

The Asymmetry and Antisymmetry of Syntax¹

A Relational Approach to Displacement

Justin Malčić

19th February 2019

Quick note on the title

Asymmetry and antisymmetry are not the same thing, but confusingly antisymmetry has been used in linguistics to refer to asymmetry, famously by Kayne (1994), but also earlier, e.g. by Sagey (1988). I define it below, following Partee et al. (1993), and repeat relevant definitions later where used.

Symmetry

$aRb \iff bRa.$

Asymmetry

$aRb \Rightarrow \neg bRa.$

Antisymmetry

$aRb \wedge bRa \Rightarrow a = b.$

Abstract

In both syntax and phonology, it has long been observed that significant restrictions exist on displacement. One such restriction ensures that displacement leads to sequences of elements which are in some sense contiguous, formalised in syntax in the concept of Feature Geometry-based Relativised Minimality by Starke (2001) and Contiguous Agree by Nevins (2007), and in Autosegmental Phonology by the Line-Crossing Prohibition (originating in the Well-formedness Condition of Goldsmith, 1976).

I argue that effects of this type, which have been called *Contiguity Effects*, are best captured by taking displacement to involve total weak orders of elements in the sense of Order Theory. Building on work taking the LCA to hold throughout the derivation, I argue that precedence relations may be the basis of phrase structure, though without claiming that linearisation is necessary for LF (as suggested by Kayne, 2013, for example). I then develop this approach to show that Order Theory may provide useful axioms for both phrase structure and displacement, and that the existence of displacement is expected given the use of Order Theory.

¹Thanks to Theresa Biberauer (who's supervising this) for useful feedback. Remaining errors my own.

Asymmetry and antisymmetry in syntax

The extent to which syntax is ‘antisymmetric’ has been a major issue for theories of phrase structure ever since Kayne’s (1994) original proposal for the Linear Correspondence Axiom. It can be argued that the requirement for antisymmetry in linearisation follows from the nature of the phonological component, which is usually taken to show a different kind of recursion to that displayed in syntax. But on the other hand it is not at all clear why the only kind of order (in the sense of Order Theory) relevant to syntax is specifically a *strict* total order (i.e. an order based on asymmetric relations).

It is important to note at this point that what Kayne calls ‘antisymmetry’ is referred to more commonly outside of linguistics as ‘asymmetry’, being based on asymmetric relations, as opposed to distinct antisymmetric relations. For clarity, symmetry holds when the existence of a relation from *a* to *b* implies its inverse also holds (from *b* to *a*). Asymmetry is the opposite: the existence of a relation from *a* to *b* implies the lack of a relation from *b* to *a*. Antisymmetry, on the other hand, holds if a symmetric relation implies equality (so for example the equality relation is both symmetric and antisymmetric). These definitions are all given in (1).

- (1) Definitions of symmetric, asymmetric, and antisymmetric relations

Symmetry

$$aRb \iff bRa.$$

Asymmetry

$$aRb \Rightarrow \neg bRa.$$

Antisymmetry

$$aRb \wedge bRa \Rightarrow a = b.$$

Two of these relations have a corresponding order: asymmetric relations form strict orders, and antisymmetric relations form weak orders (symmetric relations only lead to a lack of ordering). This is easiest shown visually: in (2), β is ordered before a as there is an asymmetric relation from β to a (indicated by the arrow). In (3) δ is ordered before γ , but in addition δ is equal to δ , and γ is equal to γ , because for each node there is a reflexive relation.

As can be seen, the key difference between asymmetric and antisymmetric relations relates to reflexivity: asymmetric relations are necessarily irreflexive, but this is not true of antisymmetric relations. And finally, (4) shows that symmetric relations cannot be used for ordering, since there is no principled

way based on the relations to determine that ζ comes before or after ε , or that θ comes before or after η .

(2) Strict order

$$\beta \longrightarrow \alpha$$

(3) Weak order

$$\begin{array}{c} \curvearrowright \\ \delta \longrightarrow \gamma \\ \curvearrowleft \end{array}$$

(4) No order

$$\zeta \longleftrightarrow \varepsilon$$

$$\begin{array}{c} \curvearrowright \\ \theta \longleftrightarrow \eta \\ \curvearrowleft \end{array}$$

So given the prominence of asymmetric c-command relations due to Kayne's (1994) Linear Correspondence Axiom, and given Moro's (1997) suggestion that symmetric c-command is also crucial, it is worth asking whether *antisymmetric* relations may play any role in syntax. More pointedly, why is (2) available to the Language Faculty but not (3)? To my knowledge the possibility that it *does* play some kind of role has not been extensively explored (perhaps in part due to the terminological confusion). And in any case, if it plays no role then it needs to be ruled out in some principled way.

While Merge certainly does not rule out the ability to extend the Phrase Marker using material internal to the Phrase Marker, it does not of itself predict that the two resulting isolated occurrences of material should be identified with each other (especially given the intervening distances often involved). Recalling that under antisymmetry, $aRb \wedge bRa \Rightarrow a = b$, weak orders could be the basis for capturing this effect, and if so there ought to be apparent idiosyncrasies of displacement following from this. In section 2, I show that well-known contiguity effects observed in syntax and phonology are indeed predicted by this requirement. But first, I further I review the role of Order Theory in syntax, and argue that it is much more wide-ranging than commonly assumed.

1 Evidence for asymmetry in syntax

The use of Set Theory as the basis of phrase structure originates in Chomsky's (2014/1995) definition of Merge, a function taking two Syntactic Objects a and β , which returns the set of these elements $\{a, \beta\}$. The original rationale for this was that this (unordered) set is the 'simplest object' formed from the two elements:

The *simplest object* constructed from a and β is the set $\{a, \beta\}$, so we

take K [the output of Merge of α and β — J.M.] to involve at least this set. (Chomsky, 2014/1995)

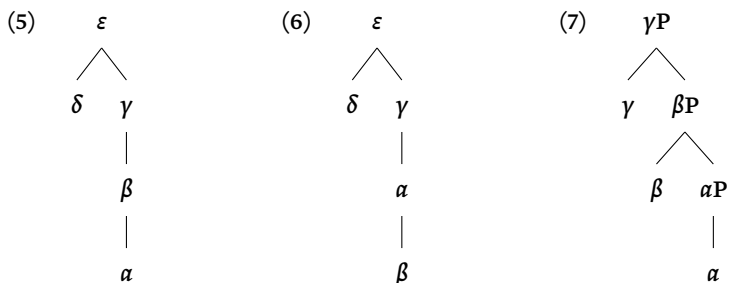
This definition of Merge, termed set Merge, is to be distinguished from another kind which yields ordered pairs, known as pair Merge, introduced by Chomsky (2004) to handle adjunction structures. It is however explicit that pair Merge is more complex than what has been termed ‘simple Merge’ (Chomsky, 2004, 2013), and as long as set Merge exists, Phrase Structure remains ultimately defined in terms of Set Theory. This becomes especially clear in more recent work by Chomsky (2013, 2015) where labelling is fully determined by the Labelling Algorithm and occurs once per phase, and is strictly independent of the phrase structure component. In this case the function of set Merge really is just to form sets comprising a head and a phrase, two phrases, or two heads.

There is apparent symmetry in this system, in that (*Merge* α, β) does not imply itself that α or β is in some sense subordinate to the other—it is the Labelling Algorithm that would be sensitive to the head-phrase distinction in, for example, $\{\alpha, \beta P\}$. Indeed, on this basis Chomsky (2013, 2015) proposes that symmetry forces movement in order to meet the requirements of the Labelling Algorithm. But the extent to which this symmetry is real or just apparent is an open question, because it is only really in cases where two phrases are merged that there is real similarity between the arguments to the function: this is obviously the case for a head and a phrase, and for two heads Chomsky (2013) suggests that the root-categoriser distinction is enough to ensure the complex is labelled by the categoriser. In the following I take the view that merging of two phrases also exhibits a distinction, and that apparent symmetry is misleading.

1.1 The status of c-command

Another possibility for characterising a possibly symmetric combinatorial operation such as Merge would be to say that the operation establishes an abstract symmetric relation between two elements α and β . And indeed representationally for the structure $\{\alpha, \beta\}$ produced by Chomsky’s (2014/1995) version of Merge, α would standardly c-command β and vice versa. On this basis Epstein (1999) proposes that the effect of Merge is precisely to establish such c-command relations derivationally, obviating the need to compute them representationally as under the standard view.

But even if c-command is not as intimately connected to Merge as this, the extent to which it seems to be relevant to extremely disparate syntactic phenomena (binding, agreement, movement, etc., but for many other examples see Uriagereka, 2011) suggests that it must be fundamental to the system in some non-trivial way. Indeed, Frank and Vijay-Shanker (2001) show that it is possible to use c-command as a primitive, and that this has the effect of restricting possible structures to a subset of those which are able to be linearised with Kayne’s (1994) Linear Correspondence Axiom. Specifically, the structures which can be defined in terms of c-command do not have nodes joined to two others by unary branching, structures which the LCA on its own does not rule out, as it addresses only terminals. Observe that there is no difference between the c-command relations in (5) and (6): crucially δ asymmetrically c-commands both β and α in both cases, no other node c-commands them, and they do not c-command any node. Hence because c-command is not extensional over these structures, taking c-command as a primitive enforces structures such as (7), which is of course LCA-compatible, and note also that a single instance of unary branching, needed for First Merge, is *not* ruled out.



But supposing c-command is indeed the basis of phrase structure, it is still unclear why this particular relation should be chosen to play such a major role in syntax. C-command is a rather curious relation in the sense that it effectively combines a symmetric relation, sisterhood, with an asymmetric relation, dominance¹, yet as Epstein (1999) points out, c-command was proposed very early (effectively in Klima, 1964, though his in-construction-with relation is really the inverse of c-command), and has outlasted many other proposed relations, notably m-command and Government (for a review

¹Assuming proper (irreflexive) dominance.

and formalisation of these and many other relations, see Barker and Pullum, 1990), suggesting that its formulation is broadly correct. However, as Bruening (2014) points out, there have in fact been a number of proposals over the years to use a slightly different relation to c-command, namely precede-and-command, which instead combines two asymmetric relations, precedence and dominance, resolving the contrast between symmetric sisterhood and asymmetric dominance. The other possibility for simplifying c-command would be to try to reduce it to only precedence or dominance, by causing these two relations to coincide.

In fact, under the LCA, asymmetric c-command already does coincide with both precedence and dominance to a large extent, which suggests that the nature of linearisation may be instructive in understanding what c-command really is. Uriagereka (2011), for example, has suggested that the existence of c-command in syntax follows from its role in linearisation under Kayne's (1994) LCA as defined in (8). But it has been argued—by Chomsky (2014/1995) among others—that the LCA only holds at PF, given the lack of convincing effects due to linear order that would be otherwise expected at LF. Admittedly Kayne (2013) has claimed that for instance, the cross-linguistic predominance of forward over backward pronominalisation could be explained by the effect of linear order at LF². But given that so many core cases of binding famously show a sensitivity to c-command over linear order, it seems wise to suppose that linearisation does not occur at LF, and to find another way to explain the pronominalisation facts.

(8) Linear Correspondence Axiom

a. Base:

When x asymmetrically c-commands y , x precedes y .

b. Induction:

If a non-terminal X dominates a terminal y , and X is linearized with regard to terminal z , then y is linearized with regard to z .

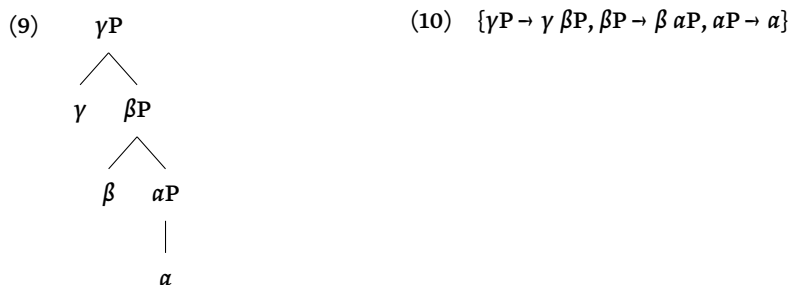
(Uriagereka, 2011)

Suppose therefore that the linearisation does not hold at LF, which is therefore not sensitive to linear order. In this case, Uriagereka's (2011) explanation for the presence of c-command in syntax—that it follows from the need to linearise structure—will have no bearing on LF of itself. Indeed it might

²Linear order because, given Principle B, restrictions on the relative ordering of pronouns must be defined in some other way than c-command.

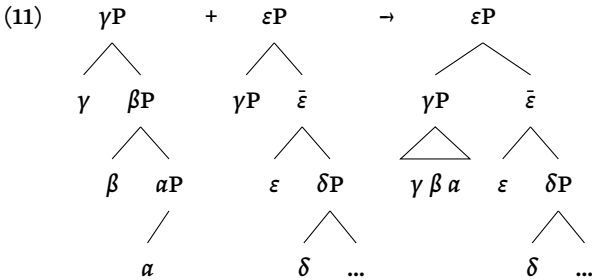
be expected that phenomena at LF and PF would make use of entirely different relations. But this appears not to be the case: the effects of c-command are as widespread at LF as at PF, notably in determining scope and in quantifier raising, and hence it seems clear that c-command is required for both PF and LF phenomena. Uriagereka (2011) therefore suggests making phenomena at LF parasitic on c-command, as it already exists in the Narrow Syntax due to the need for externalisation at PF. This goes against much Minimalist thinking which views externalisation as a somewhat peripheral aspect of language (Chomsky, 2013), and arguably replaces the so-called ‘LF bias’ with a ‘PF bias’, in that properties of LF follow from the needs of PF. By contrast, for Chomsky (2014/1995) the need to linearise at PF arises due to the inability of PF to handle the kinds of structures in the Narrow Syntax and at LF.

However, while the claim that properties of LF follow from the requirements of PF is open to dispute, there does seem to be evidence that (subparts of) structures able to be defined in terms of c-command are easier to parse. Uriagereka (2011) observes that structures such as (9) are able to be parsed by Finite State Automata, since the rules in a Regular Grammar are of the form $S \rightarrow aA$, $S \rightarrow Aa$, or $S \rightarrow a$. In particular, the structure in (9) could be produced by a Right Linear Grammar, which only has rules of the form $S \rightarrow aA$ and $S \rightarrow a$. This is significant because of the position of Regular Grammars on the Chomsky Hierarchy: Regular Grammars are the most restricted type of grammar, and consequently have the lowest time complexity in parsing. Rules for the structure in (9) are given in (10) (where non-terminals are distinguished by the suffix P rather than complete capitalisation).



Note however that taking c-command to be a primitive like Frank and Vijay-Shanker (2001) does not automatically lead to structures able to be parsed by FSAs, because specifiers are not ruled out, and Finite State Automata on their own cannot accommodate specifiers as normally conceived: if for example γ

were were a specifier γP , then the rule $\gamma P \rightarrow \gamma \beta P$ would need to be replaced with $\beta P \rightarrow \gamma P \beta P$ —but rules of the form $S \rightarrow AB$ would require (minimally) a Context-Free Grammar and a corresponding Pushdown Automaton. But adopting a Context-Free Grammar with rules of the form $S \rightarrow AB$ would predict that there should be as much left-branching as right-branching in phrase structure, since both options are available to the system. Not only is this emphatically not the case for specifiers, but abstracting away from movement it may also be the case for complements, if Kayne (1994) is right. This is especially relevant given the results of approaches which posit significantly lengthened clausal spines such as Cartography and Nanosyntax, since in this case this asymmetry in directionality of branching will significantly increase, resulting in sections of the phrase marker with no left-branching which are able to be modelled with Regular Grammars and Finite State Automata. So discounting specifiers, empirical evidence and processing concerns would suggest that the kind of structure in (9) may be on the right track.



Accommodating specifiers is a challenge because according to definition in (8) they need to appear to the system to be simplex nodes in order to be related by asymmetric c-command to other terminals in the main clausal spine. This is overcome in Kayne’s (1994) original approach (and also in Frank and Vijay-Shanker, 2001) by capitalising on the segment-category distinction introduced by May (1985) and Chomsky (1986), but the distinction is highly stipulative. Uriagereka (2011) proposes instead that a phrase marker involving complex specifiers is actually linearised in multiple cycles, known as Multiple Spell-out. Specifiers are linearised from most to least embedded, and after each cycle the specifier linearised becomes opaque. The linearised specifier is then inserted into a progressively larger structure by matching the root node of the specifier in question with the corresponding terminal node in the spine which hosts the specifier, as shown in (11), where the delta containing (γ, β, α)

indicates the opacity. This is formalised by what Uriagereka (2011) terms the Linear Correspondence Theorem, given in (12). So in addition to Uriagereka’s (2011) formulation of the LCA in (8), which is effectively covered by (12a), there is a further step which effectively ‘flattens’ any specifiers into a single strict total order, given in (12b).

(12) Linear Correspondence Theorem

a. Base:

A phrasal structure K that is weakly equivalent to the output string χ of a Finite-State automaton is externalized by directly interpreting χ as a phonetic string.

b. Induction:

A phrasal structure K that cannot be externalized as in (12a), because its sub-components L and M are weakly equivalent, respectively, to the output strings χ and ψ of different Finite-State automata, can be externalized by (12b-i) and (12b-ii):

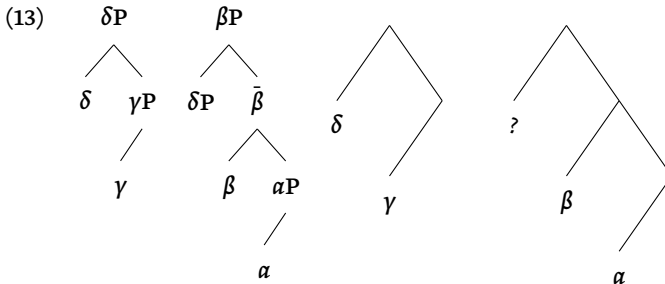
- i. applying (12b) separately to χ and ψ , as effectively as possible,
- ii. addressing L , linearized as χ , as a tributary current [string equivalent to the output string of a Finite-State automaton embedded in another such string—J.M.] of M , linearized as ψ .

(Uriagereka, 2011)

Multiple Spell-Out therefore suggests that the hypothesis that linearisation addresses strings weakly equivalent to Finite State Automata can be maintained provided that it is possible to recursively embed FSAs within these strings. But observe that if the system always manipulates the kind of structure shown in (9), as the Multiple Spell-Out approach would suggest, then for terminals hierarchy and order will always coincide with asymmetric c-command, and only the presence of phrasal nodes (and more specifically the apparent need for labelling) supports the traditional distinction between dominance and precedence. On the other hand, if phrase structure is based on a single asymmetric relation, then the phrase marker can be conceived as a single strict order of terminals based on this asymmetric relation, mapping to hierarchy, and through an analogue to (12b-ii) also to linear order. This is set out in the following section.

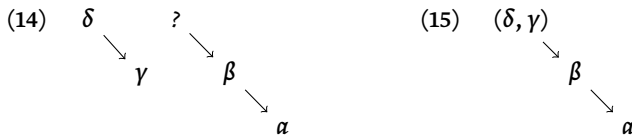
1.2 Phrase markers as strict orders

As Chomsky (2013) has observed, projection or labelling appears to be a theory-internal notion (even if externalisation sometimes manifests a sensitivity to category). One possibility advanced in Chomsky (2013, 2015) is that labelling is determined towards the end of the phase by the Labelling Algorithm, an instance of Minimal Search, so that labelling is no longer a property internal to phrase markers. Chomsky (2015) therefore advocates abandoning trees because of the absence of labelled nodes in representations, but for expository purposes it should still be possible to use an unlabelled tree diagram to represent the FSA-equivalent structures seen above, as shown in (13).



The Labelling Algorithm that Chomsky (2013, 2015) suggests works as follows: in the case of a phrase comprising a head and a phrase $\{H,XP\}$, the containing phrase is labeled by the head H. In the case of a phrase containing two phrases $\{YP,XP\}$ there are two options: either the containing phrase is labelled by the closest visible head, either Y or X (crucially, copies are invisible for the algorithm), or by a feature common to Y and X. While this solution has interesting consequences in forcing movement and successive cyclicity, it is also possible that specifiers never contribute the label to the phrase containing them because, as embedded FSAs, they are opaque to the Labelling Algorithm—much as Chomsky (2013) suggests that the root-categoriser distinction leads only categorisers to contribute labels. In this case the first head inside the specifier will not be able to provide a label for the phrase immediately containing the specifier (the labelling of the specifier itself is addressed below). If this is correct, then the root of the unlabelled tree diagram corresponding to βP in (13) would be labelled by β following Minimal Search. The next-highest non-terminal will also be labelled by β , and the following by α .

So if labelling may be predictable in this way, and phrase markers therefore contain no labels for non-terminals, are the non-terminals really necessary? Consider that from (14), the strict order derived from the asymmetric c-command relations in (13), it would be possible to derive the original structure in (13) by adding a non-terminal for each node and labelling it using the Labelling Algorithm. Moreover, by adopting structures such as (14) it would be possible to take an abstract asymmetric ordering relation to be axiomatic in syntax, and hence it would no longer be necessary to define c-command or asymmetric c-command in graph theoretic terms. This abstract ordering relation would then map to order and hierarchy in externalisation. Following Minimalist concerns, it therefore seems desirable to attempt to dispense with non-terminals in favour of strict orders of terminals, and find a way to derive the distinction between heads and phrases from the strict orders themselves.



Before doing so, an important but unresolved problem is what Uriagereka (2011) terms the Address Issue, given in (16). The difficulty is that if the specifiers are represented as separate orders as in (14), they will need to be composed with the main order, e.g. in Spell-out—but if labelling occurs late in the phase, then it will be impossible to refer to one order from another using labels. Since there is no restriction on the type of items which stand in an order, one possibility is that the node labelled ? in (14) should have the value of the entire order it represents, as shown in (15), which corresponds to the notion of conservative Spell-out of specifiers proposed by Uriagereka (2011). Since this is clearly not a head, the Labelling Algorithm may consider this kind of object opaque, just as in the case of roots.

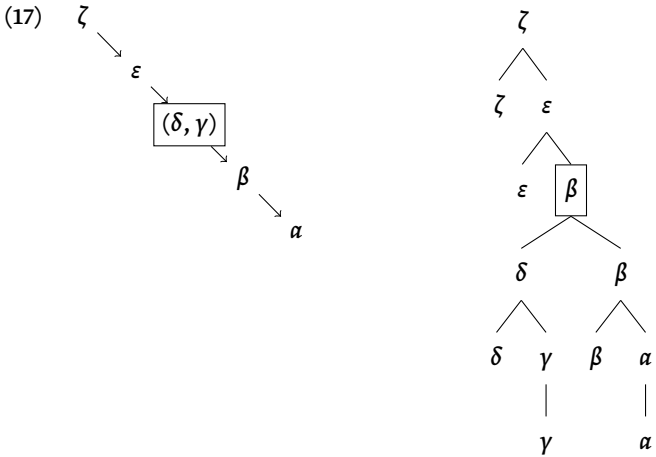
(16) The Address Issue

Whenever a phrase-marker K is divided into complex sub-components L and M , for K to meet LCA conditions of multiple Spell-out, the daughter phrase-marker M that spells-out separately must correspond to an identical term M within K .

(Uriagereka, 2011)

An interesting consequence of using Minimal Search for labelling as Chomsky (2013) suggests is that the ultimate root of the phrase marker is not

labelled (Chomsky, 2015)—but since this node by definition will not participate in any operations on the phase marker, a label may be redundant. Extending this idea, since labelling is taken to be an instance of Minimal Search, it could occur as part of any operation making use of Minimal Search which requires labels, rather than just at a particular point in the phase. Given the top-down direction of Minimal Search, the first node given a particular label will be always be the first node of a full phrase. To illustrate this point, consider (17), which shows the strict order and equivalent tree. If the head ζ is searching for some phrase headed by β , it will find (δ, γ) first. Since (δ, γ) cannot label a phrase, the label from the next node will be used, and the phrase containing (δ, γ) and everything following it will be labelled by β , as the tree shows. So the phrase identified by ζ will be a strict order containing all the nodes starting at (δ, γ) , as expected.



A major issue with this approach is that the nodes corresponding to specifiers need to have a category e.g. for selection, but this is not clear from their labels, e.g. (δ, γ) —the label could presumably be provided by Minimal Search starting at the first node of the specifier’s order, but this would require an operation to enter embedded orders to begin the search, since if this ability to enter embedded orders were automatic, then the proposed opacity of specifiers when labelling the phrases containing them could not be maintained. Since the ultimate focus here is on weak orders, and since it seems likely that any solution will require the representations posited here to

be made more complex, I leave this issue open.

Nonetheless it is important to emphasise that this is by no means the first approach to suggest that phrasal nodes should be abandoned—in fact it has been suggested that Bare Phrase Structure itself, when simplified as far as possible, ultimately leads to a dependency structure, which of course lacks phrasal nodes (for a review of this work see Osborne et al., 2011, and the works cited there). More recently, Bowers (2018) has also argued for an approach to structure based around relations, though his approach does not assume the primacy of strict orders argued for above. The approach detailed so far also implies that Merge is asymmetric, establishing an asymmetric relation between two arguments—this too is by no means a new idea, Zwart (2011) and Kayne (2013) being two rather different approaches implementing asymmetric Merge.

A more unusual feature of this approach is that it implies that the gap between structure in the Narrow Syntax and linearised structure is far smaller than normally assumed. The only function of linearisation under this view is to flatten recursively embedded specifiers into a single total order, in other words to linearise an existing order which has non-tail recursion. Because this approach also suggests that, abstracting away from the linearity of linearised structure, there is no distinction between order and hierarchy, then the Narrow Syntax, PF, and LF can all deal with the same representations, which make use of the same abstract asymmetric ordering relation. This effectively allows the requirements of the LCA to hold throughout the derivation as Kayne (1994) originally suggested (and contra e.g. Chomsky, 2014/1995), but without any claim that the kind of restrictions the LCA makes on, e.g. branching are due to the need to linearise: it is instead due to the axiomatic use of an asymmetric ordering relation. This relation is moreover far less mysterious than *c-command* in that it is completely axiomatic, as opposed to being defined in terms of Graph Theory, and maps to both order and hierarchy and hence is in evidence at both interfaces.

The fact that the syntax can then manipulate strict orders based on this relation is also significant. From a biolinguistic point of view, strict orders would seem very plausible as psychological realities due to the importance of sequencing both in humans and other species. The particular kind of sequencing that strict orders relate to has been tested experimentally for primates (Samuels et al., 2017), suggesting it would at least be available to be recruited by language. And while it might be argued that the ordered pairs representing the relations upon which orders depend can be defined in terms

of sets (e.g. Kuratowski's definition: $(a, b) := \{\{a\}, \{a, b\}\}$) and hence that relations are not sufficiently primitive to be used axiomatically in this way, this is only relevant if ordered pairs (and more generally tuples) are non-atomic as psychological realities, which is not a necessary assumption.

But most importantly, it highlights the question of why weak orders do not play any role in syntax, since discounting the distinction between total and non-total orders, orders are either strict or weak. Recalling the difference between asymmetry, $aRb \Rightarrow \neg bRa$, and antisymmetry, $aRb \wedge bRa \Rightarrow a = b$, it is possible that weak orders are used to bring about displacement, since symmetry in a weak order would cause two items to be treated as the same. The relation used to derive such orders would be an antisymmetric version of the abstract asymmetric ordering relation described above, only differing with regard to reflexivity and antisymmetry. If it is the case that antisymmetry (in this non-Kayninan sense) is what allows displacement, there ought to be apparent idiosyncrasies of this operation explained by this requirement, and in the following section I detail some predictions made for possible and impossible displacement.

2 The role of antisymmetry

Beyond the basic fact that displacement extends or modifies an existing structure using a subpart of this same structure, formalisms modelling displacement such as Internal Merge, Agree, and Autosegmental spreading are substantially different. Despite this, a strikingly similar condition on displacement has been proposed for all of them, ensuring that displacement leads to sequences of elements which are in some sense contiguous, formalised in syntax in the concept of Feature Geometry-based Relativised Minimality by Starke (2001) (and various other more recent proposals) and Contiguous Agree in multiple Agree by Nevins (2007), and in Autosegmental Phonology by the Line-Crossing Prohibition (which originates in the Well-formedness Condition of Goldsmith, 1976).

In each case, the fact that part of the structure is 'repeated' in some sense leads to a tension between identity and distinctness. These 'repeated' bits of structure, which can be termed occurrences following Chomsky (2004), are in one sense identical, given the fact that part of the structure is copied, but on the other hand also distinct, since it is possible to individuate specific copies. In the following section, I claim that this tension between identity and

In all of these cases, elements arising from displacement must form a contiguous chain with the source element: in the case of Relativised Minimality displacement must form a contiguous chain of occurrences based on the feature causing movement, in the case of Contiguous Agree again a contiguous chain of occurrences must be formed based on the feature involved in agreement, and in the case of autosegmental spreading in phonology, the node which is spreading must be linked to a contiguous stretch of segments of a particular type in the other tier, such as consonants as in (20). Importantly, contiguity in this sense is rarely with respect to the surface string: in syntax the existence of a chain does not deny that the members of the chain may be separated by large amounts of intervening structure, and in phonology that nodes to which association lines are added may not be adjacent. Rather, it is either maintained that the locality (which leads to contiguity) is relativised according to a particular feature, or that locality is in fact strict, and the fact that only certain nodes are relevant for the definition of a contiguous chain of nodes supports the postulation of distinct tiers of nodes, again according to a particular feature. But while the conditions specific to Movement, Agree, and spreading given above are able to capture these phenomena, the extent to which they are similar suggests that there is a deeper reason why displacement seems to have this characteristic, which presumably would have to be very basic.

Another, more obvious effect of displacement, is that what appears to be the same element (which may be as minimal as a feature value) appears to exist independently in multiple distinct positions. This would seem to be contradictory, and there is a range of options to account for this phenomenon, most implying it is an illusion at some level: one possibility is a copying operation which produces a new element identical to the original, which is the basis of the Copy Theory of Movement, and Chomsky's (2000) approach to Agree. Under this approach, a key issue is just how 'identical' the copies are to the original element. To some extent they must be distinct, since if they were entirely indistinguishable they would be impossible to order relative to one another, and hence impossible to linearise (Nunes, 2004). But if they were interpreted as entirely distinct, then once the copying operation had occurred, there would be no reason for the items to be construed as related unless they shared some property, such as an index, or were interpreted for some purposes as identical (e.g. interpretation) and for others (e.g. linearisation) as distinct.

Another approach is to take displacement to be a referencing operation: this is the approach taken by Unification within HPSG, and also some approaches

to agreement within Minimalism, notably Pesetsky and Torrego (2007). A referencing operation introduces a distinction between the true element (i.e. the source in a copying approach), and the references to this element, but it is not necessarily obvious to what extent this distinction is visible to the system. The Trace Theory of Movement notably adopted this approach, but in this case the ‘true element’ moved from its original position, leaving a Trace as a reference. Traces *were* interpreted as visibly different to the moved element from the perspective of the system for various purposes, but this is not a prediction of the general approach itself. However, importantly, under this approach any change to the true element should be always reflected in all of the references (and vice versa), which is not a prediction of a copying approach where the copies are basically distinct. A related approach, which is the position of Autosegmental Phonology, is that all objects in the system are reached through references, captured in this formalism by the association lines between different tiers.

Both these approaches face the same fundamental problem that identity and distinctness are contradictory, but apparently both required—as is seen in the way that the copying approach requires indices or variable interpretation, whereas the referencing approach often involves making a distinction between a single true element and the references to account for differing behaviour (e.g. with regards to (non-)pronunciation), or treating every occurrence of the element as an independent reference which can behave autonomously. To further complicate this, consider the Inclusiveness Condition in (21).

(21) Inclusiveness

A “perfect language” should meet the condition of inclusiveness: any structure formed by the computation (in particular, π and λ) is constituted of elements already present in the lexical items selected for N ; no new objects are added in the course of computation apart from rearrangements of lexical properties (in particular, no indices, bar levels in the sense of X-bar theory, etc. ...)

(Chomsky, 2014/1995)

This rules out capturing the identity aspect of displacement with indices or by using referencing, since the references will need to be added in the derivation after the true element is added. Variable interpretation would seem to be a more viable option, though defining the basis for this variability in a

principled way could pose a challenge. On the other hand, Chomsky suggests that: ‘with sufficiently rich formal devices (say, set theory), counterparts to any object (nodes, bars, indices, etc.) can readily be constructed from features’. The issue with using Set Theory to construct counterparts to indices in particular is the danger of inadvertently axiomatising the natural numbers using Set Theory, just as the mathematician Peano did—using these sorts of set-theoretic indices would imply that the system must be able to count, going against the general assumption that if grammars were able to count in this way, then there would be less reliance on structural dependence in language. But another formal device which could be used to construct a counterpart to the *effect* of indices is Order Theory, and in particular weak orders, given the property of antisymmetry.

2.2 Properties of weak orders

It has already been argued above that phrase structure is based around an abstract asymmetric ordering relation, which leads the syntax to manipulate strict orders of nodes that correspond to the terminals in a tree, but at the time no argument was made for selecting a strict order over a weak order. An important property of asymmetric relations is that they are irreflexive, meaning that the same element cannot appear twice in an order based on an asymmetric relation. Assuming that asymmetric relations form the basis of phrase structure, it is predicted that apparently identical nodes in strict orders should not be treated as identical—in other words they are coincidental repetitions of material and not copies of each other. This seems to account for the distinctness side of displacement, since as mentioned above occurrences need to appear to be distinct at some level in order to be linearised relative to one another (Nunes, 2004). But, as also established, displacement plainly involves a degree of identity. Importantly, antisymmetric relations do not enforce distinctness as they are not irreflexive, and as already mentioned actually enforce identity when symmetric. And since, abstracting away from whether orders are total, they are either strict or weak, the language faculty would be expected to make use of both these options absent some principled way of ruling out one or the other.

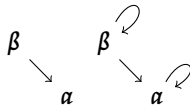
One possibility to resolve this tension between identity and distinctness would therefore be to posit that the same nodes may exist in both strict and weak orders at the same time, in parallel, and that whether a node is judged identical to or distinct from another depends on the order considered.

Nodes will always be judged distinct based on the strict order, due to the irreflexivity of asymmetric relations, but this is not necessarily true of weak orders, which may or may not show the nodes to be identical. In other words strict orders exist alongside weak orders throughout the derivation, and the fact that the two behave slightly differently sometimes leads to a tension between identity and distinctness. This is a Minimalist approach in a way that distinguishing D-structure and S-structure is not—the reason why the language faculty would putatively make use of these orders is because they follow from the use of asymmetric and antisymmetric relations, which represent the only two available options, which are equally simple. The distinction is more reminiscent of that between Internal and External Merge, which also represents a case where the Language Faculty makes use of the two available options it is given.

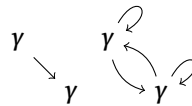
	Operation	Strict order	Weak order
(22)	External Merge	Distinct	Distinct
	Internal Merge	Distinct	Identical

(22) shows how this two-way distinction between orders would yield the difference in the effect of External and Internal Merge on the phrase marker. Crucially, Internal Merge establishes an extra antisymmetric relation when compared to External Merge, and this results in the treatment of identical occurrences as copies. This is shown graphically in (23): for External Merge as in (23a) there is one antisymmetric relation from β to a , whereas for Internal Merge as in (23b) there is both an antisymmetric relation from the first node γ to the second node γ and also from the second node γ to the first node γ . This is the full range of possibilities for establishing relations given two occurrences. One will always be ordered relative to the other in the strict order, but it may or may not be the case that the two occurrences will be interpreted as copies based on the weak order.

(23) a. External Merge



b. Internal Merge

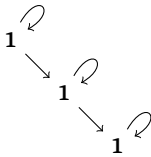


To introduce some shorthand now that two different relations are being used, the abstract asymmetric ordering relation corresponding to asymmetric c-command, (non-linear) precedence, and dominance can be represented with

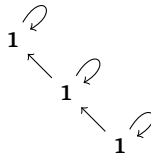
the precedence relation \prec , so $\beta \prec \alpha$ indicates that β precedes α , and $\beta \preceq \alpha$ is the equivalent antisymmetric relation indicating that β precedes or is equal to α . Obviously $\beta = \alpha$ indicates that β equals α . The difference between the \prec and \preceq relations is exactly that which distinguishes the greater than ($<$) relation from the greater than or equal to (\leq) relation, which are asymmetric and antisymmetric respectively.

With this in mind, it should now be possible to characterise chains. In (23b) the symmetric antisymmetric relations between γ and γ , i.e. $\gamma \preceq \gamma$ and $\gamma \succeq \gamma$, mean that by antisymmetry $\gamma = \gamma$. This situation corresponds to the usual notion of a chain. But unlike the use of indices, there are limits on what chain formation is possible assuming that chains represent equality relations between nodes in weak orders. Consider first of all the order $1 \leq 1 \leq 1$ shown in (24a). The inverse of \leq is \geq . Clearly it is also true that $1 \geq 1 \geq 1$ as in (24b). So since these relations are antisymmetric and the orders weak, $1 = 1 = 1$ as in (24c). But it also appears that antisymmetry rules out a large number of illicit chains, because equality, as opposed to arbitrary indexing, will not allow the formation of chains of non-identical syntactic objects. Thus the order in (25) is contradictory. Clearly $1 \neq 2 \neq 3$.

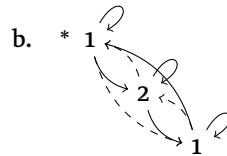
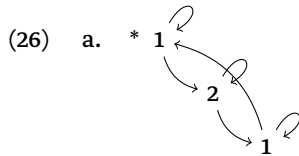
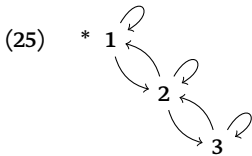
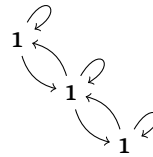
(24) a. $1 \leq 1 \leq 1$



b. $1 \geq 1 \geq 1$



c. $1 = 1 = 1$



But this is not the only restriction antisymmetry places on chains. Consider the graph in (26a). This is intended to show that $1 \neq 2$ and $2 \neq 1$ but $1 = 1$, yet by transitivity we also have the relations shown by the dashed lines in (26b). So this would actually show that $1 = 2 = 1$, which is false. Orders must be monotonic: in the case of strict orders this follows from the irreflexivity of asymmetric relations, and in the case of weak orders from antisymmetry.

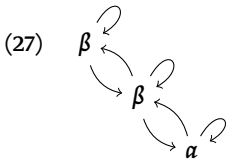
Because of monotonicity there can be no intervener between two nodes in a chain, where an intervener is a syntactic object distinct from the adjacent objects in the relevant sense. In the case of Internal Merge the relevant sense is usually the whole node, in the case of Agree usually a feature (though this will be made more precise below). Hence the properties of weak orders predict that all chains must be contiguous.

Before showing how this approach works derivationally, it is useful to compare some properties of the representations envisaged with some existing approaches to chains. Under the contextual distinctness approach of Martin and Uriagereka (2014), copies and repetitions are distinguished based on context alone: repetitions within a single phase are judged impossible, and are instead interpreted as copies. For A-chains at least, occurrences across two phases would be interpreted as repetitions—though note that, since the A-chains for the subjects of passives and unaccusatives would appear to cross a phase boundary, it will be impossible to form A-chains for these. An even more obvious issue for this approach is A-bar chains, since these uncontroversially span multiple phases, yet involve copies, hence it will be necessary for the chain in these cases to remain after the transfer of a phase. Martin and Uriagereka (2014) suggest that for A-bar chains it will be necessary to extend existing differences between Internal and External Merge:

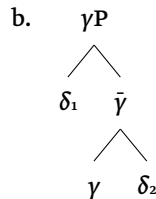
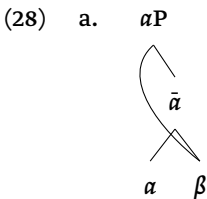
...whereas external merge creates a new term, or syntactic object, in a phrase marker, internal merge works differently in that it, in some sense, “stretches” the very same syntactic object across two different syntactic contexts. (Martin and Uriagereka, 2014)

This ‘stretching’ is what leads to the obligatory formation of a chain. This clearly captures the tension between identity and distinctness mentioned above, and is explicitly recognised in the characterisation of Internal Merge: ‘there is only one occurrence to speak of, albeit one that exists simultaneously in multiple syntactic contexts’. So in this case, the difference between copies and repetitions reflects the difference between the effect of Internal and External Merge on the phrase marker (though the authors do not explicitly abandon the phase-based characterisation for A-chains). This corresponds to the distinct effects of these operations on the phrase marker posited in (22). Martin and Uriagereka (2014) further suggest that the appearance of a chain is illusory: the system is fooled into interpreting only one syntactic object when two have been presented, because the two objects look identical to each other, and have been presented consecutively. Hence a syntactic object is interpreted

as distinct if it cannot be collapsed with another adjacent identical object. This too corresponds to what has been presented above, though in a less obvious way: the equality relations between distinct nodes in weak orders such as (24c) suggest that in the case of weak orders, the system may be fooled into treating distinct nodes as the same because there is no way to formally distinguish between a reflexive relation on a single node, as in the reflexive relation on the first node in (27), and a relation from one identical node to another, as in the relation between the first and the second node in (27). It is therefore possible that chain formation is automatic, an issue I return to shortly.



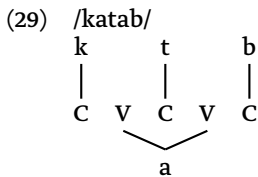
Collins and Groat (2018) note that Multidominance could also be used to achieve the effect of chain formation. In this case, a single node dominated by multiple mothers (as in (28a) appears in multiple positions in the structure, as opposed to the equivalent structure without multidominance in (28b)).



In a sense, these structures conflate the strict orders and weak orders posited above by attempting to show identity and distinctness in the same structure. As can be seen, this comes at the expense of the distinctness aspect of displacement, since it is no longer possible to distinguish occurrences of β in the structure in (28a). And as Citko (2011) notes, the real hurdle for multidominance structures is linearisation, because such structures cannot be straightforwardly linearised with the LCA. There are various ways to overcome this, some outlined by Citko (2011), and another proposed by Gračanin-Yuksek (2013). Discounting approaches abandoning the LCA, these usually involve modifying either the multidominance structures, or the definition of the LCA. However, fundamentally, these attempts create the impression that

multidominance structures are unavoidably awkward to linearise, and the persuasive simplicity of linearisation with the LCA cannot be retained. So given the linearisation difficulties encountered by multidominance due to its emphasis on identity in displacement at the expense of distinctness on the one hand, and the difficulty of showing identity without indices in approaches such as Bare Phrase Structure, on the other, it seems warranted to adopt two distinct representations, weak and strict orders, each accounting for the identity and the distinctness aspect of displacement respectively.

In claiming that strict orders from weak orders coexist in this way, a clear analogy can be made with the multiple tiers of Autosegmental Phonology, as shown in (29)³. In this formalism, a timing tier with CV skeleta indicating the order of pronunciation of segments can be distinguished from other tiers corresponding to features, in this case vowels and consonants. Nodes from these latter tiers are linked to nodes on the timing tier via association lines, and multiple linking allows displacement to be captured. Here /a/ is multiply-linked to two V nodes, which allows /a/ to be represented as a single morpheme even if it surfaces in two distinct positions in the surface string. Notice also that the association lines between consonants in the consonant tier only link to nodes in the timing tier designated as consonants, and the same for vowels. In much the same way, the weak orders in which syntactic chains form will contain a subset of the nodes in the strict order, with membership of the subset based on the presence of a feature.



The timing tier in (29) corresponds to the strict orders discussed above, which are used to extend the phrase marker with new Lexical Items just as the timing tier is extended with CV skeleta, allowing the hosting phonological features, which can then surface in pronunciation. The other tiers correspond to the various kinds of chains of structural positions which show Relativised Minimality effects (there are various suggestions for these; for now I follow Rizzi, 2001, 2011), such as head chains and chains based on feature classes. The resulting orders are in (30).

³Here I abstract away from the use of root nodes, choosing between an X-slot or moraic analysis, and Feature Geometry for expository purposes.

(30)	Order	Membership criteria
	Strict	None (all nodes)
	Weak	Head positions
	Weak	Argumental features: Person, Number, Gender, Case,...
	Weak	Quantificational features: <i>wh</i> , Foc, Neg, Measure, Frequency
	Weak	Modifier features: Evaluative, Evidential,...
	Weak	Manner, Measure, Frequency, Neg,...
	Weak	Topic

(adapted from Rizzi, 2011)

In summary, under this view the existence of Relativised Minimality, Contiguous Agree, and the Line-crossing Prohibition reduces to a weak ordering effect, and the conflicting requirements of identity and distinctness in displacement are due to the possibility for elements to be in at least two orders, which may give contradictory ordering information. Strict orders are used to extend the phrase marker, may be embedded inside each other (at least in syntax), and contain timing information used by linearisation at PF, as well as information used for determining scope and binding at LF. Weak orders on the other hand allow displacement to occur because they permit the formation of equality relations between nodes, and only contain a subset of the nodes in the strict order. Having established these representations, it is now possible to show how this approach works derivationally.

Deriving contiguity in chains

The above has focused on the representational side of asymmetry and antisymmetry, but clearly to derive contiguity effects it will be necessary to insert these representations into a derivational framework. The intention is that by adopting the representations proposed above, the derivation can proceed much as usually conceived. To highlight a number of general consequences of adopting the representations proposed, I first show how some schematic derivations would work before giving specific examples for the phenomena intended to be captured.

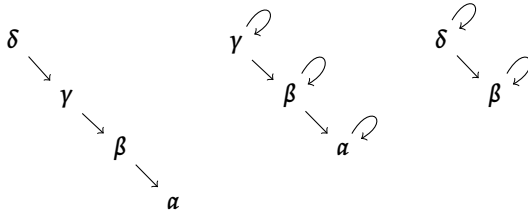
3 Schematic derivations

To begin, I illustrate the effect of External Merge and Internal Merge on the Phrase Marker, and in particular the fact that chain formation is automatic given weak orders, before extending this approach to Agree. I then consider how Multiple Spell-out and phases impact on the approach developed, and the relationship between cyclicity and Order Theory more generally.

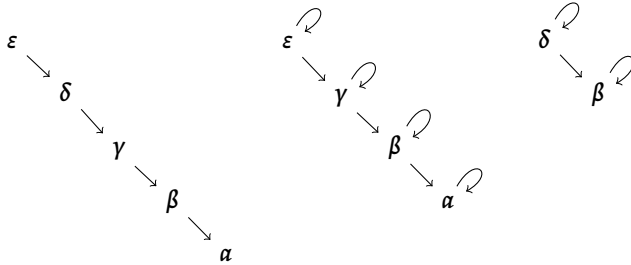
3.1 Operations on the phrase marker

Consider first External Merge. In (31a) besides the strict order there are two weak orders, each corresponding to a feature which all nodes in that particular order share, $[F_1]$ and $[F_2]$ respectively. Suppose as in (31b) a new item ε is merged with the existing structure in (31a), and that ε has $[F_1]$. Merge will need to establish a precedence relation between ε and δ in the strict order first, and from this the ordering in the weak order can be derived. So in the strict order, ε will come to be adjacent to δ , which was the first node in this order in (31a). Since ε has $[F_1]$, it also comes to be adjacent with γ in the weak order for $[F_1]$, as γ was the first node in this order in (31a), and because by transitivity ε precedes γ in the strict order. Because ε and γ are distinguishable as nodes, ε is merely ordered before γ as in the strict order. Since ε does not have $[F_2]$, it does not appear in the weak order for $[F_2]$. Linearisation would straightforwardly yield $(\varepsilon, \delta, \gamma, \beta, \alpha)$ from the strict order, and there is no conflict in ordering between the strict order and any of the weak orders.

(31) a. Strict order Weak order: [F₁] Weak order: [F₂]

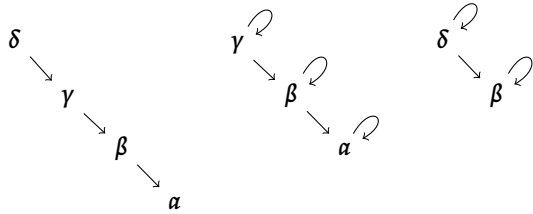


b. Strict order Weak order: [F₁] Weak order: [F₂]

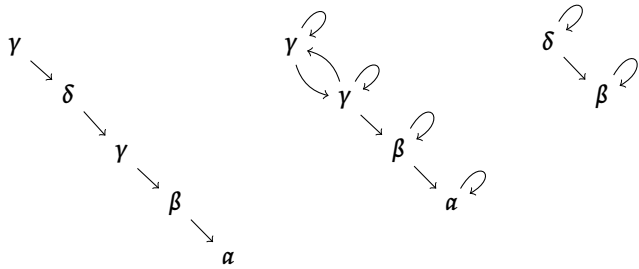


Now consider Internal Merge. In (32a) there is one strict order and two weak orders, one for [F₁] and the other for [F₂]. Suppose that this time γ , part of the existing structure in (32a) is merged with the overall structure, as in (32b), and that γ again has [F₁]. Again, in the strict order, γ will come to be adjacent to δ —recall the assumption that in strict orders every node is taken to be distinct to avoid violating irreflexivity. Additionally, since γ has [F₁], it also comes to be adjacent to the other γ in the weak order for [F₁]. But because the two γ nodes in the weak order are identical, the relation from γ to γ in the strict order leads to the vacuous establishment of a relation from (some) γ to (some) γ , i.e. a relation identical to the existing reflexive relation on γ . This results in a symmetric relation in the weak order between the two γ nodes, as shown in (32b), as they cannot be distinguished. Importantly, based on the relations and the nodes, the structure in (32b) is identical to the structure in (32c), because the relation from the first node γ to the second node γ is formally identical to a reflexive relation on a single node γ . This would suggest that the system perceives only a single γ in the weak order for [F₁] in (32b). It is possible that Internal Merge maps directly from (32a) to (32c), but for the moment I assume (32b) is merely perceived as (32c). So copy formation is obligatory, and arises from a kind of ‘collapsing’ of the phrase marker, as Martin and Uriagereka (2014) suggest. And again, since γ does not have [F₂], it does not appear in the weak order for [F₂].

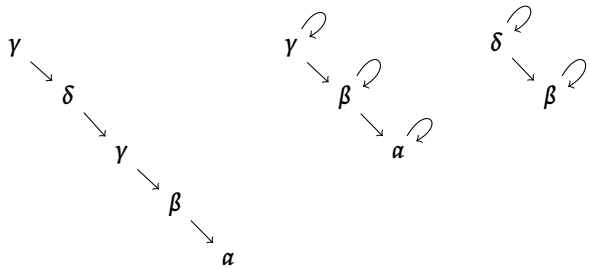
(32) a. Strict order Weak order: [F₁] Weak order: [F₂]



b. Strict order Weak order: [F₁] Weak order: [F₂]

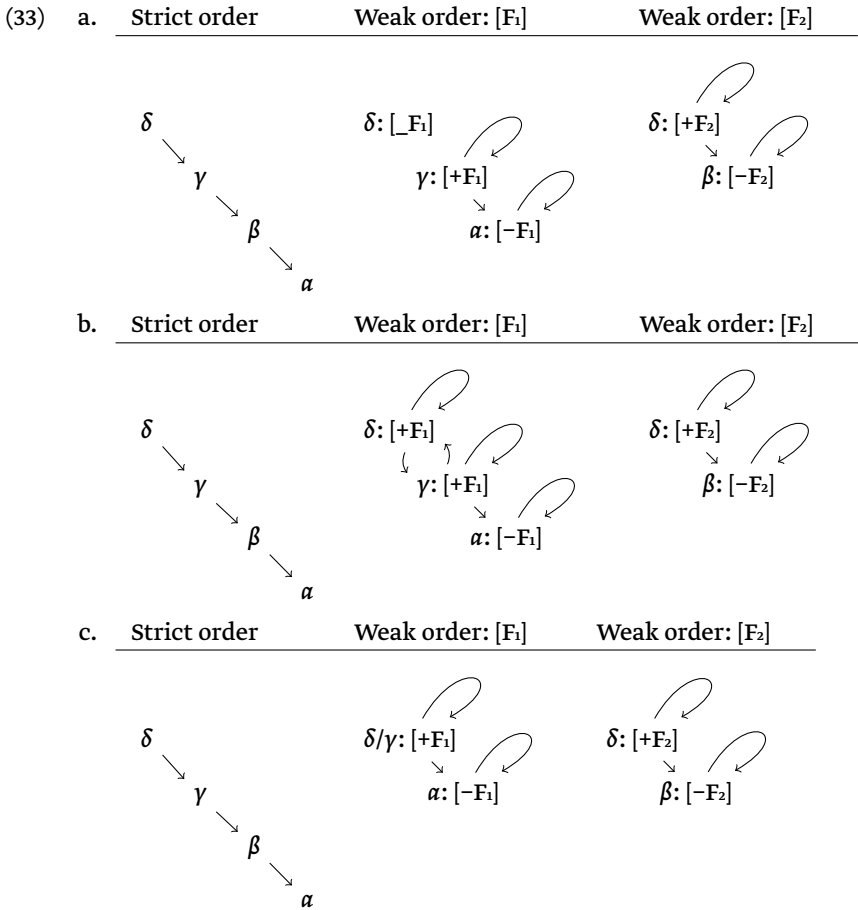


c. Strict order Weak order: [F₁] Weak order: [F₂]



This approach can now be extended to Agree. Both External and Internal Merge extend the phrase marker by adding a node to the strict order, whereas Agree crucially does not. Additionally, whereas Internal Merge creates copies of syntactic objects, following the conception of Agree as feature sharing by Pesetsky and Torrego (2007), Agree can be seen as creating copies of features or feature values. Under the approach here, Agree copies feature values specifically, and it is these which stand in weak orders, though for expository purposes the feature which the value corresponds to and the node in the strict order which the feature corresponds to are also shown. I also assume following Pesetsky and Torrego (2007) that both interpretable and

uninterpretable features may be valued or unvalued, and hence when showing feature valuation I do not distinguish interpretable and uninterpretable features. In (33a), δ has an unvalued $[F_1]$ feature, meaning that at this point there is no node for δ in the weak order for the feature value of $[F_1]$ (shown with the underscore). In (33b), the value from γ is copied and, due to identity, an equality relation is established between the copy and the original feature value. And (33c) shows how the system would treat the two identical copies resulting from this as a single node.



While this example works for downward Agree, where the unvalued probe is higher than the valued goal, it might seem more challenging to handle

upward agree in this way, because of the need to insert nodes in the middle of an existing order. Consider for example the weak order based on the \leq relation in (34). Inserting 0 in the order in (34) before the least element 1, as in (35) will minimally require the establishment of a single precedence relation $0 \leq 1$ from which all others ($0 \leq 2$, $0 \leq 4$, $0 \leq 5$) can be derived by transitivity. On the other hand, inserting 3 between 2 and 4 as in (36) will minimally require the establishment of two relations, $3 \leq 4$ and $2 \leq 3$.

$$(34) \quad 1 \leq 2 \leq 4 \leq 5$$

$$(35) \quad 0 \leq 1 \leq 2 \leq 4 \leq 5$$

$$(36) \quad 1 \leq 2 \leq 3 \leq 4 \leq 5$$

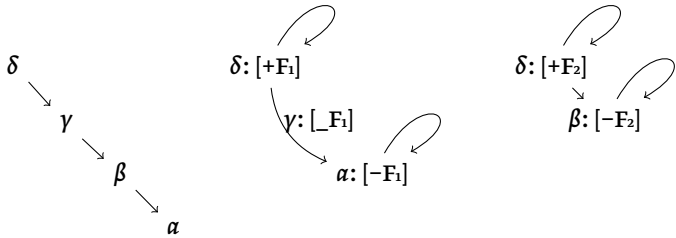
By establishing only one relation, it would not be possible to get a total order—there would either be no ordering relation between 2 and 3, or between 3 and 4. But neither External nor Internal Merge as defined above, establishes more than one relation within a single order, as would need to happen in (36), and being able to insert arbitrary material in the middle of the phrase marker as in (36) seems dubious. Yet note that neither (37) nor (38), where material is copied within the phrase marker, requires the establishment any relations that do not follow from node identity—in (37) the relation $2 \leq 2$ between the two resulting identical nodes follows from the existing reflexive relation $2 \leq 2$, and in (38) the same is true of $4 \leq 4$. So in fact there should be no problem with the insertion of a node in the middle of a weak order as long as it is a copy of an adjacent node, which is expected under Agree.

$$(37) \quad 1 \leq 2 \leq 2 \leq 4 \leq 5$$

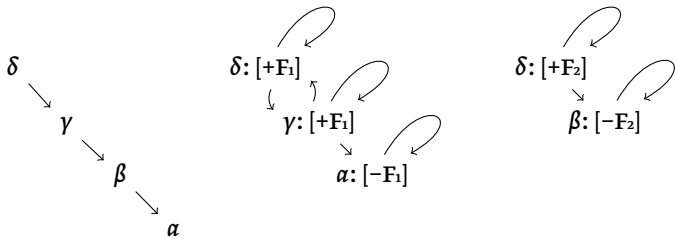
$$(38) \quad 1 \leq 2 \leq 4 \leq 4 \leq 5$$

The situation before upward Agree takes place is illustrated in (39a). In (39b), copying the value of $[F_i]$ from δ to γ results in these two nodes being indistinguishable, and hence due to identity, an equality relation is established between the values of $[F_i]$ for δ and γ . Note that the relation from $[+F_i]$ to $[-F_i]$ is maintained, but is shown in (39b) from the value of $[F_i]$ for γ because the relation from the value of $[F_i]$ for δ can be derived by transitivity. And finally, as above, (39c) shows how the two adjacent identical nodes would be treated by the system as a single node.

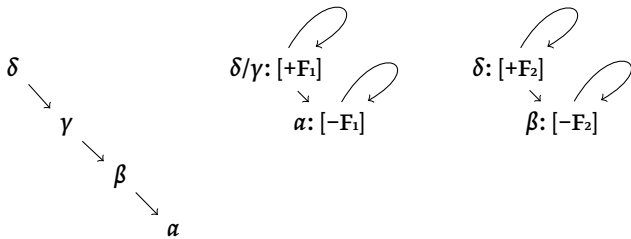
(39) a. Strict order Weak order: [F₁] Weak order: [F₂]



b. Strict order Weak order: [F₁] Weak order: [F₂]



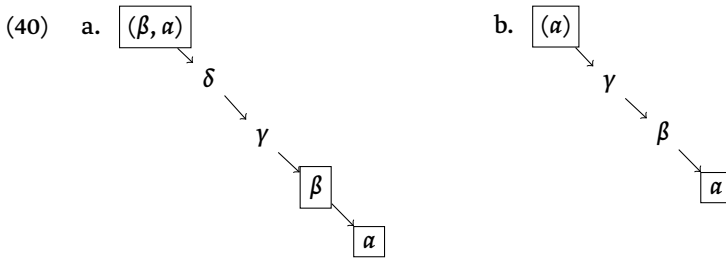
c. Strict order Weak order: [F₁] Weak order: [F₂]



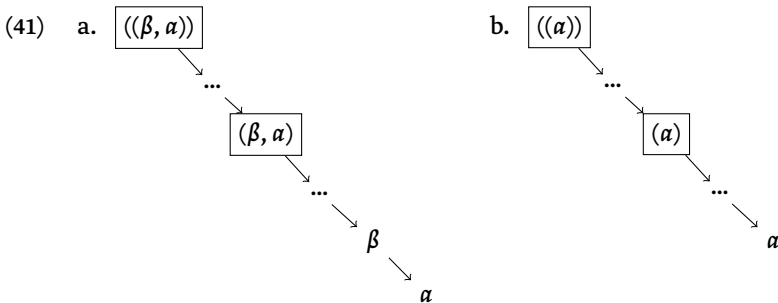
3.2 Embedded strict orders and Phases

Recall that in the case of Internal Merge, which creates identical nodes in strict orders, every node must be taken to be distinct to avoid violating irreflexivity. While it is possible to imagine how this might be implemented—nodes could be distinguished based on their relations for example—it introduces an undesirable distinction from weak orders, which must use node labels (or some analogue) to determine identity if the effects on displacement posited above are to be derived from their use. However, irreflexivity is of course only relevant when two items in a strict order are identical, and if specifiers are

represented as embedded strict orders as suggested above, then at least the first instance of phrasal movement will sidestep the irreflexivity requirement through embedding, as shown in (40a). And if Harizanov and Gribanova (2018) are correct in their distinction of syntactic head movement from what they term ‘amalgamation’, and that syntactic head movement specifically involves the movement of a head to a specifier position of some other phrase, then the first instance of head movement will also fail to violate irreflexivity, as in (40b). In other words, where material moves off the clausal spine into a specifier, there should be no violation of irreflexivity in the strict order as there will be no direct relation between the two resulting copies.



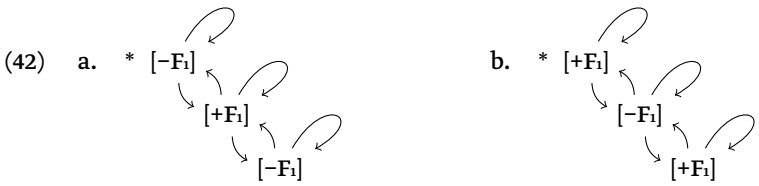
This will only work for cases involving two copies—once identical specifiers appear, then the irreflexivity problem resurfaces. One possibility for resolving this would be to take successive applications of Internal Merge to cause the content inside specifiers to become recursively embedded, reminiscent of an approach by Groat (2018) to distinguishing repetitions using recursively embedded sets (i.e. $\{a\} \neq \{\{a\}\} \neq \{\{\{a\}\}\}$). This is shown for phrasal movement and head movement respectively in (41a) and (41b).



The logic of this approach rests on the fact that the system does not treat specifiers as complex nodes, and hence copying a specifier as in (41a) and (41b) is

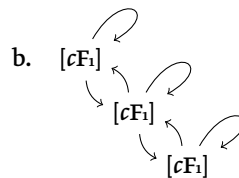
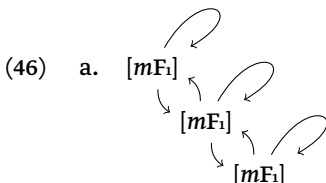
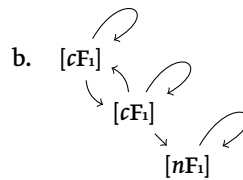
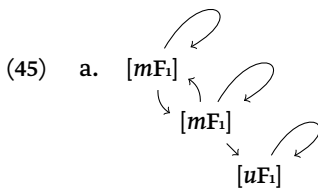
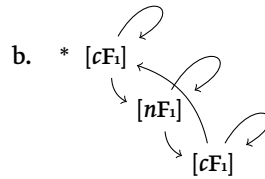
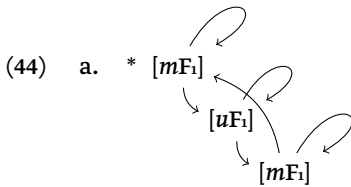
parallel to copying a head as in (40b). One possible concern about Groat’s (2018) proposal is that it appears to reconstruct the natural numbers in set-theoretic terms (i.e. $\{a\}$ is a_1 , $\{\{a\}\}$ is a_2 , $\{\{\{a\}\}\}$ is a_3 , ...), and hence distinguishing repetitions on this basis reduces to distinguishing the set-theoretic counterpart of indices. However, it is difficult to see how this would be possible without the ability to enumerate and then compare the amount of embedding of two occurrences in different parts of a phrase marker—an operation which would be surprising when compared with more familiar syntactic operations which lack this ability to enumerate elements and compare such enumerations. On the other hand, under this approach the fact that specifiers are embedded strict orders follows from the need to treat these structures as non-complex for ordering purposes, and the suggestion that the embedding is recursive for successive instances of Internal Merge follows from the fact that specifiers are treated as non-complex by the system. The effect of this simply ensures that no two items in the same strict order will be identical, satisfying irreflexivity.

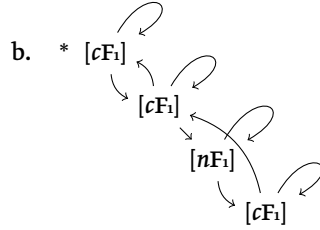
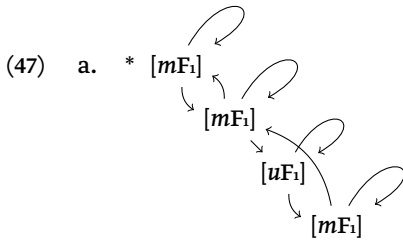
While this is able to prevent irreflexivity violations in strict orders, a similar issue arises for weak orders: consider that if the approach taken above for Agree is correct, and features are binary, then the sequences of feature values in (42a) and (42b) are impossible. Clearly, it would be undesirable for this effect to hold over the entire phrase marker. Instead, it would seem more plausible that (42a) and (42b) could hold within phases, and this would appear to be supported by one of the conditions introduced by Nevins (2007) on multiple Agree, namely Contiguous Agree, given in (43). Importantly, Nevins (2007) assumes that multiple Agree, and by extension this condition, respects phase boundaries.



- (43) Contiguous Agree: For a relativization R of a feature F on a Probe P , and $x \in \text{Domain}(R(F))$,
 $\neg \exists y$, such that $y > x$ and $P > y$ and $y \notin \text{Domain}(R(F))$
 “There can be no interveners between P and x that are not in the domain of relativization that includes x ”.
 (Nevins, 2007)

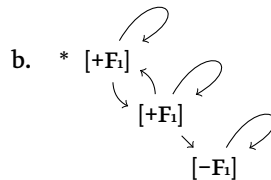
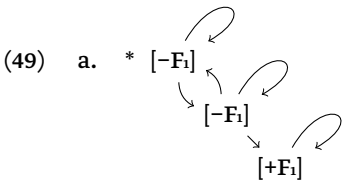
$R(F)$ or the relativisation R of a feature F is introduced to account for phenomena where Agree appears to be able to ignore otherwise legitimate goals, because the feature values of these goals are either unmarked or non-contrastive. The relativisation is supposed to affect the search, though the definition in (43) would seem to indicate that the search is also affected by values outside the relativisation. $R(F)$ may also include all the values of F , and in theory it is possible to require marked and contrastive values in combination. If $R(F)$ only has marked values, then a violation of (43) will appear as in (44a), where m designates marked values and u unmarked values, hence $[uF_i]$ is an intervener. Similarly if $R(F)$ only has contrastive values, then a violation of (43) will appear as in (44b), where c designates contrastive values and n non-contrastive values, hence $[nF_i]$ is an intervener. (45a) and (45b), however, do not violate (43), as the use of weak orders would predict. (46a) and (46b) are obviously also fine. However, (47a) and (47b) both violate (43) because of the enlarged domain of relativization. So under the weak orders approach, (43) should follow from the properties of displacement and does not need to be stipulated.





The conditions on possible sequences of actual feature values (e.g. + and -) are even stricter, due to the second condition on multiple Agree which Nevins (2007) introduces, given in (48). If $R(F)$ includes all the values of F , then (48) prohibits (42a) and (42b), but also (49a) and (49b). While weak orders themselves do not rule out (49a) and (49b), the fact that (42a) and (42b) are ruled out shows that antisymmetry is satisfied, even if vacuously. Hence, the assumption that multiple Agree respects phase boundaries adopted by Nevins (2007) should also apply relatively straightforwardly to this account, given that the configurations ruled out by weak orders are a subset of those ruled out by the combination of Contiguous Agree and Matched Values.

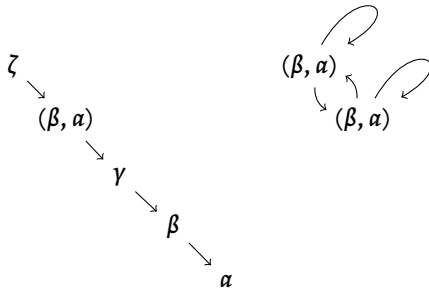
- (48) Matched Values: For a relativization R of a feature F , $\exists a, a \in \{+, -\}$,
 $\forall x, x \in \text{Domain}(R(F)), \text{val}(x, F) = a$
 “All elements within the domain of relativization must contain the same value for the feature F being agreed with”.
 (Nevins, 2007)



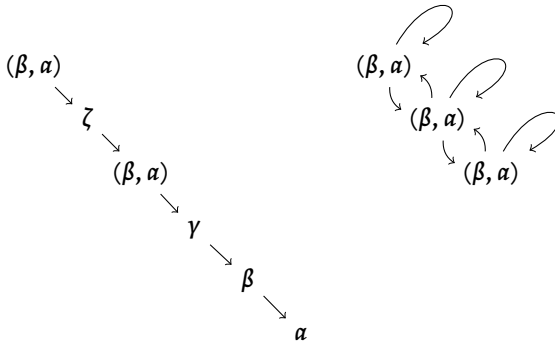
Weak orders used for Agree will therefore need to undergo transfer at each phase boundary, and the same will be true of the corresponding nodes in the strict order. This can then be extended to the weak orders used by Internal Merge—in this case all of the nodes in the weak order except those corresponding to the phase edge will be transferred. Note that this means that chains can only exist within phases, hence successive-cyclic movement will

involve as many chains as there are cycles. Consider the situation in (50a), where ζ is a phase head, and hence will undergo transfer with the elements it precedes at the end of its phase⁴. The elements in the weak order are all phrases, as represented by the embedded orders. (β, α) needs to move to a position in a higher phase, and hence in (50b) moves to the phase edge. Transfer occurs and the resulting structure is as in (50c)— $((\beta, \alpha))$ could then be displaced later, forming a new chain.

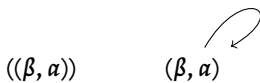
(50) a. Strict order Weak order: [F₁]



b. Strict order Weak order: [F₁]



c. Strict order Weak order: [F₁]



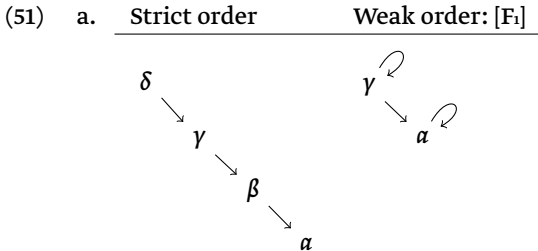
⁴In the following structures I abstract away from Anti-Locality considerations for expository purposes

4 Antisymmetry and Contiguity Effects

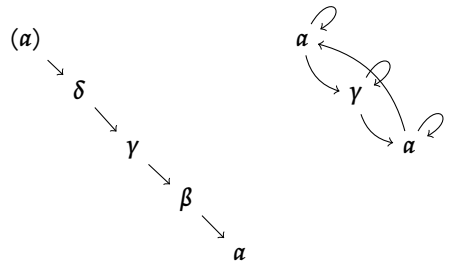
Now the derivational details are in place to show how this system is able to derive the effects traditionally characterised separately as Relativised Minimality, Contiguous Agree, and the Line-crossing Prohibition. I start with Relativised Minimality which is in a sense the most straightforward, since the effect of weak orders is broadly similar to what has traditionally been proposed. This is then followed by an example showing Contiguous Agree (where I assume the previous discussion on the relevance of weak orders to Contiguous Agree), and finally I show that the Line-crossing Prohibition can also be derived under this approach.

4.1 Relativised Minimality

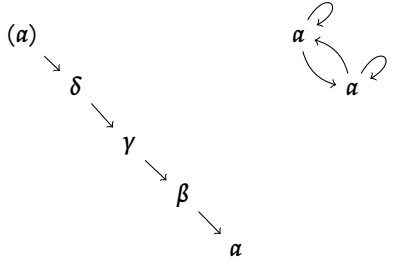
First consider Relativised Minimality involving heads, starting from the structure in (51a), where the items in the weak order are all heads. Following Harizanov and Gribanova (2018), head movement creates specifiers, hence in (51b) syntactic head movement creates a specifier (α) for δ in the strict order. The situation in (51b) constitutes a Relativised Minimality violation which can be derived from the fact that the weak order violates antisymmetry (γ is implied to be the same as α). (51c) on the other hand, does not violate antisymmetry, as γ does not have $[F_i]$ and hence is not present in the weak order.



b. * Strict order Weak order: [F₁]

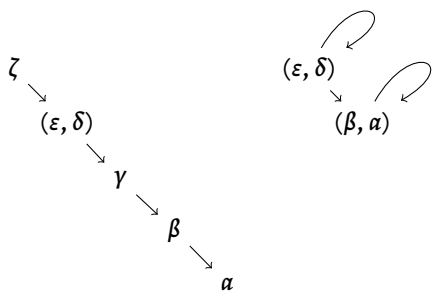


c. Strict order Weak order: [F₁]

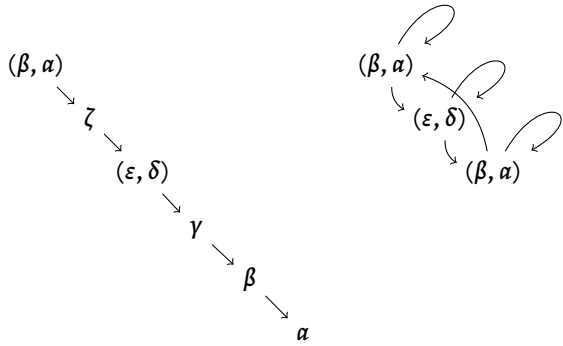


Next consider Relativised Minimality involving phrases, starting from the structure in (52a), where the items in the weak order are phrases: (β, a) is the phrase starting at β in the strict order, and (ε, δ) is the phrase starting at ε in the specifier in that order. The situation in (52b) where (β, a) becomes a specifier of ζ constitutes a Relativised Minimality violation which can be derived from the violation of antisymmetry in the weak order— (β, a) is implied to be the same as (ε, δ) . If (ε, δ) were not present in the weak order, as in (52c), there would be no violation.

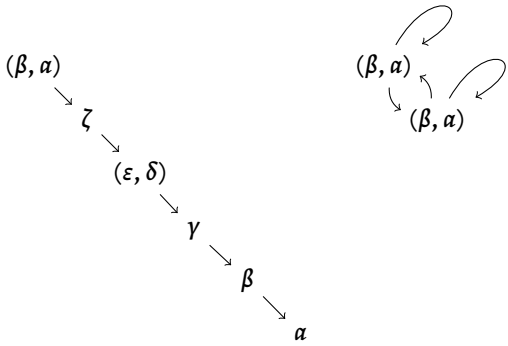
(52) a. Strict order Weak order: [F₁]



b. * Strict order Weak order: [F_i]

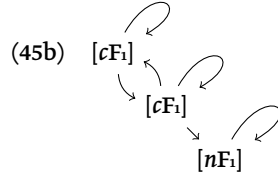
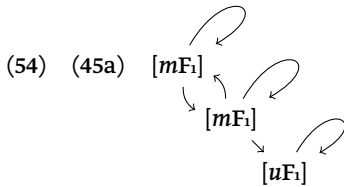
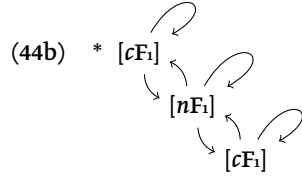
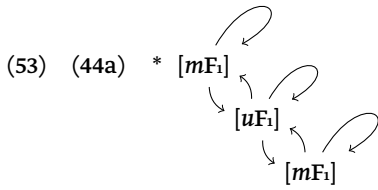


c. Strict order Weak order: [F_i]



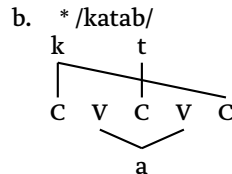
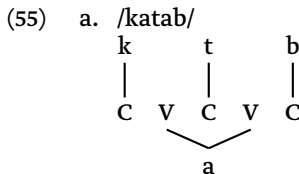
4.2 Contiguous Agree

Weak Orders can also derive Contiguous Agree: where Nevins (2007) parameterises the search started from some node N , which may seek all feature values, only marked values, or only contrastive values, this also can be seen as a parameterisation of the kind of values that N is able to take. If N must be marked, Contiguous Agree rules out sequences such as that in (44a), repeated below, which follows from antisymmetry. Likewise if N must be contrastive, a structure like (44b), repeated below, is ruled out. (54) shows licit alternatives where the marked or contrastive values are contiguous with the probe.



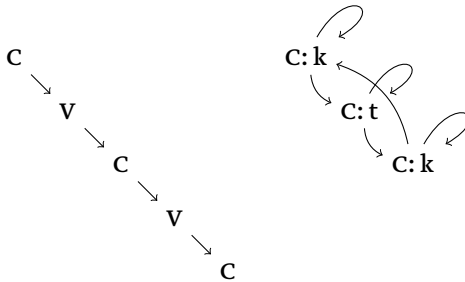
4.3 Line-crossing Prohibition

Finally, the Line-crossing Prohibition in phonology can also be derived from the use of weak orders for displacement. Suppose that strict orders in phonology can be identified with a timing tier (or root node tier in more recent work) and weak orders with the various other tiers, as was suggested above. In this case, it is unsurprising that the Line-crossing Prohibition should exist, since a restriction stating that displaced elements cannot cross over one another if they exist on the same tier, as in (55b) can be captured using weak orders, as in (56)⁵.



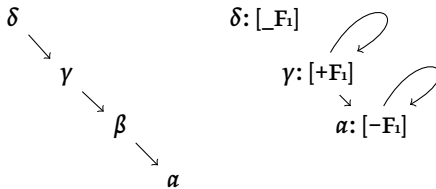
⁵This abstracts away from the problem of the apparently identical nodes in the strict order.

(56) * Strict order Weak order: [F₁]

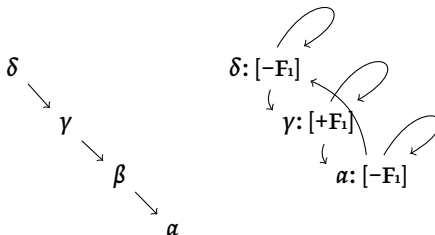


Note how similar these representations are to those used above for Agree. Nevins (2010) has indeed shown that many different kinds of vowel harmony can be understood as instantiating a search for some other vowel, and then copying the value of the vowel found, as Agree does. Violations of antisymmetry which are traditionally captured by the Line-crossing prohibition are shown in (57b) and (58b). Note from the previous discussion that it will also be impossible to insert material into the middle of the weak orders, hence vowel harmony which affects an underspecified vowel surrounded by two other vowels which can be targeted by the Agree operation will only ever result in the central vowel being valued by one or other of the surrounding two vowels.

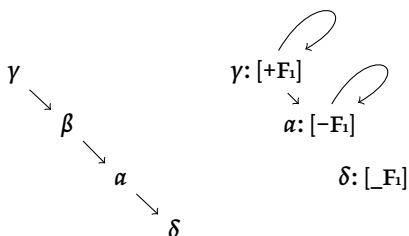
(57) a. Strict order Weak order: [F₁]



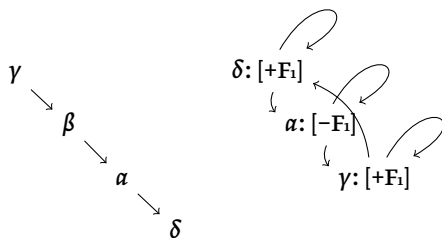
b. * Strict order Weak order: [F₁]



(58) a. Strict order Weak order: [F₁]



b. * Strict order Weak order: [F₁]



5 Conclusion

If nothing else, what should be clear from the above is that weak orders have not up to now received as much attention as they merit, if their properties are able to account for what seem to be very basic properties of displacement seen in both syntax and phonology. I have also argued that there is evidence to support a much more relational view of syntax, which, if adopted, minimises some of the architectural asymmetries introduced if weak orders are to model displacement, but strict orders become relevant to phrase structure later in the derivation—instead, phrase structure itself is based around strict orders, which explains among other things the prevalence of c-command in the characterisation of syntactic phenomena. Most importantly however, the adoption of weak orders to model displacement in Internal Merge, Agree, and spreading in Autosegmental Phonology allows a number of strikingly similar but usually separately stipulated locality conditions to be unified in a way which avoids stipulating some new metacondition—Relativised Minimality, Contiguous Agree, and the Line-crossing Prohibition all follow from the use of weak orders to model displacement, which itself arguably follows from the use of Order Theory in syntax, and possibly also phonology.

6 References

- Barker, Chris, and Geoffrey K. Pullum. 1990. A theory of command relations. *Linguistics and Philosophy* 13:1–34. URL <http://dx.doi.org/10.1007/bf00630515>. Cited page 6
- Bowers, John. 2018. Relational derivation. In *Deriving syntactic relations*, 6–48. Cambridge University Press. URL <http://dx.doi.org/10.1017/9781316156414.002>. Cited page 13
- Bruening, Benjamin. 2014. Precede-and-command revisited. *Language* 90:342–388. URL <http://dx.doi.org/10.1353/lan.2014.0037>. Cited page 6
- Chomsky, Noam. 1986. *Barriers*, volume 13. Cambridge, Mass: MIT Press. Cited page 8
- Chomsky, Noam. 2000. Minimalist inquiries: the framework. In *Step by step: Essays on minimalist syntax in honor of howard lasnik*, ed. Roger Martin, David Michaels, and Juan Uriagareka, 89–155. Cambridge, Mass.: MIT Press. Cited page 16
- Chomsky, Noam. 2004. Beyond explanatory adequacy. In *Structures and beyond*, ed. Adriana Belletti and Guglielmo Cinque, 288. Oxford, UK: Oxford University Press. Cited pages 4 and 14
- Chomsky, Noam. 2013. Problems of projection. *Lingua* 130:33–49. URL <http://dx.doi.org/10.1016/j.lingua.2012.12.003>. Cited pages 4, 7, 10, and 11
- Chomsky, Noam. 2014/1995. *The minimalist program*. Cambridge, Mass.: The MIT Press. URL <http://dx.doi.org/10.7551/mitpress/9780262527347.001.0001>. Cited pages 3, 4, 6, 7, 13, 17, and 18
- Chomsky, Noam. 2015. Problems of projection: Extensions. In *Linguistik aktuell/linguistics today: Structures, strategies and beyond*, ed. Elisa Di Domenico, Cornelia Hamann, and Simona Matteini, 1–16. Amsterdam: John Benjamins Publishing Company. URL <http://dx.doi.org/10.1075/la.223.01cho>. Cited pages 4, 10, and 12
- Citko, Barbara. 2011. Multidominance. In *The oxford handbook of linguistic minimalism*, ed. Cedric Boeckx, 119–142. Oxford, UK: Oxford University Press. URL <http://dx.doi.org/10.1093/oxfordhb/9780199549368.013.0006>. Cited page 22
- Collins, Chris, and Erich Groat. 2018. Copies and repetitions lingbuzz/003809. URL <http://ling.auf.net/lingbuzz/003809>. Cited page 22
- Epstein, Samuel David. 1999. Un-principled syntax. In *Working minimalism*, ed. Samuel David Epstein and Norbert Hornstein, 317–347. Cambridge, Mass.: The MIT Press. URL <http://dx.doi.org/10.7551/mitpress/7305.003.0014>. Cited pages 4 and 5
- Frank, Robert, and K. Vijay-Shanker. 2001. Primitive c-command. *Syntax* 4:164–204. URL <http://dx.doi.org/10.1111/1467-9612.00043>. Cited pages 5, 7, and 8

- Goldsmith, John Anton. 1976. Autosegmental phonology. Doctoral Dissertation, MIT, Cambridge, Mass. *Cited pages 1, 14, and 15*
- Gračanin-Yukse, Martina. 2013. Linearizing multidominance structures. In *Challenges to linearization*, ed. Theresa Biberauer and Ian Roberts, 269–294. Berlin, Boston: DE GRUYTER. URL <http://dx.doi.org/10.1515/9781614512431.269>. *Cited page 22*
- Groat, Erich M. 2018. \emptyset : A minimalist diacritic. *Cited pages 31 and 32*
- Harizanov, Boris, and Vera Griбанова. 2018. Whither head movement. *Natural Language & Linguistic Theory* URL <http://dx.doi.org/10.1007/s11049-018-9420-5>. *Cited pages 31 and 36*
- Kayne, Richard S. 1994. *The antisymmetry of syntax*. Cambridge, Mass.: MIT Press. *Cited pages 1, 2, 3, 5, 6, 8, and 13*
- Kayne, Richard S. 2013. Why are there no directionality parameters. In *Theoretical approaches to disharmonic word order*, 219–244. Oxford University Press. URL <http://dx.doi.org/10.1093/acprof:oso/9780199684359.003.0008>. *Cited pages 1, 6, and 13*
- Klima, Edward S. 1964. Negation in english. In *The structure of language*, ed. Jerry Alan Fodor and Jerrold Jacob Katz, 246–323. Englewood Cliffs: Prentice Hall. *Cited page 5*
- Martin, Roger, and Juan Uriagereka. 2014. Chains in minimalism. In *Language faculty and beyond: Minimalism and beyond*, 169–194. Amsterdam: John Benjamins Publishing Company. URL <http://dx.doi.org/10.1075/fab.11.07mar>. *Cited pages 21 and 26*
- May, Robert. 1985. *Logical form*, volume 12. Cambridge, Mass: MIT Press. *Cited page 8*
- Moro, Andrea. 1997. Dynamic antisymmetry: Movement as a symmetry-breaking phenomenon. *Studia Linguistica* 51:50–76. URL <http://dx.doi.org/10.1111/1467-9582.00017>. *Cited page 3*
- Nevins, Andrew. 2007. The representation of third person and its consequences for person-case effects. *Natural Language & Linguistic Theory* 25:273–313. URL <http://dx.doi.org/10.1007/s11049-006-9017-2>. *Cited pages 1, 14, 15, 32, 34, and 38*
- Nevins, Andrew. 2010. *Locality in vowel harmony*. Cambridge, Mass.: The MIT Press. URL <http://dx.doi.org/10.7551/mitpress/9780262140973.001.0001>. *Cited page 40*
- Nunes, Jairo. 2004. *Linearization of chains and sideward movement*. The MIT Press. URL <http://dx.doi.org/10.7551/mitpress/4241.001.0001>. *Cited pages 16 and 18*
- Osborne, Timothy, Michael Putnam, and Thomas M. Gross. 2011. Bare phrase structure, label-less trees, and specifier-less syntax. is minimalism becoming a dependency grammar. *The Linguistic Review* 28. URL <http://dx.doi.org/10.1515/tlir.2011.009>. *Cited page 13*

- Partee, Barbara H., Alice Ter Meulen, and Robert E. Wall. 1993. *Mathematical methods in linguistics*. Dordrecht: Springer Netherlands.
URL <http://dx.doi.org/10.1007/978-94-009-2213-6>. Cited page 1
- Pesetsky, David, and Esther Torrego. 2007. The syntax of valuation and the interpretability of features. In *Linguistik aktuell/linguistics today: Phrasal and clausal architecture*, ed. Simin Karimi, Vida Samiiian, and Wendy K. Wilkins, 262-294. Amsterdam: John Benjamins Publishing Company. URL <http://dx.doi.org/10.1075/la.101.14pes>. Cited pages 17 and 27
- Rizzi, Luigi. 2001. Relativized minimality effects. In *The handbook of contemporary syntactic theory*, ed. Mark Baltin and Chris Collins, 89-110. Oxford, UK: Blackwell Publishers Ltd. URL <http://doi.wiley.com/10.1002/9780470756416>. Cited page 23
- Rizzi, Luigi. 2011. Minimality. In *The oxford handbook of linguistic minimalism*, ed. Cedric Boeckx, 220-238. Oxford, UK: Oxford University Press.
URL <http://dx.doi.org/10.1093/oxfordhb/9780199549368.013.0010>. Cited pages 23 and 24
- Sagey, Elizabeth. 1988. On the ill-formedness of crossing association lines. *Linguistic Inquiry* 19:109-118. URL <https://www.jstor.org/stable/4178577>. Cited page 1
- Samuels, Bridget D., Marc Hauser, and Cedric Boeckx. 2017. Looking for ug in animals. In *The oxford handbook of universal grammar*, ed. Ian Roberts, 527-546. Oxford, UK: Oxford University Press. URL <http://dx.doi.org/10.1093/oxfordhb/9780199573776.013.22>.
Cited page 13
- Starke, Michal. 2001. Move dissolves into merge: A theory of locality. Doctoral Dissertation, University of Geneva, Geneva. Cited pages 1, 14, and 15
- Uriagereka, Juan. 2011. *Spell-out and the minimalist program*. Oxford, UK: Oxford University Press. URL <http://dx.doi.org/10.1093/acprof:oso/9780199593521.001.0001>. Cited pages 5, 6, 7, 8, 9, and 11
- Zwart, Jan-Wouter. 2011. Structure and order: Asymmetric merge. In *The oxford handbook of linguistic minimalism*, ed. Cedric Boeckx, 96-118. Oxford, UK: Oxford University Press. URL <http://dx.doi.org/10.1093/oxfordhb/9780199549368.013.0005>. Cited page 13