond to the standard functional heads (FPs) in the lexical head's extended projection (Grimshaw 1991). Importantly, the LICENSOR features associated to a given lexical head are limited in number and are hierachically ordered, much as in the cartographic approach proposed by Cinque (1999), Rizzi (1997, 2004). The general schema is then the following:

(7) $[_{\text{LICENSOR features (+X)}}\ F_1 \ldots F_n\ ]$ **head** $[_{\text{SELECT features (=Y)}}\ C_1 \ldots C_n\ ]$

[3]It is hardly plausible that we would speak a language using a particular grammatical competence and that we would produce the very same language using a different knowledge.

[4]The statement "A *immediately dominates* B" would correspond to the result of a merge operation where A projects over B: $[_A\ A\ B]$.

[5]Here *selection* means both *C(ategorial)-selection* and *S(emantic)-selection* (Pesetsky 1982).

Chesi (2004) then defines a general top-to-bottom algorithm which can be exploited both in generation and in parsing; specifically, in generation, the algorithm converts a set of immediate dominance relations among semantic/formal feature structures into a set of immediate precedence relations among lexicalized phonological feature structures; vice versa, in parsing, it converts a set of immediate precedence relations among phonological structures into a set of immediate dominance relations among lexicalized semantic/formal structures). From this perspective the structure building operations can be redefined as follows:[6]

(8) *Merge* is a binary function (sensitive to temporal order) which takes two feature structures and unifies them (in the sense of unification grammars, Shieber 1986).

(9) *Phase Projection* is the minimal set of dominance relations introduced in the SD based on the expectations triggered by the SELECT features of the currently processed lexical head.

(10) *Move* is a top-down oriented function which stores an unselected element in a memory buffer[7] and re-merges it at the point of the computation where the element is selected by a lexical head.

An *unselected* element is any element that is processed before the lexical head is found, and hence temporally and linearly precedes the head itself, according to the following Linearization Principle (inspired by Kayne's (1994) LCA):

(11) *Linearization Principle*
 a. <A, B> if A (is a lexical head and) selects B as an argument;
 b. <B, A> if B is a functional specification of A.

Though limitations of space prevent us from fully describing the proposed model, we will illustrate how the structure building operations work in a simple example, where all the basic ingredients are involved:

[6]See Chesi (2004), Ch. 3.3.2 for an explicit and thorough formalization.

[7]Limitations of space do not allow us to fully characterize the memory buffer (the reader is referred to Chesi 2004). Let us simply emphasize two points. First, the memory buffer must be *multidimensional*, i.e. different kinds of elements are stored in separate lists; this will account for the selectivity of intervention (Relativized Minimality) effects, cf. Rizzi (1997, 2002). Second, the minimality effect itself can be captured by assuming a Last In First Out memory, so that at a given point of the computation only the last element that was inserted in the buffer can be retrieved, and the previously inserted ones cannot.

(12) The boy kissed the girl.
  i. As the initial step, the system projects a top-down expectation of a verbal phase (i.e. a CP),[8] whose lexical head will have to be a verb.
  ii. The constituent [the boy] is processed[9] and, being compatible with the functional Tense-related specification,[10] it is inserted at the corresponding functional level. Since the element is not selected in this position, it is also stored in the memory buffer.
  iii. The lexical item *kissed* (analysed as *kiss* +T) is processed; this introduces in the derivation the verb's SELECT features, here abbreviated as =S (external argument) and =O (internal argument) which are projected, according to Phase Projection, starting from the most external one.[11]
  iv. The constituent [the boy] previously stored in the memory buffer is re-merged as a sister to the verb to satisfy the verb's feature.
  v. As a final step, the computation proceeds by processing the direct object.

We return immediately to the special status of the lowest selected complement, which follows from a novel definition of phase.

2.2 Phases

Chesi (2004) argues that in order to gain computational tractability, the derivation must be broken up into *phases*, i.e. subparts of the computational process with a fixed upper bound in complexity. The phase can be roughly defined as follows:

(13) A *phase* is the minimal part of a top-to-bottom computational process in which all the functional and selectional specifications associated to a given lexical head are satisfied.

[8]This root application of Phase Projection is obviously not triggered by any SELECT feature.

[9]This actually constitutes a separate and "nested" computational phase, as will become clear in §2.2.

[10]The exact nature of the subject position is irrelevant for the present discussion.

[11]This is because we want to preserve scope relations and, as Phillip's (1996) Merge Right, from a derivational perspective, we expect these intermediate constituents to be built in the following order: $[_V \text{S V}] \rightarrow [_V \text{ S } [_V \text{V O}]]$.

Intuitively, each phase corresponds to the computation of a "minimal chunk" of syntactic structure like (7) above. Importantly, each phase will have a fixed upper bound in depth, determined by a limited number of possible functional specifications (Cinque 1999) and of selected arguments (Pesetsky 1982). Note however that, contrary to the standard bottom-to-top derivation, here a phase does not correspond to a complete subtree. In fact, when Phase Projection is triggered by the last SELECT feature of the lexical head, the current phase gets closed, and the computation of the complement constitutes the next phase. Thus, a phase corresponds to a subtree whose lowest selected element is not yet expanded. For the sake of simplicity, we assume here that only V and N can head a phase, and accordingly, phases correspond to the computation of a CP or DP chunk.[12]

Crucial to our argument is the distinction between *sequential* and *nested* phases.[13] As we have just said, when a phase reaches the lowest position selected by the lexical head, it is closed off: the expansion of the complement constitutes the next, *sequential* phase. A sequential phase thus follows the phase of the selecting head, and is separated from it.[14]

On the other hand, any DP or CP within a phase P_n that does not occur in the lowest position selected by the lexical head of P_n constitutes a *nested* phase, which must be processed while P_n is still incomplete. Hence, all unselected DPs or CPs preceding the lexical head of P_n are necessarily nested phases: preverbal subjects and fronted wh- or topical phrases can only be nested phases (and additionally, when their computation is completed, they are stored in the memory buffer of P_n). In (12), for instance, the subject DP [the boy] constitutes a nested phase within the matrix CP phase.

If phases are minimal chunks of the syntactic computation, it is reasonable to assume that each phase has its own local memory buffer for Move.[15] However, since long-distance movement can cross phase boundaries, it is necessary to devise a way to transmit the content of a phase's memory buffer to that of another phase. We adopt the following Success Condition:

[12]From our perspective, vP is not a separate phase from CP.

[13]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).

[14]In double complement structures, we can take both complements to be computed as sequential phases, or only the lowest one; the second assumption would derive a version of Kuno's (1973) "clause nonfinal incomplete constituent constraint"; see Bianchi & Chesi (2005) for discussion. Here we will not consider double complement structures.

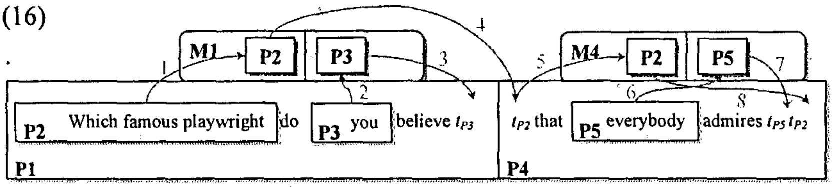
[15]This is our way to reconstruct Chomsky's "Phase Impenetrabilty Condition".

(14) *Success Condition*
At the end of each phase the local buffer is empty, or else its content is inherited by the memory buffer of the next sequential phase (if any).

Crucially, this condition only allows for a communication between the memory buffer of two adjacent sequential phases. (Obviously, at the end of the final phase, the local buffer will have to be empty). This accounts for the transparency of the lowest recursive branch of the tree.

To see this, consider for instance a computation for (15), as schematically represented in (16) (where the boxes identify phase boundaries):

(15) Which famous playwright do you believe that everybody admires?
[$_{CP}$ [$_{DP}$Which famous playwright]$_i$ do you believe [$_{CP}$ *whP$_i$* [that everybody admires *whP$_i$*]?

(16)


The algorithm initializes a CP phase 1 (P1). Then it computes the wh-phrase, which constitutes a separate nominal phase 2 (P2). Since the wh-phrase is not selected, it is stored in the local memory buffer (M1) of P1 by Move (step 1). Then, the computation of P1 proceeds, down to the complement position of the matrix verb *believe* (we disregard the computation, storage and retrieval of the subject phase P3: step 2, 3, 6, 8). At this point P1 is closed and the wh-phrase (P2) in its memory buffer is discharged into the complement CP phase 4 (P4), since the latter is sequential and selected. We propose that this takes place by re-merging the content of the memory buffer of P1 in the left periphery of the complement CP, P4 (step 4); since this position is unselected, the wh-phrase is re-stored in the local memory buffer of P4 (step 5). As a result, the "inheritance" mechanism leaves an intermediate copy/trace in the edge of the complement CP phase.[16] The computation proceeds down to the object position of the

[16]Although this assumption is not strictly necessary for the algorithm to work, it seems fairly natural and it allows us to capture various successive cyclicity effects, like e.g. Irish complementizer alternations or French stylistic inversion. We thank Luigi Rizzi for discussion of this point.

verb *admires*, where the wh-phrase P2 is discharged from the local memory buffer of P4 and re-merged (step 8): the Success Condition is thus satisfied at the end of the computation.

To summarize, the following points of Chesi's (2004) model will be crucial for the development of our analysis:

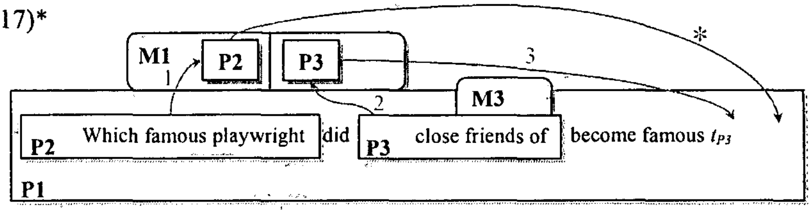
a. Every computation is a top-down process divided into phases of fixed maximal size.
b. A phase gets closed when the lowest selected position of its head is processed; the lowest selected complement constitutes the next sequential phase.
c. All unselected constituents are instead nested phases: they are processed while the superordinate phase has not been closed yet.
d. The Move operation stores an unselected element found before (i.e. to the left of) the head in the local memory buffer of the current phase, and discharges it in a selected position if possible; if not, when the phase is closed the content of the memory buffer is inherited by the next sequential phase. The memory buffer of the last phase must be empty at the end of the computation.

3 Left-branch Islands are Computationally Nested Phases

With this background, we can now go back to our initial problem, namely, left branch islands and the connectedness effect, as in (1).

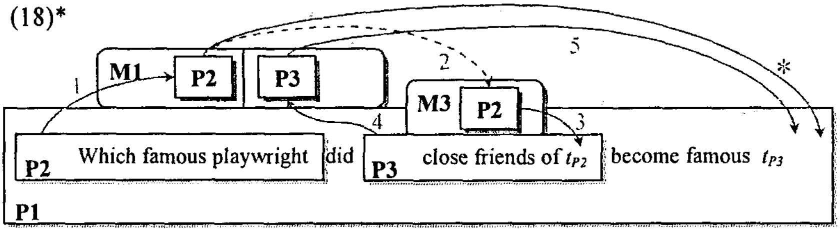
Consider now a computation for (1b), as schematically represented in (17):

(17)*



Once again, the algorithm initializes a CP 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 (M1). The computation of phase 1 proceeds, inserting *did* in C. As a 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 (M3) does not contain the wh-phrase which was stored in the memory buffer of phase 1 (M1): hence, the wh-phrase cannot be discharged in the selected gap position within the subject DP. The wh-phrase also remains undischarged at the end of the computation of phase 1, violating the Success Condition (14). This accounts for the strong island effect.

(18)*



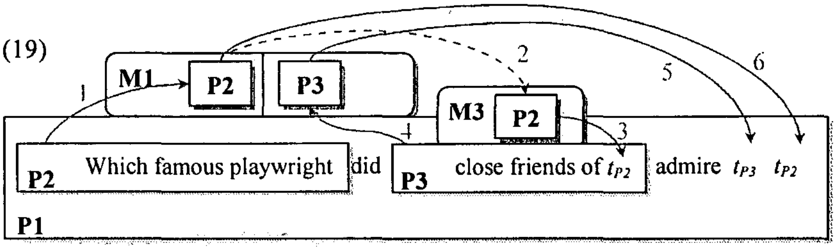
Suppose now that we optionally allow the memory buffer of the nested subject DP phase 3 to "copy" the buffer of the immediately superordinate phase 1, which contains the wh-phrase (this "parasitic copying" is represented by a dotted line in (18), step 2).[17] Then, the wh-phrase can be discharged in the gap position within the DP phase 3. However, this step will only empty the local memory buffer of phase 3. We crucially assume that copying into the memory buffer of the nested phase cannot discharge the memory buffer of the superordinate phase. As a result, even after the "parasitic gap" is computed, the local memory buffer M1 of the yet incomplete matrix phase 1 still contains the wh-phrase. This remains undischarged at the end of the computation, violating the Success Condition.

On the other hand, the copying mechanism does lead to a successful computation in the case of (1b). As in (18), the "parasitic" copy of the wh-phrase in the memory buffer of the subject DP phase 3 is discharged in the first gap position (step 3); however, the matrix CP phase 1 contains another selected position where the wh-phrase can also be discharged from the memory buffer of phase 1 (step 6). This derivation complies with the

[17]Optional copying would actually introduce non-determinism in the computation. In order to avoid this, we can assume the possibility of backtracking: when the computation of phase 3 reaches the position selected by the noun *friends*, since there is no more lexical material available in phase 3 and the local memory buffer is empty, the system backtracks and copies in the memory buffer of phase 3 the content of the buffer of the immediately superordinate phase 1. The crucial point is that this "parasitic copying" in the buffer of a nested phase can not discharge the local buffer of phase 1.

Success Condition, as shown in (19). This accounts for the connectedness effect.[18]

(19)



Before closing this section, let us summarize the main aspects of the proposed analysis. The Connectedenss Condition has been recast in derivational terms, by assuming

a. a top-to-bottom derivation divided in phases;
b. a "storage" conception of the Move operation, which stores an unselected element in the local memory buffer of the current phase and re-merges it in a selected position;
c. a distinction between sequential phases (corresponding to the "canonically governed" branches on the recursive side of the tree) and nested phases (corresponding to the "non canonically governed" branches on the non-recursive side of the tree).

The crucial element in our account of left branch islands is the idea that the content of the memory buffer of a phase can only be inherited by the next sequential phase, and not by a nested phase; in other terms, the content of the memory buffer can be "bequeathed" only after the relevant phase has been completed. In order to account for parasitic gaps licensed under connectedness, we have allowed for the possibility of parasitically copying the content of the buffer of a matrix phase into the buffer of a nested phase; this parasitic copy, however, cannot empty the matrix memory buffer, whence the necessity of *another* (selected) gap within the matrix phase itself (or within a phase that is sequential to the matrix one).

[18] Our analysis cam also account for the lack of a connectedness effect in configurations where the parasitic gap is embedded in two strong islands, nested in one another: this will give rise to a double application of "parasitic copying", but only one of the two "parasitic" memory buffers will be discharged, whereas the other one will lead to a violation of the Success Condition. For limitations of space we cannot illustrate the computation of such examples.

4 Further Prospects and Conclusions

In this paper, we have proposed an approach to left branch islands and to the connectedness effect within a top-to-bottom derivational framework (formalized in Chesi 2004). In conclusion, we wish to point out some more general consequences of our approach. First, the top-to-bottom orientation of the computation allows for a relatively straightforward solution to the problem of phase-by-phase linearization, since nested and sequential phases are processed in a well defined order, driven by the LICENSOR and SELECT features of the relevant phase heads. Second, the storage conception of Move avoids the "teleological" mechanism of raising to the edge of each phase, which is required in bottom to top successive-cyclic movement: such moves are teleological in that in the lower phases the final trigger of movement, i.e. the probe/EPP head, has not been inserted yet.[19]

To the extent that our proposal is tenable, it supports a general conception whereby considerations of computational efficiency and cognitive plausibility at the interface with the performance tasks directly constrain the architecture of the grammar itself.

References

Bianchi, Valentina, and Cristiano Chesi. 2005. Phases, strong islands, and computational nesting. Ms., University of Siena.

Chesi, Cristiano. 2004. Phases and Cartography in Linguistic Computation: Toward a Cognitively Motivated Computational Model of Linguistic Competence. Doctoral Dissertation, University of Siena.

Cinque, Guglielmo. 1990. *Types of A' Dependencies*. Cambridge, MA: MIT Press.

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

Culicover, Peter W. and Paul M. Postal (eds.). 2001. *Parasitic Gaps*. Cambridge, MA: MIT Press.

Fox, Danny and David Pesetsky. 2004. Cyclic linearization of syntactic structure. Ms., MIT.

Grimshaw, Jane. 1991. Extended projection. Ms., Brandeis University.

[19]As far as we can see, the problem is not really solved in Fox & Pesetsky's (2004) approach: the wh-phrase moves to the edge of the phase (linearization domain) in order to be linearized to the left of the other internal constituents of the phase; however, the necessity to alter the linearization derives from the ultimate goal of reaching the final left-hand landing site. A common solution is to stipulate uninterpretable wh- or EPP features, which are however justified only by internal constraints of the bottom to top derivation. See McCloskey (2001) for a defense of such features and Rizzi (2002) for a critical discussion.

Haider, Hubert. 2003. Pre- and post-verbal adverbials in OV and VO. *Lingua* 114.

Levine, Robert D., Thomas E. Hukari, and Michael Calcagno. 2001. Parasitic gaps in English: Some overlooked cases and their theoretical implications. In *Parasitic Gaps*, ed. P. Culicover and P. Postal, 181–222. Cambridge, Mass.: MIT Press.

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

Kayne, Richard S. 1994. *The Antisymmetry of Syntax*. Cambridge, Mass.: MIT Press.

Kuno, Susumu. 1973. Constraints on internal clauses and sentential subjects. *Linguistic Inquiry* 4(3):363–385

Levine, Robert, and Ivan A. Sag. 2003. Some empirical issues in the grammar of extraction. In *Proceedings of the HPSG03 Conference*. ed. S. Müller, Michigan State University, East Lansing: CSLI.

Longobardi, Giuseppe. 1985. The theoretical status of the adjunct condition. Ms., Scuola Normale Superiore, Pisa.

McCloskey, James. 2001. The morphosyntax of WH-extraction in Irish. *Journal of Linguistics* 37: 67–100.

Nunes, Jairo, and Juan Uriagereka. 2000. Cyclicity and extraction domains. *Syntax* 3:20–43.

Pesetsky, David. 1982. Paths and Categories. Doctoral Dissertation, MIT.

Phillips, Colin. 1996. Order and Structure. Doctoral Dissertation, MIT.

Pollard, Carl, and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. Stanford: CSLI.

Postal, Paul M. 1994. Contrasting extraction types. *Journal of Linguistics* 30:159–186.

Rizzi, Luigi. 1990. *Relativized Minimality*. Cambridge, MA: MIT Press.

Rizzi, Luigi. 1994. Argument/adjunct (a)symmetries. In *Paths towards Universal Grammar*, ed. Guglielmo Cinque et al., 361–376. Washington, DC: Georgetown University Press.

Rizzi, Luigi. 1997. The fine structure of the left-periphery. In *Elements of Grammar*, ed. Liliane Haegeman. Dordrecht: Kluwer.

Rizzi, Luigi. 2002. Locality and left periphery. In *Structures and Beyond. The Cartography of Syntactic Structures*, ed. A. Belletti, Vol. 3. Oxford: Oxford University Press.

Rizzi, Luigi. 2004. *The Structure of CP and IP. The Cartography of Syntactic Structures*, Vol. 2. Oxford: Oxford University Press.

Shieber, Stewart M. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Stanford, CA: CSLI Lecture Notes.

Stabler, Edward. 1997. Derivational minimalism. Ms., UCLA.

Starke, Michael. 2001. Move dissolves into Merge. Doctoral Dissertation, University of Geneva.

Centro Interdipartimentale di Studi Cognitivi sul Linguaggio
Università di Siena
8 Piazza S. Francesco
I-53100 Siena
Italy
bianchi10@unisi.it
cri@mit.edu