University of Groningen

Recursion in language:

Zwart, Cornelius

Published in:
Biolinguistics

Publication date:
2011

Recursion in Language:
A Layered-Derivation Approach

Jan-Wouter Zwart

This paper argues that recursion in language is to be understood not in terms of embedding, but in terms of derivational layering. A construction is recursive if part of its input is the output of a separate derivational layer. Complex clauses may be derived recursively in this sense, but also iteratively, suggesting that standard arguments for or against recursion in language are misdirected. More generally, we cannot tell that a grammar is recursive by simply looking at its output; we have to know about the generative procedure. Using the new definition of recursion in terms of derivational layering, we once again inspect the recorded data of Pirahã, arguing that there is reason to believe that the grammar of Pirahã is recursive after all.

Keywords: cyclicity; derivation; layered derivation; Pirahã; recursion

1.   Spotting Recursion

A procedure is recursive if part of it involves running the entire procedure anew. The output of such a procedure is also called recursive. For example, the procedure that draws the famous Droste can picture in (1) below at some point involves calling the very same procedure, in order to draw that can *in* the picture.

However, one cannot tell that an object is recursive by simply looking at it: One has to know by what procedure it was derived. For example, the picture in (1) could have been generated in a non-recursive way (and it probably was), for instance by assigning a color to each pixel.

In another example, consider the situation in (2), from the comic strip 'Calvin and Hobbes', where the boy Calvin finds himself surrounded by five duplicates of

himself. As Calvin exclaims, the procedure bringing this situation about was recursive: Calvin made a duplicate of himself, and the duplicate proceeded to make a duplicate of *him*self, etc. However, Calvin could have created five duplicates by simply repeating the duplication process five times, yielding an iterative rather than a recursive procedure.

(1)
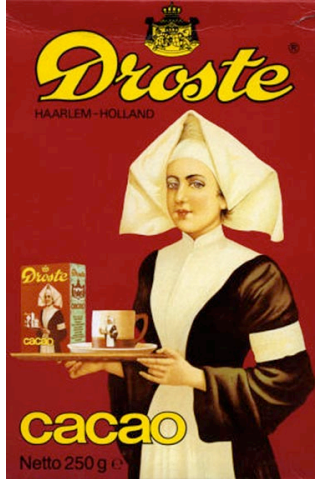
(2)
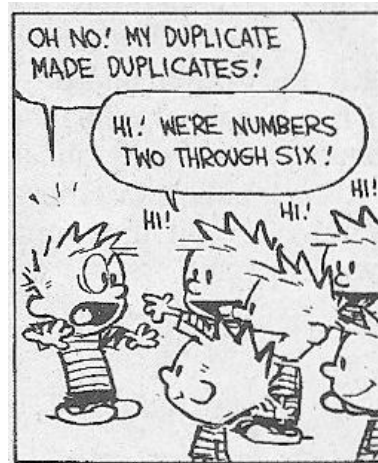
In language, it is commonly taken for granted that structures involving some kind of embedding are recursive. Typical examples are in (3):

(3)   a.   John thinks that Mary said that it was raining.
       b.   The height of the letters on the cover of reports from the government.

The structures in (3) are recursive if they are generated by a recursive procedure, such as the rewrite rules of pre-minimalist generative grammar. The procedure generating (3a) may be paraphrased as in (4), showing recursion in that to interpret 'clause' in (4b), one must run (4a) again.

(4)   a.   clause = subject + predicate
       b.   predicate = verb + clause

Similarly with the procedure in (5) generating (3b):

(5)   a.   determiner phrase = determiner + noun phrase
       b.   noun phrase = noun + preposition phrase
       c.   preposition phrase = preposition + determiner phrase

For this reason, embedding is taken to signal recursion, underlying much of the debate between Everett (2005, 2009) and Nevins *et al.* (2009) on recursion as a

defining property of natural language. It will be recalled that Everett (2005) observed that the Amazonian language Pirahã lacks embedding, which was contested by Nevins *et al.* (2009) but upheld by Everett (2009). In the original 2005 article, the connection of embedding and recursion was not made (except in a comment by Michael Tomasello), but in his 2009 reply to Nevins *et al.*, the absence of recursion in the grammar of Pirahã was made the centerpiece of Everett's claim. At the background of this discussion is the hypothesis of Hauser *et al.* (2002) that recursion is the defining property of the faculty of language.

Meanwhile it should be clear that the model of grammar as understood in current minimalism no longer involves rewrite rules of the type in (4)/(5). Instead, structure is created by a single operation Merge, which Chomsky (1995) defines as taking two elements and combining them in a set:

(6)   *Merge*

    i.   select x
    ii.   select y
    iii.   create $\{x, y\}$

Merge may be (re)written as a rewrite rule:

(7)   *Merge*

    x, y = {x, y}

It is (mostly implicitly) understood that the next time Merge runs, it takes the output of the previous run as part of its input. Thus, x or y on the left-hand side in (7) may be {x, y}, yielding recursion. For example, if the next run combines {x, y} with z, the structure in (8) results.

(8)   {z, {x, y}}

         z   {x, y}

                x   y
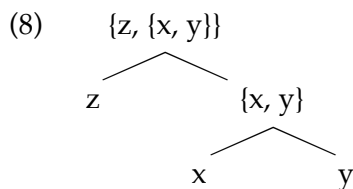
(8) is a familiar binary branching phrase structure (where {x, y} may be written as the projection of x, X', and {z, {x, y}} as the maximal projection *XP*). Such molecular structures are arguably the building blocks of all natural language phrases (including those in (3)), suggesting that every phrase consisting of more than two elements is the output of a recursive procedure (Nevins *et al.* 2009: 366).

However, just like Calvin's band of duplicates, the phrase structure in (8) may be generated by an iterative procedure as well. For instance, we may think of the structure-building process as a transfer procedure, moving elements from a resource (the lexicon, or a subset of the lexicon, or some other collection of elements called *Numeration* in Chomsky 1995) to a workspace (the structure under construction).

This procedure is spelled out in (9):

(9)   *Transfer*
        i.   Move y from the Numeration to the workspace.
        ii.   Move x from the Numeration to the workspace.
        iii.   Move z from the Numeration to the workspace.

One could then define a constituent as the state of the workspace after each step.

An operation like Transfer is clearly iterative. It simply ticks off members of the resource set (the Numeration), and yet it generates the same structures as the recursive procedure that yields (8), essentially because the workspace is defined as the list of elements ticked off, and constituents are defined derivationally as subsequent stages of that list. The example illustrates that we cannot take for granted that binary branching structure involves a recursive procedure.

The point of the example is not to argue that Merge does not exist or that Transfer is superior to Merge. The point is that Merge and Transfer are equivalent in important respects, and though Merge may be a useful shorthand for Transfer, recursion should be identified in terms of the process, not its notation.

More generally, we cannot decide that a language (or natural language) 'is recursive' by simply looking at its structures. We have to know about the procedure by which these structures are derived (see also Everett 2009: 438).

2.   Layered Derivations

There is ample reason to believe that the procedure generating clauses must be simple. Constituency tests generally yield the result that structures are binary branching (Kayne 1994), and structure-to-order conversion is not random but more or less automatic (Zwart 2011). This suggests a simple, stepwise procedure involving no more than two elements in each step, creating structure incrementally in a way that can easily be tracked by interpretive components of the mind/brain.

I have argued elsewhere that transfer (or a similar procedure) is in fact simpler than Merge, in that it identifies in each step a single element to be manipulated (transferred), while keeping the destination of transfer constant throughout the derivation (i.e. there is a single workspace). This yields binary branching structures with asymmetric sister pairs, essentially a nest of ordered pairs which can be linearized straightforwardly (along the lines of Fortuny 2008; see Zwart 2009, 2011).

One can think of transfer as an iterative procedure selecting (or identifying) one element at a time from the Numeration, creating a record of elements identified. As argued in Zwart (2009), this procedure can be viewed as working from the top down, splitting the Numeration in an element identified and a residue set. For example, (8) can be derived by starting from the Numeration in (10), via the steps in (11).

(10)   Numeration = {x, y, z}

(11)   *Transfer*

    i.   split off *z* yielding ⟨z, {x, y}⟩
    ii.   split off *x* yielding ⟨z, ⟨x, {y}⟩⟩
    iii.   split off *y* yielding ⟨z, ⟨x, ⟨y, {∅}⟩⟩⟩

       It is easy to see that transfer can yield embedding structures without recursion (see also Christiansen 1994, referred to in Parker 2006: 184). For example, (3a) can be derived from the Numeration in (12) by iterative splitting:

(12)   Numeration = {John, thinks, that, Mary, said, that, it, was, raining}

Embedding, then, does not betray recursion if the structure building procedure is as simple as suggested here.
       However, it is clear that not all sequences can be derived by iteratively splitting off the elements in the sequence from a Numeration containing them. Essentially, every complex non-complement (e.g., a subject) must be included in the Numeration as an atom. To see this, consider the derivation of (13).

(13)   The man kicked the ball.

Constituency tests show that *the man* and *kicked the ball* in (13) are constituents (Chomsky 1964: 983). For example, *kick the ball* can be isolated in VP-fronting constructions like (14).

(14)   The man said he would kick the ball, and *kick the ball* he did.

Likewise, *the man* can be isolated in the backgrounding construction in (15).

(15)   He kicked the ball, *the man*.

But the sequence *man kick(ed) the ball* cannot be identified as a constituent by any known test.
       The procedure generating (13), then, must yield a structure in which *the* and *man* are joined together before *the man* is joined with the rest of the clause. Iterative transfer starting from the Numeration in (16) could not yield this result, as can be seen after the first application of split in (17a).

(16)   Numeration = {the, man, kicked, the ball}

(17)   *Transfer*

    i.   split off *the* yielding ⟨the, {man, kicked, the ball}⟩

It must be, then, that *the man* is included as a single item in the Numeration under-

lying (13):

(18)   Numeration = {[the man], kicked, the ball}

Transfer then yields the correct result in terms of constituent structure:

(19)   *Transfer*

    i.   split off *the man* yielding ⟨[the man], {kicked, the, ball}⟩

It follows that *the man* must have been created in a separate derivation (or derivational layer), with the Numeration in (20) and the iterative transfer procedure in (21).

(20)   Numeration = {the, man}

(21)   *Transfer*

    i.   split off *the* yielding ⟨the, {man}⟩
    ii.   split off *man* yielding ⟨the, ⟨man, {∅}⟩⟩

The output of this derivation is included in the Numeration for the next derivation (i.e. (18)).

It should be clear now that the derivation as a whole, including the layers (20)/ (21) and (18)/(19), is recursive: The output of the procedure (20)/(21) is part of the input (the Numeration) of the same procedure in (18)/(19). If this is the correct approach, recursion is not evidenced (necessarily) by embedding, but it is evidenced (necessarily) by left branch embedding (sometimes called 'left-tail recursion'). (An alternative derivation starting from the Numeration in (16) merges *the* and *man* and *kicked, the*, and *ball* in two parallel derivations with outputs of the parallel derivations to be joined afterwards. In this alternative, a Numeration is associated with multiple workspaces, creating added complexity.)

3.   Center-Embedding

Recursion in language is also typically illustrated by examples like (22), involving center-embedding.

(22)   The dog the cat bit barked.

Here, *the cat bit* is a restrictive relative clause modifying *the dog*. The constituency of (22) is as indicated in (23):

(23)   [[the dog [the cat bit]] barked]

In (22)/(23), we again see a complex subject (*the dog the cat bit*) which must have been derived in a separate derivational layer. The structure of (22)/(23), then, is recursive already for the same reason that the structure of (13) is. Center-embedding reduces to left branch embedding (and 'nested recursion' is in a natural class with 'left-tail recursion', if we are right).

It has been observed that center-embedding cannot be performed indefinitely, unlike right-branch embedding (Yngve 1961). Consider the triple embedding in (24), contrasting markedly with the triple embedding in (3a).

(24)   [[the dog [the cat [the man kicked] bit]] barked]

It seems, then, that recursion (as understood here) comes with a cost (perhaps in memory load, as Yngve suggested). Much better again are (25) and (26):

(25)   The dog that the cat bit that the man kicked barked.

(26)   The dog that bit the man that kicked the cat barked.

The difference may be accounted for by the fact that (25) and (26) contain fewer derivational layers (due to the right-branch embedding) and hence less recursion.

4.   Interface Effects between Derivational Layers

If recursion in language is correctly defined as the interaction between derivational layers, it is easy to identify other phenomena that signal recursion in this sense. Assuming the model of grammar entertained in minimalism, each derivation (sequence of operations Merge) feeds the interface components dealing with sound and meaning, and hence we may expect the output of a derivational layer to show idiosyncratic sound/meaning properties. Moreover, we expect the output of a separate derivational layer to behave as an atom in the context of the next derivational layer: Constituents of the output of a separate derivation are not themselves in the numeration for the next derivation, and therefore cannot be merged individually in the context of that next derivation. This, I believe, derives a wide range of opacity effects, as first observed in Toyoshima (1997).

Among the interface effects affecting derivational layer outputs are (cf. Zwart 2009):

(27)   a.   morphological effects (incorporation, conflation, fusion)
       b.   atomization (opacity)
       c.   idiosyncratic meaning (idioms)
       d.   linearization effects (idiosyncratic order, template effects)
       e.   categorization effects (reanalysis)

The effects in (27) all show a mix of syntactic and lexical properties. For instance, complex verbs (e.g., involving a transitivizing morpheme, possibly covert, like the 'little *v*' of Chomsky 1995) are clearly structured, but also clearly lexical (in obeying lexical integrity, for instance). The mixed properties are accounted for if such elements are created in a separate derivational layer, and pass through the interface components before being enlisted in the Numeration for the next derivational layer. If this is the correct approach, then every structured lexical item that is used in a larger syntactic context betrays recursion in the sense understood here.

In view of this, I would like to define 'lexical' in relative terms, i.e. in the context of a derivational layer, as in (28). (Note that under this definition, complex subjects are lexical items.)

(28) *Lexical*

  x is a lexical item for derivation D of numeration N if x is included in N as a single item.

5.   Recursion in Pirahã

The question now arises to what extent a language like Pirahã shows signs of recursion in the sense of derivational layering. I will try to answer the question, using data from Everett (1986) that are uncontested (and confirmed by Dan Everett, p.c.), by looking for (i) complex subjects and (ii) structured lexical items (in the sense of (28)) showing interface effects.

5.1.   Complex Subjects

I believe it is uncontested that Pirahã does have phrases, and therefore, if Pirahã lacked recursion (in our sense), we would expect phrases to show up only in complement position, not in subject or adjunct position. Indeed, the large majority of the subjects in the examples of Everett (1986) appear to be single-word items (though they may be nominalizations or compounds, for which see below). But even so, several examples of complex subjects can be found in the examples (numbers refer to the example numbers in Everett 1986):

(29) *Complex subjects in Pirahã* (not including nominalizations / compounds)

  a. **Xipoógi hoáoíi** hi xaagá.   (22)
      Xipoógi shotgun 3 be
      'That is Xipoógi's shotgun.'

  b. **Xoogiái hi xapisí** biga aí big-á.   (43)
      Xoogiái 3 arm   thick be thick-EMPH
      'Xoogiái's arm is thick (i.e. strong), very strong.'

c. **giopaí gáihi** kapióxio xigiábií. (85)
   dog that other like
   'That dog looks like another (dog).'

d. **ti xahaigí** gáihi. (196a)
   1 brother that
   'That (one) is my brother.'

e. **kaoí xahaigí** gáihi. (196b)
   who brother that
   'Whose brother is that?'

f. **baaí xaíbaí** pii ap-ái-p-i pii bo-ó gai kob-á.
   wild pig many water enter-ATELIC-IMPF-PROX water up-LOC DEM see-REM
   'A herd of pigs is entering the water upriver, look!' (277)

g. **Xoogiái hi go-ó hoasígikoí bíib-i híx** hoasígikoí
   Xoogiái 3 WH-OBL lead shot send-PROX C lead shot
   koab-áo-b-í-i. (282)
   run.out-TELIC-PERF-PROX-EVID
   'The lead shot which Xoogiái sent ran out.'

Of these, (29a,d,e) are copular constructions, which might allow another analysis in which the boldface noun phrase is a predicate. I have no idea about the plausibility of either analysis. Everett (2009: 419) comments on the copula-less type in (29d) that it is not a clause but just a string of nouns. But this does not affect the argument, unless the string looks like (30a) rather than (30b).

(30) a. Me, brother, that.
     b. [My brother], that.

If it is (30b), then, assuming iterative Merge and layered derivations, *my brother* must still be the output of a separate derivation.

Example (29g) involves a relative clause where it looks like the boldface material is not actually the subject, but a pre-posed topic, perhaps juxtaposed to the main clause. In that case, the boldface material would constitute a complex adjunct, which would have to be the output of a separate derivation under the assumptions entertained here.

5.2. *Complex Lexical Items*

Pirahã verbal morphology is fairly complex, and described in Everett (1986: 288–289) as templatic, featuring 18 slots following the verb root. None of these slots are occupied by inflectional morphemes. The categories represented include aspect, negation,

interrogativity, deixis, mood, but not tense. The complete template as proposed by Everett (1986) is in (31).

(31) *Pirahã verbal template* (all following the root)

1. incorporation	7. continuative	13. frustrative
2. duration	8. interrogative	14. intensive
3. telicity	9. ingressive	15. emphatic
4. perfectivity	10. deictic	16. complementizer/nominalizer
5. desiderative	11. iterative	17. evidential
6. negation	12. certainty	18. result

The first slot behind the root is reserved for incorporation (mainly of verb roots), "an extremely productive method of forming new verbs" (Everett 1986: 300–301).

Examples of complex verbs derived via incorporation are:

(32) *Pirahã incorporation*

a. xab   op   (388a)
    turn   go
    'return, arrive'

b. xiga   hoag   (388b)
    take   come
    'bring'

c. xig   ab   op   (388c)
    take   turn   go
    'bring back'

As Everett (1986: 301) notes, the complex verb is treated as a single unit: Neither root can take any affixes, and "suffixes are added to the entire stem as one element". I take this to entail that the incorporated verbs in Pirahã are prototypical structured lexical items as discussed above, i.e. outputs of separate derivational layers.

The etymology of the affixes is not generally elucidated in Everett (1986), but in a few cases he notes that the verbal suffixes are grammaticalized lexical items. For example, the evidential suffix *-xáagahá* 'OBSERVATION' is analyzed as involving *xaagá* 'be' and *–há* 'COMPLETE CERTAINTY' (Everett 1986: 298).

(33) *–xáagahá*   < *xaagá* + *–há*.
        OBSERVATION   *be*   *COMPLETE CERTAINTY*

This suggests that the verbal complex is the result of conflation as discussed in Hale & Keyser (2002), a typical syntactic process creating lexical items.

The complementizer/nominalizer slot 16 in (31) is occupied by *–sai*, a very productive morpheme for nominalizing verb phrases (Everett 1986: 277f.) (there is

also another nominalizer, *si*). One of its uses is quotative, affixed to a verb of saying, suggesting that it may be an embedding complementizer (cf. Nevins *et al*. 2009: 382).

(34) ti gái-sai   kó'oí   hi kaháp-ií.   (Everett 2005: 629)
      1   say-NOM Kó'oi   3   leave-INTENTION
      'I said that Kó'oí intends to leave.'

But Everett (2009: 418) takes *–sai* to be a marker of old information, allowing a juxtaposition rather than subordination analysis of the type in (34), where the first clause represents information that is familiar from the discourse setting (presumably akin to backgrounded quotatives in English).

Far more important to the discussion at hand, it seems to me, is the observation that *sai* turns verb phrases into nouns, the kind of process that betrays derivational layering (27e). Examples abound:

(35) *Pirahã nominalization*

      a.   kahaí   kai-sai.   (80b)
            arrow   make-NOM
            'arrow making, arrow maker'

      b.   **xií kai-sai**   hiaba.   (200)
            thing make-NOM   NEG
            'This is not a factory.'

      c.   agaoakait-i-sai   (243)
            canoe   bore-LINK-NOM
            'canoe-boring-thing'

      d.   **tiobáhai hóoi ai-sai**   xabahíoxoi.   (262)
            child   bow   make-NOM   incorrect
            'Children's bow making is incorrect.'

      e.   **ko kab-i-si**   baósaápisi bag-áo-b-á-há.   (280)
            eye NEG-LINK-NOM hammock   sell-TELIC-PERF-DIST-EVID
            'The man without eyes (blind one) sold the hammock.'

      f.   gahió   **pi-ó**   **xabaip-i-sai**   (288)
            airplane   water-LOC   sit-LINK-NOM
            'Hydroplane'

In a layered derivation approach, nominalizations are perfectly regular syntactic constructions, merged with a nominalizing morpheme, and then turned over to the interfaces, acquiring idiosyncratic sound/meaning properties. At the interface, the output receives a new categorial feature (N), not transparently derived from any of

its constituent parts. The resulting unit can be included as a single item in the Numeration for a next derivational layer.

In this context, it is important to note that the nominalizing process is instrumental in creating Pirahã names:

> All names for people are derived from verbal constructions, animal names, nominal phrases, etc. In about 90% of these cases, *–si* occurs optionally in morpheme final position, as though marking a change in the basic reference or function. (Everett 1986: 279–280)

A final category that suggests derivational layering in the grammar of Pirahã involves compounds. Everett (1986) lists numerous compounds, but Everett (2009: 423–424) appears to argue that these are not really compounds, admitting that:

> If there were compounding in Pirahã, this would be clear evidence for recursion. (Everett 2009: 423)

Let us first look at the examples:

(36) *Compounds in Pirahã*

   a. **xagí gahióo** xogií ái-xi-xi   pii   xigiábií.   (86)
       path   airplane big   be-EMPH-EMPH water like
       'The **airstrip** is big, like a river.'

   b. xogaogií   < xogaí + ogií   (389)
       big field   field   big

   c. xabagisoixaoxoisai < xabagi   + soixaoxoisai   (477)
       saw   toucan   beak

   d. xapaítoii   <   xapaí + toii   (478)
       ladder   foot   handle

   e. pigáía ‹ pi   + gáía   (481)
       scissors   thorn   crooked

   f. kaogiái   <   kao   + ogiái   (482)
       [kind of bass]   mouth   big

These all seem clear cases of compounds, in most cases acquiring the idiosyncratic (non-compositional or metaphoric) meaning suggestive of derivational layering.

Everett (1986: 322) describes his grounds for classifying formations like those in (36) as compounds as semantic: The non-compositional meaning suggests they are lexical items rather than phrases. Later, Everett (2009: 423–424) withdraws the semantic argument and details the prosodic properties of the formations in (36), suggesting to him that they are not compounds. With the remarks of Everett (1986: 322) in mind (see below), we may conclude that he considers (36) not to involve compounds but (syntactic) phrases.

> The criterion to classify the examples to follow as compound words rather than merely phrasal constructions is semantic. (Everett 1986: 322)

In the layered derivations approach, there is no fundamental distinction between 'compound words' and 'phrasal constructions'. What is relevant is the complexity of the string (suggesting derivation output status) and its behavior as a single item in the context of a (subsequent) derivation. The idiosyncratic meaning merely provides an additional argument for these elements' derivational history, regardless their status as words or phrases.

In this context we may also point to complex locatives in Pirahã, suggestive of the kind of fusion that we expect to occur at the interface between derivational layers (cf. (27a)).

(37)   *Complex locatives in Pirahã*

   a.   xoí   'jungle'   (326)
   b.   xo-ó   'in the jungle'
   c.   xo-ó-xio   'into the jungle'

6.   Conclusion

In conclusion, if recursion is identified in terms of derivational layering, as proposed here, then it seems clear that the grammar of Pirahã is recursive. Both complex subjects and complex lexical items are attested, some in average quantity, some in abundance.

This raises the further question what the observations in Everett (2005), substantiated in (2009), about the absence of embedding imply. Clearly, the wide-ranging implications having to do with the nature of the faculty of language vanish, but the original conclusions of Everett (2005) were considerably less bold and may still be valuable. One suggestion to make, within the model of grammar considered here, is that while interaction among derivations is unaffected by whatever cultural constraints are at play, the size of the Numeration is. Recall from section 2 that embedding structures like (3) can be derived by iterated Merge, starting from a large enough Numeration. Perhaps the 'immediacy of experience principle' that Everett (2005) suggested to capture the cultural constraints on Pirahã grammar (and cognition) constrains the Numeration in ways that virtually preclude ordinary right-branch embedding.

References

Chomsky, Noam. 1964. The logical basis of linguistic theory. In Horace G. Lunt (ed.),

Proceedings of the Ninth International Congress of Linguists, Cambridge, Mass., August 27–31, 1962, 914–1008. The Hague: Mouton.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Christiansen, Morten H. 1994. Infinite languages, finite minds: Connectionism, learning and linguistic structure. Edinburgh: University of Edinburgh dissertation.

Everett, Daniel. 1986. Pirahã. In Desmond C. Derbyshire & Geoffrey K. Pullum (eds.), *Handbook of Amazonian Languages*, vol. 1, 200–325. Berlin: Mouton de Gruyter.

Everett, Daniel. 2005. Cultural constraints on grammar and cognition in Pirahã: Another look at the design features of human language. *Current Anthropology* 46, 620–646.

Everett, Daniel. 2009. Pirahã culture and grammar: A response to some criticisms. *Language* 85, 405–442.

Fortuny, Jordi. 2008. *The Emergence of Structure in Syntax*. Amsterdam: John Benjamins.

Hale, Ken & Samuel J. Keyser. 2002. *Prolegomenon to a Theory of Argument Structure*. Cambridge, MA: MIT Press.

Hauser, Marc D., Noam Chomsky & W. Tecumseh Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science* 198, 1569–1579.

Kayne, Richard S. 1994. *Connectedness and Binary Branching*. Dordrecht: Foris.

Nevins, Andrew, David Pesetsky & Cilene Rodrigues. 2009. Pirahã exceptionality: A reassessment. *Language* 85, 255–404.

Parker, Anna R. 2006. Evolution as a constraint on theories of syntax: The case against minimalism. Edinburgh: University of Edinburgh dissertation.

Toyoshima, Takashi. 1997. Derivational CED. *WCCFL* 15, 505–519.

Yngve, Victor. 1961. The depth hypothesis. *Proceedings of Symposia in Applied Mathematics* 12, 130–138.

Zwart, Jan-Wouter. 2009. Prospects for top-down derivation. *Catalan Journal of Linguistics* 8, 161–187.

Zwart, Jan-Wouter. 2011. Structure and order: Asymmetric merge. In Cedric Boeckx (ed.), *Oxford Handbook of Linguistic Minimalism*, 96–118. Oxford: Oxford University Press.

Jan-Wouter Zwart
Rijksuniversiteit Groningen
Center for Language and Cognition Groningen
P.O. Box 716
9700 AS Groningen
The Netherlands
c.j.w.zwart@rug.nl