One dimensional syntax

Michael Brody

0.

I explore in this paper the idea that syntax is one dimensional, its domain is the p-string, the path from the initial symbol to the terminal element. P strings consist of nodes/features linearly ordered by the precedence relation. Sentences correspond to a set of (connected) syntactic structures. Constituents in the usual sense, invisible for syntax, are visible only for the interpretive systems. In the approach sketched c-command and antisymmetry are consequences of a necessary monotony property of dependencies in an impoverished grammar, the property that requires fillers to precede gaps. Furthermore mirror symmetry properties, the centrality of c-command and the existence and ubiquity of the puzzling duplications involved in the various labeling relations (local agreement, Criteria, projection) become expected and in part conceptually near-inevitable.

1. Antisymmetry

Let us assume (for the moment) that the Kaynean (1994) descriptive assumptions about antisymmetry are correct: hierarchy maps into precedence as specified in (1):

(1) If x asymmetrically c-commands y, then x (and whatever x dominates) precedes y

Of course if antisymmetry holds, we would like to know why. The original explanations given in Kayne (1994) were problematic in various respects, as was discussed at the time in Brody (1996) (1997), Chomsky (1995) and others and more recently rehearsed in Abels and Neeleman (2012). Kayne himself has since essentially abandoned his early approach and proposed an alternative account. Kayne (2010) proposes to derive the antisymmetry of syntactic structures from syntactic spec-head complement order in direct contrast to his early attempt where he attempted to derive the latter from the former. This is exactly the way mirror theory (Brody 1997, 2000) approached the problem.

Kayne suggest (i) that probes precede goals in syntax because they must do so in production and parsing systems and (ii) that the complement always contains a goal hence complements must be on the right of the head.

The assumption that there is a probe goal asymmetry is widely shared but its truth is much less certain than that of the factual existence of antisymmetric effects in syntax. Even if there is a probe-goal asymmetry it is not clear that indeed probes search for goals and not the other way around. Also one wonders why it is precisely the probe goal order of linguistic performance rather than any other randomly chosen performance order that needs to be reproduced in syntax. The assumption that selection is accomplished by a probe goal mechanism, while interesting, brings its own problems, in particular those having to do with the strict locality of the selection relation. It is perhaps not unfair to summarize these remarks by saying that of the two orders of spec-head and head-complement Kayne stipulates one, perhaps with a potentially interesting rationalization. Again this is exactly parallel to mirror theory where also one of the two orders, there the spec-head order, was explicitely stipulated.

Kayne continues by assuming that a spec directly merges with the head (rather than with a projection of the head. In other words abandoning in this respect the standard assumptions, he effectively adopts the tripartite spec-head-complement structures of mirror theory. As was emphasized in that framework, on the assumption that specs and complements both must be adjacent to the head, they will necessarly be on opposite sides of the head, deriving spec-head-complement order from a stipulation of either the spec-head or of the head-complement order. Kayne's suggestion to this effect repeats the corresponding proposal of mirror theory. Notice also that for Kayne the tripartite structures are only a rather ad hoc means to ensure the result of spec-head-complement order, while mirror theory provided significant independent motivation in terms of the maximal simplification, i.e. the elimination, of the X' projection line.
In sum Kayne's (2010) account of antisymmetry, which unlike mirror theory, only addresses the problem of antisymmetry and ignores the equally prominent symmetry of morphosyntactic structures, is largely identical to the mirror theoretical explanation, --and in the relatively minor respects where it differs, it does not seem to be a significant improvement. The mirror theoretical approach, whether in its original form or in Kayne's version is not fully satisfactory however, primarily because in both incarnations it crucially involves an essentially stipulative component. (There is also a a central empirical issue with Kayne's proposal, see next section.)

2. Specifier ordering

Still keeping (temporarily) to the assuptions of universal spec-head-complement order and to the descriptive correctness of antisymmetry (1) let us approach their explanation differently. Suppose an early step in the spellout component tucks in specifiers and heads before their sisters. (Specifiers and heads form a natural class, heads can be viewed as a subcategory of specifiers or conversely specifiers can be taken to be heads, see Brody 2006.) Kaynean antisymmetry with universal spec-head-complement order now follows trivially.
Why do specifiers and heads tuck in before their sister? We might try to explain this in various ways but it is clear that a more minimal, hence better, hypothesis would be to assume that since specifiers and heads are not ordered by the syntactic trees with respect to their sisters, they can in principle either precede or follow it.
A reasonable amount of evidence has accumulated, that contrary to the Kaynean and Cinquean assumptions about antisymmetry, specifiers may indeed occur in principle either on the left or on the right side of their sister.
Brody and Szabolcsi (2003) argued this in connection with Hungarian quantificational structures in the context of mirror theory, which made a certain class of structures with specifiers on the right possible. Adger et al. (2008) applied and elaborated this approach in a detailed description and careful analysis of Kiowa, providing further evidence. Abels and Neeleman (2012) showed that insisting on the universal specifier-head complement order would result in a less restrictive overall theory, where various well-motivated restrictions would have to be given up.
There is also empirical evidence then to support the theoretically desirable minimal hypothesis: Since the specifier and its sister are not syntactically ordered, in principle spellout has the option of creating either order, --which ordering option is chosen is presumably a matter of parameter setting. We can speculate that elements of the

extended word (aka. extended projection, f-sequence etc.), silent or not, are listed in the lexicon and such inseparability could therefore be a lexical property of certain elements of the extended word. A lexical redundancy rule could ensure that groups of extended word elements (sometimes, but by no means always, all lexical heads in a language) have this property. Assuming as the default the specifier tuck in (spec on the left) option, the lexical requirement of not interrupting the extended word would result in specifiers on the right.
Recall that phrasal specifiers on the right must stack up in inverse order (Cinque 2005, 2008). For example in a nominal projection the apparently basic pre-nominal order Demonstrative-Numeral -Adjective-Noun becomes the mirror image Noun-Adjective-Numeral-Demonstrative, when these specifiers are on the right.

(2) Demonstrative $_C$[Numeral $_B$[Adjective $_A$[Noun]]]

In (2) the Adjective cannot be placed inside the A node or outside the C node by spellout, since that would change constituency relations. It may only be placed to the left or to the right of its sister, the A node. If to the right, then we get the Noun Adjective order. If Numeral and Demonstrative are then also placed to the right of their respective sisters, this results in the Noun-Adjective-Numeral sequence.
Recall that non-phrasal specifiers/heads on the right also stack up in inverse order, (eg. Hungarian *al-hat-t-am, sleep-may-past-I, I may have slept*), --the mirror generalization. This generalization then apparently covers both phrasal and non-phrasal specs/heads. It is straightforwardly captured here by the spellout ordering theory. (Suffixes on auxiliaries would presumably need to involve a different mechanism, perhaps, as Bjorkman (2014) argues, *have/be* support.) The generally tighter nature of suffixation, the nonseparability of suffixes compared to prefixes, might follow from the assumption that suffixation is a consequence of the nonseparability of the corresponding extended word elements involved in the structure.

3. Movement to the left

As Cinque (2005) and (2008) make clear, although specifiers in the basic order to the left of the terminal element of an extended word can appear in the inverse basic order to the right of this terminal, rearranged orders on the left do not get mirrored on the right. Abels and Neeleman (2012) point out that the NP internal orders that, as Cinque observes, are prohibited, remain impossible even if specifiers in the base order can in principle be either to the right or to left of their sister, --provided that the movement operation that rearranges the order of specifiers is always to the left. So for example the Demonstrative-Numeral -Adjective-Noun order can be rearranged to Demonstrative-Noun-Numeral -Adjective or Demonstrative-Adjective-Noun- Numeral by moving the constituent contaning the noun, or the adjective and the noun, respectively to the left. Multiple rollup leftward movements can create orders like for example Demonstrative-Noun-Adjective-Numeral by moving first the noun to the left of the adjective, and then moving the constituent contaning the the moved noun and the adjective to the left of the numeral. (I follow Cinque, as do Abels and Neeleman, in assuming that all relevant movement operations here must apply to a constituent containing the head noun.)
Crucially, if no rightward movement is possible then there is no way to derive the non-occuring mirror images of these orders: *Adjective-Numeral-Noun-Demonstrative (mirroring Demonstrative-Noun-Numeral –Adjective) , *Numeral- Noun-Adjective-Demonstrative (mirroring Demonstrative-Adjective-Noun- Numeral), or *Numeral-Adjective-Noun-Demonstrative (mirroring Demonstrative-Noun-Adjective-Numeral).

The necessity of the assumption that movement must always proceed leftwards however is puzzling in the context of the standard approach in which a core tenet is that movement must always proceed upwards, i.e. to a c-commanding position. The statement that movement is both upwards and leftwards appears to clearly signal a missed generalization. Abels and Neeleman consider the options of stating the left-directional requirement in the spellout component or as part of the parsing sytem and argue for the latter. However, neither of these approaches appears to have the potential to mitigate the sense of missed generalization. They in fact exacerbate the situation in that they imply in addition to the missed generalization a puzzling conspiracy between the different components of our linguistic abilities.
Furthermore, under the parsing proposal Abels and Neeleman argue for, it would not be enough to postulate their parsing requirement that movement fillers must be leftwards of their gaps. It would be necessary to ensure also that fillers must be higher than their gaps. Otherwise movement to the right would be possible with low copy spellout/filler on the left and a high copy gap on the right as in (3):

(3) (…filler …) … gap…

A structure like (3) can directly result in various impossible NP orders on the right of the base position of the N. If for example we take the filler to be the Noun and the gap is on the right of the Adjective that is generated on the right of the Noun (ie. “N Adj N(gap) Num Dem”) then via rightward multiple rollup we get the order N Num Adj(N-gap) Dem, --by moving the Adjective-Noun(gap) constituent to the right of the Numeral.
(In addition, under the parsing requirement that movement fillers must be leftwards of their gaps, the impossibility of rightward movement would not follow in general. See eg. Abels 2007 for arguments that rightward movement does not exist. The parsing approach would allow rightward movement of a category not only in cases like (3) but also where this movement is followed by a longer leftward movement of the same element.)
Notice that the requirement that the filler must be higher than the gap does not follow from the syntactic requirement that movement must be upwards, under either the multiple copy or the remerge/multiple domination theory of movement. While the extension requirement on syntactic derivations ensures that movement is from bottom to top, this only ensures usefully that movement links c-commanding/ dominating positions, and does not determine the identity of the filler/spellout position. (In a bottom to top derivation in addition to connecting c-commanding positions movement must always proceed from lower to higher positions, --a property of the derivational approach that similarly does not contribute to the identification of the filler/spellout position under either the copy or the remerge/multiple domination approaches to movement.)
So under the assumption that the upwards directionality of movement is the consequence of syntax but leftness of movement is a parsing requirement, we need to exchange the puzzling disjunction according to which movement is upwards and leftwards to the even less promising tripartite statement: movement is upwards and fillers are upwards and leftwards of their gaps.

4. Ninety degree

On an intuitive level antisymmetry suggests that the relationship between c-command and precedence is closer than one might have thought. As is by now generally appreciated, c-command is a complex and unnatural notion, best thought of as an epiphenomenal composition of two simpler concepts that play a role in the grammar:

domination/containment and the *spec-X/head-X* immediate domination or sisterhood relation (Brody 2002, also Chomsky 2006).
This might suggest another way of looking at the problem. Suppose that what we normally think of as abstract containment is in fact precedence. Our trees have their initial symbols at the top and branch from top to bottom. If containment is precedence, then it would be more perspicuous to draw the trees with the initial symbol leftmost and branching from left to right, --call these p(recedence)-trees. By this proposal I do not mean of course that a claim about how the tree is represented is ultimately of theoretical consequence. What is of potential consequence is what concepts and relations the tree is taken to express, and a tree that is rotated by 90 degrees to the left with respect to its standard orientation would more clearly indicate the hypothesis that instead of pure containment we need to use the stronger notion of precedence.
Under the present hypothesis there is then no separate pure domination/ containment. The syntactic linear ordering provided by domination/ containment does not *correspond to* a (PF) linear ordering but *is* a linear ordering by precedence.
It is often stated that there is no evidence for linear precedence being relevant for syntax. But perhaps precedence *is* relevant, as antisymmetry phenomena directly suggests, and domination/containment is just an aspect of this concept, --a concept that was mistaken for pure domination/containment. There is indeed little evidence for another independent dimension and to this extent the pronouncement remains true.
Given the understanding of domination as precedence, the need for a core disjunctive condition on movement disappears. Consider first the single step movement case. Under the multiple domination („remerge") theory a moved constituent C may end up immediately preceded (instead of immediately dominated as in the standard orientation of trees) by two nodes, say A and B. If A precedes B then A will be the position of the filler (it will immediately precede the filler) and B will be the position of the gap (B will immediately precede the gap). Let us call the nodes A and B that immediately precede a moved constituent C the addresses of the moved constituent C. A is then the address of the filler and B the address of the gap. The only requirement we need is (4):

(4) The address of the filler must precede the address of the gap

In the context of p-trees, condition (4) ensures the effects of the more standard disjunctive statement that movement targets a position that c-commands and is on the left of the launching site. It ensures also the effects of the tripartite condition necessitated by Abels and Neeleman's approach, that movement targets c-commanding positions and fillers c-command and precede their gaps.
In the context of the hypothesis of section 3 above, according to which spellout orders specifiers and heads with respect to their sisters, it is of course necessary to assume that (4) remains relevant also for this ordering, to ensure that a moved specifier does not get linearized to the right of its sister. Since spellout ordering shows no syntactic movement properties, there would be no obvious ways to achieve this, if we used instead of (4) the requirement that the landing address of movement/ remerge must precede the launching address .
Nothing of substance in this account need to change if instead of multiple domination we assumed a multiple copy theory of movement.
If (4) is added to a standard bottom to top derivational theory then we may assume that movement landing sites (remerge sites) are filler positions and launching sites are gap positions. (In successive step movement an intermediate site S is both a filler

for the movement /remerge step that landed in S and a gap for the movement / remerge step launched from S. The assumption that landing sites are fillers and launching sites are gaps brings into relief the well known strange duplication inherent in bottom-up derivational theories: Derivational movement steps need to be duplicated by filler gap dependencies (in reverse direction) in the perfomance system.
A core advantage of the representational approach is that it can be paired without the duplication and odd redundancy of the bottom to top derivation with a natural theory of performance in which structures are assembled left to right.
In a representational theory which aims to generate (in the mathematical sense) a set of LF-PF pairs in abstraction from the computational steps necessary for assembling (as opposed to merely characterising) these structures we can deal with successive step phenomena by defining an ordered set of addresses (ie. a chain) such that each member immediately precedes the multiply attached element. We can assume that such a chain is the representational image of a serial filler gap dependency that corresponds to the successive step movement of the standard approach. In other words each chain member (except the last) is the representational image of a filler for the next member and each (except the first) is the image of a gap for the immediately preceeding one.
The representational approach has a further advantage.C-command constrains a set of structures that properly include movement constructions. Hence basing the explanation of c-command on properties of movement will necessarily miss a core generalisation of the grammar. Since c-command is relevant also to various island crossing constructions we clearly cannot explain c-command in terms of movement in general. This would necessitate eliminating the property of island sensitivity from the definition of movement, effectively emptying the notion.
The representational approach makes it possible and natural to generalize (4) to the wide class of non-movement dependency relations that require c-command (eg. island violating quantifier-variable binding or resumptive structures) by requiring the address of the antecedent to precede the address of the dependent.
To sum up so far: Antisymmetry is a consequence of the leftness condition on chains. Given p(recedence)- trees, the leftness condition entails the c-command condition.

5. Merge?

In the context of p-trees, the operation of Merge invites the questions of why the container set precedes the elements it contains. If trees are constructed by Merge, then precedence of the container is additional and unexpected.
The problem is reminiscent to the problem of labeling: If trees are constructed by Merge, then labeling is additional and unexpected. I take labeling to express the fact that the container set shares some property with one (or more) of its members. Labeling is partly synonymous with and partly enters into various syntactic notions and processes like (categorial) projection, spec-X-agreement, Criteria etc. All these involve a core component of local feature duplication, which I assume consists in the highest node of the contained constituent and the container being copies of each other. I shall use the term labeling to refer to this local feature duplication. Merge contributes nothing to the explanation of why in apparent contradiction to general parsimony expectations, such a duplication of the relevant feature(s) should so pervasively exist, since it is a quasi set-theoretical operation expressing containment, a concept that should involve no property sharing between the set, the container and

a member of this set, the contained element. Labeling needs to be added in an ad-hoc fashion to Merge.
Labeling, like precedence indicates that Merge does not do enough.
On the other hand in the context where features correspond to nodes, labeling also suggests that Merge does too much: Merge adds the container set to a set of (contained) contituents but the existence of this additional element is ensured independently also by labeling, which produces a copy of a feature/node of one of these constituents.
Another area where Merge seems to do too much has to do with c-command. Merge creates trees, but syntax seems exclusively concerned with narrower structures, where each element c-commands the next. The fact that movement/remerge always proceeds to a c-commanding position has been attributed to the extension condition on derivations, but this condition by itself is not strong enough to ensure that interveners of remerge, whether relativized minimality or barrier type, are also in a c-command relation with the members of the movement chain. More generally the extension condition does not entail that constraints on move/remerge, like relativized minimality, barriers, nesting- crossing- and improper movement conditions all involve only c-commanding positions. In general although we have trees in syntax by virtue of Merge, syntax does not seem to make use of them. There are no 'triangulating' constraints or operations. No prohibition exists for example that would prohibit movement over or under a non-c-commanding intervener. The *wh*-movements in (5) exemplify that syntactic movement chains not related to each other by c-command never affect each other.

(5) What did (the man who just called you) want to see?

Various conspiracy theories are imaginable and several have been proposed to explain the pervasiveness of c-command. For example in a bottom to top derivational approach incorporating the extension requirement, the lack of such interactions can in general be attributed to the phasal nature of the derivation and the derivational non-transparency of completed phases. It remains unclear however, why phases and the extension requirement conspire to together ensure the simple c-command generalization, --that for syntactic operations and conditions on them, only c-commanding positions matter. There are also some doubts about the validity of the extension condition, relating to „tucking in", countercyclic LF and head movements, and redundancy with the conception of strength features (Richards 1999, Boscovic and Lasnik 1999) and to my mind more worrying general questions concerning the necessity of derivations in characterizing sentence-structure competence as distinct from theories of performance (Brody 1993, 1998, 2002). Derivations are of course necessary if grammar is the parser or the competence theory of the parser, but then the bottom to top direction is clearly and obviously unreasonable.(cf. e.g. Phillips 1996, Chesi 2004, 2007). Without derivations or with top down left to right derivations the extension condition cannot be invoked.
Furthermore, to repeat the point rehearsed also in the previous section, c-command constrains a set of structures that properly include movement constructions. Hence basing the explanation of c-command on properties of movement will necessarily miss a core generalisation of the grammar.

6. P-strings. One dimensional syntax.

The precedence condition in (4) and the considerations relating to c-command and the lack of 'triangulating' in the previous section might suggest that what is really relevant to syntax proper is not trees but sequences of elements in c-command

relation, where each element c-commands the next. Asymmetric c-command is a transitive and of course antisymmetric relation. In a syntactic tree it is not total. But it is total in the sequences of elements in c-command relation, which therefore constitute a linear order. Suppose then that syntax is in fact so impoverished that it can only create linear orders. A sentence will then have to correspond to not one but a collection of syntactic structures.
As noted in section 4 above, it is better to proceed using the more basic concept of domination instead of c-command. In this spirit we define a domination sequence (*d-string*) associated with a given terminal element T as the set of all nodes dominating T, ordered by domination. We can assume that the representation of a sentence with initial symbol S is a set of (overlapping) *d-strings starting with S*. Constituents only exist in /are visible by the interpretive components. For LF interpretation we can take to be a constituent any node N, together with all constituents N dominates in the set of strings that together constitute the representation of the sentence.
In view of the discussion in sections 1-5 above, we immediately modify this picture: instead of d-strings we postulate p(recedence)-strings: the set of all nodes preceding a terminal T, ordered by precedence. A constituent at LF will be any node N, together with all constituents N precedes.
So p-strings are sets, linearly ordered by the precedence relation. If we wish to have an operation to create them, it is not Merge, but Concatenate, which we may also take to apply freely, with the results filtered by interface requirements. However, since I do not see any clear advantage in generating additional representational levels in the competence theory of syntax by grafting a stepwise derivational machinery onto the LF representation, I will not consider concatenate as a derivational operation either. I will attempt to define permissibility of structures representationally, in terms of licensing. (Note that recursion and derivation are distinct notions and nothing prevents a single level representational characterization from being recursive.)
The assumption that syntactic structures are not trees but p-strings entails that syntactic relations or operations can involve no ’triangulating’, that they never make use of the additional dimension that trees offer.

7. Licensing P-strings

What are valid p-strings, licenced at the interface? Clearly, strings that express the concept of extended word, the f(unctional) or cartographic sequence, --I shall henceforth refer to these as c(onceptual)-strings --, should be well formed segments of p-strings. I make no specific assumptions about what enters into constraining c-strings; -- whether this is a purely semantic or syntactic matter, or as seems more likely, some combination of the two. The crucial assumption is just that c-strings exist and they are made use of by the grammar: they are licensed segments of p-strings. But the p-string, from the initial symbol to the terminal, the domain of syntactic relations, potentially spanning an infinitely large number of sentences is thus often much longer than a single c-string. We thus need to assume that there is a way to combine c-strings into longer strings. To take concrete example, consider the junction of a p-string P, say the spine of a main clause, with the main, i.e. initial, c-string C of the nominal that is the subject of the sentence. Focusing on the element X of P to which C attaches (in standard terms, the subject is the daughter of XP), we assume that a p-string Q is licensed as the concatenation of the initial segment of P, up to and including X with the c-string C.

More generally we might take the property of beginning with the initial symbol and ending with a terminal a condition on p-strings and license p-strings recursively as in (7) :

(6) A valid p-string starts with the initial symbol S and ends with a terminal T

(7) A string that is the concatenation of a segment of a p-string with a c-string is a p-string (where segments are continuous sequences of p-string members)

Since p-strings must start with the first element of the sentence by (6), the segment of the p-string in (7) with which the c-string/terminal is concatenated must be an initial segment of that p-string. To make sure that the concatenation of a full c-string with the initial segment of a p-string results in a p-string in accordance with (6), we might assume that the last element of a c-string is always a terminal, either the relevant lexical head or, more likely, a special terminal symbol T.

To start off the recursion, we might take the starting symbol S to be a degenerate one member p-string.

It may seem problematic to allow multiple attachment (move/chain/remerge/etc.) in one dimensional syntax since the multiply attached element E appears to be external to the p-string that contains E's addresses. However the initial (top)node of E (and indeed the initial c-string of E starting with this top node) will be in at least one p-string that contains E's addresses. This is enough to ensure that the construction is fully detectable p-string internally.

To allow for multiple attachment we then also want to license the concatenation with the initial segment, SEG1, of P a later segment of P, SEG2. To explicitely allow for this we revise (7) as (7'):

(7') a string that is the concatenation of two licenced strings is a p-string

Where c-strings and segments of p-strings are licensed, the latter by virtue of being part of a licensed p-string.

Since p-string Q, the concatenation of SEG1 and SEG2 of P, like all p-strings, must end with a terminal, SEG2 of P must end with a terminal, it must be a final segment of P. The effects of the standard requirement that movement operates on constituents thus follow here largely from the assumption that p-strings span the distance from S to T, from the initial symbol to a terminal.

Notice that I do not assume the competence theory internal existence of an actual operation that concatenates SEG1 and SEG2 or more generally competence theory internal operations that produce p-strings. Production is a matter of performance and I take syntactic structures to be a representations of the structures that are created by whatever production system proves to be the correct (abstract, core) theory of performance.

It follows from the precedence condition in (4) that the final segment of a given p-string P can only be in an additional concatenation with the initial segment of P, the addresses of the multiple attached element cannot otherwise be in a syntactic (precede) relation with each other. Hence the effects of the standard c-command condition continue to fall out. More generally, the lack of triangulation, that nothing can be syntactically relevant for the remerge of the final segment of a p-string P that is not part of P is also a natural immediate consequence if nothing external to P can have syntactic access to the relevant (syntactic) licensing rule.

Further in the one dimensional syntax of p-strings, we can attribute (4) to a simple monotony condition. Assuming that the relevant dependency relations cannot involve storage of the unlinked dependent, in other words that a dependent must be linked to its antecedent immediately upon being introduced into the structure, the antecedent will necessarily have to precede.

8. Interpreting sentences

We can think of a p-string as a set of local (and transitive) precedence statements (first member of the string precedes the second, the second the third etc). There is a lot of redundancy in the set of precedence statements that correspond to a sentence. The precedence statements in an initial segment of a p-string will be repeated in the p-string of every terminal whose p-string includes this segment . So to take our earlier example consider again the forking configuration, the junction of a p-string P, the spine of a main clause with the initial c-string C of the nominal phrase that is the subject of the sentence. All the initial precedence statements of the p-string P down to the element X of P to which C attaches will be repeated in in the p-string Q that results from the concatenation of the initial segment of P and C.

It seems natural to assume that such redundancies are automatically eliminated when the p-strings of a sentence are collected for interpretation. Accordingly, I assume that, as part of the unification of p-strings, in forking configurations, members of a p-string that duplicate members of another p-string are expunged .
But here we encounter a problem. In the case of the example with the subject, if all members of the p-string Q of the subject are deleted that correspond to (the initial) members of the p-string P of the main clause, in other words all initial members of P up to and including X, then we lose the information of where C attaches and we have failed to put together a viable input for spellout.
Thinking in terms of precedence statements instead of string members suggests a solution. Suppose unification of p-strings involves deletion of all duplicate precedence statements. This means that after unification X will remain both in P and in Q since in each it is part of a distinct precedence statement. This will ensure that P and Q will be appropriately positioned.
We now seem to have obtained also a non-stipulative potential explanation of the existence of labeling, of the fact that heads and specifiers must participate in feature sharing with the immediately dominating node. To exemplify, consider the junction of a p-string P, the spine of a main clause with the initial c-string C of a fronted wh-phrase. The Q-node in P to which the wh-phrase attaches is now the X that needs to duplicate and hence will be present also in the wh-phrase. We obtain categorial labeling by a head H if in a c-string P, X is the relevant category feature/node to which the c-string Q of H attaches which therefore needs to duplicate on the top of the c-string Q of H. We can obtain spec-head agreement if the spec and the head attach to the same node X, hence X will need to be duplicated in both. This may involve spec and head attaching to the same token of X as in the tripartite structures of mirror theory (or of Kayne 2010 discussed in section 1. above) or to the same category type X as in the more traditional binary theories of phrase structure.
If nodes correspond to single features then agreement in multiple features will need to involve multiple attachments („movement"). Notice that the fact that the direction of agreement might sometimes seem inverted, for example a subject inherits its agreement features from the node in the c-string of the verbal node to which it attaches rather than, as generally assumed, the verbal ending inheriting features from the nominal. But the usual assumption might be mistaken. Lexical items inherently specified with agreement features can insert in the structures generated, giving the illusion where the subject is lexically specified but the verb stem is not, that there is a grammatical process that is directly responsible for making the verb agree with the subject rather than the other way around.

While many basic questions remain, and we are very far even from a detailed descriptively adequate account, the present approach might help us to at last begin to have some understanding of the reason for the genuinely mysterious pervasive duplications of grammar involved in agreement, criteria and categorial projection. The existence of such apparently non-parsimonious duplications seems to be another natural consequence of the assumption that syntactic structures are one dimensional objects and the sentential interpretive systems access collections of connected syntactic structures.

References

Bjorkman Bronwyn M. 2014 Verbal inflection and overflow auxiliaries. Ms. Queen's University

Bošković, Željko. and Howard. Lasnik.1999. How Strict Is the Cycle? *Linguistic Inquiry* 30: 691-703

Brody, Michael. 1993. Theta theory and Arguments. *Linguistic Inquiry* 24:1-23

Brody, Michael 1998. The minimalist program and a perfect syntax. *Mind and Language* 13:205-214.

Brody, Michael. 2002. On the status of derivations and representations. in *Derivation and Explanation in the Minimalist Program*. Eds.Samuel Epstein and Daniel Seely. Blackwell Publishers.

Chomsky, N. 1995. Bare phrase structure. In *Government and binding theory and the Minimalist Program*. 383–439. Ed. Gert Webelhuth. Oxford: Blackwell.

Chesi, Cristiano. 2004. *Phases and cartography in linguistic computation: Toward a cognitively motivated computational model of linguistic competence*. PhD thesis, Università di Siena.

Chesi, Cristiano. 2007. An introduction to phase-based minimalist grammars: Why move is top-down from left-to-right. *Studies in Linguistics*. 1, 49–90.

Cinque, Guglielmo. 2005. Deriving Greenberg's Universal 20 and its exceptions. Linguistic Inquiry 36:315–332.

Cinque, Guglielmo. 2009. The fundamental left-right asymmetry of natural languages. In *Universals of language today*. Eds. Sergio Scalise, Elisabetta Magni, Antonietta Bisetto. Berlin: Springer.

Richards, Norvin. 1999. Feature cyclicity and ordering of multiple specifiers. In Working Minimalism. 127–158. Eds. Epstein, Samuel D. and Norbert Hornstein. Cambridge, MA: MIT Press.

Kayne, R. 1994. *The antisymmetry of syntax*. Cambridge, MA: MIT Press.

Phillips, Colin. 1996. *Order and structure*. PhD thesis, Massachusetts Institute of Technology.