Agreement as information transmission over dependencies

Marina Ermolaeva   Gregory M. Kobele

March 15, 2021

Abstract

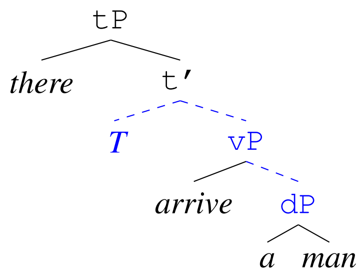
In contemporary minimalism agreement is the result of a syntactic operation. In contrast to Merge, Agree does not build structure, its role being to transmit morphological features from one head to another. We provide an alternative perspective on agreement in a minimalist idiom, one which cuts the ontological pie in a different way. Syntax has as its only operation Merge, and agreement, now divorced from syntax, transfers purely morphological information from head to head along *channels* established via syntactic feature checking. Factoring agreement from structure building seems to allow for more elegant descriptions of even complicated agreement phenomena.

Keywords: agreement, grammatical architecture, minimalism, dependencies

1. Introduction

In contemporary minimalism agreement is the result of a syntactic operation, Agree. While Merge constructs structure, Agree does not. Instead, Agree allows morphological features to be valued between heads. Agree is thought to work by searching through a syntactic tree until it finds an appropriate head to transmit features from. Consider the ultimately successful path from probe (the head of the root) to its goal in the tree below.

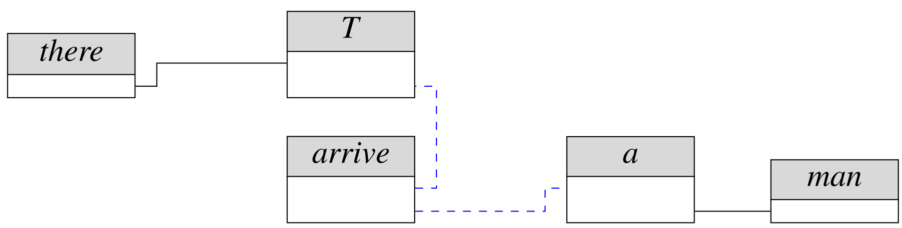
(1)



The search for a goal in the tree above ultimately succeeds along the dashed path in blue. (There may well have been wrong turns and dead ends during the search process.) As the structure in this tree is a reflection of the Merge steps used to construct it, we can view this path from probe to goal as a series of local feature transmissions over Merge steps. If we view Merge steps as being driven by syntactic features (as in (Stabler 1997; Adger 2003; Müller 2010)), we can describe these local feature transmission steps at the level of lexical items: a DP transmits its ϕ-features to whatever merges with it, a vP transmits these features to whatever merges with *it*, and so on.

In order to obtain a more modular theory of grammar, we propose to factor out syntactic structure building (via Merge) from agreement (via Agree).[1] This has been proposed before—Bobaljik (2008) suggests we treat Agree as post-syntactic—here we work out how this might look. In particular, we focus on the idea of the search space of Agree, reformulating it in terms of paths of *dependencies* between lexical items. Such a dependency is not an object in its own right, but rather a shorthand for the more unwieldy: "Merge applying to two phrases whose heads are these lexical items." (We go into more detail on this in §2.) With this abbreviation in mind, the tree above induces the dependency structure below. Here the path used by Agree is the dashed one in blue from *T* to *D*.

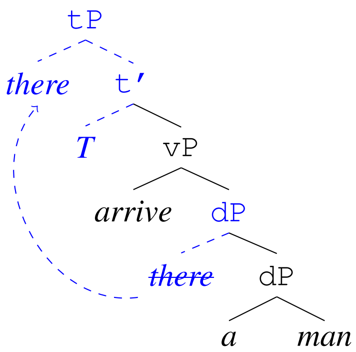
(2)



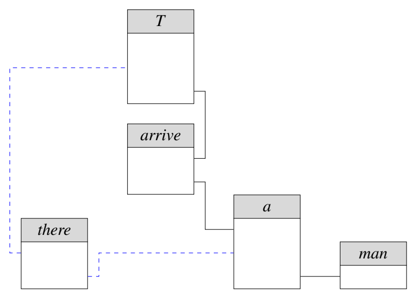
[1]Kaplan (1987) provides a compelling defense of modularity, and of factoring out (at the competence level) logically distinct processes. Especially his discussion of the "procedural" and "interaction" 'seductions' is relevant.

Having separated the process of agreement from the process of creating the structure it is defined over, we note that agreement is fundamentally a matter of choosing a path between the controller of agreement (the goal) and its target (the probe). As the search space for agreement is completely determined by the structure assigned to the sentence, as the analysis changes so too do the possible paths between controller and target. If we adopt an associate-internal analysis of expletive *there*, as shown below, a new path between *D* and *T* becomes available, as shown in the neighboring dependency structure.

(3)



(4)



Our fundamental claim is that this is a useful way of thinking about agreement in minimalism. By focusing on the agreement paths actually chosen rather than the procedure for constructing paths, we can give natural and elegant direct statements about even complicated cases of agreement, as we show in for the case of the 'raising puzzle' in Lubukusu (as described in Diercks (2013)). Perhaps in contrast to what we might expect if Agree used a domain independent search strategy (like the depth-first-like search currently used) to construct agreement paths, in the cases we consider the obvious statement about the simplest agreement paths can be lexicalized — each lexical item contributes in a regular way to the actual agreement paths taken in a given derivation. Making syntax solely about structure building has the additional benefit of making available a simple and novel analytic strategy, which we exemplify both for Lubukusu as well as for expletive *there*. While we consider the primary goal of this paper to be the perspective on Agree (and thus on the nature of syntax) that it offers, we also believe the methodological contribution (in the form of the analytic strategy offered) also to be valuable. While agreement is at the forefront of current syntactic theory in the Chomskian tradition, the literature on minimalist grammars (Stabler 2011) has largely ignored these fundamental questions (but see Ermolaeva (2018) which is a direct predecessor of this paper). A further contribution of this paper is to extend the minimalist grammar framework so that it more directly addresses the interests of today's syntactician.

We begin in §2 by fleshing out the notion of the search space for Agree, and its streamlined depiction in terms of paths of dependencies. As expletive *there* was instrumental in the development of the Agree theory, we use it (in §3) to introduce our dependency-based analysis strategy, demonstrating how not only the high origin and the associate internal, but also the low origin, analyses emerge as simple choice points during analysis construction. We introduce and successively refine here a language for expressing the agreement path contribution of lexical entries. We turn next in §4 to the reanalysis of Lubukusu (as described by Diercks (2013)), demonstrating both the dependency-based analysis strategy, as well as the utility of our approach to agreement. In §6 we revisit the relation of the traditional formulation of Agree and ours, demonstrating how dynamic restrictions on its search strategy can be reformulated in terms of restrictions on lexical path descriptions.

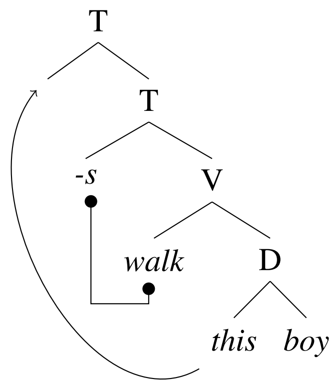
2. Agreement

We describe here in more detail our static reformulation of the usual procedural characterization of Agree. Our working example will be the very simple sentence "this boy walks." There are here two agreement relations present, one between "boy" and "this" (in number), and the second between "boy" and "walks."[2] The subject "this boy" must both move to Spec-TP, as well as agree with T. A (somewhat simplified) traditional analysis is sketched in (5). Here the verb and tense heads form a morphological word, the details of which are orthogonal to the present discussion.[3]

[2] This might sound unusual — isn't it the DP "this boy" which controls agreement on "walks?" We are here intending to focus on "boy" as the originator of the ϕ-features, so as to provide as theory-neutral a characterization of events as possible.

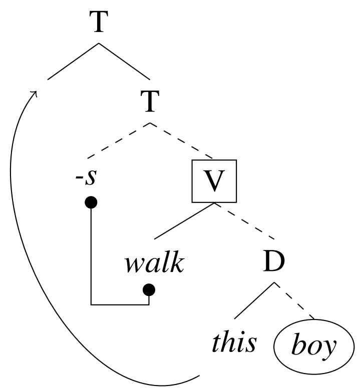
[3] We will implicitly adopt a spanning approach to morphological word-formation (Brody 2000; Williams 2003; Svenonius 2016), which can roughly be thought of as classical head movement with more flexible pronunciation options.

(5)



One analysis of this sentence has it that the Agree step precedes the (internal) Merge step. Agree thus searches through the sister of the tense head until it finds "this boy." This is illustrated in figure (6), where the (dashed) agreement path starts at the probe -*s*, leads to its sister (the boxed V), and ends at the circled noun.

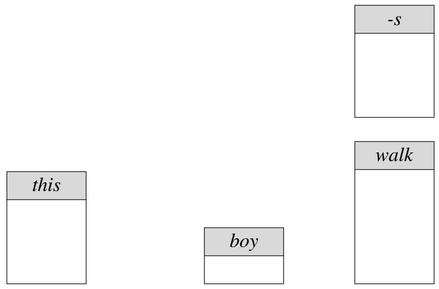
(6)



This path reflects a series of Merge steps taken in the derivation of this expression. We walk through the derivation below, ignoring non-structure-building (i.e. Agree) steps. At each step, we write a dependency structure, which succinctly represents the derivational dependencies — an edge connects two nodes just in case Merge applied to constituents whose heads were those nodes. We will write our dependency graphs in such a manner that items that form the same morphological word are directly above one another, in the same column.

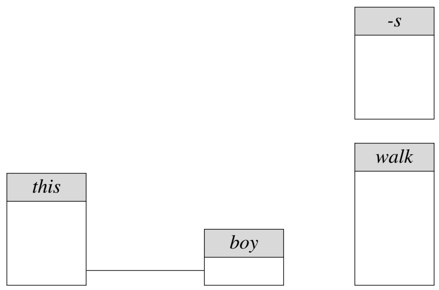
The sentence is derived in four steps, beginning with a numeration of the lexical items to be used.

0. Construct numeration



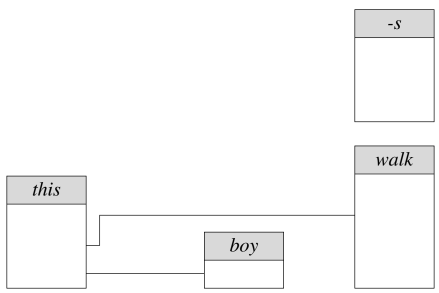
1. Merge *this* and *boy*: [$_D$ *this boy*]

   We draw an edge connecting these two lexical items.



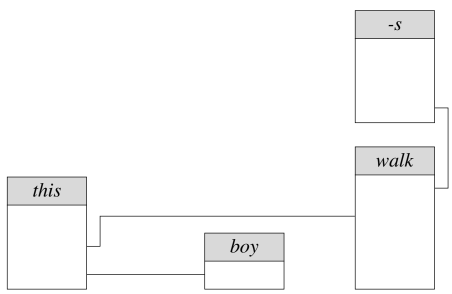
2. Merge the DP with *walk*: [$_V$ *walk* [$_D$ *this boy*]]

   We connect the head of the DP (i.e. *this*) with *walk*.



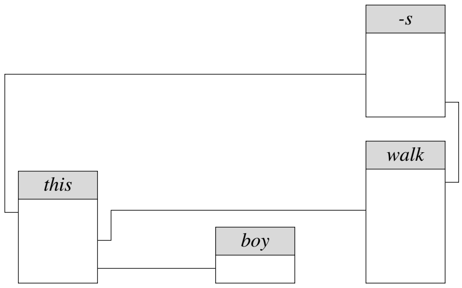
3. Merge the VP with tense: [$_T$ *-s* [$_V$ *walk* [$_D$ *this boy*]]]

   We connect the head of the VP (*walk*) with *-s*.



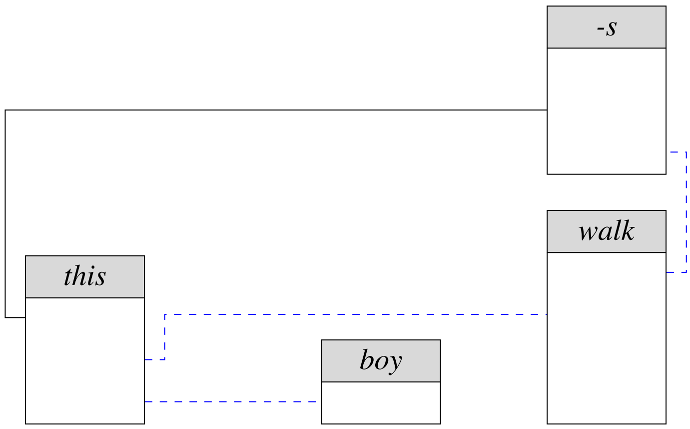
4. (Re)Merge the DP and the TP: [$_T$ [$_D$ *this boy*] [$_T$ *-s* [$_V$ *walk* ~~[$_D$ *this boy*]~~]]]

We draw an edge between the head of the moved DP (namely *this*) and the head of the TP (namely -*s*).



The dependency structure above thus directly represents the Merge operations performed over the course of the derivation; one Merge operation per edge, acting on phrases headed by those LIs.[4] Viewed in terms of this dependency structure, Agree works by constructing a path from the probe to a goal, and then transferring morphological information from the latter to the former. The path chosen by Agree in figure (6), moving first to the sister of -*s* (the VP), then to the DP and from there to *boy*, can be recast in terms of Merge operations, and thus in terms of this dependency structure. The sister of -*s* (the VP), is the first merged argument to -*s*, which in the dependency structure is the edge connecting -*s* and *walk*. This VP is the result of merging *walk* and the DP, which in the dependency structure is represented as the edge connecting *walk* and the head of the DP, *this*. The DP is the result of applying the Merge operation to *this* and *boy*, which is represented as the edge connecting *this* and *boy*. This path is depicted by the dashed edges in the dependency structure, shown below.

(7)

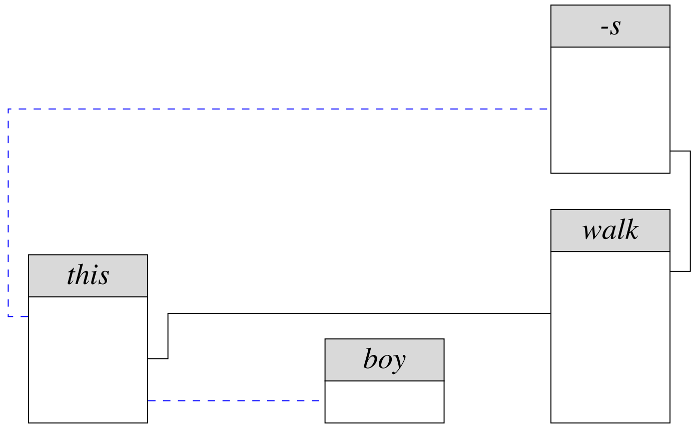


This path connects the probe/target -*s* with the goal/controller *boy*. We see that there is also

[4] Stabler (1999) calls these 'proof nets,' emphasizing their relation to categorial grammars.

another path from the probe -*s* to the goal *boy* — this path begins and -*s*, and then moves directly to *this*, and then from there on to *boy*. From a more derivational perspective, it corresponds to first moving the DP to Spec-TP, and *then* having T agree with the DP in its specifier (an instance of upward agree).

(8)



The other agreement relation which is established in the course of the derivation is that between the demonstrative and its noun. Here, there is but one possibility, which is to apply (right) after *this* and *boy* undergo Merge. This corresponds to a dependency between these two words. This dependency is already part of both routes from -*s* to *boy*, making it clear that the agreement between tense and noun is *factored* through the determiner.

2.1 *Syntactic Analyses*

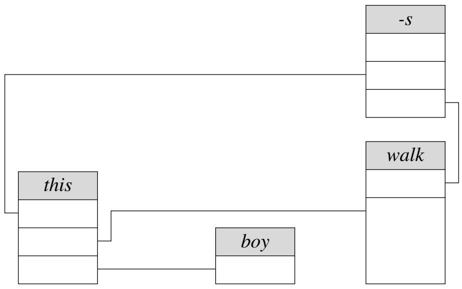
A syntactic analysis provides a recipe for assigning structures (analyses) to expressions. Given that we want to assign the structure above to the sentence "this boy walks," with the derivation as represented in the associated dependency structure, how are we to encode this into a precise analysis? A simple and flexible approach was proposed in Stabler (1997), based on the idea that all Merge steps should be triggered by syntactic features. We assume that Merge is triggered by *matching* features of *opposite* polarity, and when it applies it *checks* (i.e. deletes) both triggering features.[5] We will assume that feature bundles are *ordered lists* of features, which must be

[5]We can exert very fine control of the derivation by manipulating these basic assumptions. Stabler (2011) summarizes a number of variations on this basic theme, including *persistent* features, which can survive checking, and which we will make use of in our analysis of Lubukusu.

checked from left to right.[6] Then an analysis is given by providing a set of lexical items with their syntactic feature bundles. Starting from a dependency structure, which is here nothing more than a record of Merge steps, we can obtain a set of lexical items which can participate in such a derivation in four steps.

Step one: Decide the order in which the Merge steps occured.

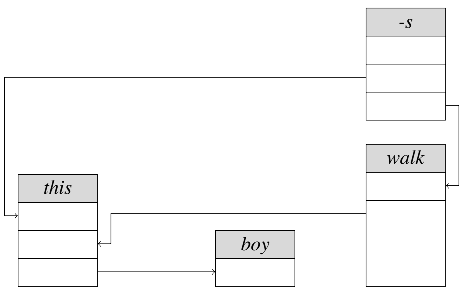
For example, did *this* Merge with *walk* before or after Merging with *-s*? We have already implicitly represented this in our dependency structures with the vertical arrangement of the edges attached to a single node, with higher edges corresponding to later Merge steps. We make this concrete by explicitly segmenting the nodes.



We have added an extra position on *-s*, the head of the entire structure, because it can be subject to further Merge steps, and must therefore have a further syntactic feature.

Step two: Determine which of the two LIs each edge connects projects over the other after that Merge step.

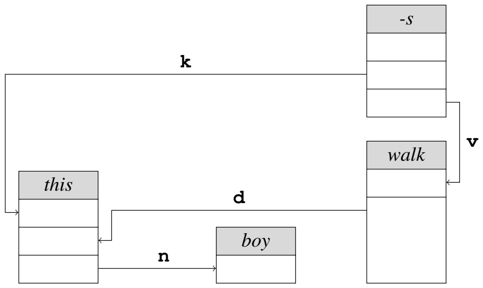
We can represent this by directing the edges, so that they point from the projector (the head) to the projectee (its dependent).



[6]This is for convenience only. If we have some way of deciding which feature is checked *first*, then we can do away with order among the rest (by breaking apart lexical items with too many features into a sequence of smaller functional heads). If the order of first merged expressions can be derived without reference to features, perhaps given by a *functional sequence* (Starke 2001), then we can eliminate order from our feature bundles altogether.

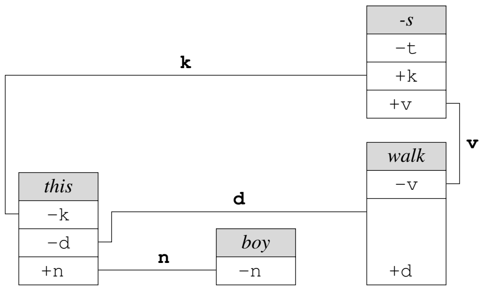
Step three: Determine which syntactic feature type triggered each Merge step.

We can record this by writing on each dependency a feature name, which we decide is responsible for that step of the derivation.



Here we have decided that the syntactic feature type allowing Merge to apply to *this* and *boy* is an **n** feature, the one allowing Merge to apply to *this* and *walk* is a **d** feature, the one permitting it to apply to *walk* and *-s* is a **v** feature, and the one enabling it to apply to *this* and *-s* a **k** feature (reminiscent of 'case').

Step four: For each edge, put the positive version of its feature on its source LI, and the negative version of its feature on its target LI.



The lexical items we need to reconstruct this derivation along with their feature bundles can be read off of this structure; the feature bundle of each lexical item has its first feature on the bottom, and the last one on top. It is often more convenient to write lexical items on a single line; the lexical items we arrived at are shown below.

(9) *this* :: +n.−d.−k *walk* :: +d.−v

boy :: −n *-s* :: +v.+k.−t

Using current terminology, all syntactic features are *uninterpretable*, and must be checked in order to avoid a crash at the interfaces. This simplifies the identification of a syntactically well-formed expression — we simply need verify that all features have been checked, with the exception of a single −x feature at the head. While this may appear to be a radical departure from orthodoxy, it is due primarily to the fact that we are distinguishing syntactic features from morphological ones (in line with our stance on modularity in this paper), whereas current work in syntax adopts a strong reductive stance on features, assuming that what we are calling syntactic features can (at least in part) be reduced to morphological ones. We neither mean nor desire to take a position on this interesting proposal in this paper.

2.2 Dependencies and channels

Now that we have described how to view the search space of Agree in a static way, we would like to take it out of syntax altogether. Instead of being a core operation of the syntax, agreement merely supervenes on the derivation, with the syntactic dependencies between lexical items acting as the channels through which morphological information can be ferried.

Our example lexicon is repeated in (10), augmented with morphological feature bundles.

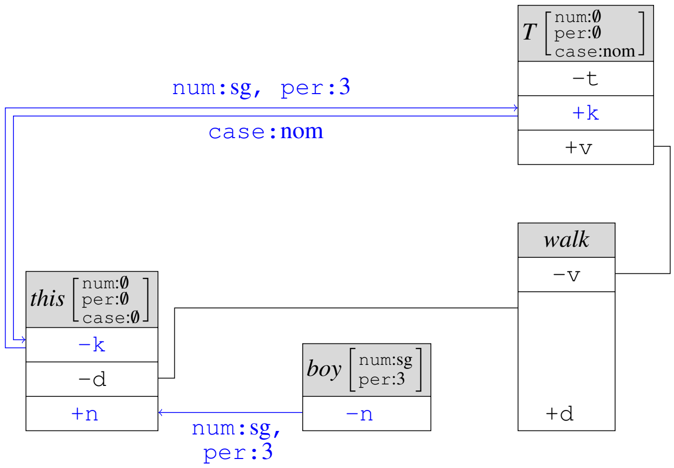
(10)

this [num:∅, per:3, case:∅]	::	+n.−d.−k
boy [num:sg, per:3]	::	−n
walk	::	+d.−v
-s [num:∅, per:∅, case:nom]	::	+v.+k.−t

Two of these lexical items have unvalued morphological features: *-s* needs both number (num) and person (per) values, and *this* lacks both case and number values. (The lexical item *this* will be realized as *these* if it is plural.)

We assume that all syntactic dependencies are eligible for use as channels. The only point of contact between *this* and *boy* is the dependency that formed by Merge checking the +n/−n feature pair. Subject-verb agreement can be resolved (as in figure (8)) over the +k/−k dependency between *this* and *-s*. These two links are sufficient to pass number and person information from *boy* to *this* and subsequently to *-s*. However, the same links can be used to transmit case information from *-s* to the subject. These agreement channels, along with the morphological messages passed through them, are shown in (11).

(11)



While it is arguably more in line with minimalist desiderata, we find the most compelling argument for the channel approach to agreement the fact that it offers a novel and visually appealing method of developing analyses. In what follows, we explore and refine this idea with two case studies. Section 3 examines the commonalities between a number of existing accounts of *there*-insertion in English, while section 4 compares a traditional Minimalist analysis of complementizer agreement in a Bantu language with a straightforwardly obtained alternative.

3. *There*-insertion in English

The English subject-verb agreement system is straightforward to begin to describe: a tensed verb agrees in person and number with the DP in its specifier. This rule of thumb appears to break down in constructions, such as the expletive *there* construction as in (12a), where the element in the specifier is not the DP agreed with. One natural strategy is to postulate that the agreed with DP is, despite appearances, in the specifier of T (Chomsky 1986). More recent work has instead taken this construction to motivate a long-distance Agree operation (Chomsky 2000). In this section, we will show how three prominent analyses of expletive *there* sentences can be arrived at systematically — by manipulating a basic, word-based, dependency representation for the sentence. We will introduce two simple operations for manipulating dependency structures, dependency introduction and word decomposition. In particular, the operation of word decomposition creates what are in effect complex heads, or spans, in the reconstructed minimalist analysis. From a dependency structure, possible agreement transmission corridors can be read off of the arcs connecting the two agreeing words to one another.

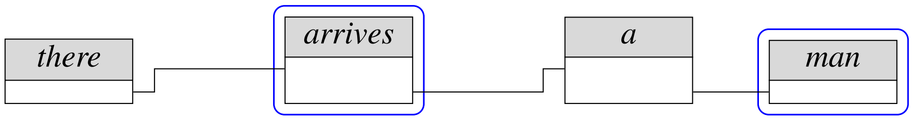
Consider the following simple sentences:

(12) a. There arrives a man.

b. There arrive three men.

We mainly focus on just two relevant aspects of expletive constructions with *there*: syntactic dependencies between the expletive and other lexical items and agreement between the expletive's associate and the verb. The graph in (13) shows the bare-minimum Merge dependencies within the construction in question, color-coding the words that form an agreement relation.

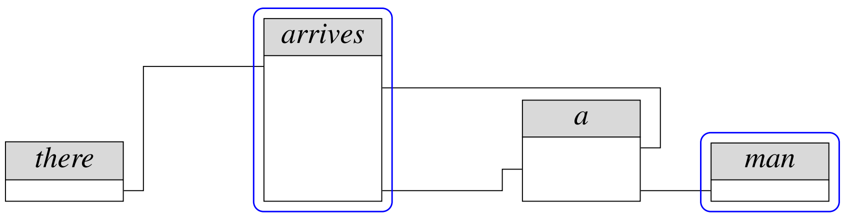
(13)



This graph depicts the following, word-based, derivation: *a* and *man* are merged together, then this unit is merged with *arrives*, which next merges with *there*. The lexemes *arrives* and *man* are in blue, as we assume that the ϕ-features which are realized by the word *arrives* are inherited from *man*.

The minimal, word-based, dependencies given in (13) can be refined in various ways, as we shall see. One mode of refinement is to add an additional dependency between two elements, reflecting a theoretical commitment that these two elements be syntactically related. One such is the addition of an additional dependency between the determiner (i.e. the DP) and the inflected verb (14), in line with the common assumption that DPs enter into two syntactic dependencies (corresponding to their θ and case positions, in GB terms).

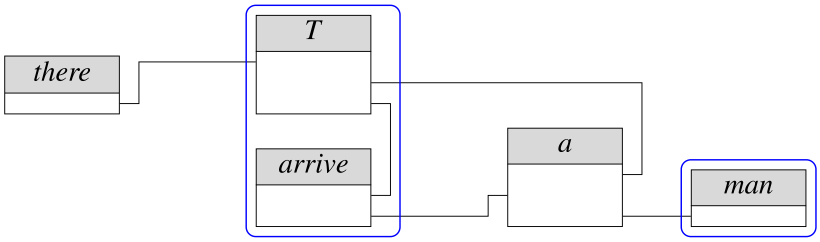
(14)



This graph now represents a derivation in which *a man* merges with *arrives*, and then is immediately remerged with it.

We may in addition manipulate dependency structures by decomposing one node into two, which requires us to distribute the dependencies entering and leaving this node among the two new ones (as well as making these nodes dependent on one another). This corresponds to recognizing a word as the realization of multiple heads. As we believe that the inflected verb *arrives* is the realization of (minimally) a T head and a V head, we decompose this word into two, with just one dependency to the Det linked to the V head, and the other dependencies linked to the T head (15).

(15)



This graph represents a derivation where an abstract lexeme *arrive* merges with *a man*, and then this complex is merged with an abstract tense head, whereupon *a man* is remerged with this tense head.[7] Both of the lexemes the original *arrives* has been decomposed into (*T* and *arrive*) are marked in blue, indicating a temporary agnosticism about which of these two heads should inherit the ϕ-features of *man*. Based on familiar facts not visible in this single structure (namely, that *arrive* may appear uninflected elsewhere), we assume that it is in fact *T* that is the recipient of *man*'s ϕ-features.

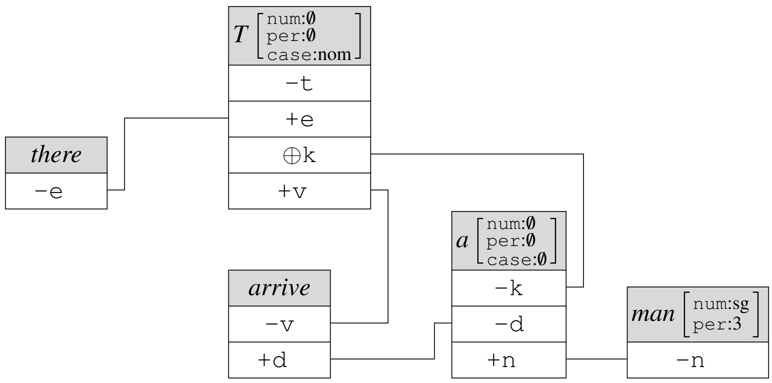
We have arrived at dependencies underlying the so-called *high-origin* account (Chomsky 2000), according to which expletives freely merge into the specifier of TP. We associate therefore each dependency with a feature type to be checked, decide which end of the dependency played the active role in that checking relationship (i.e. which was the head), and what order the features appear in in the lexical feature bundles.

Looking at the example, we see that, for example, the lexical item *T* has three dependencies, which we uniformly interpret as active dependencies. As it is the head of the entire structure,

[7]The family of tense heads we have hereby created shadow the already existing tense heads, which do not further combine with expletive *there*. This introduces an unpleasant redundancy into our lexicon, which can be eliminated by decomposing tense into a head *T*, which just introduces tense, and a head *AgrS*, which hosts a subject position, and optionally an expletive. The previous redundancy between expletive selecting and non-expletive selecting tense heads is reduced to a lexical choice between two versions of *AgrS* with feature bundles `+t.+k.-agrS` and `+t.⊕k.+e.-agrS`.

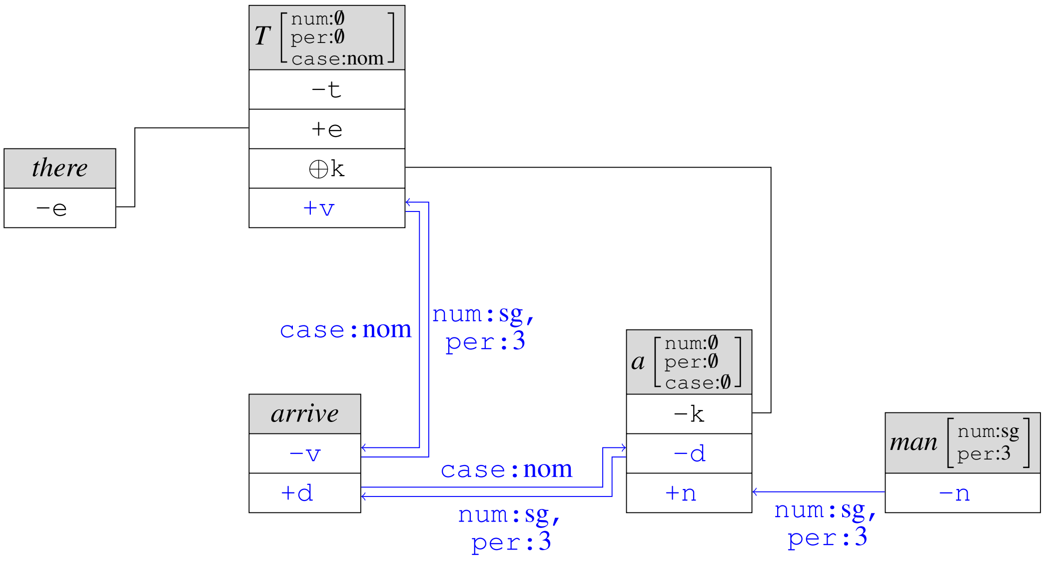
its categorial feature, -t remains unchecked. The dependency with *there* will be associated with an expletive selecting feature +e, the dependency with *a* with a case feature +k, and the span-forming dependency with the verb *arrive* with a verb selecting feature +v. Given the word order of the sentence for which this is the dependency structure, we see that the DP *a man* could not have moved overtly to the left of the tense head. Accordingly, we introduce a variant of a +k feature, written ⊕k, which forms a Move dependency without actually changing the linear order of morphological feature bundles (i.e. covert movement). The resulting configuration for the expletive construction is as shown in (16).

(16)



The high-origin configuration can be considered the simplest (in terms of the number of lexical items and dependencies) approach to introducing an expletive into the structure. Still, it provides two ways to pass information between the noun phrase and *T*, either using external Merge dependencies or bypassing the lexical verb completely and establishing a direct Move channel. We see that, as there are two arcs leading out from the determiner, there are in fact two basic paths between the tense head and the noun: 1. *man* — *a* — *arrive* — *T*, and 2. *man* — *a* — *T*. The first path (shown in figure (17)) follows the θ dependency between determiner and verb, and embodies a version of long-distance agreement (factored through local steps).

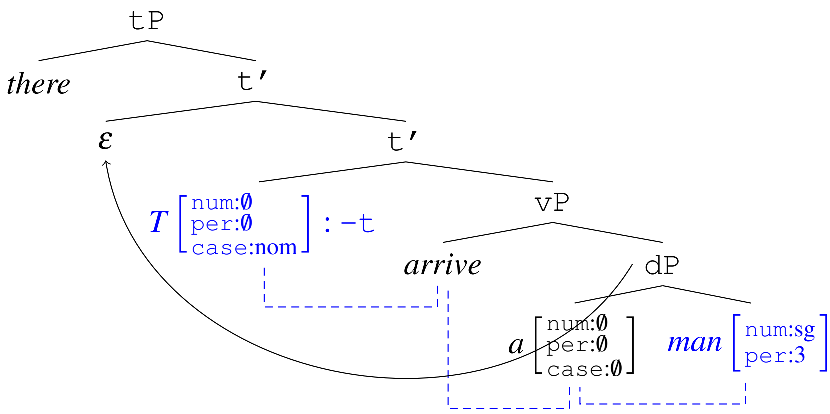
(17)



High origin: agreement via Merge

This structure can be presented in more standard terms as the tree in figure (18). The movement arrow from the dP to the lower specifier of t indicates a covert movement, which we have represented by writing an ε in the moved-to position. The dotted lines indicate the path over which morphological information is exchanged between *T* and *man*. Head movement (or rather spanning) is not explicitly represented in this tree, but *T* and *arrive* are intended to form a span.

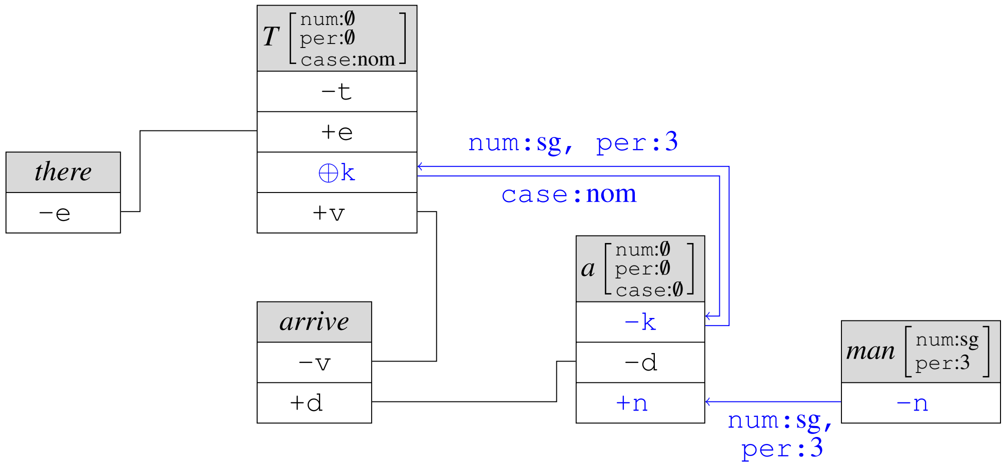
(18)



High origin: agreement via Merge; derived tree

The second path (in figure (19)) connects the determiner a directly with the tense head, and has the property that its length does not increase as the distance between the θ position and tense grows, as in, for example, a raising construction (e.g. *There seem*(s) to have arrived a man*). This approach essentially reanalyzes the expletive *there* construction so as to render it compatible with the naïve agreement rule formulated at the beginning of this section: T agrees with the DP in its specifier. Here of course, being in T's specifier is recast as being directly connected with T via its ⊕k feature.

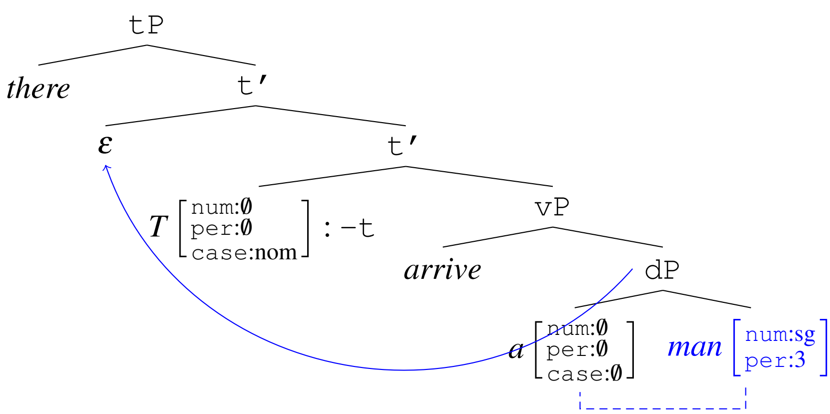
(19)



High origin: agreement via Move

A more standard tree-based representation of this structure is given in figure (20). The movement arrow from the dP to the lower specifier of t now ferries morphological information between *T* and *a*, with the dotted line between *a* and *man* allowing this information to get to *man* and back.

(20)



High origin: agreement via Move; derived tree

Importantly, neither of the above solutions establishes a relation between *there* and its associate, nor between the expletive and the lexical verb. The expletive is present in the structure but excluded from any agreement-related processes.

3.1 Controlling channels

Before continuing with the other extant analyses of expletive *there* sentences, we pause to address the mechanical question of how to state either of the two channel perspectives on the high origin account above. Until now, we have simply observed that the three lexical items in need of morphological feature values are indeed connected to one another in one of two ways, and the pictures we drew suggest that we intend information to flow along just one of these paths. The attentive reader will note that we have not yet said anything about how to enforce this - it would be consistent with what we have said thus far to imagine information taking all possible paths at once!

We intend the path over which information should flow to be a part of an analysis. As our analyses are currently given by presenting a set of lexical items, we will lexicalize our desired channels. Looking back at example (19), we see that we want morphological information to flow between T and a over the **k** feature dependency. This information flow needs to be bidirectional, as ϕ-features are transmitted *to*, and case transmitted *from*, T. As the dependency in question comes about because the lexical items it connects have matching **k** features, we will mark on these features that they permit bidirectional information flow. We do this here by adding an outgoing arrow $^{\rightarrow}$ and an incoming arrow $_{\leftarrow}$ to the features in question: $\oplus\mathtt{k}^{\rightarrow}_{\leftarrow}$ and $-\mathtt{k}^{\rightarrow}_{\leftarrow}$.

The arrow markings on a feature serve as sluices which permit information to flow (or not) along a channel. A feature marked with an outgoing arrow $^{\rightarrow}$ indicates that information is able to flow out along that dependency. A feature marked with an incoming arrow $_{\leftarrow}$ indicates that information is able to flow in along that dependency. An unmarked feature acts as a dam, completely blocking information from flowing past it (in either direction).

Figure (21) gives the lexical items needed for the move channel alternative of the high-origin analysis presented above.

(21)

there	::	$\mathtt{-e}$
man $\begin{bmatrix}\mathtt{num}:\text{sg}\\ \mathtt{per}:3\end{bmatrix}$	::	$\mathtt{-n}^{\rightarrow}_{\leftarrow}$
arrive	::	$\mathtt{+d.-v}$
-*T* $\begin{bmatrix}\mathtt{num}:\emptyset\\ \mathtt{per}:\emptyset\\ \mathtt{case}:\text{nom}\end{bmatrix}$	::	$\mathtt{+v.}\oplus\mathtt{k}^{\rightarrow}_{\leftarrow}\mathtt{.+e.-t}$
a $\begin{bmatrix}\mathtt{num}:\emptyset\\ \mathtt{per}:\emptyset\\ \mathtt{case}:\emptyset\end{bmatrix}$	::	$\mathtt{+n}^{\rightarrow}_{\leftarrow}\mathtt{.-d.-k}^{\rightarrow}_{\leftarrow}$

We can impose yet more control over the flow of information by allowing lexical items to stipulate the information that they wish to send and receive over their various channels. It is often assumed that T (in English) gets its ϕ-features from whatever it assigns nominative case to, not just that T opens itself up indiscriminately to any and all information that might flow along the `-k` channel. This can be written in the following way (borrowing from notation popular in the LFG community - the $\downarrow$ can be read as 'my'):[8]

$\downarrow\mathtt{case} \Longrightarrow \oplus\mathtt{k}$	send my case value to the checker of my $\oplus$`k` feature
$\downarrow\mathtt{per} \Longleftarrow \oplus\mathtt{k}(\mathtt{per})$	get my person value from the checker of my $\oplus$`k` feature
$\downarrow\mathtt{num} \Longleftarrow \oplus\mathtt{k}(\mathtt{num})$	get my number value from the checker of my $\oplus$`k` feature

These minimalist path equations should then replace the channel annotations on the individual features of a lexical item (in this particular case, the *T* lexical item). When this level of control is unnecessary, it is simpler to annotate features with arrows, and we will move between these notations throughout the present paper.

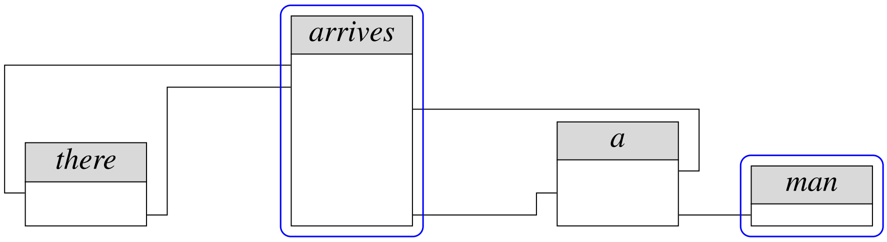
3.2 *The low-origin account*

One prominent alternative is the *low-origin* account (Deal 2009; Alexiadou and Schäfer 2010). Instead of originating in Spec-TP, *there* appears earlier in the derivation, coming into local contact with the verb and subsequently moving to its surface position. We arrive at this analysis by

[8]Just as LFG path equations describe paths through f-structures (which are themselves labeled graphs), these 'directed equations' describe (very local) paths through dependency structures. Whereas LFG uses the $\uparrow$ to speak about a parent of the current node, we here use feature names to speak about the neighbor of the current node connected to it at that feature. This notation is unambiguous so long as no lexical item has the same feature twice, which holds in this paper.

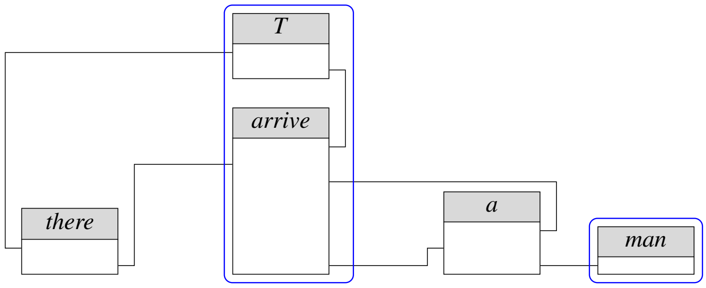
operating on our basic dependency structure in (14) — the naive dependency structure in (13) with an additional dependency between tensed verb and DP. We first add another dependency between *there* and *arrives*, as shown in (22), reflecting our analytic decision to have *there* enter into a prior relationship with the verb.

(22)



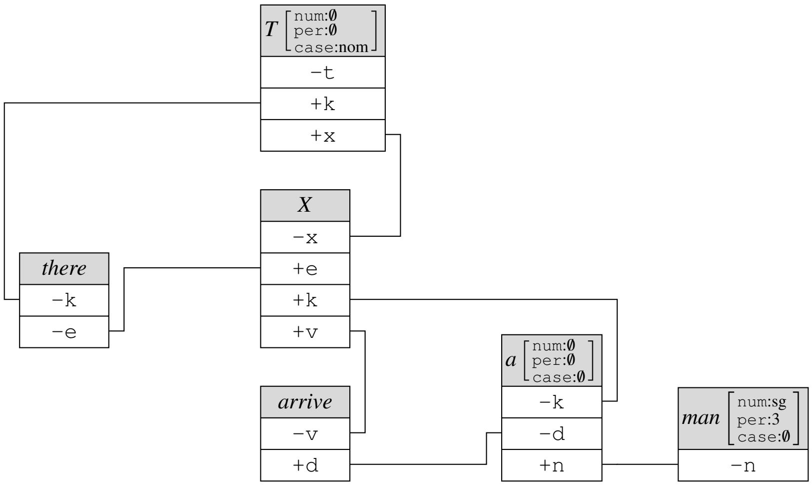
We then again decompose *arrives* into a head expressing tense, and one expressing the lexical verb, as shown in (23). We decompose *between* the two dependencies of *there*. This creates a complex span of the form T-V expressing the inflected verb, but simplifies the lexicon (instead of having an atomic form *arrives* we decompose it into the familiar present tense head and a tenseless verb).

(23)



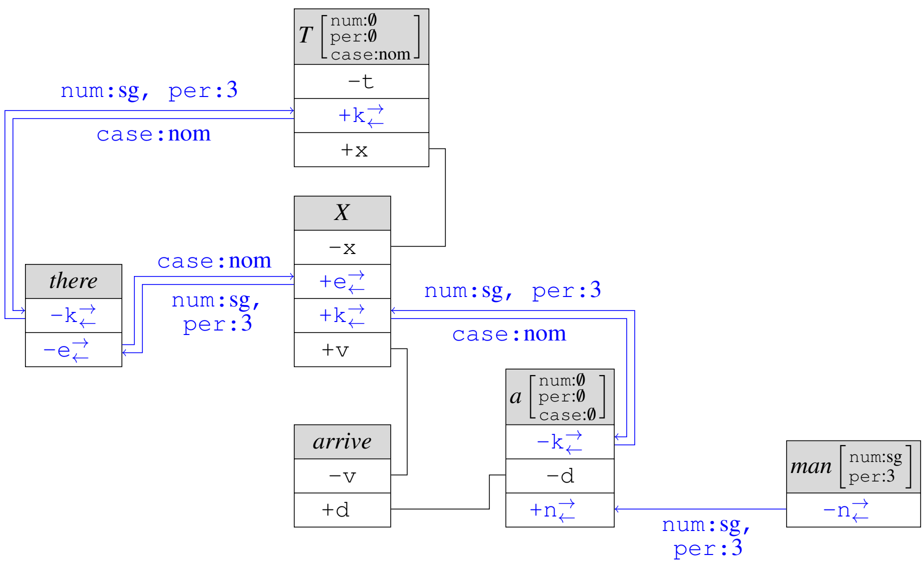
In order to maintain that there is a formal similarity between the usual verb *arrive* and this expletive introducing one, we decompose yet again, viewing the expletive introducing *arrive* as a combination of an expletive introducing head and the usual verb. We reconstruct features on the lexical heads as described previously, arriving at the structure in (24).

(24)



There are again multiple ways of transmitting information about the noun to T: 1. *man* — *a* — *X* — T, and 2. *man* — *a* — *X* — *there* — T.[9] Although the path running from noun to determiner to X and then directly to T is the shortest in this particular example, it grows with the number of clausal embeddings between matrix T and the θ position of the pivot DP. Indeed, any path which connects X directly to T transmits information along the merge dependency between X and T, and thus will of necessity grow in length proportional to the distance between X and T. Only paths which take the 'detour' from X to *there* and then back to T have a constant length regardless of the depth of embedding. The shortest path of this kind goes directly from *a* to X, bypassing the lexical verb. This is shown in (25).

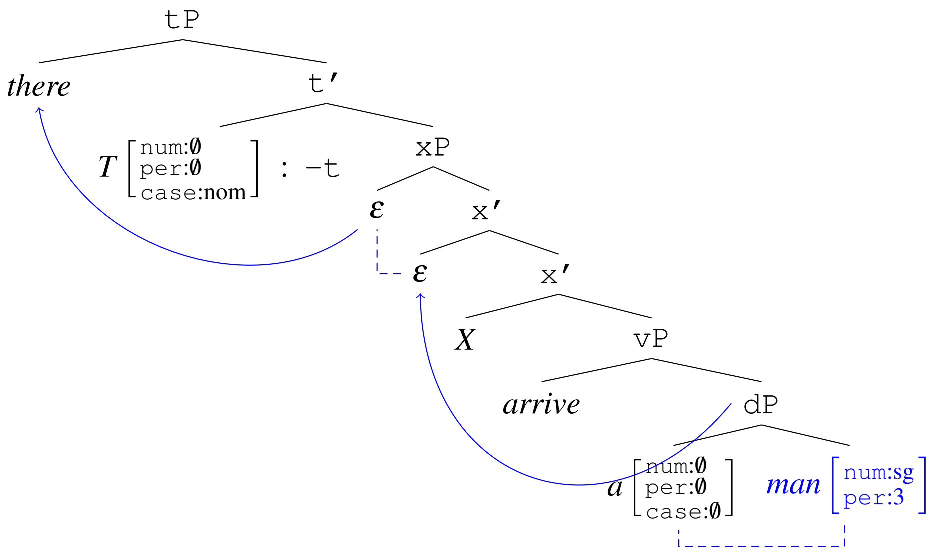
(25)



[9]For each of these paths there is another where the determiner transmits its information to the verb via the merge dependency, and then the verb passes this information on to the expletive introducing head (X) via their merge dependency.

The tree derived by the derivation described by this dependency structure is given in (26), where the agreement relations are marked in blue, with a solid line indicating agreement taking place via a movement dependency, and a dotted line via a merge dependency. We assume for simplicity here that the span *arrive*-*X*-*T* is pronounced at *T*. (Otherwise, we would need to make the DP-movement to *X* covert, or decompose additional abstract heads above *X*.)

(26)



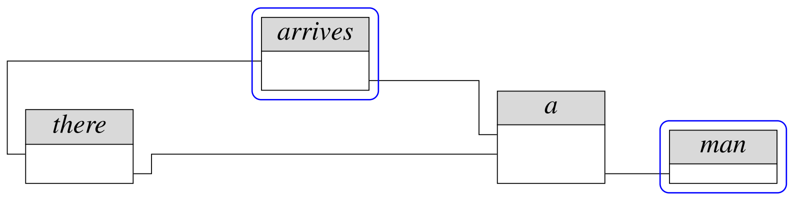
In this analysis we see that information enters the head *X* on one channel (+k) and exists along another channel (+e). This 'horizontal' transfer of information can be accommodated in one of two ways. 1. We populate the morphological feature bundle of *X* with unvalued case, person, and number features. 2. We make use of explicit path equations to pass information from one channel to another. Example path equations for the lexical item *X* are shown below.

$+\mathtt{e}(\mathtt{case}) \Longrightarrow +\mathtt{k}$	send the case value received from +e to the checker of my +k feature
$+\mathtt{k}(\mathtt{per}) \Longrightarrow +\mathtt{e}$	send the person value received from +k to the checker of my +e feature
$+\mathtt{k}(\mathtt{num}) \Longrightarrow +\mathtt{e}$	send the number value received from +k to the checker of my +e feature

3.3 *The associate-internal account*

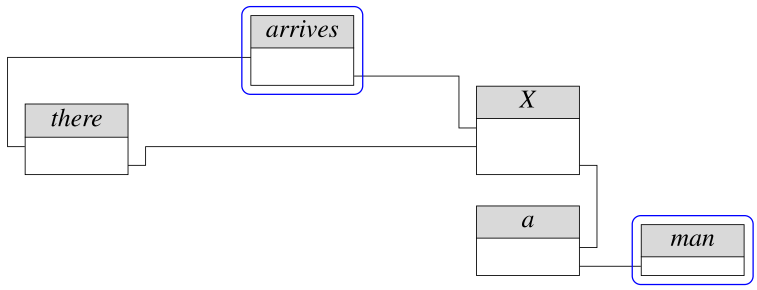
A third analytic alternative can be described as the *associate-internal origin* approach (Basilico 1997; Sabel 2000). Here, *there* and its associate form a constituent together, from which the expletive is subsequently extracted. We again are able to arrive at this analysis via a simple manipulation of dependencies, beginning this time with the basic structure in (13). Again we introduce another dependency from *there*, but this time connecting it to the determiner (27).

(27)



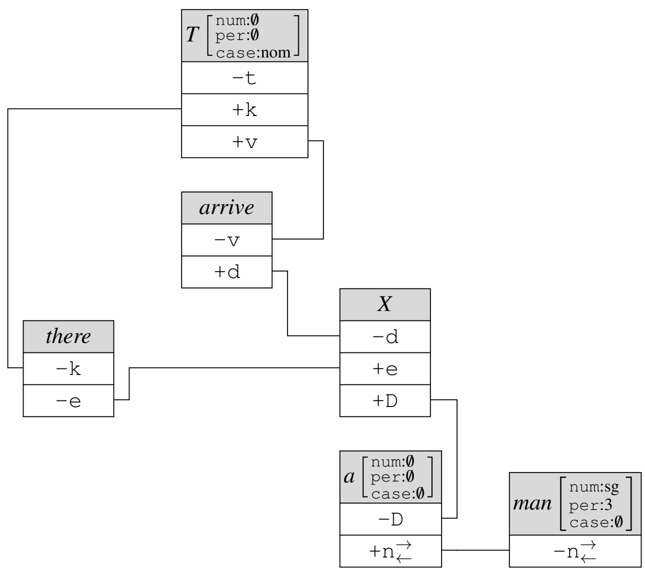
This time the determiner is more complex than usual, and so we decompose it (after the noun dependency) so as to simplify the morass of dependencies it enters into, as shown in (28).

(28)



Another decomposition of the tensed verb into T and V heads leads to our final dependency structure, the syntactic features underlying which can be reconstructed as in (29).

(29)



A main problem faced by any associate-internal approach to *there*-insertion might be called the *case-transmission problem*, after Chomsky (1986). Here the case requirements of the DP must be satisfied vicariously by *there*. In the present context, this may be rephrased as: how does the `-k` feature of the associate DP get transferred to the expletive? Our analysis here does this by divorcing the `-k` feature from the lexical entry of the determiner. What we have been viewing as determiners are then decomposed throughout the lexicon into big-D and little-d pairs, with the lexical head big-D (features: `+n -D`) selecting the nominal complement, and the functional head little-d (features: `+D -d -k`) introducing the case requirement. Our associate internal analysis in effect postulates that there is an expletive selecting little-d head, which we are here calling *X*.

There are again two basic routes along which information can be transferred from N to T: 1. *man* — *a* — *X* — *there* — *T*, and 2. *man* — *a* — *X* — *arrive* — *T*. Although both routes are, in this simple example, of equal length, only the length of the route via *there* remains constant irrespective of the syntactic structure intervening between T and X.

Using explicit path equations, we can specify how lexical items permit features to be transmitted along the channels they are part of, without requiring that these same lexical items themselves have these morphological features in their morphological feature bundles. Figure (30) gives the minimal lexical items needed for the *there*-channel alternative of the associate-internal analysis presented above. Because *T* does not exhibit any kind of morphological sensitivity to `case`, there is no need for to possess a morphological `case` feature. In addition, at least in the context of the present analysis, there is no need for *a* to have any morphological features at all.

(30)

$$
\begin{array}{rcll}
\textit{there} & :: & \texttt{-e.-k} & \left\{\begin{array}{l}\texttt{e(num)}\Longrightarrow\texttt{-k}\\ \texttt{e(per)}\Longrightarrow\texttt{-k}\\ \texttt{-k(case)}\Longrightarrow\texttt{e}\end{array}\right.\\
\textit{man}\begin{bmatrix}\text{num:sg}\\ \text{per:3}\end{bmatrix} & :: & \texttt{-n} & \left\{\begin{array}{l}\downarrow\texttt{num}\Longrightarrow\texttt{n}\\ \downarrow\texttt{per}\Longrightarrow\texttt{-n}\end{array}\right.\\
\textit{arrive} & :: & \texttt{+d.-v} & \\
T\begin{bmatrix}\text{num:}\emptyset\\ \text{per:}\emptyset\end{bmatrix} & :: & \texttt{+v.+k.-t} & \left\{\begin{array}{l}\downarrow\texttt{num}\Longleftarrow\texttt{+k}\\ \downarrow\texttt{per}\Longleftarrow\texttt{+k}\\ \text{nom}\Longrightarrow\texttt{+k}\end{array}\right.\\
a & :: & \texttt{+n.-D} & \left\{\begin{array}{l}\texttt{+n(num)}\Longrightarrow\texttt{-D}\\ \texttt{+n(per)}\Longrightarrow\texttt{-D}\end{array}\right.\\
X & :: & \texttt{+D.+e.-d} & \left\{\begin{array}{l}\texttt{+D(num)}\Longrightarrow\texttt{+e}\\ \texttt{+D(per)}\Longrightarrow\texttt{+e}\\ \texttt{+e(case)}\Longrightarrow\texttt{+D}\end{array}\right.
\end{array}
$$

While it is salient to focus on which features are being sent along which channels, it is also of importance to specify where information is *not* being transferred. In this regard, the lexical entry *X* is the most interesting, as it transmits information between `+D` and `+e`, but *not* via `-d`. This lexical item is here blocking information from traveling over the Merge dependency which will check its `-d` feature (i.e. via the verb).

In figure (30) the path equations for *there*, *a*, and *X* merely serve to pass information along. It may seem unparsimonious to make these rules sensitive to the content of the information they provide a conduit for. A simpler description would simply state that, for example, *there* just connects its `-e` channel to its `-k` channel, like so: `-e` $\Longleftrightarrow$ `-k`. This allows *man* and *T*, as the only active users of the channels in this sentence, to specify which information they send, and which information they wish to receive.

3.4 Summary

The three different analyses arrived at here exemplify three prominent types of analyses in the minimalist literature. While the analyses may appear profoundly different, representing them as dependency graphs shows how the differences boil down to two analytic choices: whether there is an additional dependency between 1. *there* and either the verb or the determiner, and 2. the tensed verb and the determiner.The high-origin analysis presented here is a hybrid of Chomsky (1986) and Chomsky (2000) - it involves (covert/feature/LF) movement of the DP to *T* (as in Chomsky (1986)), thereby establishing a direct transmission corridor between the DP and *T*. Chomsky (2000) postulates that agreement proceeds not over this corridor, but via a search downwards in the structure, beginning at *T* and ending at the DP (or, here in our example, *man*). This corresponds to the *merge-path* connecting the DP and *T*, as shown in example (18).

We end this section with an example that we hope emphasizes the separation between morphology and syntax that the channel-based approach to agreement that we are offering allows. As Deal (2009) points out, *there*-insertion in English is allowed only in the context of non-inchoative unaccusative verbs (31). Other verbs — transitives, unergatives (32a), and inchoative unaccusatives (32b), (32c) — are incompatible with this construction.

(31) a. There appeared a shadowy figure in the doorway.

b. There arrived a train in the station.

(32) a. *There laughed a man in the hallway.

b. *There melted a block of ice in the front yard.

c. *There slowed a train on the eastbound track.

Both low origin and associate-internal origin analyses can enforce this in a natural way, avoiding overgeneration. In both accounts, the locus of the existential construction resides in the head we have called *X* (though this head has quite different properties across the two analyses). Accordingly, a restriction on the type of verb which can appear in such constructions is most naturally imposed at the point where *X* and the verb come in to contact. This can be achieved in the low origin account by refining the `+v`/`−v` features so as to discriminate between different verb classes, and allowing *X* to only select non-inchoative unaccusatives. In the associate internal account, we refine instead the `+d`/`−d` features, giving the existential-introducing *X* a special kind of `−d` feature, which only non-inchoative unaccusatives can select.

The high origin account, on the other hand, places the locus of the existential construction in the matrix *T*. This can be arbitrarily distant from the item (the verb) the type of which must be constrained (in for example raising constructions). This is of course exactly the kind of configuration we might use a mechanism of long-distance agreement for! Still, this kind of long distance 'syntactic checking' does not involve morphological features (in any obvious way), and thus should be excluded in the present modular (i.e. exclusively morphological) approach to agreement.

4. Complementizer agreement in Lubukusu

Lubukusu (Bantu, Kenya) displays an interesting instance of complementizer agreement, extensively documented in (Diercks 2010, 2013). Descriptively speaking, the complementizer of an embedded clause agrees upwards in person and noun class with the subject of the matrix clause.

The paradigm of the agreeing complementizer AGR-*li* is shown in (33), with a point of comparison provided by the generic non-agreeing complementizer *mbo*.

(33)

	Singular	Plural
1st person	n-di	khu-li
2nd person	o-li	mu-li
3rd person	a-li	ba-li
Noun class *N*	*N*-li	*N*-li

Complementizers in Lubukusu (Diercks 2013, p.363)

Lubukusu complementizers only agree with the subject of the most local superordinate clause; in particular, indirect objects of ditransitive verbs (34) and higher matrix subjects (35)) are not acceptable goals.

(34) Ewe w-a-bol-el-a Nelsoni o-li (*ali) ba-keni ba-rekukha.
you 2SG.S-PST-say-AP-FV 1Nelson 2SG-that 2-guests 2S-left
'You told Nelson that the guests left.'[10]

(35) Alfredi ka-a-loma a-li ba-ba-andu ba-mwekesia bali (*ali) o-mu-keni k-ola.
1Alfred 1S-PST-say 1-that 2-2-people 2S-revealed 2-that 1-1-guest 1S-arrived
'Alfred said people revealed that the guest arrived.'

Notably, Lubukusu complementizer agreement is subject to further restrictions. A number of interesting phenomena arise from its interaction with raising. Perception verbs may occur with an expletive-type subject or in a raising-to-subject construction. For some speakers, expletives can trigger class agreement on the complementizer (36). However, complementizer agreement with a raised subject is ruled out (37).

(36) Ka-lolekhana ka-li Tegani ka-a-kwa.
6S-seems 6-that 1Tegan 1S-PST-fell
'It seems like Tegan fell.'

(37) Michael a-lolekhana mbo (*ali) a-si-kona.
1Michael 1S-appears that (*1-that) 1S-PRES-sleep
'Michael appears that he is still sleeping.'

Embedded subjects may be produced to the left of the complementizer, providing evidence for raising to object. These constructions allow complementizer agreement with the matrix subject (38). When a raising-to-object verb is passivized, complementizer agreement with the derived

[10] All Lubukusu examples in this section are from (Diercks 2013); glosses and translations are as in the source.

subject is impossible (39), even though generally allowed with derived subjects of passives (40).

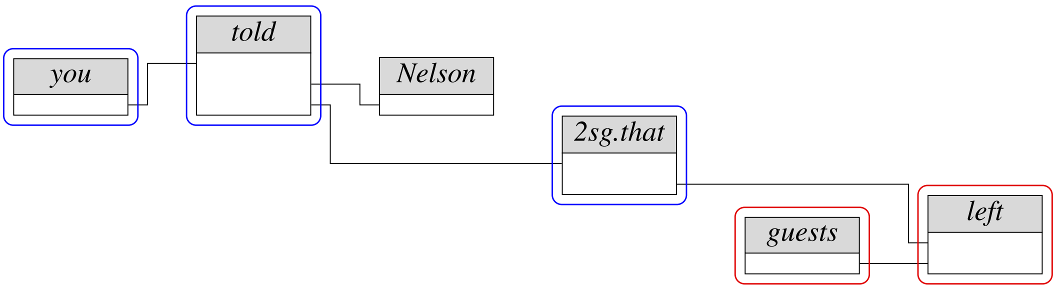
(38) N-enya Barack Obama n-di a-khil-e.
1sgS-want 1Barack.Obama 1sg-that 1S-win-SBJ
'I want Barack Obama to succeed.'

(39) Barack Obama k-enyi-bwa (*ali) a-khil-e.
1Barack.Obama 1S-want-PASS (*1-that) 1S-win-SBJ
'Barack Obama is wanted to succeed.'

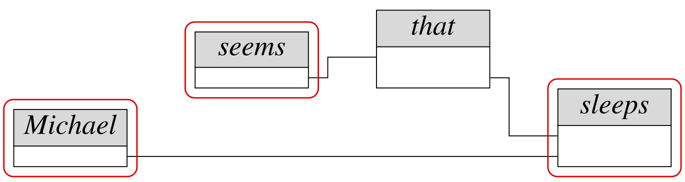
(40) Sammy ka-bol-el-wa a-li ba-keni b-ola.
1Sammy 1S-say-AP-PASS 1-that 2-guests 2S-arrived
'Sammy was told that the guests arrived.'

The empirical generalization encompassing the main case and the facts about raising is that complementizer agreement is only possible with a subject that *originated* in the superordinate clause (Diercks 2013, p.388). The graphs below represent the main Merge dependencies and complementizer requirements of the base case (41) and raising-to-subject (42). As in section §3, color-coding in these graphs indicate items which share morphological features. In (41) the embedded subject and verb agree with one another, while the matrix subject, verb and complementizer agree. In (42) the raised subject agrees with both matrix and embedded verbs, but not the complementizer. The *raising puzzle* is how to make complementizer agreement obligatory in (41), but forbidden in (42).

(41)



(42)



4.1 Indirect Agree

In order to deal with Lubukusu complementizer agreement, Diercks introduces the notion of *Indirect Agree*: a type of agreement configuration where the Agree relation is mediated by some other syntactic element. While this intermediate step does not necessarily correspond to a single theoretical operation, in the case of Lubukusu the agreement relation is mediated by Binding. The features of *C* are valued by local agreement with a null anaphor *OP* that originates in Spec-CP. *OP* goes on to adjoin to the superordinate *T*, coming into a local relationship with the subject in Spec-TP and establishing a Binding relation with it (43).

(43)

$$[_{TP}\ Subject_1\ \ldots\ [_{CP}\ OP_1\ [\ \ldots\ C\ \ldots\ ]\ \ldots\ ]\ \ldots\ ]$$

Binding *Agree*

A crucial element of this analysis is Local Antecedent Licensing (Safir 2004), stating that the anaphor must (covertly) move to the local domain of its antecedent to be bound. This setup allows restrictions on complementizer agreement to be explained as more general restrictions on clitic movement and Binding relations. In particular, complementizer agreement is limited to the most local superordinate clause by the Tensed Sentence Condition (Chomsky 1973) that restricts dependencies across clause boundaries. This ensures that *OP* adjoins to the most local superordinate *T* and is anteceded by its subject.

The raising puzzle is explained with a version of the Chain Condition (Rizzi 1986), whereby a coindexed phrase intervening between the Case position and the theta-position of an argument leads to ungrammaticality. This idea has been reconstructed by McGinnis (2004) as Lethal Ambiguity. According to her proposal, a moved element must be unambiguously linked to its lower copy, and elements are identified by their index and 'address' in the structure, in such a way that specifiers of the same head share the same address. Ungrammaticality arises when a phrase moves through a specifier position where it shares an address with a coindexed phrase.

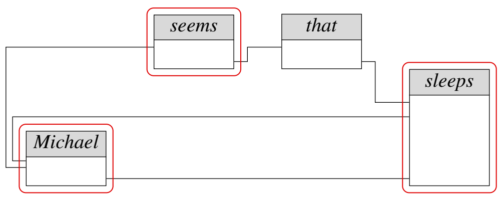
In the Lubukusu case, raising moves an argument over *OP*. Under the assumption that the subject must move to the edge of the CP phase to raise out of it (Chomsky 2000), the moving element lands in the specifier of the embedded *C*, where it shares the address with *OP*. In the raising-to-subject configuration *OP* is coindexed with the subject, resulting in an instance of Lethal Ambiguity. Raising-to-object, on the other hand, moves a phrase that is not coindexed with *OP* and creates no ambiguity.

For convenience, we will refer to this analysis as the *Binding approach*. A faithful translation of it into channel terms is presented in Appendix A.

4.2 Deriving an analysis from dependencies

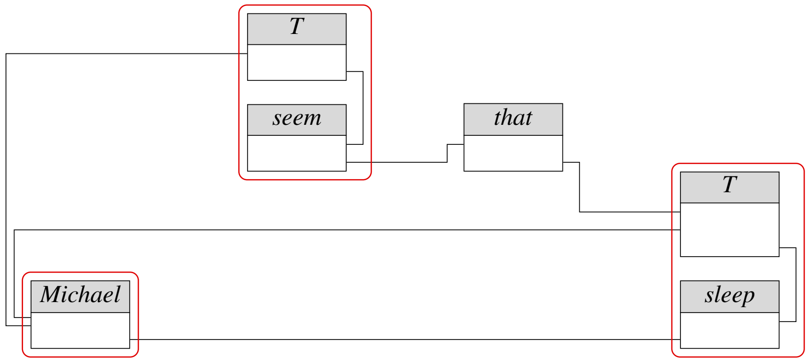
In this section we refine the basic dependencies of Lubukusu into a direct analysis of the previous data. For each dependency structure we consider, We systematically introduce a second dependency between nominals and the heads we associate with their case positions, and will decompose inflected verbs into T-(v-)V spans. The interesting aspects of the analysis will revolve around the reasoning about information flow. We begin with example (37), involving raising to subject, whose basic dependencies were given as (42). In this example, we saw that only a non-agreeing complementizer was possible. As stated previously, we insert additional dependencies between the DP and each inflected head we think it agreed with, giving rise to the refined structure in (44).

(44)



We next decompose the inflected verbs (*sleeps* and *seems*) into T and V heads, as shown in (45), which is the refined dependency structure on the basis of which we will reconstruct lexical feature bundles (and hence our syntactic analysis).

(45)



We extract a *non-agreeing* lexicon from this dependency structure, as shown in (46); this lexicon contains lexical items with the correct feature bundles, but these features have not (yet) been marked up to control the transfer of information along channels. Instead of assigning two `-k` features to the subject DP, we allow that a single `-k` feature may be checked multiple times (a persistent feature in the sense of Stabler (2011)).

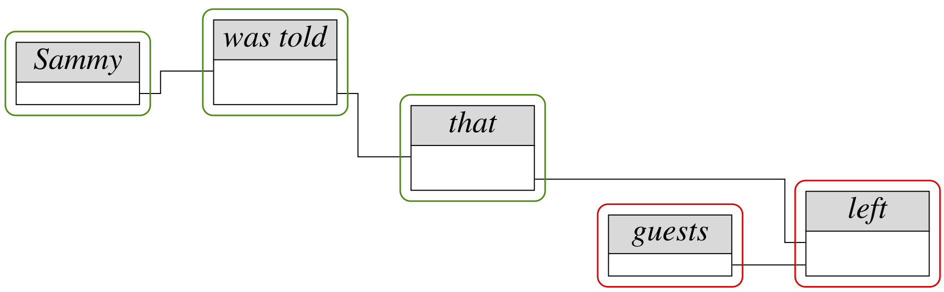
(46)

Michael	`::`	`-d.-k`	*sleep*	`::`	`+d.-v`	*T*	`::`	`+v.+k.-t`
			seem	`::`	`+c.-v`	*that*	`::`	`+t.-c`

Before doing that, we next must decide how information flows through this structure. The DP subject *Michael* must transmit information to both inflected verbs via dependencies, but crucially avoiding the complementizer, which cannot agree. While there are two paths to the lower span (D – V and D – T), the only path to the higher span involves D – T. For reasons of simplicity **we assume that this agreement is uniformly conducted over the D – T dependency**.

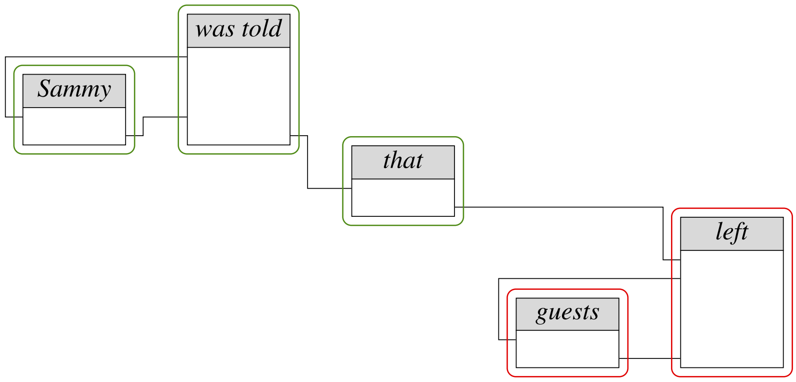
We now turn to sentence (40), which is at the level of basic dependencies a minimal variation of the previous example, in which the complementizer agrees with the subject. The basic dependencies are represented in (47).

(47)



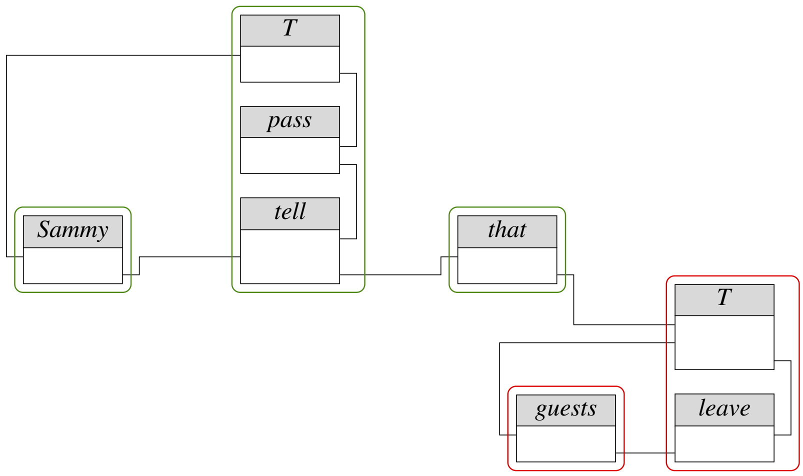
As before, we add dependencies between nominals and their assumed case positions. Doing this, we arrive at the structure in (48).

(48)



We now decompose the inflected verbs into complex spans involving tense and, for the matrix verb, passive heads. This represents our final syntactic analysis, and is shown in (49).

(49)

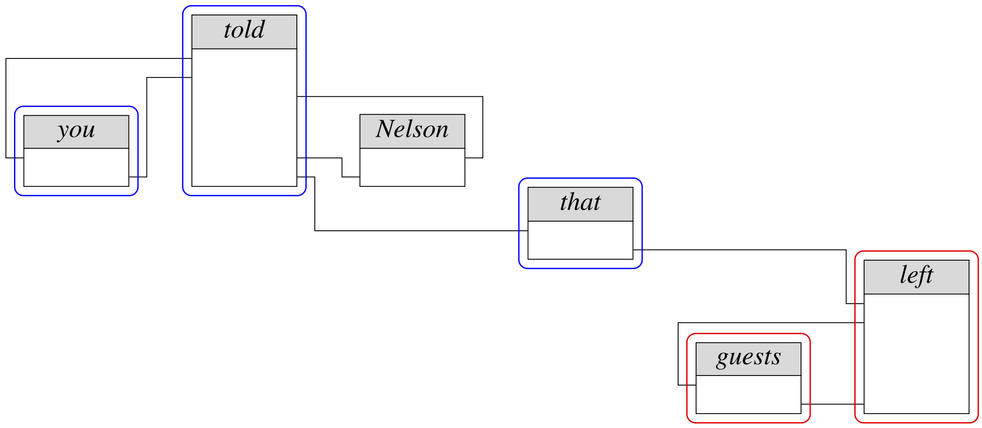


The information flow considerations inherited from the above analysis of sentence (40) allow for the transmission of information between nominals and tense. In contrast to that analysis, where crucially information did *not* flow to the (non-agreeing) complementizer, here information *must* be transmitted from noun to the (agreeing) complementizer. There are here two possible paths from DP *Sammy* to the complementizer; one making use of the dependency between D and T: D – T – Pass – V – C and the other using the dependency between D and V: D – V – C. Of these two paths, the first would have been present in sentence (37) as well, whereas the second is unique to the present sentence. We choose to make the second path the bearer of inflectional tidings to the complementizer, assuming that **agreement between nominals and**

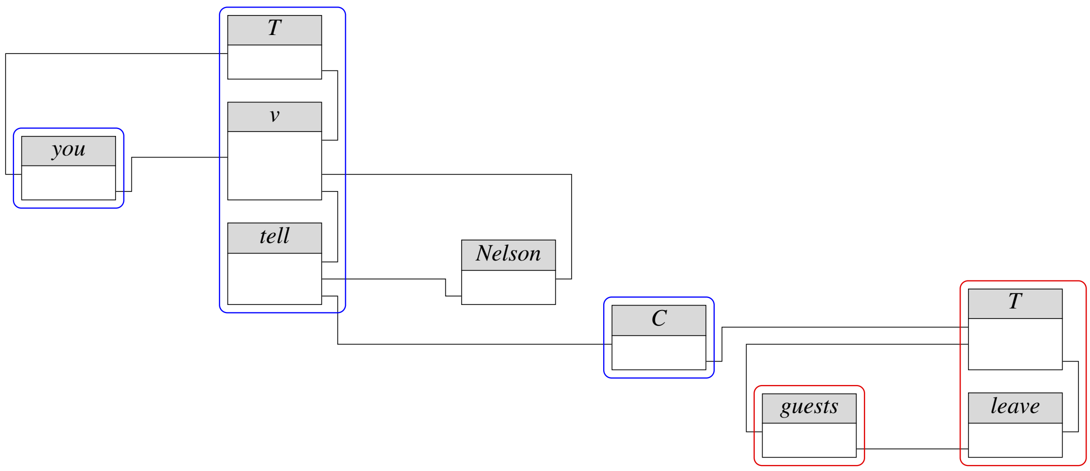
complementizers is uniformly conducted over the D – V – C dependency.

Turning now to sentence (34), the active version (modulo lexical identity) of sentence (40), its basic dependencies are given in (41). As before, we insert additional dependencies between nominals and their supposed case assigners, as shown in (50). We next decompose inflected verbs into tense, little-v (for *told*) and V heads, as shown in (51), which represents the final structural analysis for this sentence.[11]

(50)



(51)



Given our assumptions about information flow inherited from the analyses of the previous sentences, agreement between nominals and verbs is transmitted to the verbal span from D to T, and agreement between nominals and complementizers is transmitted from D to V to C. In this case, however, the complementizer should agree with the subject nominal *you*, and so we must revise our previous assumption to allow agreement to pass as well from D to v on to V to

[11] A more fine-grained analysis would further decompose *tell* into the root and the applicative morpheme introducing the indirect object. While this step is morphologically motivated, we do not perform it here for convenience and space reasons. Similarly, the subjunctive suffix *-e* (37), (38) is left as part of the root.

C. Intuitively, agreement with C takes place from a θ position in the verbal domain. Now there are two nominals with direct channels to C; the object *Nelson* and the subject *you*. We see that it is the structurally higher subject which actually agrees with C, and thus must explain why this is the case. From a structural perspective, the 'winner' of the agreement possibilities is the expression whose information enters the verb last. We accordingly adopt an 'overwriting' principle, which can be informally stated as such:

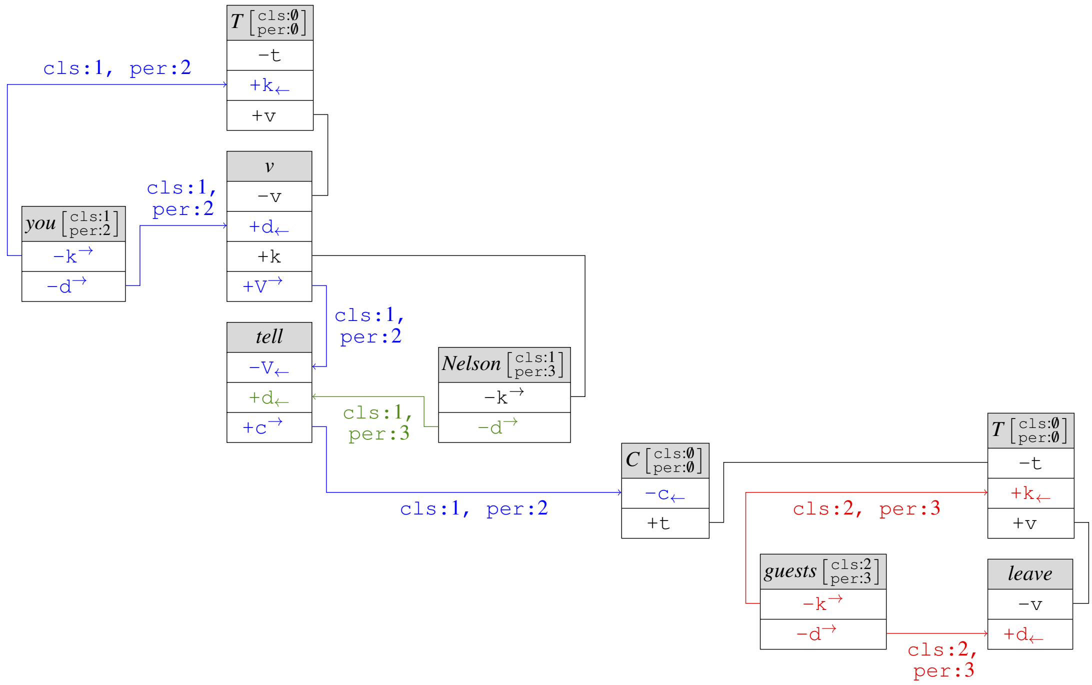
Overwriting if a lexical item receives conflicting values for a single morphological feature, the value which is received along the later channel takes priority

We can now reconstruct the features on our lexical items from the refined dependency structures we have been discussing. We begin with our most recent example, which exhibits all the complexity of the examples to date. The graph in (52) presents the syntactic features establishing the dependencies from (51), together with the morphological information flowing along these dependencies. Note that person and number information from the object *Nelson* flows in to the verb *tell* (at its +d feature), but that the same information from the subject *you* enters the verb afterwards (at its −V feature), and thus the subject's information takes priority over the object's, in that the only former is transmitted on to C. This is perhaps clearest at the level of the path equation for *tell*. This item collects information from two channels (its −V and +d features), and passes this on to a third channel (its +c) feature. Schematically, we want to write something like the following: $-\mathtt{V} + +\mathtt{d} \Longrightarrow +\mathtt{c}$, where $+$ is some way of combining the information received from the −V and +d channels. Pursuant to the discussion above, we want the information to be combined in such a manner that values from the −V channel take priority over those from the +d channel. The operation of *priority union* (Kaplan 1987) does exactly this:

$$\begin{bmatrix} \mathtt{m}_1\mathtt{:v} \\ \mathtt{m}_2\mathtt{:v}_2 \end{bmatrix} / \begin{bmatrix} \mathtt{m}_1\mathtt{:v'} \\ \mathtt{m}_3\mathtt{:v}_3 \end{bmatrix} = \begin{bmatrix} \mathtt{m}_1\mathtt{:v} \\ \mathtt{m}_2\mathtt{:v}_2 \\ \mathtt{m}_3\mathtt{:v}_3 \end{bmatrix}$$

Priority union combines morphological feature bundles via union, but where two values conflict, the first one is used.

(52)

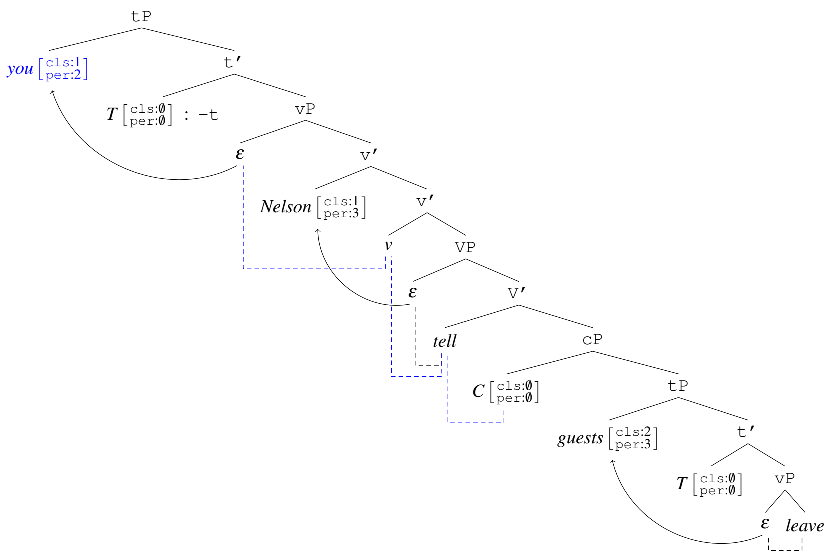


The lexicon in (53), assembled in the usual way, give rise to the derived syntactic structure in (54). Of crucial importance in this analysis (given by the lexical entries) is the fact that *T* is an informational island: it opens *neither* an information channel to the verb, it's +v feature blocks information from moving between *T* and *v*, *nor* one to its selecting complementizer, its -t feature keeps information from being transfered to whatever selects it. Similarly, *C* accepts information only from above (from whatever might select it), not from below (from its selected TP).

(53)

Nelson [cls:1 per:3]	::	-d$^{\rightarrow}$.-k$^{\rightarrow}$	*v*	::	+V$^{\rightarrow}$.+k.+d$_{\leftarrow}$.-v
leave	::	+d$_{\leftarrow}$.-v	*Pass*	::	+V$^{\rightarrow}$.-v$_{\leftarrow}$
tell	::	+c$^{\rightarrow}$.+d$_{\leftarrow}$.-V$_{\leftarrow}$	*T* [cls:∅ per:∅]	::	+v.+k$_{\leftarrow}$.-t
seem	::	+c$^{\rightarrow}$.-v$_{\leftarrow}$	*C* [cls:∅ per:∅]	::	+t.-c$_{\leftarrow}$

(54)



Now consider the other two constructions used above to drive the analysis: passive clauses (55) and raising-to-object (56).

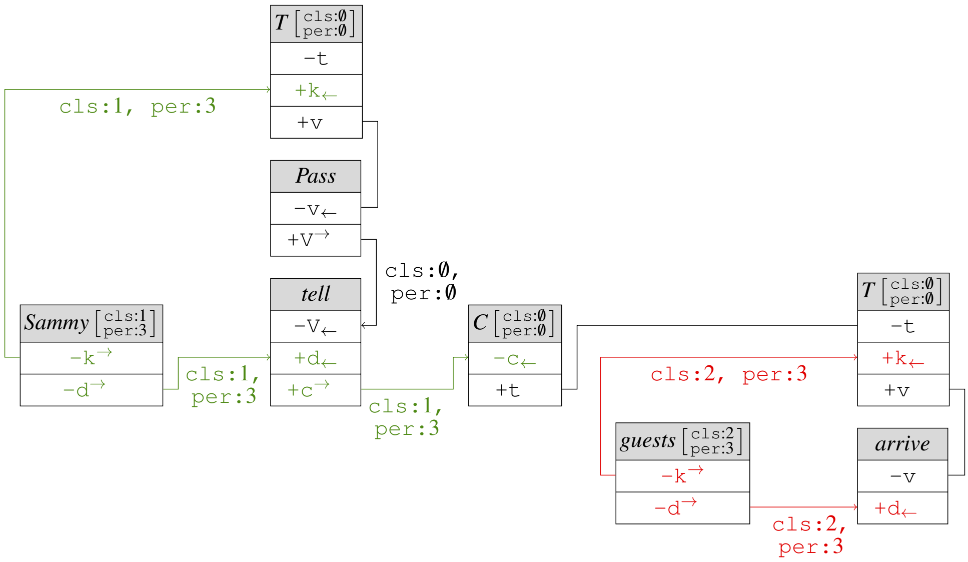
(55) Sammy ka-bol-el-wa a-li ba-keni b-ola.
1Sammy 1S-say-AP-PASS 1-that 2-guests 2S-arrived
'Sammy was told that the guests arrived.' =(40)

(56) Michael a-lolekhana mbo (*ali) a-si-kona.
1Michael 1S-appears that (*1-that) 1S-PERS-sleep
'Michael appears that he is still sleeping.' =(37)

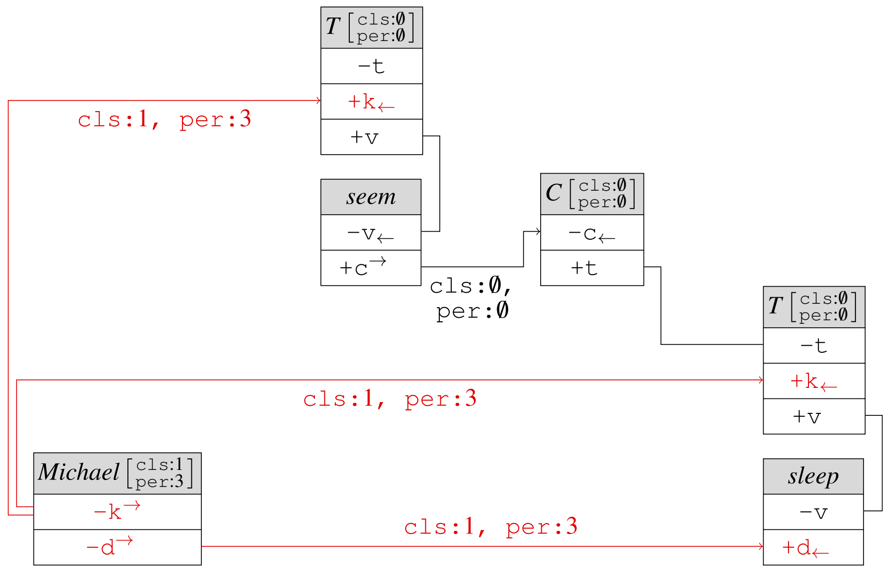
In the former case, the complementizer agrees with the derived subject. Whichever argument is merged last is promoted to subject and ends up transmitting its feature values to the embedded *C* (57).[12] In the latter case, no arguments are merged in the superordinate clause. This leaves the embedded *C* with unvalued features – to be realized by morphology as the non-agreeing complementizer *mbo* (58).

[12]We have not yet explicitly stated how agreement proceeds if a head has a receiving channel but no values are sent through it. This is what takes place in (55): the highest receiving channel of *tell* connects to *Pass*, which however receives no data through its own receiving channel on $-v_{\leftarrow}$. One intuitive way to resolve this, which produces the desired outcome, is to say that *nothing happens*: channels that transmit no features have no effect on agreement. In other words, morphological features may be overwritten but not erased.

(57)



(58)



Diercks' generalization about agreement in Lubukusu distinguishes between arguments that originated in the matrix clause and arguments that have been moved there; while both successfully trigger verbal agreement, only the former are capable of valuing the features of the embedded complementizer. This distinction translates easily into channel configurations. As before, verbal agreement is transmitted via **k** dependencies and can be safely factored out. However, complementizer agreement spreads exclusively via the dependencies of external Merge.

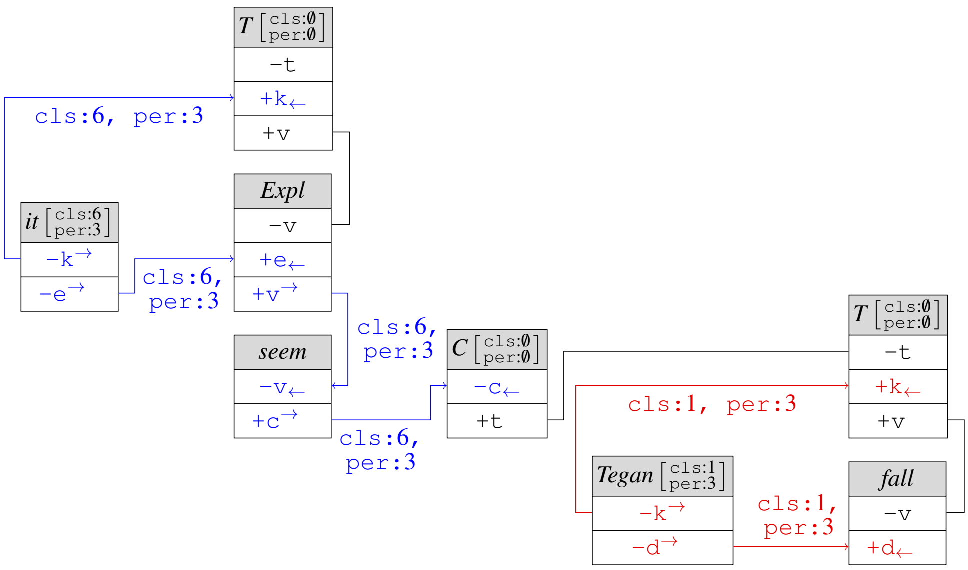
This can be further exemplified by the contrast between raising-to-subject (56) and expletive constructions with perception verbs (59). Both exhibit normal agreement with the subject on *T*.

(59) Ka-lolekhana ka-li Tegani ka-a-kwa.
6S-seems 6-that 1Tegan 1S-PST-fell
'It seems like Tegan fell.' =(36)

We follow (Diercks 2013) in adopting Bowers' (2002) assumption that expletives are merged in the same position as the external argument and move to the subject position in Spec-TP — essentially the same low-origin configuration that has already been discussed for the English *there* in section 3. The phonetically null expletive-type subject *it* transmits its features to the *Expl* head that selects it and, being the highest argument generated in the matrix clause, to the embedded complementizer (60), (61).

(60) *it* $\begin{bmatrix}\texttt{cls:6}\\\texttt{per:3}\end{bmatrix}$ :: $\texttt{-e}^{\rightarrow}\texttt{.-k}^{\rightarrow}$ *Expl* :: $\texttt{+v}^{\rightarrow}\texttt{.+e}_{\leftarrow}\texttt{.-v}$

(61)



Finally, the agreeing complementizer can occur with raising-to-object verbs (62), (63). Here the raised object (or subject, if later passivized) is not able to agree with the complementizer. Rather, it is again only the highest argument base generated in the matrix clause which triggers agreement.

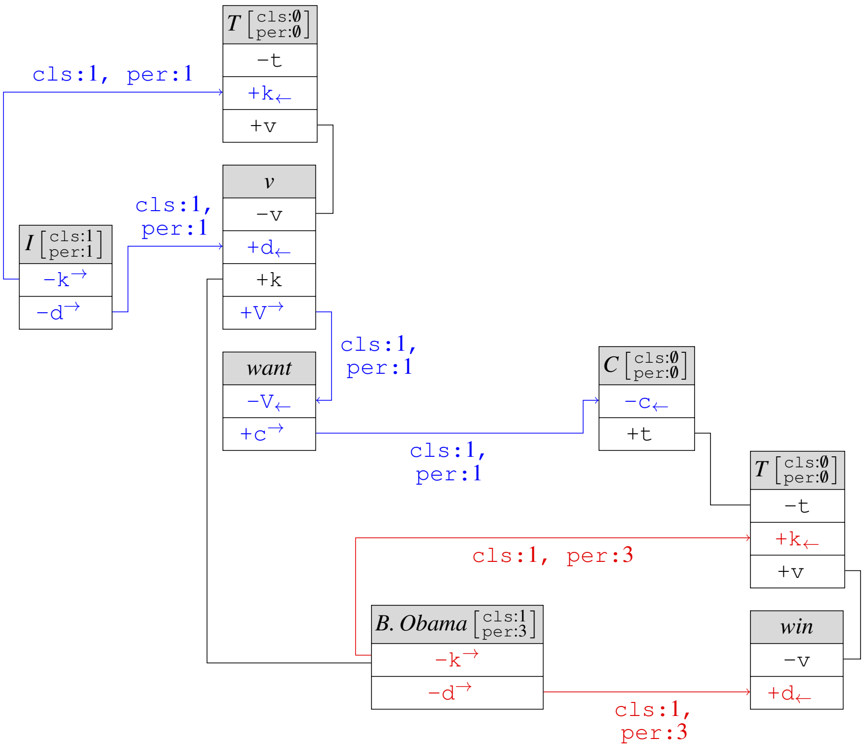
(62) N-enya Barack Obama n-di a-khil-e.
1sgS-want 1Barack.Obama 1sg-that 1S-win-SBJ
'I want Barack Obama to succeed.' =(38)

(63) Barack Obama k-enyi-bwa (*ali) a-khil-e.
1Barack.Obama 1S-want-PASS (*1-that) 1S-win-SBJ
'Barack Obama is wanted to succeed.' =(39)

The lexical verb *want* (64) simply passes whatever features it receives from *v* down to the complementizer, yielding for (62) the channel structure in (65). The features of the raised argument are not passed into the higher clause and don't come into contact with *want*.

(64) *want* :: $+\mathtt{c}^{\rightarrow}.-\mathtt{V}_{\leftarrow}$

(65)



The subject of the matrix clause is the only possible goal for complementizer agreement. As expected, passives of raising-to-object verbs where no arguments are generated in the matrix clause (63) can only occur with the non-agreeing complementizer.

4.3 Summary

Unlike the Binding approach, the Overwriting solution gives each argument a chance to contribute to complementizer agreement. Every noun phrase transmits its `cls` and `per` values to the expression that selects it. Since newer values overwrite older ones, as per the overwriting principle, *C* receives class and person from the highest NP merged in the immediately superordinate clause.

As the channel approach does not make use of indices or operations other than basic Merge and Move, no additional machinery is required to express agreement. There is no feature exchange between T and C, which means that complementizer agreement processes are restricted to the most local superordinate clause. Ruling out agreement at this level can be essentially thought of as reconstructing the Tensed Sentence Condition in terms of MG-like features.

Transmitting complementizer agreement exclusively via the dependencies of external Merge ensures that its source originates in the matrix clause: an argument moved into the superordinate subject position would not be an eligible goal for complementizer agreement. Recall the generalization in (Diercks 2013, p.388) stating that the complementizer can only agree with a subject that originated in the superordinate clause, as opposed to one which was moved there. The Overwriting approach deals with raising by directly capturing this generalization. It is worth noting that we treat the complementizer as an upward-looking probe, however, instead of agreeing with the closest c-commanding goal, the complementizer agrees with the farthest one.

5. Extensions

Splitting Agree off from the syntax proper in the way we propose in this paper makes the search space of agreement quite salient, and allows us to explore different ways of allowing probes and goals to find each other. We have seen that we are in fact able to lexicalize these agreement paths, at least for the cases we have considered; this means that each lexical item is able to contribute to potential agreement relations between words in a sentence in its own particular way. This is a significant departure from the usual, universal and therefore purely geometric approach which is standard. Here we would like to explore whether and how various aspects of the more standard approach can be expressed in terms of path equations, as well as to briefly discuss how more sophisticated interactions between morphological features (such as resolution) can be dealt with in our lexicalist approach.

5.1 Interaction between morphological feature bundles

Our analysis of Lubukusu was based on the assumption that when a head receives multiple values of a morphological feature from different sources, it keeps the last one — in other words, inherited features can be overwritten by later instances of agreement. This solution worked well for the Lubukusu puzzle and may prove useful for other phenomena (Ermolaeva (2018), using a system similar to the present one, makes the same assumption in her analyses of Icelandic).

Bejar and Massam's (1999) analysis of multiple case checking suggests a similar mechanism at play. When a DP is assigned multiple structural cases, it realizes the one received last; when a quirky case and a structural case are assigned, the quirky case wins. The second part of this observation can be handled in our formalism by separating the `case` feature into two components, feeding both quirky and structural case values to the morphological component (which will realize the more highly specified case). The 'last value wins' restriction automatically takes care of the first part. Another example is found in Richards (2013). In Lardil, inflected nominals drop semantically uninterpretable case (such as accusative) when further morphology is added; however, semantically interpretable case (such as instrumental) is retained, and the new morphology attaches after the case morpheme. Once again, the channel system can potentially handle this by splitting the `case` feature and allowing the uninterpretable case component to be overwritten by later agreement.

At the same time, certain phenomena require a more complex interaction between morphological features from different sources. A number of interesting cases revolve around *feature resolution* in coordinate noun phrases — a process by which the features of a coordinate structure are computed from those of individual conjuncts (Corbett 1983). Consider a common pattern of gender agreement that is found, for instance, in French: the coordinate structure is feminine just in case both conjuncts are feminine, and masculine otherwise. In channel terms, this would simply require the *and* head to either allow the feminine gender feature `gdr`:fem to be rewritten with any value, or to allow the masculine `gdr`:mas to rewrite any value, regardless of the order (66). This is easily achieved by identifying fem and mas to be the bottom and top elements of a boolean algebra respectively, and taking boolean joins (i.e. disjunction).[13] Dalrymple and Kaplan (2000) extend this idea by using join-semilattices (represented as sets of atomic features under set union).

(66) *and* $\begin{bmatrix} \texttt{num:}\emptyset \\ \texttt{per:}\emptyset \\ \texttt{gdr:}\emptyset \end{bmatrix}$ `::` `+D.+D.-D` $\begin{cases} \downarrow\Longleftarrow \texttt{+D}_1 \vee \texttt{+D}_2 \\ \downarrow\Longrightarrow \texttt{-D} \end{cases}$

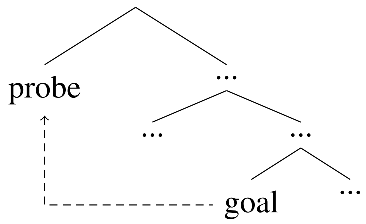
A slightly more complex example comes from the domain of number agreement. In Slovenian (again from Corbett (1983); for a more recent discussion see Marušič et al. (2015)), the coordinate structure is dual if both arguments are singular, and plural otherwise. Here again we need to allow features to be combined in a manner different from simple priority union. We might think of the conjunction head — or, in fact, any head that has multiple receiving channels — as a morphological function that takes multiple feature bundles as its arguments and outputs a new morphological feature bundle based on their properties. Priority union then becomes a special case of such a function. Formal constraints on such functions, such as monotonicity (Graf 2019), can be imposed to better circumscribe the attested forms of resolution in the languages of the world (Corbett 2006).

5.2 *Directionality of Agree*

The standard version of Agree, following Chomsky (2000), takes place between a probe *P* and goal *G* such that: 1. *P* and *G* have matching features, 2. *P* c-commands *G*, and 3. there is no other eligible goal for *P* that is closer than *G*. This is known as downward Agree (67): the probe looks *downward* into its domain (sister), and once the Agree relation is established, feature values are transmitted upwards from the goal to the probe. One notable alternative, proposed by Zeijlstra (2012), is upward Agree (68): the probe is c-commanded by the goal and must look *upward* to find a matching feature. Feature values are then transmitted downwards.

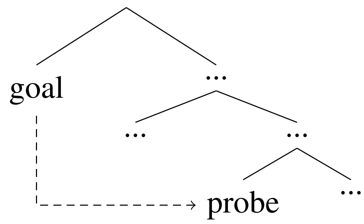
[13] Dually, fem could be identified with the top and mas the bottom elements, and then boolean meet (i.e. conjunction) could be used.

(67)



Downward Agree

(68)

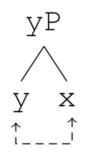


Upward Agree

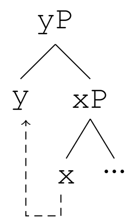
Which of these works better is a matter of debate; see, for example, Bjorkman and Zeijlstra (2014) for arguments for and an elaboration on downward valuation, and Preminger (2013) for arguments against it. From our abstract perspective on Agree as searching through a structure to find a path from probe to goal, both upward and downward Agree are restrictions on the possible paths which may be selected. For example, if we adopt downward Agree, we are in effect ruling out all paths which do not pass through the sister of the probe. This restriction can simply be adopted without change, as a constraint on well-formed agreement paths. While logically distinct from the proposal to dissociate Agree from syntax, we have chosen in this paper to lexicalize agreement path constraints using path equations (simple cases of which we can more concisely represent using channel notation on features). One might well wonder whether the path constraints which embody restrictions to either downward or upward Agree can also be lexicalized, and if so, whether they are natural to state in these terms.

We begin with downward Agree, which imposes the constraint that all agreement paths must pass through the sister of the probe. As path equations refer to dependencies via the syntactic features in a lexical item's feature bundle, we must identify a property of feature bundles which allow us to speak of the sister of a lexical item. There are in fact two distinct configurations in which a lexical item may probe into its sister. In the first, the atomic lexical item is selected by a phrase which projects over it, as in examples a and c of figure (69); here the focus is on the atomic lexical item `x`, which is selected by `y`. In the second case, the atomic lexical item selects another phrase, as in examples a and c; here the focus is on `y`, which selects `x` as its first merged argument. What is absolutely ruled out is when a complex phrase either selects or is selected by another, as in examples b (the complex `x` cannot probe into `y`), c (the complex `y` cannot probe into its argument `x`), and d (neither complex phrase can probe into the other).

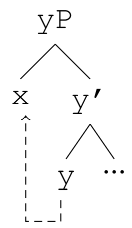
(69)



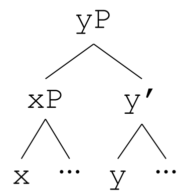
a. Both expressions are atomic: agreement in any direction



b. Atomic selects complex: the head agrees with the dependent



c. Complex selects atomic: the dependent agrees with the head



d. Both expressions are complex: no agreement

What unifies these two cases is the distinction between atomicity and complexity; an atomic expression (a lexical item) can probe into the first expression it combines with via Merge. The dependency connecting it with this expression is the one linked to its first feature. To implement downward Agree we then must stipulate that only information from a lexical item's first feature can be used to value its own features. We can effect this by restricting path equations of the form $\downarrow\Longleftarrow$ **x** to be permitted only when **x** is the first feature in the feature bundle. However there is an additional complication, namely, that the atomic `x` in case c of figure (69) might not have been base generated there (as we have heretofore assumed) but may have moved there. This is a case of movement feeding downward Agree.[14] This can be accommodated by allowing path equations of the form $\downarrow\Longleftarrow$ `-x` for `-x` not the first feature in a feature bundle only in case the entire feature bundle consists of negative features. Feature overwriting then implements the closest c-command condition.

The path constraint imposed by upward Agree is more complicated that that imposed by downward Agree, because not just any path not passing through the sister of the probe is a possible agreement path, but rather only those which end at (the maximal projection of) a goal which c-commands the probe. Inside the maximal projection of a lexical item, this can be

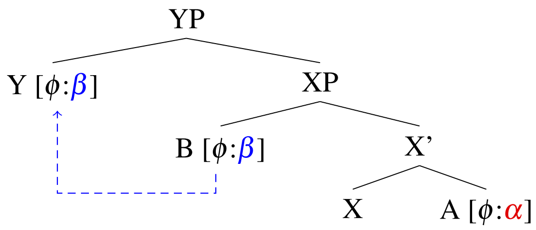
[14]Movement can feed downward Agree only in case the mover is atomic (i.e. a lexical item). This is due to our formulation of the search space of downward Agree, which requires that it search through the probe's sister. In case a complex phrase moves (as in the case of `xP` in case d), the head `x`'s sister remains the same.

enforced by restricting path equations of the form $\downarrow\Longleftarrow$ +x to be permitted only when +x is **not** the first feature in the feature bundle; this allows a probe only to agree with a goal which is *not* in its complement. This is exemplified by y in the configurations c and d. Here y may agree with x and/or xP, as neither of these are in its complement. The additional complication of upwards Agree comes from the restriction that only c-commanding (maximal projections of) goals may agree with probes. This may also be dealt with inside of feature bundles. We first note that within the maximal projection of a head, later merged dependents c-command earlier merged dependents. Translated into feature bundles, this says that later +x features can pass their information to earlier ones. We thus require that in a path equation of the form +x $\Longrightarrow$ +y, +y must precede +x in the feature bundle. Finally, a maximal projection c-commands everything inside of its sister. In terms of feature bundles, the sister of a maximal projection is anything that checks one of its negative features. Accordingly, a lexical item can serve as a goal to anything which checks one of its negative features. In terms of path equations, we allow information about a LI to be sent off along its negative features (but also its first positive feature i.e. its complement): in $\downarrow\Longrightarrow$ **x**, **x** is either a negative feature, or the first positive feature in a feature bundle. What this does not account for is the closest c-command condition on upward agree. In fact, feature overwriting as we have it currently implemented ensures a default *farthest* c-command condition! We want a condition we might call feature *under*writing, which has lower channels take priority over higher ones. This can be implemented using priority union as a condition on the well-formedness of path equations. Parallel to the implementation of overwriting, which we did by requiring that channels hosted by later features take priority over those hosted by earlier ones, we implement underwriting by requiring that the channels hosted by earlier features take priority over those hosted by later ones.

5.3 Locality and intervention effects

Chomsky's (2000) probe-goal system requires Agree to be local, where locality is understood as 'closest c-command'. In Adger (2010) this restriction is incorporated as the Minimal Link Condition (MLC) demanding that the features in a probe-goal relation have no other matching feature intervening between them. This is represented schematically in (70): the probe *Y* agrees with the closest goal *B*, but not with the lower goal *A*.

(70)

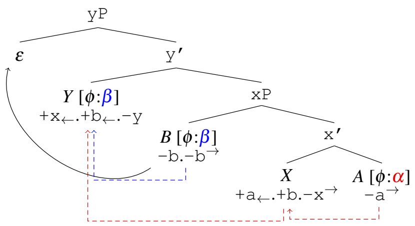


Agreement-as-channels already incorporates the prerequisites for this constraint. Recall that each instance of Agree is tied to a specific syntactic dependency created by Merge. A long-distance morphological dependency between two heads can be represented as a series of local information exchanges across Merge dependencies involving, step by step, each of the intervening heads. At each step, information exchange occurs directly between the heads that enter a feature-checking relationship, and the probe does not have access to other goals in the structure.

Towards implementing an MLC-like restriction, we observe that the crucial configuration arises only when a given node along the search path dominates both possible goals (in figure (70), this is the node XP, which contains B in its left subtree, and A in its right). Translated into channels, this obtains when a lexical item (here again X) has multiple receiving channels and comes into direct contact with multiple morphological feature bundles (from different arguments it selects/licenses). The channel system redefines locality in terms of syntactic dependencies rather than linear or structural adjacency. The overwriting principle (implemented via priority union) expresses that the features of the structurally higher argument (here B) are the ones that are visible to the exclusion of those of the lower argument (here A). In case A have features B does not, those features continue to be visible.

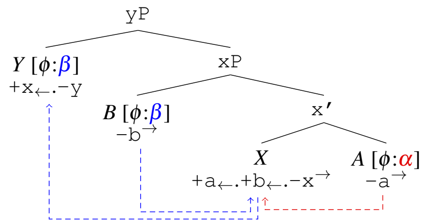
Two abstract examples below illustrate the intuition behind this. In (71), Y forms a Move dependency with B and agrees with it, rewriting the features it received earlier from A. In (72), the probe Y does not come into direct contact with either A or B and can only agree with them through the mediation of X. In its turn, X receives morphological information from both A and B and passes the latter to Y at the next step of the derivation. In traditional terms, the argument of X which was merged last becomes the closest goal for Y.

(71)



Covert movement

(72)



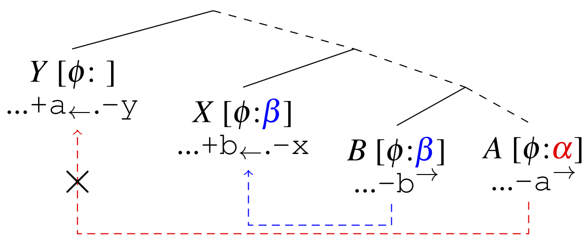
Merge

In both cases, overwriting ensures that Y receives morphological features from whatever goal was merged last — which is also the closest goal for Y.

5.3.1 *Defective Intervention*

A large body of work, starting with Chomsky (2000), examines configurations where an Agree relation between a probe and a goal is apparently blocked by a closer goal which itself is inactive due to a prior Agree with another probe. While intervention in general, in the sense of every head between the probe and the goal being potentially able to block agreement, is necessarily built into long-distance Merge dependencies of the channel system, this sort of "Defective Intervention" is not. As recent work questions traditional analyses based on this phenomenon or even its very existence (Broekhuis 2007; Bruening 2014), this may be viewed as desirable. Defective intervention, should it exist, can be added to the present system via an island-like constraint ruling out agreement (for a given morphological feature) across an existing dependency that carries a value of the same feature, as in figure (73).

(73)



Of course, particular instances of defective intervention can be added piecemeal simply by postulating abstract heads which do not let information pass through them (as we might do for experiences defectively intervening between a raised existential *there* and its subordinate clause) on the agreement path.

6. Conclusion

The previous sections have presented channels as an alternative way to conceptualize agreement in minimalist syntax. Syntax is thereby reduced exclusively to structure building — the only operation is (internal and external) Merge — and agreement is, while parasitic on the structure building process, nevertheless not part of it. By making this logical split between structure building (i.e. syntax) and agreement, we are able to separate syntactic from morphological feature bundles.[15] This in turn allows syntax to be formulated so as to completely satisfy the no tampering condition of Chomsky (1995). Reducing the operations of syntax to just Merge allows us to use the easy to manipulate dependency-like representations to develop analyses, as it is simple to obtain from them a detailed lexicon.

We do not view this approach to agreement as formally divorced from the more familiar Agree operation. Instead, we see channels as a static representation of the way that Agree is thought to search through the syntactic structure, trying to find a path from probe to goal. As the syntactic structure is the result of Merge operations, the search path of Agree of necessity flows along these Merge dependencies. We can thus see that Agree conflates a particular, syntactic, implementation of information flow with a universal proposal about how to restrict the available channels. We would like to disentangle these two, pushing Agree out of the syntax, while entertaining the question of whether available channels and information flow can be derived from something more principled.

We also see this work as contributing to a rapprochement between minimalism as she is practiced, and the formal framework of minimalist grammars. To our eyes, the focus on agreement in minimalism contrasts sharply with its complete omission in minimalist grammars. The present work is our attempt to fill this gap. By design, it is light on formal universals and sufficiently permissive 'out of the box', so as to allow for the definition of a variety of restrictions (as substantive universals) and to model their consequences. Our formalism sets the stage for an important line of future work — determining which constraints or additions are feasible and/or useful. This work, purely linguistic in nature, involves implementing the modifications in the channel system and constructing precise grammar fragments to observe how well they

[15]This assuages worries about using feature checking on features potentially relevant to morphology (Corbett 2006).

perform on language data.

References

D. Adger. *Core Syntax*. Oxford University Press, 2003.

D. Adger. A minimalist theory of feature structure. *Features: Perspectives on a key notion in linguistics*, pages 185–218, 2010.

A. Alexiadou and F. Schäfer. There-insertion: an unaccusativity mismatch at the syntax-semantics interface. In *Poster presentation at West Coast Conference on Formal Linguistics*, volume 28, 2010.

D. Basilico. The topic is "there". *Studia Linguistica*, 51(3):278–316, 1997.

S. Bejar and D. Massam. Multiple case checking. *Syntax*, 2(2):65–79, 1999.

B. Bjorkman and H. Zeijlstra. Upward agree is superior. *Ms., University of Toronto and Universität Göttingen*, 2014.

J. D. Bobaljik. Where's phi? agreement as a postsyntactic operation. In D. Harbour, D. Adger, and S. Béjar, editors, *Phi Theory: Phi-features across modules and interfaces*, chapter 10, pages 295–328. Oxford University Press, 2008.

J. Bowers. Transitivity. *Linguistic Inquiry*, 33(2):183–224, 2002. doi: 10.1162/002438902317406696.

M. Brody. Mirror theory: Syntactic representation in perfect syntax. *Linguistic Inquiry*, 31(1):29–56, 2000.

H. Broekhuis. Does defective intervention exist? *Linguistics in the Netherlands*, 24(1):49, 2007.

B. Bruening. Defects of defective intervention. *Linguistic Inquiry*, 45(4):707–719, 2014.

N. Chomsky. Conditions on transformations. *A Festschrift for Morris Halle, Holt, Rinehart & Winston, New York*, pages 232–286, 1973.

N. Chomsky. *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group, 1986.

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

N. Chomsky. Minimalist Inquiries: the framework. In R. Martin, D. Michaels, and J. Uriagereka, editors, *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, pages 89–156. MIT Press, Cambridge, MA, 2000.

G. Corbett. Resolution rules: agreement in person, number, and gender. In G. Gazdar, E. Klein, and G. K. Pullum, editors, *Order, Concord, and Constituency*, chapter 8, pages 175–206. Walter de Gruyter, 1983.

G. Corbett. *Agreement*. Cambridge University Press, 2006.

M. Dalrymple and R. M. Kaplan. Feature indeterminacy and feature resolution. *Language*, pages 759–798, 2000.

A. R. Deal. The origin and content of expletives: Evidence from "selection". *Syntax*, 12(4): 285–323, 2009.

M. Diercks. *Agreement with subjects in Lubukusu*. PhD thesis, Georgetown University, 2010.

M. Diercks. Indirect agree in lubukusu complementizer agreement. *Natural Language & Linguistic Theory*, 31(2):357–407, 2013.

M. Ermolaeva. Morphological agreement in minimalist grammars. In *International Conference on Formal Grammar*, pages 20–36. Springer, 2018.

T. Graf. Monotonicity as an effective theory of morphosyntactic variation. *Journal of Language Modelling*, 7(2):3–47, Sep. 2019.

R. M. Kaplan. Three seductions of computational linguistics. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett, editors, *Linguistic Theory and Computer Appications*, pages 149–188. Academic Press, 1987.

F. Marušič, A. I. Nevins, and W. Badecker. The grammars of conjunction agreement in slovenian. *Syntax*, 18(1):39–77, 2015.

M. McGinnis. Lethal ambiguity. *Linguistic inquiry*, 35(1):47–95, 2004.

G. Müller. On deriving CED effects from the PIC. *Linguistic Inquiry*, 41(1):35–82, 2010.

O. Preminger. That's not how you agree: A reply to zeijlstra. *The Linguistic Review*, 30(3): 491–500, 2013.

N. Richards. Lardil "case stacking" and the timing of case assignment. *Syntax*, 16(1):42–76, 2013.

L. Rizzi. On chain formation. *Syntax and semantics*, 19: The Syntax of Pronominal Clitics: 65–95, 1986.

J. Sabel. *Expletives as features*. Univ.-Bibliothek Frankfurt am Main, 2000.

K. Safir. *The syntax of anaphora*. Oxford University Press on Demand, 2004.

E. P. Stabler. Derivational minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 Nancy, France, September 23–25, 1996 Selected Papers*, pages 68–95. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

E. P. Stabler. Remnant movement and complexity. *Constraints and resources in natural language syntax and semantics*, 2:299–326, 1999.

E. P. Stabler. Computational perspectives on minimalism. In C. Boeckx, editor, *The Oxford Handbook of Linguistic Minimalism*, Oxford Handbooks in Linguistics, chapter 27, pages 616–641. Oxford University Press, New York, 2011.

M. Starke. *Move Dissolves into Merge: A Theory of Locality*. PhD thesis, University of Geneva, 2001.

P. Svenonius. Spans and words. *Morphological metatheory*, 229:201, 2016.

E. Williams. *Representation Theory*. MIT Press, Cambridge, Massachusetts, 2003.

H. Zeijlstra. There is only one way to agree. *The Linguistic Review*, 29(3):491–539, 2012.