MERGE
The derivation of structure and order

Groningen, June 5, 2008

PROGRAM
&
ABSTRACTS

diug

Merge: the derivation of structure and order

a workshop, organized by the research project
'Dependency in Universal Grammar' (DIUG)
Center for Language and Cognition, University of Groningen

June 5, 2008

P R O G R A M

Academy Building (Broerstraat 5), Zernikezaal

09:45 welcome; 10:00 opening

10:10-10:50 Jordi Fortuny (University of Groningen), The nesting machine

10:50-11:30 Jan-Wouter Zwart (University of Groningen), Prospects for top-down derivation

11:30-12:10 Cristiano Chesi (University of Siena), Minimalist merge, destructive feature-checking, and sequential unification

12:10-14:00 lunch break

14:00-14:40 Ian Roberts (University of Cambridge), A phase-based approach to the cross-linguistic distribution of "mixed" word-order types

14:40-15:20 Ángel Gallego (Universitat Autónoma de Barcelona/ University of Cambridge), Label-free bare phrase structure

15:20-16:00 Aritz Irurtzun (University of the Basque Country), Merge and asymmetric structure building

16:00-16:20 break

16:20-17:00 Barbara Citko (University of Washington), Symmetry in syntax? Symmetric merge and symmetric move

17:00-17:40 Robert Truswell (Tufts University), Order but no structure in Bonobo English

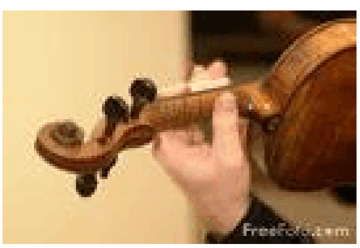
17:40 closing

organizers: Jordi Fortuny & Jan-Wouter Zwart

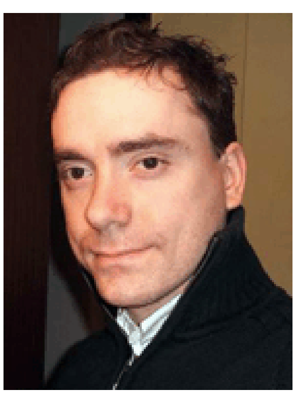
Participants

Cristiano Chesi
Centro Interdipartimentale di Studi Cognitivi sul Linguaggio, University of Siena
chesi@media.unisi.it
www.ciscl.unisi.it/persone/chesi.htm

Barbara Citko
Department of Linguistics, University of Washington (Seattle)
bcitko@u.washington.edu
faculty.washington.edu/bcitko/



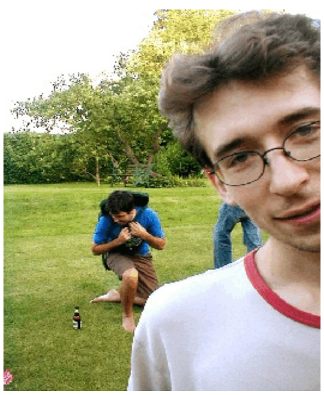
Jordi Fortuny Andreu
Department of Linguistics, University of Groningen
j.fortuny.andreu@rug.nl



Ángel Gallego
Centre de Lingüística Teòrica, Universitat Autònoma de Barcelona
University of Cambridge
angel.gallego@uab.cat
http://seneca.uab.es/ggt/membres/professors/gallego.html

Aritz Irurtzun
Department of Linguistics and Basque Studies, University of the Basque Country (Vitoria-Gasteiz)
fvbirsva@vc.ehu.es
http://www.ehu.es/hitt/irurtzun.htm

Ian Roberts
Department of Linguistics, University of Cambridge
igr20@cam.ac.uk
www.ling.cam.ac.uk/people/roberts/



Robert Truswell
Center for Cognitive Studies, Tufts University
robert.truswell@tufts.edu
www.tufts.edu/~rtrusw01/

Jan-Wouter Zwart
Department of Linguistics, University of Groningen
c.j.w.zwart@rug.nl
www.let.rug.nl/zwart

‘The nesting machine’. Jordi Fortuny
Center for Language and Cognition Groningen. Rijksunversiteit Gronigen
---

In this talk I shall present the central idea of Fortuny & Corominas (forthcoming). On the basis of Kuratowski’s (1921) general theory of order, itself defined on the foundamental terms of axiomatic set theory, I shall propose a precise definition of the combinatorial procedure by which the syntactic component of the faculty of language generates hierarchically structured expressions.

Given a finite alphabet $A = \{a_1, a_2, \ldots, a_n\}$, at the first step s_0 of a derivation, the ‘nesting machine’ generates the set M_0, which is an element of A. At the following step s_1, it generates a new set, M_1, by forming the union of M_0 and a member of A. At step s_n, it generates the set M_n, which is the union of M_{n-1} and an element of A. When $a_i \in A$ comes into the computation at the step s_k, it becomes an occurrence a_i^k of a_i. This can be summarized through the recursive operation:

$$M_0 = a_i^0$$
$$M_{n+1} = a_k^{n+1} \cup M_n$$

Let the final outcome of this machine be $\mathbf{N} = \{M_0,\ldots, M_{n+1}\}$, a set whose elements are the Ms generated through the successive derivation. $\mathbf{N}$ has the important property of being a ‘nest’, i.e., a set whose elements are sets linearly ordered by inclusion.

If the syntactic algorithm is a nesting machine, the common syntactic relations can be readily defined with no need to introduce any idiosyncratic grammatical element. Not only the linear order among the terminals of an expression can be properly defined on the basis of Kuratowski’s set-theoretical definition of order, as advanced in Fortuny (2008), but also the constituency relationship and the dominance relationship.

We shall also allow the nesting machine to perform $A \cup B$ when:

(a) A or B are nests, and
(b) when B belongs to the domain of A.

If the circumstance (a) is allowed, the nesting machine has the capacity to comprise not only a sole derivational space (‘single nesting’) but also multiple derivational spaces (‘complex nesting’), in which case the outcome of a particular derivational space D_i feeds a different derivational space D_j. Some basic conditions on complex nesting will be presented. Circumstance (b) is equivalent to Chomsky’s (2001, 2005) intuition that Merge can be internal, not only external. We shall provide the means to represent chains comprising multiple copies of an occurrence and to distinguish among copies, again without resorting to further assumptions or postulating grammatical idiosyncracies.

We shall conclude by noting that the importance of nestedness as a unifying concept may be beyond the general theory of order and the syntactic configurational theory of linguistic expressions, by suggesting the new course we begin to undertake.

Prospects for top-down derivation

Jan-Wouter Zwart

Merge: the derivation of structure and order
(June 5, 2008, Groningen)

Simplest Merge

Every derivation of syntactic structure needs (a) a set of elements N manipulated in the course of the derivation, called ‘numeration’, and (b) a procedure establishing relations among the members of N, called ‘merge’. Simplicity considerations then demand:

(1) *Simplicity requirements on the derivational procedure*
a. merge manipulates a single element of N at each step of the derivation
b. merge manipulates each element from N only once

These requirements are not met in standard conceptions of the derivational procedure, which describe merge as an operation combining two elements (hence manipulating more than one element), and which allow a merged element to be merged again (‘internal merge’, i.e. movement). While these deviations from the simplicity requirements seem minimal or perhaps unavoidable, it should be pointed out that they introduce stipulations unwanted in a truly minimalist approach.

Thus, a derivational procedure that allows merge to manipulate two elements at a single step in the derivation can disallow merge to manipulate more than two elements only by stipulation (since 2, unlike 1, is not the absolute minimal number). This stipulation is nevertheless needed, if the system is supposed to derive only binary branching structures. Likewise, a procedure that allows a merged element to be moved (remerged) essentially states that one of the two elements manipulated maybe contained within the structure being derived. But then the possibility that the other element being manipulated at that step of the derivation is also contained within the structure being derived can be excluded only by stipulation (the ‘extension condition’). Yet this stipulation is needed, if we want the system to be unable to derive bizarre and in fact endlessly looping structures not attested in human language.

Adhering to the simplicity requirements in (1), then, eliminates seemingly inevitable stipulations and yields a closer match between the structures generated by the derivational procedure and the actual phenomena of human language.

Proposal

Concretely, I would like to propose that each step in the derivational procedure turns the numeration N (a set, or, perhaps more appropriately, an array) into the ordered pair $P = \langle x, y \rangle$, where $x \in N$ and $y = (N - x)$. In other words, merge splits N into a pair consisting of a (designated) member of N and its residue in N. The numeration, then, is reduced at each step, and each next step in the derivational procedure targets the residue of the numeration created by the previous step, until N is empty. This procedure meets the requirements in (1): a single element from N is split off from N and further operations target the residu of N.

Structure

The system described here takes the derivation to be a procedure that transforms an unordered or unstructered collection of elements (i.e. a set or an array) into a structure. This differs slightly but significantly from the standard view, which takes merge to involve a transfering process taking elements out of a resource (the numeration) into a structure.

The derivational procedure operates in a top-down fashion, in the sense that the first pair created corresponds to the highest pair in the tree structure representation. This implies that properties of lexical heads (having to do with argument structure or subcategorization features) play no role in driving the derivation.

Top-down derivations within generative grammar have been proposed and defended earlier, most notably by Phillips (2003) and Chesi (2007). In Phillips's system, merge adjoins material to the most deeply embedded right branch of the structure, essentially splitting that branch by the addition of new material from the numeration. This system differs from the one contemplated here in that it takes merge to be an operation importing material into the structure from some resource, as in the more traditional bottom-up derivational system.

Order

In the system proposed here, ordering is not a function of features of the elements merged, but a function of the derivation itself. More precisely, the circumstance that the steps in the derivation are temporally ordered yields an ordering of the elements affected by these steps.

The relation between the temporal ordering of the steps in a derivation and the linear ordering of the elements involved in the derivation can be made precise in the following way. Adopting the derivational system in (4), we observe that the set of elements merged (i.e. split off from the numeration) grows with each step in the derivation. If we consider only the first two steps, the sets of elements merged at each step are:

(2) 1. after the first step: { a }
    2. after the second step: { a, b }

The set of sets of elements merged after step 2 is:

(3) { a, { a, b } }

which is the set-theoretical notation of the ordered pair ⟨ a, b ⟩. It follows that after a sequence of steps we obtain a set of sets of elements merged, which is equivalent to an ordered n-tuple as in (4):

(4) { a, { a, b }, { a, b, c}, { a, b, c, d }, { a, b, c, d, e } }

The idea of deriving order from nested sets originates with Fortuny (2008) (though details of the implementation differ). As Fortuny (2008) shows, the actual linear order of the words and phrases involved in the derivation, which is established at the interface component dealing with sound, may be derived straightforwardly from the output of the derivational procedure if that output is an ordered n-tuple. The simplest implementation of that idea appears to be (5):

(5) *Linear Correspondence Axiom (revised from Kayne 1994)*
    $\langle \alpha, \beta \rangle = / \alpha\ \beta /$

Consequences

This paper discusses a number of consequences of the top-down derivational approach, having to do with the definition of syntactic relations, morphological dependency, and the phenomena commonly described in terms of movement.

References

Chesi, Cristiano. 2007. Five reasons for building phrase structure top-down from left to right. *Nanzan Linguistics Special Issue 3*, 71-106.
Fortuny, Jordi. 2008. *The emergence of order in syntax*. Amsterdam: Benjamins.
Kayne, Richard. 1994. *The antisymmetry of syntax*. Cambridge: MIT Press.
Phillips, Colin. 2003. Linear order and constituency. *Linguistic Inquiry* 34, 37-90.

Minimalist merge, destructive feature-checking and sequential unification

Cristiano Chesi
CISCL – University of Siena

In 1967, Ross tried to characterize the upper bound of the set of well-formed sentences in a natural language focussing on a collection of constraints on, powerful enough, recursive procedures (i.e. rewriting/transformational rules). After 40 years we still face the very same problem: *move* α and the *projection principle* before (Chomsky 1965), *merge*, *move* and *agree* now (Chomsky 2005a) are very general and (too) powerful devices that perhaps allow us to figure out very important universal principles, but that are practically insufficient to constrain many relevant empirical phenomena.

Within the spirit of the minimalist initiative, in this talk I will try to show that a more restrictive definition of *merge* can be successfully rephrased in top-down (*phase*-based), left-right terms, attaining superior results in terms of computational economy and empirical adequacy, at least with respect to a relevant set of phenomena such as argument cluster coordination Vs. fronting/scrambling/clefting asymmetries, (1)-(2) (Phillips 1996, Choi and Yoon 2006), and (time permitting) "spec"-head/multiple agreement, (3) (Chesi 2007):

(1) a. Wallace gave [Gromit a biscuit] and [John some cheese] for breakfast
b. *[Wallace a biscuit] and [Gromit some cheese] gave to John
c. *[Gromit a biscuit] Wallace gave for breakfast
d. *What Wallace gave is [Gromit a biscuit]

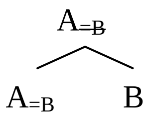
(2) a. Mary-ga [[John-ni ringo-o 2-tu] to [Bob-ni banana-o 3 hon]] ageta(koto)
M-Nom J-Dat apple-Acc 2-Cl and B-Dat banana-Acc 3-Cl gave (fact)
'Mary gave two apples to John and three bananas to Bob'
b. [[John-ni ringo-o 2-tu] to [Bob-ni banana-o 3-bon]] Mary-ga ageta (koto)
J-Dat apple-Acc 2-Cl and B-Dat banana-Acc 3-Cl M-Nom gave (fact)
'(the fact that) [two apples to John and three bananas to Bob] Mary gave'
c. Mary-ga age-ta no-wa [John-ni ringo-o 3-tu] da
M-Nom give-Pst Nm-Top J-Dat apple-Acc 3-Cl be
'It is [three apples to John] that Mary gave'

(3) a. La grande palla rossa
$The_{sg, f}$ $big_{sg, f}$ $ball_{sg, f}$ $red_{sg, f}$
'The big red ball'
b. $Gianni_{3,m,sg}$ $ha_{3,sg}$ paura di essere $arrivato_{m,sg}$ troppo tardi.
G. has fear to be arrived too late
'G. fears to have arrived too late'
c. $Maria_{3,f,sg}$ $li_{3,m,pl}$ $ha_{3,sg}$ già $mangiati_{m,pl}$ $tutti_{3,m,pl}$ (i panini)
M. them has already eat all (the sandwiches)
'M. has already eaten them all'
d. *$Maria_{3,f,sg}$ $ha_{3,sg}$ già $mangiati_{m,pl}$ [tutti i panini] $_{3,m,pl}$
M. has already $eaten_{m,pl}$ all the $sandwiches_{3,m,pl}$

In English, but not in Japanese, constituency tests (e.g. coordination, fronting, anaphor binding) conflict with one another when they target the Subject-Object argument cluster. Phillips explains this fact, deriving phrase structure from left to right (by means of *merge right*): for instance in Japanese (SOV), but not in English (SVO) the intermediate constituent [SO] is built and can be targeted for coordination. On the other hand multiple (selective, non-local) agreement and order effects reported in (3) are not easily implementable within the standard minimalist phase edge driven re-merge.

I will build on Stabler's (1997) proposal discussing some formal problems related to his conception of *merge* and to the *destructive feature checking* (selection-based) algorithm exemplified below ($A_{=B}$ indicates that A selects B in the standard sense; as a result of the merge operation, =B is deleted):

(4) merge ($A_{=B}$, B) =



In the end I will propose that *merge* would better be described as a partial function that unifies (in the sense of Shieber 1986), sequentially, lexical items (i.e. clusters of features) with (a structured set of) features already present in the tree.

The proposed operation departs from the simplest assumptions discussed within the standard minimalist approach (Chomsky 1995-2005b): *minimalist merge* (and other standard structure building operations) leads to solutions that are only seemingly simpler and more essential than the ones proposed within the government and binding framework: as Stabler's formal implementation of a minimalist grammar suggests, theoretical (and empirical) complexity is mainly projected onto lexical items/features, recasting potential universal properties to language specific properties (much as in transformational, rule-based, approach, Chomsky 1965). This way UG computational core would be really minimal but also problematic for explanatory adequacy purposes.

References

Chesi, C. (2007) *Five reasons for building phrase structures top-down from left to right*, Nanzan Linguistics, Special Issue 3, 71-106

Choi, Y., Yoon, J. (2006) *Argument Cluster Coordination and Constituency Test (Non)-Conflicts*, NELS 37**,** University of Illinois at Urbana-Champaign.

Chomsky, N. (1965) *Aspects of the theory of syntax,* M.I.T. Press, Cambridge.

Chomsky, N. (2005a) "Three factors in language design", *Linguistic Inquiry* 36**,** 1-22.

Chomsky, N. (2005b) "On Phases", *Manuscript, MIT*.

Phillips, C. (1996) *Order and Structure,* Ph.D. Thesis, Massachusetts Institute of Technology.

Stabler, E. (1997) *Derivational Minimalism*, Logical Aspects of Computational Linguistics: First International Conference, LACL'96.**,** Nancy, France**,** 68.

A Phase-based approach to the cross-linguistic distribution of "mixed" word-order types

A. Assume:

(i) the strict version of the Phase Impenetrability Condition (PIC) (Chomsky, 2000: 108). Hence in the configuration in (1), where H^0 is a phase head, only **bold material** is accessible to X^0, Z^0, etc; material in outline font has already been sent to Spellout;

(1) $[_{ZP} \ldots Z^0 \; [_{XP} \ldots X^0 \; [_{HP}$ **… [H^0** $[_{YP} \ldots Y^0 \; [_{WP} \ldots [W^0 \ldots$

(ii) Kayne's (1994) Linear Correspondence Axiom (LCA): asymmetric c-command determines linear precedence;

(iii) following Chomsky (2005), phase heads determine many properties of the heads in their phasal domain (i.e. all the heads they c-command, subject to (1)), including whether they are probes, but

(iv) any head may independently have an EPP-feature (a movement-inducing linearization feature), such that the movement diacritic on head X (written as X_{EPP}) signals that X's complement must move to X's specifier, thereby resulting in the order *complement* > *X*.

B. Now consider (2), a consequence of assumption (iii), for a given phase PH:

(2) If phase-head H_{EPP}, then all heads Y, W… in H's domain are Y_{EPP}, W_{EPP}, etc.

Applying (2) to the vP phase, we arrive at the following:

(3) a. v_{EPP} V_{EPP} → $[\; [_{VP} \text{ O V }] \text{ v }]$ (consistent head-final order)
b. v V → $[\text{ v } [_{VP} \text{ V O }]]$ (consistent head-initial order)
c. v V_{EPP} → $[\text{ v } [_{VP} \text{ O V }] \;]$ (disharmonic order I: initial-over-final)
d. $*_{EPP}$ V → $[\; [_{VP} \text{ V O }] \text{ v }]]$ (disharmonic order II: final-over-initial)

The "harmonic" orders in (3a,b) are of course widely attested. Disharmonic order I is also fairly widely-attested: if auxiliaries are in v, then Germanic "verb-projection raising" is a case of this, as illustrated in (4):

(4) .. dat Jan [$_{vP}$**wil [$_{VP}$[$_{DP}$ 'n huis] koop]]]** (Modern Spoken Afrikaans)
that John want a house buy ".. that John wants to buy a house"

This is a consequence of assumption (iv) above. But disharmonic order II, as in (3d), is extremely rare. This is primarily a consequence of assumption (iii).

C. The rarity of final-over-initial order is supported by the following observations:(a) Old and modern Germanic varieties exhibit a mix of head-initial and head-final orders in VP and IP, with all permutations of Aux, V and Object attested except VOAux (den Besten & Edmondson 1983, Hróarsdóttir 2002). The unattested VOAux order is a case of (3d). Finnish, which also permits mixed orders, shows the same pattern. (b) Sentence-final complementisers are not found in VO languages (Hawkins 1990). This is since both $[_{CP} [_{TP} [_{VP} \text{ V O }] \text{ T }] \text{ C }]$ and $[_{CP} [_{TP} \text{ T } [_{VP} \text{ V O }]] \text{ C }]$ parallel (3d). (c) In the nominal domain, Finnish has mixed projections too: it has both pre- and postpositions and N-Complement (N-O) as well as Complement-N order. All permutations of P, N and O are found except N-O-P, the order in (3d). (d) The pattern in (3d) is also relevant to Greenberg's Universal 20 ("When any or all of the items (demonstrative, numeral and descriptive adjective) precede the noun, they are always found in that order. If they follow, the order is either the same or its exact opposite" Greenberg 1963:87). Disregarding the order of A and N, supposing the universal first-merged order *D(em) > Num > N*, and taking D to be a phase head, we see that this order corresponds to (3b), while *N > Num > D(em)* corresponds to (3a) and *D(em) > N > Num* (not predicted by Greenberg, but attested according to Cinque 2005) corresponds to (3c). *Num > N > D(em)* corresponds to (3d), and is very rare.

D. We are not predicting an absolute ban on structures like (5), however (unlike Holmberg 2000, Julien 2000):

(5) $[_{Head1'} [_{Head2P} \text{ Head}_2 \text{ Compl }] \text{ Head}_1]$

The ban on final-over-initial order instead holds only where (i) Head_1 is a phase head, (ii) Head_2 is a PIC-accessible non-phase head in the complement of Head_1. Cases of (5) where Head_1 is a not phase head include many VO languages with clause-final negation (notably, a range of genetically distinct languages in Western-Central Africa and many Papuan languages; cf. Dryer 2006, Reesink 2002). If negation is either not a head or not a phase-head, these cases are consistent with our assumptions. Similarly, a range of VO languages, primarily in East Asia or in Central Africa, show final tense-aspect particles (e.g. Logbara, Mamvu [cf. Tucker & Bryan 1966] and a range of other languages in the same geographical area [cf. Dryer 2006]; and also see Duffield 2001 and Simpson 2005 on this phenomenon in East Asian languages, including Chinese). Here the tense/aspect particle is provably non-verbal – in many cases, it alternates with a fully inflected counterpart which occupies a clause-internal position; hence it cannot be v or a V-related non-phase head. A third kind of case is represented by clause-final "force" particles in VO languages: again roughly the same groups of languages show these (cf. Dryer 2006 on the "clustering" of Final-over-Final violations). Here it may be possible to invoke a "split-C" system, with the lower C triggering movement of TP to its specifier, while the higher C is the true phase head – these orders would then be cases parallel to (3c) (see Shih & Sybesma 2007 for an analysis of Mandarin along these lines). Finally, circumpositions of the type found in West Germanic (***auf*** *den Berg* ***hinauf*** - "up the mountain DIR-up") probably fall under this umbrella too (see Svenonius, to appear for arguments that the "postpositional" element is not a phase head, and Aboh 2004 on "fake postpositions" in similar constructions in the Gbe languages).

Cases of (5) where (ii) doesn't hold fall into two kinds: (a) those where $Head_2$ is PIC-accessible to $Head_1$ but a phase head; this would be the case of head-initial DPs in head-final VPs in languages like German, and (b) those where $Head_2$ is not PIC-accessible to $Head_1$: hence we might expect that, for example, the C-system requires one surface head-complement order and the v-system or the D-system another. This kind of case is familiar from Germanic and elsewhere.

E. There is some evidence that (5) is also allowed where $Head_1$ and $Head_2$ are categorially distinct. If P is not a phase head, this is our account of prepositional complements to final verbs in German and similar languages (if P is a phase head, these fall under the account of head-initial DPs in head-final VPs just given). If this is correct, then we must modify (2) as follows:

(2') If phase-head $H[\alpha N, \beta V]_{EPP}$, then all heads $Y[\alpha N, \beta V]$, $W[\alpha N, \beta V]$… in H's domain are $Y[\alpha N, \beta V]_{EPP}$, $W[\alpha N, \beta V]_{EPP}$, etc.

In other words, phase heads which are categorially distinct from their complements do not impose their EPP-feature on the heads in their phasal domain.

F. Assumption (ii) above (the LCA) is crucial since there really appears to be an asymmetry in the attestation of initial-over-final orders, (3c), and final-over-initial orders, (3d). The evidence from mixed typologies in Germanic, Finnish, Basque and many other languages supports this. This also implies that parsing-based approaches, relying on the relative ease of processing harmonic vs disharmonic orders are inadequate. Both (3c) and (3d) are disharmonic, but (3c) is not hard to find, while (3d) is. A parsing-based approach like that of Ackema & Neeleman (2002), which appeals to severe restrictions on backward localisation – here of the selected head in a selection dependency – also fails, despite the fact that it draws the necessary distinction between different types of disharmonic orders (cf. Cecchetto 2007 for a proposal along these lines): the counterexamples mentioned above make it clear that such backward localisations are possible and cannot therefore lie at the root of the problem.

G. Assumption (i) is vital for our account. Crucially, unlike Chomsky and others, we assume that completion of a phase leads to the *radical removal* from the computation of the material in the spellout domain associated with that phase (VP, TP, etc.). Thus VOAux order cannot be derived by raising a vP of the kind in (3b) containing no Aux (Aux being merged in T) to SpecTP, for example. Such a derivation will give rise to surface AuxVO order, thanks to our interpretation of the PIC (or VAuxO order if V moves to v). The unmoved VP contained in vP is no longer present in the computation after the completion of a (nondefective) vP phase.

H. Taken together, the four assumptions given above give rise to a general, phase-based account of linearization which is rich in typological implications, particularly in its potential for giving a more principled account of the nature and cross-linguistic distribution of "mixed" word-order types than has hitherto been possible.

Merge: the derivation of structure and order
University of Groningen, Groningen (The Netherlands)
June 5th 2008

Label-free Bare Phrase Structure
Some consequences for Hale & Keyser's lexical-syntax

Ángel J. Gallego
CENTRE DE LINGÜÍSTICA TEÒRICA
Universitat Autònoma de Barcelona
University of Cambridge

In this talk I discuss Hale & Keyser's (1993 and subseq.) hypothesis that the limits of argument structure follow from restrictions on X-bar templates –i.e. from *unambiguous projection* of complements and specifiers– in the context of Chomsky's *Bare Phrase Structure* (BPS).

With Boeckx (2008a), I assume that Chomsky's BPS makes HK's pivotal claim virtually unstateable, a situation compounded by the lack of consensus about how to formulate the 'bare' theory. Here I want to develop the Boeckx–Chomsky label-free approach to BPS, where labels are not actually created / projected, but identified by minimal search (a third-factor metric), as shown by the algorithms below.

(1) Labelling algorithms
a. {H, α}, H an LI, H is the label
b. If α is internally merged to β, forming {α, β} then the label of β is the label of {α, β}
[from Chomsky 2008:11]

The label-free formulation above adopts the main insight of Collins' (2002) (iff *label* = Chomsky's *probe* = Collin's *locus*), though couched within a phase-cycle system where labels (like everything else; e.g. minimality) are identified at the phase-level, after the derivation has unfolded a bit. Assuming this conception, I take (2) to be a consequence of the weakly derivational nature of Chomsky's *Phase Theory*.

(2) Phase level labelling
The label of K is decided at the phase level

The presentation explores Boeckx–Chomsky's proposal in the context of HK's framework, and its consequences for their *l-syntax* vs. *s-syntax* cut, a distinction that presupposes a D-structure *level of representation* (or *component*, see Uriagereka 2008).

REFERENCES

Boeckx, C. 2008a. *Bare Syntax*. Oxford (New York): Oxford University Press.

Boeckx, C. 2008b. Elementary Syntactic Structures. A Minimalist Inquiry. Ms., Harvard University.

Chomsky, N. 2008. On Phases. In *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, R. Freidin, C. Otero, and M.L. Zubizarreta (eds.). Cambridge (Mass.): MIT Press.

Collins, C. 2002. Eliminating Labels. In *Derivation and Explanation in the Minimalist Program*, S. Epstein and T. Seely (eds.), 106-132. Malden (Mass.): Blackwell.

Hale, K. & Keyser, S. J. 1993. On the argument structure and the lexical expression of syntactic relations. In *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*, K. Hale and S. J. Keyser (eds.), 53-109. Cambridge (Mass.): MIT Press.

Uriagereka, J. 2008. *Syntactic Anchors. On Semantic Structuring*. Cambridge: Cambridge University Press.

MERGE AND ASYMMETRIC STRUCTURE BUILDING

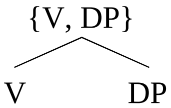
In this talk I provide a reanalysis of the labeling mechanism of *Bare Phrase Structure* (*cf.* Chomsky (1995)). I propose that labels are created derivationally to ‘repair’ illegible phrase structures. As a corollary, I will argue that argument structure is essentially an 'emergent' property, derived from the asymmetric nature of the labeled phrase structure.

Chomsky (1995) proposes that the output of the merger of two elements α and β is a labeled set {γ, {α, β}} where γ represents the label of the phrase. According to him, the justification of the need of the labeled structure would lie in interface phenomena. Contrarily, some recent works have challenged this idea arguing for a label-free syntax (*cf.* Collins (2002), Seely (2006)). However, I will show that the proposal of the label-free syntax has serious empirical drawbacks commenting briefly the properties of displacement, ellipsis, case marking, p-phrasing, incorporation, and argument structure arguing that all these phenomena make reference to syntactic labels. But given that, the following questions arise (1).

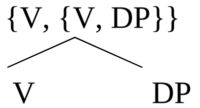
(1) (i) Why do syntactic phrases have labels? (ii) How do labels appear derivationally?
(iii) How do labels identify the set they label?

The idea that Merge is a purely symmetrical set-formation operation (*cf.* Chomsky (2005), Hinzen (2006a/b)) entails that in and of itself, the merger of (α, β) cannot give a labeled structure like {γ, {α, β}}, but a simpler {α, β} set. So, I will argue that the only way to get a labeled structure using just Merge and the lexicon is to have two instances of Merge where the first one creates a bare set and the second one provides it with the label (2).

(2a) Step1: α↔β = {α, β}



(2b) Step2: β↔{α, β} = {β, {α, β}}



However, since the notion of ‘labelhood’ is vague (after all, V is just one of the members of the {V, {V, DP}} set of (2b)), the ontology and consequences of labelhood will have to be explained. To answer the essential questions of (1i) and (1iii), my proposal will rely in the formulation of the hypothesis in (3).

(3) *Labels as a 'third factor' Hypothesis:* C_{HL} requires intensionally decidable sets/phrases.

Given such a restriction, labeling operations can be explained as *repairing* strategies: the label provides a set/phrase with a coherent intension (*i.e.* all of the members of the set are of a given nature). For instance, in the step1 of (2a), the simple {V, DP} set is created. The question is that, at this step, the set {V, DP} is heterogeneous: there is no grammatical category that can provide it a coherent type, and hence, by (3), it is illegible (assuming a Neodavidsonian conjunctivist semantics, in (2a) we have two unrelated monadic predicates (something like {kiss(e) & Mary(y)}). I will argue that the labeling mechanism provides the step from this adjunct-like syntax of conjunction of independent predicates to the hierarchical predicate-argument syntax based on labels (*cf.* also Hornstein (2005), Hornstein & Nunes & (2008) and Hinzen (2006a/b)): having {V, DP} in (2a), the verbal head (the syntactically active *locus*) is remerged with the structure to give it a coherent type (2b). Now an asymmetry emerges in the new set; crucially, both members of {V, {V, DP}} will have a verbal character (both contain a [+V] categorial feature). Thus, the set {V, {V, DP}} labeled with an event type (*i.e.*, a verbal intension) is readable at the interfaces.

We are left with a last problem though: the primitive {V, DP} of (2a) (now, a member of {V, {V, DP}}) is still an illegible object as defined by (3). And obviously, recursion on the labeling strategy won't solve the problem, since this only breaks the symmetry among the members of the highest phrase. Here my proposal is a purely repairing strategy: the DP that as such is interpretable as an individual (*i.e*, Val(y, *Mary*) *iff* Mary(y)) is now in a verbal/eventish environment at the highest phrase (a VP), that converges at the interfaces. Thus, the solution to the VP-contained DP is to lift its type (*à la* Pietroski (2005)) to accommodate its type to that of the intension of the highest set that contains it: this is a lifting of the type of the DP complement of V from an individual-denoting type to an *event-participant* one (4).

(4) Val(y, Mary) *iff* Mary(y) → Val(e, int-Mary) *iff* Theme(e, Mary))

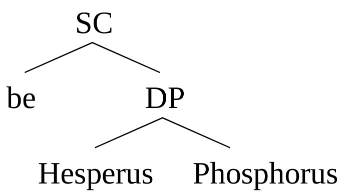
Thus, I argue, it will be this type-lifting what will turn an individual-denoting expression like 'Mary' into an argument of a predicate when it is contained within the projection of the predicate.

As I will show, this analysis accounts naturally for the mysterious necessity of a strictly local configuration for predicate-argument relations: it won't be the case that verbs have to take arguments in local configurations but quite the opposite; argument structure *emerges* as the outcome of a repair strategy when heterogeneous {V, DP}-like sets (semantically {kiss(e) & Mary(y)}) are readjusted for convergence at the interfaces. The *configurationality* of predicate-argument relationships (*cf.* Hale & Keyser (2003), Uriagereka (2008)) is, thus, derived automatically.

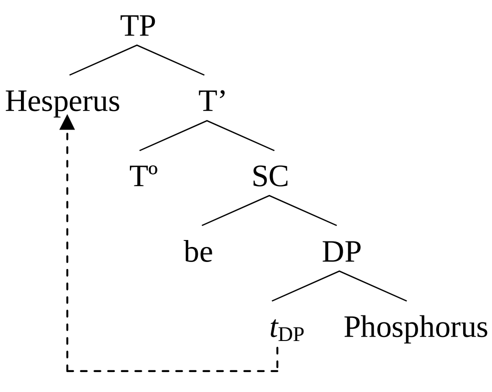
Furthermore, I will also show that the system can account for the opposite pattern; the lack of theta role assignment in some constructions like copulatives (*cf.* Hurford (2007)). Assuming that in a copulative construction like (5a) we have a Small Clause-like configuration (5b) with further raising of one of the DPs (5c) (*cf.* Moro (2000)), we can account for the fact that the DPs of these constructions lack any theta-role. None of the DPs but the phrase containing both of them will be merged with the copula, hence, no type shifting will be required for any of the DPs, which will make them not to be separate arguments of any verb.

(5a) Hesperus is Phosphorus.

(5b)



(5c)



Finally, I will extend this analysis to account for the problematic nature of the argument structure of constructions with *unsupported clauses* like those in (6a/b) (*cf.* Higginbotham (1983)).

(6a) I consider John a fool.
(6b) I found Mary in the library.

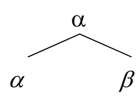
Symmetry in Syntax? Symmetric Merge and Symmetric Move

Barbara Citko (University of Washington)

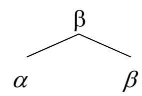
While there has been a lot of research on asymmetry and antisymmetry in syntax, symmetry has been mostly ignored or claimed to be outright non-existent (Kayne 1994, Chomsky 1995, Moro 2000, Di Sciullo 2005). This focus on asymmetries and antisymmetries is somewhat surprising from a biolinguistic perspective, which seeks to integrate linguistics with the natural sciences (Boeckx and Piattelli-Palmarini (in press), Chomsky 2005, Jenkins 2000, among others), in which symmetry tends to be the norm and departures from symmetry are something that needs an explanation (Brody 2006). My main goal in this talk is to remedy this gap, by motivating the need for research on symmetry in syntax, and providing evidence for symmetric aspects of the two fundamental syntactic operations: Merge and Move. I will start by reviewing the arguments that led syntacticians to believe that Merge and Move are asymmetric.

The asymmetry of Merge is twofold. First, given Kayne's (1994) Antisymmetry Theory, Merge can only create structures in which all terminal nodes are in asymmetric relationships, since only such structures can be linearized. This excludes structures with mutual c-command between terminal nodes, for example. Second, labeling, which is thought to be part of Merge, is asymmetric in that only one of the two merged elements projects as the label of the newly formed constituent (Chomsky 1995:244). For example, if two elements α and β merge, either α or β can project as the label, as shown in (1a-b).

1) a.



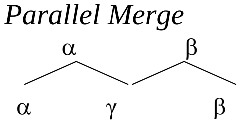
b.



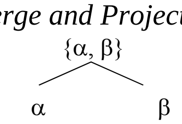
Various relativized minimality effects led us to believe that Move is also asymmetric. Well-known cases involve superiority effects, head movement constraint, and superraising. In all of them, only the higher of the two potentially movable elements may move.

In this talk, I show that, contrary to standard assumptions outlined above, both Merge and Move can be symmetric as well. I discuss cases two cases of symmetric Merge: *Parallel Merge* (Merge that creates structures with symmetric c-command relationships) and *Merge and Project Both* (Merge in which the labels of both α and β symmetrically project as the label of the new constituent). The two are illustrated in (2a) and (2b), respectively. Empirical evidence in favor of Parallel Merge comes from accross-the-board wh-questions (Citko 2005), conjoined wh wh-questions (Citko 2008, Gracanin-Yuksek 2007), serial verb constructions (Hiraiwa and Bodomo 2008), and right node raising constructions (Wilder 1999, Bachrach and Katzir to appear).

2) a. *Parallel Merge*



b. *Merge and Project Both*



Chomsky (1995) suggests *Merge and Project Both* is a theoretical possibility, as long as the two projecting elements do not conflict in relevant features. To the best of my knowledge, the empirical consequences of this theoretical possibility have not yet been investigated. I argue that this is what is involved in the derivation of comparative conditionals.

Next, I turn to cases of symmetric Move. Symmetric Move is Move that can target two elements with equally grammatical results. I discuss two cases of symmetric Move: passive movement in double object constructions and wh-movement in multiple wh-questions. In some languages, passivization is symmetric in that it can target either the direct or the indirect object, whereas in others it is asymmetric in that it only target one of them (Anagnostopoulou 2003, McGinnis 2001, Woolford 1993, Marantz 1993, Baker 1988, among many others). Wh-movement in multiple wh-questions exhibits similar variation; in some languages it is symmetric in that it target any wh-element in a clause (with no superiority violations), whereas in others it is asymmetric in that it has to target a higher one. The issue I conclude the talk with is the correlation (or lack thereof) between the availability of symmetric A movement and the availability of symmetric A-bar movement.

Order but no Structure in Bonobo English

Robert Truswell, Tufts University
robert.truswell@tufts.edu

Generative research since the 1950s has focused on the relationship between order and structure: for example, one major difference between regular grammars and context-free grammars is that the latter, but not the former, can assign some nontrivial hierarchical constituent structure to a string of terminal elements. However, Merge, as originally defined by Chomsky, is notably incapable of representing either order or nontrivial structure. Chomsky's Merge is an operation that, initially, takes two elements from the numeration and forms a set from them, and subsequently, takes one element from the numeration and forms a set from that element and the previously formed syntactic object. This recursive application of Merge is already sufficient to yield a discrete infinity, but it gives no representation of order (these are unordered sets) and, if the numeration consists only lexical items, no representation of nontrivial constituent structure. In fact, in one sense, structures created by Merge alone are weaker than those that can be generated by regular grammars.

Not surprisingly, then, most of the action in minimalist research concentrates on other operations. Most importantly for our purposes, order is created by linearisation of the terminals of the unordered set created by Merge. This linearisation can also in principle create some non-finite-state structures, if the LCA doesn't hold (e.g. (1a), created by Merge, may be linearisable as (1b), with a mixture of left- and right-branching which cannot be represented in a regular grammar).

(1a) {a, {b, {c.d}}}
(1b) [a [[c d] b]]

However, constituent structure can be most fully represented by considering the interactions of syntactic objects created in multiple derivational workspaces. Multiple derivational workspaces are necessary whenever we have two syntactically complex sisters, as in (2): the subject NP is assembled in one workspace and then joined to the VP which has been independently assembled in the other.

(2) {{The, engineers}, {build, aeroplanes}}

This requires a more nuanced conception of recursion in the minimalist program. Much of the expressive power of natural language can only be captured by assuming something broadly along the lines of Renumeration in the sense of Johnson (2002): the numeration feeds structure-building, but this structure-building can feed back into the numeration, to be subsequently added to a *different* structure. This is recursion at a coarser grain from that given by repeated application of Merge alone.

In this talk, I contend that this coarser notion of recursion is necessary if we are to make any sense at all of Hauser, Chomsky & Fitch's (2002) claim that recursion is the core part of FLN, the uniquely human, domain-specific elements of the language faculty. My evidence comes from responses made by Kanzi, the bonobo, to 660 spoken English instructions (Savage-Rumbaugh *et al* 1993). Kanzi shows clear evidence for sensitivity to linear order, and so we have to attribute *some* syntactic ability to him. However, his comprehension falters drastically in two areas, namely NP-coordination and function words. These are the only two cases in the corpus where any reasonable syntax-semantics mapping procedure requires a notion of hierarchical constituent structure in the syntax. The natural conclusion is that Kanzi is unable to assign such structures to input strings: he can deal with order (as regulated by Merge and linearisation), but not with the type of structure that requires multiple derivational workspaces.