Syntactic Identity in Survive-Minimalism

Ellipsis and the Derivational Identity Hypothesis*

Gregory M. Kobele

Humboldt-Universität zu Berlin

Abstract

Over the years, a number of counter-examples to the hypothesis that ellipsis resolution is mediated via syntactic identity have been identified. However, in the same time evidence which seems to require comparison of syntactic structures in ellipsis resolution has also been unearthed. On top of this empirical puzzle, survive minimalism places an additional theoretical constraint: syntactic structures, once assembled, are opaque to further search or manipulation. In this paper, I show that a simple perspective shift allows us both to view the purported counter-examples as providing glimpses into the nature of the operations which build syntactic structure, and to satisfy the theoretical constraints imposed by survive minimalism's derivational take on syntactic structure.

1 Syntactic identity in ellipsis

One of the basic tasks of a theory of ellipsis is to explain what elliptical sentences can mean.[1] An observation so obvious it hardly bears mentioning is that the interpretation of an ellipsis site is not free, but rather is constrained in various ways by aspects of its surrounding context. As an example, in an empty discourse context, sentence 1 has only one meaning, which is synonymous with 2, and sentence 3 means only what 4 does, although only the matrix verb differs between the two.

(1) Jesús praised every girl that Adam did.

*My thanks to Hans-Martin Gärtner, Mike Putnam, and an anonymous reviewer for their helpful suggestions. The usual disclaimers apply.

[1]A complete theory of ellipsis must account not only for how utterances of elliptical sentences are interpreted, but also for the distribution of ellipsis sites in discourse. It seems fruitful to approach the study of ellipsis by pursuing these two tasks independently. I will here largely ignore the task of specifying *when* ellipsis is licensed (though see Hardt (1993); Lobeck (1995); Murguía (2004)).

(2) Jesús praised every girl that Adam praised.

(3) Jesús met every girl that Adam did.

(4) Jesús met every girl that Adam met.

1.1 Two perspectives on ellipsis

There are roughly two approaches to determining the meaning of an elliptical sentence. The first view is that the meaning of 1 is computed differently from the meaning of 2. In particular, 1 is taken to be significantly syntactically different from 2 in that the embedded verb phrase in 1 has no internal structure (it can be thought of as a pronoun as long as one does not adopt the position that pronouns have internal structure (Parsons, 1978; Elbourne, 2002)). This first perspective I will refer to as the *proform theory of ellipsis*. The opposing view holds that the meaning of elliptical sentences is computed in exactly the same way as that of non-elliptical sentences; in other words, that ellipsis sites are syntactically structured objects.[2] According to this view, sentences 1 and 2 have the same meaning in virtue of the fact that they have the same or similar syntactic structure; the major difference between them is that some of this syntactic structure was not pronounced in 1. I will call this view the *non-pronunciation (or deletion) theory of ellipsis*. Note that the non-pronunciation theory of ellipsis has reframed the task of explaining what elliptical sentences mean as explaining under what conditions parts of sentences can be left unpronounced. Two options suggest themselves. Namely, that a sentence part can be left unpronounced just in case it is similar to an antecedent either syntactically, or semantically. I will refer to these different realizations of the non-pronunciation theory as the *deletion under syntactic/semantic identity theory of ellipsis*.

1.2 Deletion vs the pro-form theory

As the major structural difference between the deletion and pro-form theories lies in the syntactic representation they assign to the ellipsis site (with the deletion theory assigning to it the standard articulated structure, and the pro-form theory taking it to be syntactically simplex), the obvious way to argue for the deletion theory is to show that the ellipsis site exhibits characteristically syntactic behaviour, and to argue for the pro-form theory, that it doesn't.

[2]This, of course, is predicated on the widely held assumption that meaning is read compositionally off of syntactic structure, or, in other words, that a syntactic description of an expression is simultaneously a description of its meaning.

1.2.1 Island sensitivity

It is well known that expressions can be dependent upon, descriptively speaking, positions internal to an ellipsis site, as in 5.

(5) Jesús loves Susan, but I don't know who Adam does.

Intuitively, *who* in 5 saturates a semantic argument of the recovered verb phrase meaning. While this has a natural description in terms of syntactic structure, we might think of equally natural semantic characterizations, and so the mere existence of sentences like 5 can't be taken as an argument for the deletion theory. However, characteristic of syntactic dependencies is that they are subject to various geometrical ('island') constraints, which do not seem to have a natural characterization in semantic terms. Merchant (2001) has shown that expressions (like *who* in 5) which seem to be dependent on ellipsis-site internal elements are subject to island constraints, as in 6, where *which subject* would be related to a gap within a relative clause (*girls that like* **t**), as is explicit in 7.

(6) *Jesús goes for girls that like geometry, but I don't know which subject Adam does.

(7) *Jesús goes for girls that like geometry, but I don't know which subject Adam goes for girls that like.

It is well-known that ellipsis sometimes ameliorates island violations. Although this might seem to favor the pro-form theory, the fact that there are *ever* island effects in elliptical contexts poses a severe problem for the pro-form theory, as it doesn't have access to the syntactic distinctions which seem necessary to correctly discriminate between those sentences which violate island constraints, and those which do not.

1.2.2 Voice mismatches

Passive and active sentences (as in 8 and 9) are semantically equivalent, although clearly they differ syntactically.[3]

(8) Gorbachev could have released this information.

(9) This information could have been released by Gorbachev.

[3]Following standard terminology, semantic equivalence boils down to identity of truth conditions. Thus, to claim that passives and actives are semantically equivalent is to claim that a passive sentence cannot be true while its active counterpart is false, or vice versa. In the early days of generative syntax, 'passivization' was a canonical example of a meaning preserving syntactic transformation. Note that it is perfectly reasonable to hold that actives and passives are semantically equivalent, while maintaining that they are information structurally distinct.

As we have here syntactic distinctness, but semantic identity, if ellipsis is sensitive to the distinction between active and passive sentences, the pro-form theory would be at a loss to explain this. Although voice mismatches in verb phrase ellipsis (as in 10, from Hardt (1993)) are not exactly the clear cases upon which linguistic theories should be based (Chomsky, 1956), there is psycholinguistic (Kobele et al., 2008) and corpus (Hardt, 1993) evidence which provide independent confirmation of their grammaticality.[4,5]

(10) This information could have been released by Gorbachev, although he decided not to.

And so at first glance, it would seem that the facts favor the pro-form theory. However, closer inspection reveals this not to be the case. As noted by Merchant (2007, 2008), neither sluicing (11, 12) nor (psuedo-)gapping (13, 14, 15, 16) allows for voice mismatches between antecedent and ellided VP.

(11) *Someone released the information, but I don't know by whom.

(12) The information was released (by someone), but I don't know by whom.

Sentence 11 has an active antecedent (*someone released the information*), and a passive ellided VP (*by whom the information was released*). In sentence 12, we see both that passive ellided VPs are acceptable when their antecedents are passive, and that prepositional phrases (*by whom*) are acceptable even without an explicit antecedent, thus suggesting that the reason for the unacceptability of 11 stems indeed from the voice mismatch.

(13) *Mary was praised by Adam, and Jesús, Susan.

(14) Adam praised Mary, and Jesús, Susan.

(15) *Adam praised Mary, and Susan, by Jesús.

[4] Recall that a sentence is grammatical if and only if it is generated by our theory of grammar, and that it is acceptable if and only if people like it. These properties needn't be coextensional: center-embedded sentences (such as *The boy the girl the cat the dog chased licked likes died*) are frequently claimed to be, although unacceptable, grammatical. Arregui et al. (2006) claim that sentences like 10 are, although acceptable, ungrammatical.

[5] To allay any confusion, this is just a convenient way of saying that if we decide to treat voice mismatches in verb phrase ellipsis as ungrammatical (and therefore write our grammars in such a way as to rule them out), their corpus-attestedness and relative acceptability would be tedious to account for. Arregui et al. (2006) start with the assumption that voice mismatches are ungrammatical (i.e. not generated by the grammar), and are led to introduce various complicated psycholinguistic 'repair rules' to explain why people find voice mismatches relatively acceptable. Note that their account is not able to explain why people ever produce such sentences in the first place.

(16) Mary was praised by Adam, and Susan, by Jesús.

In the unacceptable 13, the antecedent clause is passive, and the ellipsis clause active. Sentence 14 shows that the unacceptability of 13 is not due to a ban against the ellipsis clause being in the active voice. Sentences 15 and 16 show the same for passive ellipsis clauses.

The fact that purely syntactic (i.e. non-semantic) distinctions can be relevant in determining the well-formedness of elliptical sentences argues against not only the pro-form theory, but also the deletion under semantic identity theory.

1.2.3 Conclusions

We have seen two examples (island effects and voice mismatches) where purely syntactic properties internal to the ellipsis site are crucial in determining the grammaticality of their containing sentences. Others are presented in Merchant (2001) and Chung (2006). Only the deletion under syntactic identity theory is currently capable of correctly accounting for the patterns of grammaticality assumed above. It may well not be the whole story, but, as we have seen, it must be part of it. And so, in the remainder of this paper, I will see exactly how far we can get with just the deletion under syntactic identity theory, in its most restrictive form.

2 On syntactic identity

I propose to take seriously the idea that 'deletion under syntactic identity' is licensed by *exact* syntactic identity. As I intend it here, syntactic structure is what is constructed during the derivational process, and, in line with the tenets of survive minimalism, 'destructive' operations are not allowed (this is also known as the non-tampering condition). This has the consequence that operations like *trace conversion* (Sauerland, 1998; Fox, 1999), which changes syntactic copies to (something like) traces, are not permitted. Indeed, given the copy theory of movement, and the non-tampering condition, we are led immediately to the following conclusion:

> If a syntactic object SO_1 is elided under identity with SO_2, then SO_1 and SO_2 have been derived in *exactly the same way*

It is important to note that, without the ability to alter already built structure, syntactic identity boils down to being derived in the same way, or *derivational identity*.[6] Shifting our perspective from identity of structure, to identity of the processes which construct this structure allows us to conceptualize deletion under syntactic identity in such a way as to render it

[6] As far as I am aware, the view that 'syntactic identity' should be understood as derivational identity was first articulated by Lees (1960).

compatible with another, even more fundamental, tenet of survive minimalism, which is that operations which search through already created structure should be banned. Derivational identity can be checked while still respecting the principles of survive minimalism simply by building identical expressions simultaneously—by synchronizing derivations (see Kobele (2006)).

Most importantly, viewing syntactic identity in terms of derivational identity helps us determine the right question to ask when we are confronted with a case of ellipsis which seems to fly in the face of our current analysis. Given that we can't change the structure we have built up (by applying a destructive operation), we must instead revise our analysis so as to allow us to directly construct the right structure.[7] This has a number of interesting methodological consequences, which I will talk more about later, among them that, as derivational identity can be formulated independently of any particular syntactic theory, ellipsis can be used to decide between different analyses of a particular phenomenon written in different frameworks.

Although simple, and for that reason appealing, the hypothesis that ellipsis is deletion under derivational identity is simply a programmatic assumption, and needs to be shown worthwhile to be interesting. Accordingly, in the remainder of this section I show how the derivational identity hypothesis (DIH) can be fruitfully applied even to complicated data. Before we begin, a note is in order. The DIH is a methodological assumption, and is not the type of thing which can be true or false. It can be only fruitful, in which case we should continue to use it, or not, in which case we should abandon it. It is my goal to show that it is indeed a fruitful assumption.

2.1 Derivational identity vs data

Although the DIH cannot bear a truth value, it *can* be *incompatible* with other assumptions, such as those of survive minimalism, Tree-Adjoining Grammar (TAG), or Combinatory Categorial Grammar (CCG). It is, in this sense, like a constituency test, which, as a way to interpret data, cannot possibly be true or false, although it certainly *can* be incompatible with a particular analysis of that data (whence its great value).

[7] This is similar to the *deforestation* program transform in the functional programming literature (Wadler, 1990). Deforestation is a general technique that allows two functions g and f that apply in serial to be interleaved. This allows the input i to be directly mapped to the output $g(f(i))$ without first building an intermediate representation $f(i)$ (as the representations in question are usually trees, deforestation amounts to getting rid of intermediate trees, whence the name). In our case, we have a derivation which results in a structure d, and then a destructive operation *trace-conv*, which applies to this structure to yield a structure *trace-conv*(d) which is appropriate for syntactic identity. What we are doing, then, is attempting to find an alternative way of deriving the sentence in question, so that we directly arrive at this desired structure; the mechanism proposed later on can be viewed as performing the trace-conversion derivationally.

2.1.1 The reality of the verb phrase

Consider sentence 17, in which the verb and object of the second conjunct are deleted.

(17) Jesús will praise Mary, but Adam won't ~~praise Mary~~.

We have two analytical options. First, we can assume that there were two independent applications of a deletion operation, one targeting the verb, and one targeting the object. Second, we can assume that both elements were deleted in one fell swoop, by a single application of the deletion mechanism. These options are schematized in figure 1. As a general rule, we will

S ~~V~~ ~~O~~ S ~~V O~~

Figure 1: Analytical options

prefer analyses that involve fewer applications of the deletion mechanism. This is not forced upon us by anything, but seems reasonable to assume as a default for the following two reasons. First, although we are currently working on building a theory of *how* ellipsis works, we ultimately hope to extend it to a theory of *when* ellipsis is licensed. If we analyze the sentences above as involving independent applications of deletion to the verb and to the object, it seems like it will be difficult to rule out a sentence like 18 below, which also involves two applications of deletion.

(18) *Jesús will praise Mary, but Adam will ~~praise~~ the man who loves ~~Mary~~.

Second, if we don't make this assumption (at least as a default), we can't do anything interesting.

Preferring, then, to assume as few applications of deletion as possible, we are led to the conclusion that the sentences above (and more generally, transitive sentences) can be derived so that the verb and object form a constituent to the exclusion of tense, and the subject. Note that we are *not* entitled to conclude anything about what derivations are *not* available, only which ones are. Thus, we cannot conclude, for instance, that transitive sentences *always* are constructed with the verb and object coming together before the tense and subject.

As a consequence, any analysis of English according to which transitive sentences are built up exclusively from left to right (such as would seem to arise in a system like that proposed by Philips (2003)) would be incompatible with the interpretation of the above data forced upon us by the DIH. More interestingly, analyses such as Steedman's [2000] CCG analysis of English which, although it allows the verb and object to be derivational constituents some of the time, disallows it in cases like 19, are ruled out by the DIH.

(19) Jalapeños, Jesús hates. Adam does ~~hate jalapeños~~, too.

Passive is Phrasal, not (sentential or lexical) Turning now to the voice mismatches in VP-ellipsis which had once been interpreted as an argument against a syntactic account of ellipsis (examples are given as 20 and 21), we see that the DIH leads us to the conclusion that passivization is a VP level process. In other words, whatever structural differences there may be between active and passive sentences, the derivation of a sentence begins in the same way, regardless of its voice, up to at least the point where the object is merged.[8]

(20) This information could have been released by Gorbachev, although he decided not to ~~release this information~~.

(21) The children asked to be squirted with the hose, so we did ~~squirt the children with the hose~~.

Unlike the previous conclusion (that verb and object may be derivational constituents), the present one places us on one side of an important theoretical disagreement, ruling out nearly all current analyses of English passivization. In particular, all analyses that have been written in the TAG framework are incompatible with this conclusion, as are all previous CCG analyses, as in both of these grammar formalisms analyses of passivization have been uniformly lexical. The kind of analysis of sentences like 20 and 21 forced upon us by the DIH is given in figure 2. Note that the very same

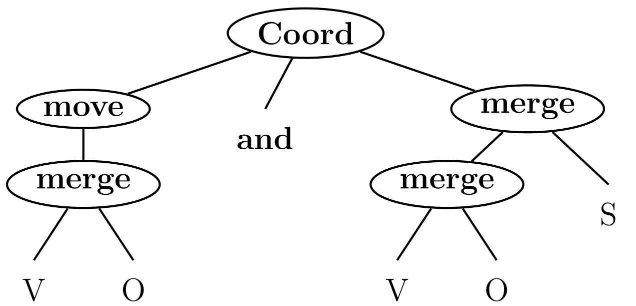

Figure 2: The gross structure of voice mismatches

conclusion (that passivization applies to verb phrases) has been argued for independently on very different grounds (Bach, 1980; Keenan, 1980).

2.1.2 A paradox?

We have seen above that, far from being incompatible with a syntactic approach to the identity condition in ellipsis, voice mismatches (together

[8]Note as well that, given the morphological differences between the passive and active form of the verbs above, the DIH is incompatible with the position that words come into syntax fully inflected (Chomsky, 1995), and, more generally, with any grammatical organization in which the output of morphology is the input to syntax.

with the DIH) provide us with a powerful argument for a transformational approach to voice, leading us to the position that actives and passives have a common initial subderivation (as in figure 3). That this cannot be the

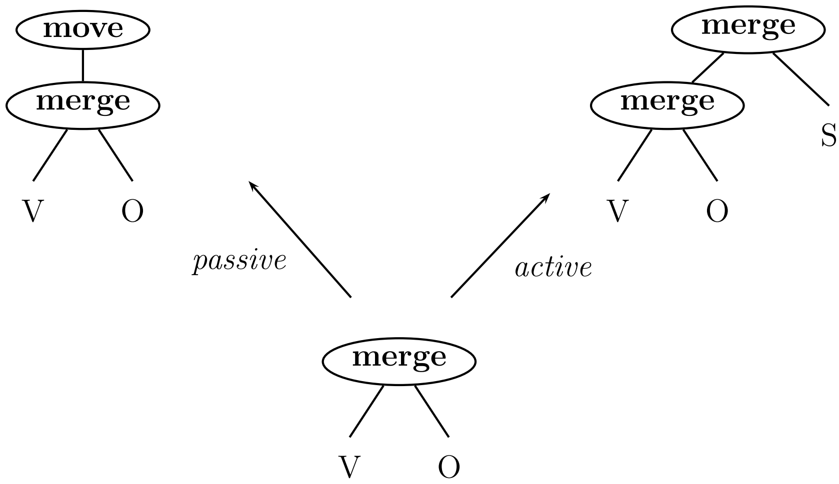

Figure 3: The relation between actives and passives

entire story is confirmed by the fact that passives can antecede passives, as in the sentences below.[9]

(22) Mary was praised, and Susan was ~~praised~~, too.

(23) Mary doesn't seem to have been praised, but Susan does ~~seem to have been praised~~.

Although sentence 22 could be treated as simple deletion of the verb (although non-trivial assumptions about morphology would have to be made), in sentence 23 the entire verb plus non-finite complement is deleted. According to the DIH, because the surface subjects *Mary* and *Susan* are different, they cannot be present in the derivation at the point where ellipsis occurs. In other words, we need a second structure for passives, as shown in figure 4. Although the structure on the left in figure 4 depicts the derivation that we expect to have given current (default) assumptions about the nature of passivization made in the context of government and binding theory, and its successors, the structure on the right is more mysterious. Still, it is not

[9]Matched-voice ellipsis is by far more acceptable than mismatched voice ellipsis (Tanenhaus and Carlson, 1990; Arregui et al., 2006; Kobele et al., 2008). Thus, even if we decide to remove the mismatched examples from the purview of syntactic theory (i.e. call them ungrammatical, and explain their attestedness some other way), we are left with the same problem, as active-active ellipsis forces us to assume that transitive sentences can be constructed with a derivational VP, and passive-passive ellipsis that they can be constructed without a derivational VP. Note that this is precisely the problem that led Sag (1976) to abandon the deep structure identity condition of his contemporaries. The solution proposed in this paper can be thought of as a syntactic reimplementation of his.

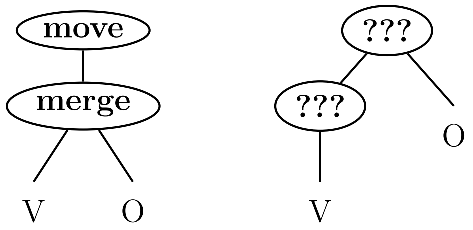

Figure 4: Two derivations of passive sentences

completely without precedent in the minimalist community, something like it having been proposed by Manzini and Roussou (2000) in the context of A-movement. Slightly farther afield, the structure on the right is precisely the kind of derivation one would expect, if we had slash-feature percolation instead of movement. In the remainder of this paper, I will present a system which allows for both kinds of passive derivation demanded by the DIH.

3 (Survive) Minimalism

As presented in Stroik (2008), survive minimalism departs from more traditional variants of minimalism primarily in that movement is viewed not as being driven by the attractor, but rather by the moving object itself. This means that it is known whether an expression will stay in its current position as soon as the expression is merged there. This allows Stroik to adopt a strong 'no-tampering' condition on syntactic objects, treating them as black boxes which are opaque to search and manipulation. Intuitively, whereas the 'standard' minimalist would derive sentence 24 along the lines of 25, necessitating a search for a goal with features matching the probe in C (25.4), the survive minimalist would already have prepared the 'goal' for movement as soon as it was merged (26.1), and have it on-hand for remerge into SPEC-CP as soon as the C head is introduced (26.4).

(24) What did Jesús eat?

(25)
1. [eat what$_{wh}$] (merge eat and what)
2. [Jesús [eat what$_{wh}$]] (merge Jesús)
3. [C$_{wh*}$ [Jesús [eat what$_{wh}$]]] (merge C)
4. [what$_{wh}$ C$_{wh}$ [Jesús [eat ~~what$_{wh}$~~]]] (probe for and remerge what)

(26)
1. [eat ~~what~~]; what$_{wh*}$ (merge eat and what; what 'survives')
2. [Jesús [eat ~~what~~]]; what$_{wh*}$ (merge Jesús)
3. [C$_{wh}$ [Jesús [eat ~~what~~]]]; what$_{wh*}$ (merge C)

4. [what$_{wh}$ C_{wh} [Jesús [eat ~~what~~]]] (remerge what)

Because we will be here concerned very much with the structure of derivations, and little differences will matter a great deal, we will have to flesh out a theory of grammar well beyond what is usually done in Stroik's work and elsewhere. For this purpose, we will use minimalist grammars (MGs), introduced in Stabler (1997) as a working out of some of the core ideas in Chomsky (1995).[10] There is an interesting connection between minimalist grammars and the intuitions in Stroik (2008). Although Stabler's original presentation of minimalist grammars was in terms of 'destructive' operations on trees (as in the standard picture on derivations outlined in 25), it was proven in Michaelis (2001) that we could adopt a perspective on derivations similar to the one in 26 without changing any of the properties of the formalism. In other words, MGs can be thought of as a precise variant of survive minimalism.

3.1 Derivations

There are two basic operations in minimalist grammars; one that combines two structures, which we will call **merge**, and one that operates over a single structure, which will be called **move**. Entire derivations can be represented as single trees. Given three lexical items, A, B, and C, the following derivation can be represented by means of the single derivation tree in figure 5.

1. $\mathbf{merge}(A, B) = [_A\ A\ B]$
2. $\mathbf{move}([_A\ A\ B]) = [_A\ B\ [_A\ A\ t_B]]$
3. $\mathbf{merge}(C, [_A\ B\ [_A\ A\ t_B]]) = [_C\ C\ [_A\ B\ [_A\ A\ t_B]]]$

In the case of the **merge** operation, we adopt the convention that the first of its two arguments is the one which projects over the other. Although the

[10]One advantage of working within such a formalism is that everything—every mechanism, derivational step, and data structure—is perfectly explicit (it is a *generative grammar*, in the sense of Chomsky (1965)). Only with this kind of precision is the theoretician able to demonstrate both why a particular sentence is derivable in his or her system, and also (a point too often neglected) that a particular sentence is *not* derivable. Furthermore, it allows for construction of learning algorithms (Stabler et al., 2003), parsing algorithms (Harkema, 2001), and serious psycholinguistic processing models (Hale, 2003), as well as meaningful comparisons with other grammar formalisms (Harkema, 2001; Michaelis, 2001; Kobele, 2002; Stabler, 2003).

The fact that everything is completely worked out does not imply that it is set in stone. Frey and Gärtner (2002) introduce asymmetric feature checking to account for scrambling and adjunction, Gärtner and Michaelis (2007) study the result of adding various locality constraints to the formalism, Kobele (2005) studies a certain kind of feature percolation, Stabler (2006) investigates a version of sideward movement, and Kobele (2002) investigates Brody-style mirroring (Brody, 2000).

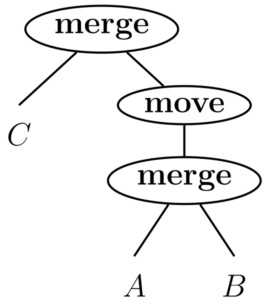

Figure 5: Representing derivations with trees

concept of a **numeration** will play no role in the theory developed here, it can be straightforwardly defined in terms of the **yield** of derivation trees.[11]

3.2 Features

As the generating functions **merge** and **move** are taken to be universal and invariant, differences between languages reside in the lexicon. Lexical items have various features, which determine how they behave in the derivation. In addition to movement being feature driven, I will assume that merger is as well, and that, for ease of presentation, the kinds of features which are relevant for the **merge** and the **move** operations are distinct, and will be called **selection** and **licensing** features, respectively.[12] Each feature has an **attractor** and an **attractee** variant (figure 6), and these must match in order for an operation to apply. Each time an operation is applied, it checks

	attractor	attractee
merge	`=x`	`x`
move	`+y`	`-y`

Figure 6: Features

both an attractor and an attractee feature, of the appropriate variety.

Although the features used here will be suggestively named (such as `V` for VP, `v` for little-vP, `n` for NP, and so on), the features themselves have no content apart from that which is given to them by the role they play in the formal system. In other words, in a minimalist lexicon, all that matters is

[11] By a result of Hale and Stabler (2005), in the context of minimalist grammars a string of lexical items uniquely determines a derivation tree.

[12] Treating syntactic features as falling into these two distinct groups is really for simplicity only. (It is simpler in the sense that it cuts down on the bookkeeping needed to ensure that lexical items that shouldn't interact don't.) We could just as well treat there as being only one kind of feature relevant for both **merge** and **move** operations (say, `*z` and `z`) without changing the essential formal properties of the system.

which features match which others, and not what they are called.[13] We will also not attempt here to model inflection or agreement, which is perhaps best seen as a side-effect of formal feature checking. The hope is, of course, that the (too) simple system studied here will synthesize seamlessly with more sophisticated accounts of agreement.

3.3 Lexical Items

Syntax relates form and meaning. Lexical items are the building blocks of this relation, atomic pairings of form and meaning, along with the syntactic information necessary to specify the distribution of these elements in more complex expressions. Here, simplifying somewhat, we take lexical items to be pairings of abstract lexemes such as dog, cat, bank_1,... with feature bundles. Feature bundles are not unstructured sets, but are rather ordered, so that some features can be available for checking only after others have been checked. We will represent feature bundles as lists, and the currently accessible feature is at the beginning (leftmost) position of the list. An example lexical item is shown in figure 7. Its feature bundle '`=d V`' indicates that it first selects a DP argument, and then can be selected for as a VP.[14] There

$$\langle \mathsf{praise}, \texttt{=d V} \rangle$$

Figure 7: A lexical entry for *praise*

are a number of advantages of treating feature bundles as lists, instead of as sets. First, certain lexical items, such as the Saxon genitive `'s`, are naturally thought of as selecting two syntactic arguments, an NP complement and a DP specifier. As the notions of 'complement' and of 'specifier' reduce to 'first-merged' and 'not-first-merged' in the context of the minimalist program, we need a way of ensuring that the first merged argument of `'s` is the NP, and not the DP. This not being obviously derivable from anything else, here we can simply structure the feature bundle for `'s` so as to have the noun phrase selected before the determiner phrase.[15] A second advantage

[13] It is interesting, in this context, to think of how to argue that a feature in one language should bear the same name as a feature in another language. Intra-sentential code-switching, as implemented in minimalist grammars by MacSwan (2000), is one way of imbuing this question with determinate empirical content.

[14] More specifically, in order to be checked, its first feature `=d` requires that it be merged with an expression which has a matching attractee feature, `d`. After this, the resulting feature bundle will have first feature `V`, in order to check which, the expression must be merged with another which has the matching attractor feature, `=V`, as its first feature.

[15] Another option is to have the semantics take over responsibility for this distinction, perhaps by assigning to `'s` the meaning s of type $(eet)((et)t)(et)t$, such that $\mathsf{s}(R)(Q)(B) = Q(R^{-1}[B])$, where $R^{-1}[B] := \{a : \exists b \in B.\ aRb\}$, and attributing the unacceptable **doctor's every boy* to a semantic type mismatch. The problem with this is that we no longer have a direct description of the well-formed sound-meaning pairs; they are now

is that we can state directly that (for example) the case feature of a DP must be checked before its *wh*-feature.

3.4 Syntactic Objects

I am writing lexical items using the notation $\langle\alpha, \delta\rangle$, where α is a lexeme (such as praise), and δ is a feature bundle (such as '=d V'). Complex expressions I will write using a labeled bracket notation, as per the following:

$$[_\delta\ \alpha\ \beta]$$

The above expression has feature bundle δ. As an example, if we assign the lexical entries ⟨'s, =n =D D⟩ and ⟨brother, n⟩ to the Saxon genitive and to brother respectively, then the complex expression 's brother, which is the result of merging brother as the complement of 's, is represented as the below.

$$[_{\texttt{=D D}}\ \text{'s brother}]$$

As can be seen, the above expression has feature bundle '=D D', which means that after it merges with an expression with first feature D, it will itself be such an expression.

$$[_{\texttt{D}}\ \text{Jesús [}\text{'s brother]]}$$

By representing the relevant syntactic information about an expression in its feature bundle, we technically no longer need to make constituency distinctions, and can (and sometimes for convenience will) write the above expression as per the following.

$$[_{\texttt{D}}\ \text{Jesús 's brother}]$$

Moving expressions, that is, expressions which have 'survived' from earlier merger, will be written after a semi-colon following the 'main' expression. For example, the expression below is a sentence (IP) which contains a *wh*-phrase which has not yet checked its -wh feature.

$$[_{\texttt{i}}\ \text{Jesús will praise}];\ [_{\texttt{-wh}}\ \text{who}]$$

3.5 Merge

The **merge** operation applies to two arguments, A and B, resulting in the new object $A + B$, just in case the head of A has some selector feature =x as the first unchecked feature in its feature bundle, and the head of B has

defined indirectly as the result of applying a partial function to a set. (The desire to have a direct description of the well-formed sound-meaning pairs is expressed in the guise of *crash-proof syntax* by Frampton and Gutmann (2002).) Kracht (2007) has argued in this vein that constituency (order of application) is a fundamentally syntactic notion, and should be excised from the semantics.

the corresponding `x` as the first unchecked feature in its feature bundle. In the resulting $A + B$, both first features used in this derivational step are checked, making available the next features of both feature bundles. There are three cases of the **merge** operation, depending first on whether the B argument will continue moving, and then, if not, whether B will surface as a specifier or as a complement of A (figure 8). The first case corresponds

$$\mathbf{merge}(A, B) = \begin{cases} A;\ B & B \text{ survives} \\ [_A\ A\ B] & B \text{ is a complement} \\ [_A\ B\ A] & B \text{ is a specifier} \end{cases}$$

Figure 8: Cases of **merge**

to the operation **Survive!** in survive minimalism. Let us assume that all DPs move for case, which we will here represent with an abstract feature '`K`', the attractor version of which is `+k`, and the attractee version is `-k`. Then, provisionally, we assign the feature bundle '`d -k`' to DPs like Jesús, Adam, Mary,... Now, whenever a DP is selected by an expression (with first feature `=d`), we know that it will not surface in that position, as it still has a case feature that must be checked (via movement to some later specifier position). Consider the derivation step below, in which praise selects Jesús as its complement.

$$\mathbf{merge}(\langle \mathsf{praise}, \texttt{=d V}\rangle, \langle \mathsf{Jesús}, \texttt{d -k}\rangle) = [_{\texttt{V}}\ \mathsf{praise}\ t_{Jesús}]\,;\,[_{\texttt{-k}}\ \mathsf{Jesús}]$$

Because Jesús still needs to move for case, it 'survives' by not being trapped within the main expression. As Kobele (2006) shows how to directly model-theoretically interpret each derivational step, obviating thereby any semantic need for traces, I will usually leave out explicit mention of the trace in the expression above, writing instead:

$$\mathbf{merge}(\langle \mathsf{praise}, \texttt{=d V}\rangle, \langle \mathsf{Jesús}, \texttt{d -k}\rangle) = [_{\texttt{V}}\ \mathsf{praise}]\,;\,[_{\texttt{-k}}\ \mathsf{Jesús}]$$

Important to note is the fact that in the resulting expression, both matching features of the arguments (`=d` and `d`) are checked (which for present purposes means gotten rid of).

The next case of **merge** is the merger of a complement, or 'first-merge'. Let us represent the head assigning case to the object in English (AgrO) with the following lexical item: $\langle$-ϵ, `=V +k agrO`$\rangle$.[16] Because the first feature

[16] The symbol ϵ is the empty string. The hyphen in front of it indicates that it is a suffix, and thus triggers head movement; its complement's head raises to it. Its feature bundle `=V +k agrO` indicates that it first merges with a VP (`=V`), then provides a case position for the object to move to (`+k`), and finally becomes itself selectable as an AgrOP (`agrO`).

in the feature bundle of this lexical item is =V, it can be merged with the VP *praise Jesús* we derived above. There are two things of note to watch out for here. First, one of the arguments to merge has ‘surviving’ expressions attached to it. These will simply continue to survive in the result. The same would be true if both arguments had surviving expressions attached to them. Second, the selecting lexical item is a suffix (marked by the hyphen preceding the lexeme). This triggers head movement from its complement.[17]

$$\textbf{merge}(\langle\text{-}\epsilon, \texttt{=V +k agrO}\rangle, [_{\texttt{V}}\ \mathsf{praise}]\,;[_{\texttt{-k}}\ \mathsf{Jes\acute{u}s}])$$
$$= \left[_{\texttt{+k agrO}}\ \mathsf{praise}\text{-}\epsilon\ \left[\ t_{praise}\right]\right]\,;[_{\texttt{-k}}\ \mathsf{Jes\acute{u}s}]$$

head movement

Leaving out the traces, and unnecessary internal structure, this expression can be abbreviated as the below.

$$[_{\texttt{+k agrO}}\ \mathsf{praise}\text{-}\epsilon]\,;[_{\texttt{-k}}\ \mathsf{Jes\acute{u}s}]$$

Note again that both of the matching first features of the arguments to **merge** (=V and V) have been checked in (i.e. deleted from) the result.

The last case of **merge** is merger of a specifier into its surface position. This happens whenever the first argument to **merge** is not a lexical item, and the second argument has no unchecked features (other than the matching selectee feature which drives the **merge** operation). This will not occur in any of the examples we will encounter in this paper.

3.6 Move

The **move** operation applies to a single syntactic object A just in case it contains a surviving expression B, and the first unchecked feature of A is +y and the first unchecked feature of B is -y. In order to rule out indeterminacy, **move** will only be defined if *exactly one* surviving expression begins with a matching -y feature.[18] Just as with the **merge** operation, **move** checks the matching features of the expression which it applies to. There are two cases, according to whether the moving element survives or not. As an example of the first case, consider the expression below (which

[17] See Stabler (2001) for more details. The basic idea is that head movement is treated as a quasi post-syntactic operation, along the lines of Matushansky (2006). In other words, head movement is not feature-driven movement.

[18] This is a radical version of the shortest move constraint (Chomsky, 1995), and will be called the SMC – it requires that an expression move to the first possible landing site. If there is competition for that landing site, the derivation crashes (because the losing expression will have to make a longer movement than absolutely necessary). Using this as a constraint on movement has desirable computational effects (such as guaranteeing efficient recognizability–see (Harkema, 2001; Michaelis, 2001)), although other constraints have been explored in Gärtner and Michaelis (2007), who show that adding constraints on movement can sometimes actually increase the expressivity of the formalism.

is similar to *praise Jesús* derived above, but with who having been merged instead of Jesús).

$$[_{\texttt{+k agrO}}\ \textsf{praise-}\epsilon]\,;[_{\texttt{-k -wh}}\ \textsf{who}]$$

The **move** operation applies to this expression, as the main expression has as its first feature a licensor +k, and there is exactly one surviving expression with first feature the matching licensee -k. In the result, because the moving expression still has features left to check (-wh), it survives.

$$\textbf{move}([_{\texttt{+k agrO}}\ \textsf{praise-}\epsilon]\,;[_{\texttt{-k -wh}}\ \textsf{who}]) = [_{\texttt{agrO}}\ t_{who}\ \ [\textsf{praise-}\epsilon]]\,;[_{\texttt{-wh}}\ \textsf{who}]$$

Again, leaving out traces and internal structure, we write the expression above as per the below. Note again that both features involved in the **move** operation (+k and -k) are deleted/checked in the result.

$$[_{\texttt{agrO}}\ \textsf{praise-}\epsilon]\,;[_{\texttt{-wh}}\ \textsf{who}]$$

The second case of the **move** operation is when the moving element is moving to what will be its surface position; in other words, its last feature being checked, it no longer needs to survive. The expression *praise Jesús* we derived earlier illustrates this.

$$\textbf{move}([_{\texttt{+k agrO}}\ \textsf{praise-}\epsilon]\,;[_{\texttt{-k}}\ \textsf{Jesús}]) = [_{\texttt{agrO}}\ \textsf{Jesús}\ \ [\textsf{praise-}\epsilon]]$$

Ignoring internal structure, the expression above becomes the below. This case of the **move** operation permits surviving expressions to become reintegrated with the main expression.

$$[_{\texttt{agrO}}\ \textsf{Jesús praise-}\epsilon]$$

3.6.1 Eliminating Movement

In the minimalist syntactic literature, there has been much interest in unifying the operations of **merge** and **move**. In a purely set theoretical sense (i.e. viewing functions as sets of pairs), this is easy to do: **merge** is a binary function and **move** is unary, their domains are therefore disjoint, and so their union **merge**∪**move** is also a function. Equally clear, however, is the fact that this technical trick 'should not count.' It seems that what is going wrong is that the complexity of description of the grammatical operations is not changed by this particular 'unification'—the easiest way to describe the function **merge** ∪ **move** remains as the union of the functions **merge** and **move**, not directly. Thus, the real question at issue is whether there is a way of describing the action of the generating functions according to which it is simpler to describe **merge** and **move** as special cases of a single function, rather than as operations in their own right. Another question which should then be asked is whether such a unified perspective on **merge** and **move** is simpler than the current, non-unified perspective.

Stroik (2008) has attempted to unify **merge** and **move** by making extensive use of a numeration (or ‘work space’); in other words, by treating the inputs to and outputs of the generating functions not as trees, but as (multi-)sets of trees (Chomsky, 1995). According to this perspective, ‘surviving’ expressions, which we have here written as part of a single expression, are reinserted into the work space. The **merge** operation is given an entire work space as its argument: it selects two objects therefrom to be merged, these are removed from the work space, and in their place are inserted the result of merging these objects, as well as any surviving expressions. ‘**Move**’ is simply what happens when a surviving expression is chosen to be merged.

Although this interpretation of movement eliminates it as an operation in its own right (it has been reduced to a special case of a much more complex presumably non-functional relation we continue to call **merge**), some mechanism must be in place to ensure that merger of surviving expressions results in well-formed chains.[19]

The expressions used in this paper, with their explicit record of which expressions survive within them, needn’t be thought of as concrete proposals of what grammatical objects look like. Instead, we can understand them as abstract representations not only of these objects, but also of whatever mechanism is used to ensure that only well-formed chains are built. In other words, when I write an expression like $[_{\texttt{agrO}}\ \textsf{praise}]$; $[_{-\texttt{wh}}\ \textsf{who}]$, I am indicating that we have an expression $[_{\texttt{agrO}}\ \textsf{praise}]$, which must be involved in a larger derivation which uses the surviving expression $[_{-\texttt{wh}}\ \textsf{who}]$ which is currently in the work space. Therefore, the present system is perfectly compatible with the survive minimalist’s reduction of **move** to a subcase of **merge**. It is however equally compatible with an alternative perspective on survival (also proposed in Stroik (2008)), according to which an expression survives by moving to adjoin to a projection of the current head of the tree, as well as with a host of other, equally valid yet heretofore undiscovered alternatives. As these concerns are orthogonal to the main point of this paper (which is to argue for adoption of the DIH), I continue to treat **merge** and **move** as different. The reader whom this discomforts should simply substitute each occurrence of the word **move** with the phrase “the subcase of **merge** where one of the two arguments is a surviving expression.”

4 Passivization in English

In this section, I will present a grammar for a fragment of English comprising passivization. This will serve simultaneously as an introduction to the

[19]‘Chains’ as used here is a descriptive term: the dependencies into which a single expression enters must be related to each other in a certain way. This, as a fact about language, must be ensured even by grammar formalisms which do not reify chains as grammatical objects.

minimalist grammar formalism (and the fragment-style of language analysis), as well as a reintegration with the theme of this chapter, of providing a grammatical analysis of English consistent with the demands of the DIH.

Analyses in the transformational tradition tend to assign similar derivational histories to semantically related construction types. Passives and actives are related in virtue of having been constructed in the same way up to a point (as per figure 3). It is precisely this perspective which we are pushed toward by the DIH.

The basic clause structure I will be assuming is as in figure 9. I will

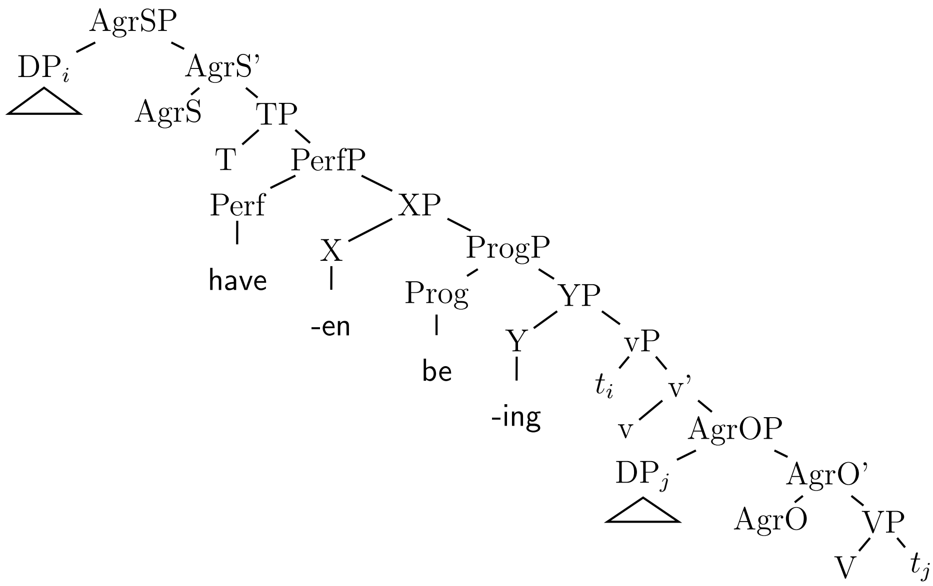

Figure 9: Basic clause-structure

assume that objects move overtly to an AgrO position located underneath the subject θ-position (Koizumi, 1995), which is introduced by little-v. This assumption is forced by the SMC constraint on the **move** operation: as all argument DPs will be assumed to have the feature bundle '`d -k`',[20] as soon as the subject is merged into its θ-position, it survives with feature bundle '`-k`'. If the object has not checked its case feature by that point, there will be two surviving expressions with the same first feature (`-k`), which is precisely the configuration ruled out by the SMC. We can derive sentences with this clause structure with the lexical items in figure 10. Using the lexical items in figure 10, we can derive sentences like *Adam will have been criticizing Jesús*, as shown below.[21]

[20] An alternative is to treat nominative and accusative cases as syntactically distinct, and then DPs might have either `-acc` or `-nom` features. The DIH actually argues against this alternative: as passives may antecede actives (as in 20), the DP inserted in a passive sentence must be identical to the DP inserted in an active one. Therefore, case differences must be left to the morphology.

[21] In fact, we can derive exactly eight sentences (four of which violate condition C of the binding theory).

$\langle \epsilon$, =t +k agrS$\rangle$	$\langle$will, =perf t$\rangle$	$\langle$have, =x perf$\rangle$	$\langle$be, =y prog$\rangle$
	$\langle$-ed, =perf t$\rangle$	$\langle$-en, =prog x$\rangle$	$\langle$-ing, =v y$\rangle$
$\langle$-ϵ, =agrO =d v$\rangle$	$\langle$-ϵ, =V +k agrO$\rangle$	$\langle$praise, =d V$\rangle$	$\langle$Jesús, d -k$\rangle$
		$\langle$criticize, =d V$\rangle$	$\langle$Adam, d -k$\rangle$

Figure 10: Lexical items for figure 9

We begin by merging together the lexical items criticize and Jesús. This is defined because the first features of the respective feature bundles are matching attractor (=d) and attractee (d) features. Because Jesús still has a feature to check (-k), it survives.

1. **merge**($\langle$criticize, =d V$\rangle$,$\langle$Jesús, d -k$\rangle$)

$$[_{\mathtt{V}} \text{ criticize } t_d] \,;\, [_{\mathtt{-k}} \text{ Jesús}]$$

The next step is to merge the lexical item $\langle$-ϵ, =V +k agrO$\rangle$ with the expression just derived. Because it is a suffix, it triggers the head of its complement (here: *praise*) to raise up to it.

2. **merge**($\langle$-ϵ, =V +k agrO$\rangle$,1)

$$[_{\mathtt{+k\ agrO}} \text{ criticize-}\epsilon \ [\ t_V\ t_d]] \,;\, [_{\mathtt{-k}} \text{ Jesús}]$$

The first unchecked feature of the above expression is the attractor feature +k, which is checked by movement of the surviving expression *Jesús*, whose first feature is the matching attractee feature -k.

3. **move**(2)

$$[_{\mathtt{agrO}} \text{ Jesús } [\text{ criticize-}\epsilon\ [\ t_V\ t_d]]]$$

Next little-v is merged, triggering head movement of *criticize*.

4. **merge**($\langle$-ϵ, =agrO =d v$\rangle$,3)

$$[_{\mathtt{=d\ v}} \text{ criticize-}\epsilon\text{-}\epsilon\ [\text{Jesús } [\ t_{agrO}\ [\ t_V\ t_d]]]]$$

Next the external argument, Adam, is merged, checking the =d feature of little-v. Because Adam has an unchecked case-feature, it survives.

5. **merge**(4,$\langle$Adam, d -k$\rangle$)

$$[_{\mathtt{v}}\ t_d\ [\text{criticize-}\epsilon\text{-}\epsilon\ [\text{Jesús } [\ t_{agrO}\ [\ t_V\ t_d]]]]] \,;\, [_{\mathtt{-k}} \text{ Adam}]$$

In the next five steps, the tense and aspect heads are merged, triggering head movement when applicable.

6. **merge**(⟨-ing, =v y⟩,5)

$$[_{\mathtt{y}}\ \text{criticize-}\epsilon\text{-}\epsilon\text{-ing}\ [t_d\ [t_v\ [\text{Jesús}\ [\ t_{agrO}\ [\ t_V\ t_d]]]]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

7. **merge**(⟨be, =y prog⟩,6)

$$[_{\mathtt{prog}}\ \text{be}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing}\ [t_d\ [t_v\ [\text{Jesús}\ [\ t_{agrO}\ [\ t_V\ t_d]]]]]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$
$$\rightsquigarrow [_{\mathtt{prog}}\ \text{be}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

8. **merge**(⟨-en, =prog x⟩,7)

$$[_{\mathtt{x}}\ \text{be-en}\ [t_{prog}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

9. **merge**(⟨have, =x perf⟩,8)

$$[_{\mathtt{perf}}\ \text{have}\ [\text{be-en}\ [t_{prog}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

10. **merge**(⟨will, =perf t⟩,9)

$$[_{\mathtt{t}}\ \text{will}\ [\text{have}\ [\text{be-en}\ [t_{prog}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

Finally, we merge agrS, which allows the case feature of *Adam* to be checked.

11. **merge**(⟨ϵ, =t +k agrS⟩,10)

$$[_{\mathtt{+k\ agrS}}\ \epsilon\ [\text{will}\ [\text{have}\ [\text{be-en}\ [t_{prog}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]]]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

12. **move**(11)

$$[_{\mathtt{agrS}}\ \text{Adam}\ [\epsilon\ [\text{will}\ [\text{have}\ [\text{be-en}\ [t_{prog}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]]]]]]$$

The entire derivation of this sentence can be compactly represented as a single tree, shown in figure 11. In order to allow for the optionality of *be -ing* and *have -en*, we need to explicitly introduce lexical items which allow for vP to be selected by XP, and ProgP by TP, respectively. This can be done with the lexical items below.

⟨-ϵ, =v prog⟩ ⟨-ϵ, =prog perf⟩

With these additions to our lexicon from figure 10, we can derive sentences without either *be -ing*, *have -en*, or both. To derive the sentence *Adam will be criticizing Jesús* (i.e., our previous sentence without *have -en*), instead of steps 8 and 9, we substitute 8′ below.

8′. **merge**(⟨-ϵ, =prog perf⟩,7)

$$[_{\mathtt{perf}}\ \text{be-}\epsilon\ [t_{prog}\ [\text{criticize-}\epsilon\text{-}\epsilon\text{-ing Jesús}]]]\,;[_{\mathtt{-k}}\ \text{Adam}]$$

The derivation can then continue with 10–12. Note that the expression in 8′ has exactly the same featural make-up as the expression in 9 (the main expression has the feature bundle '`perf`', and the single surviving expression has the feature bundle '`-k`').

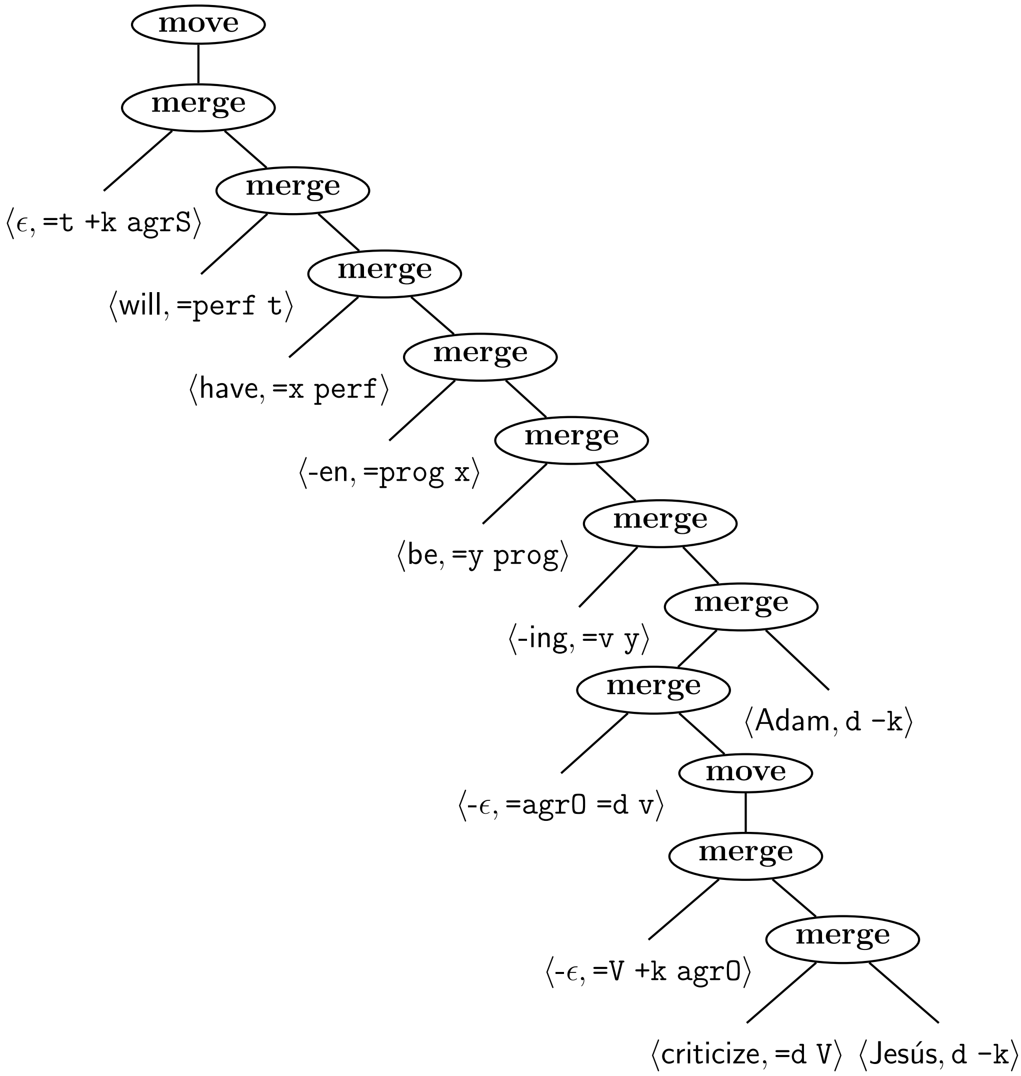

Figure 11: A derivation tree for *Adam will have been criticizing Jesús*

4.1 The Passive

One major point of difference between passives (28) and actives (27) in English is that in the passive, the external argument is not required to be present (if it appears at all, it is in an optional *by*-phrase), and that the internal argument raises to the subject position and assumes all the canonical subject properties, such as nominative case.

(27) Adam was criticizing Jesús.

(28) Jesús was being criticized.

In the context of our present analysis (i.e. the lexicon in figure 10), objectual case is assigned in AgrO, and the external argument introduced in v. In order to block both case assignment to the object and assignment of an external θ role (selection of the subject), we simply allow lexical items particular to the passive to select the VP, do their thing, and then result in a vP. We introduce the two new lexical items in figure 12. The first,

$$\langle \textsf{-en}, \texttt{=V z}\rangle \quad \langle \textsf{be}, \texttt{=z v}\rangle$$

Figure 12: Lexical items for the passive

$\langle \textsf{-en}, \texttt{=V z}\rangle$, selects a VP, triggers head movement of the verb (causing it to appear as a participle), and then returns a ZP.

1′. **merge**($\langle \textsf{-en}, \texttt{=V z}\rangle$,$[_{\texttt{V}}$ criticize $t_d]$;$[_{\texttt{-k}}$ Jesús$]$)

$$[_{\texttt{x}}\ \textsf{criticize-en}\ \ [t_V\ t_d]]\ ;\ [_{\texttt{-k}}\ \textsf{Jesús}]$$

The next, $\langle \textsf{be}, \texttt{=z v}\rangle$, selects a passive participle phrase (ZP), and returns a vP.

2′. **merge**($\langle \textsf{be}, \texttt{=z v}\rangle$,1′)

$$[_{\texttt{v}}\ \textsf{be}\ \ [\textsf{criticize-en}\ \ [t_V\ t_d]]]\ ;\ [_{\texttt{-k}}\ \textsf{Jesús}]$$

The resulting expression (2′) has the same featural make-up as the expression in 5 (the main expression has the feature bundle '`v`', and the only surviving expression the feature bundle '`-k`'), and thus is syntactically interchangeable with it. The derivation of the sentence *Jesús was criticized* is as in figure 13. Our current fragment can define passives and actives of sentences with various tenses and aspects. While not particularly exciting, it will suffice for the purposes of this paper.[22] We now return to these.

5 Satisfying the DIH

The derivational identity hypothesis, in response to sentences 20 and 21, forced upon us the conclusion that actives and passives must have a derivational constituent in common. Comparing the derivations of active and passive sentences shown in figures 11 and 13 respectively, we see that they share the subderivation shown in figure 14. Ignoring for the moment that our fragment isn't expressive enough to generate sentences like (the unelided version of) 20 (we do not have modal auxiliaries *could*, nor control verbs *decide*), if we render phonologically null just the lexical items shown in figure 14 in 29, we obtain a sentence parallel to 20.

[22] Kobele (2006), from which these lexical items are drawn more or less verbatim, shows how this fragment can be extended naturally to account for raising to subject and object, control, expletive-*it*, and quantifier scope ambiguities.

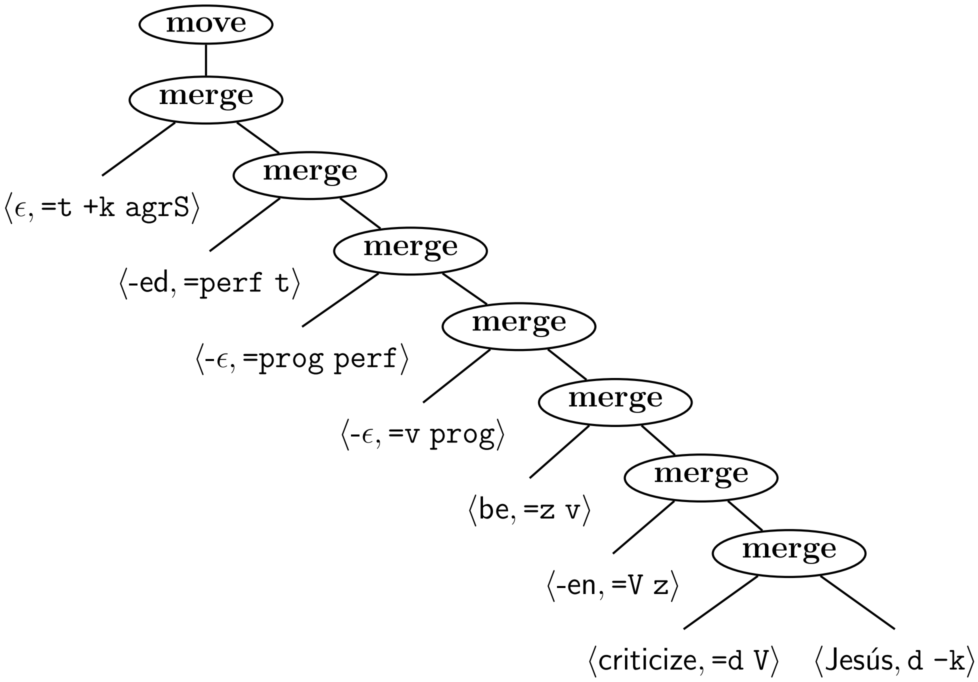

Figure 13: A derivation tree for *Jesús was criticized*

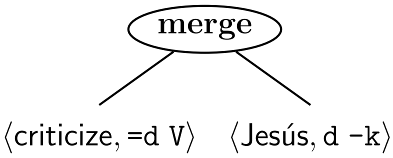

Figure 14: The common subderivation of figures 11 and 13

(29) Jesús could have been criticized, but Adam decided not to ~~criticize~~-ϵ-ϵ ~~Jesús~~.

Allowing for deletion under derivational identity at the VP level, we can generate the dialogues given in 30 and 31 (extending our fragment in the obvious way for proper names). Both B responses (they are the same) should be read as *Adam did.*

(30) A: Mary criticize-ϵ-ϵ-ed Jesús.

B: (No,) Adam ~~criticize~~-ϵ-ϵ-ed ~~Jesús~~.

(31) A: Jesús be-ϵ-ϵ-ed criticize-en.

B: (Yes,) Adam ~~criticize~~-ϵ-ϵ-ed ~~Jesús~~.

Thus, it seems that minimalist-style analyses support the kind of derivational constituents demanded by the DIH. However, they are not able to deliver *all* of the requisite derivational structure, even when we limit ourselves to the relatively simple case of voice. As seen in examples 22 and 23 (repeated below as 32 and 33), passive verb phrases can be elided, despite having different internal arguments.

(32) Mary was praised, and Susan was ~~praised~~, too.

(33) Mary doesn't seem to have been praised, but Susan does ~~seem to have been praised~~.

The problem is that according to the present analysis, the largest derivational constituent shared by both antecedent and elliptical clause is the lexical item ⟨praise, =d V⟩. Looking at figure 15 (which, substituting Mary and Susan, respectively, for the blank, yields the respective structures of the clauses in 32), we see that the derivation trees for the passive clauses in 32 would have large enough a subtree in common if we were able to abstract away from the difference in internal argument. Although we could simply

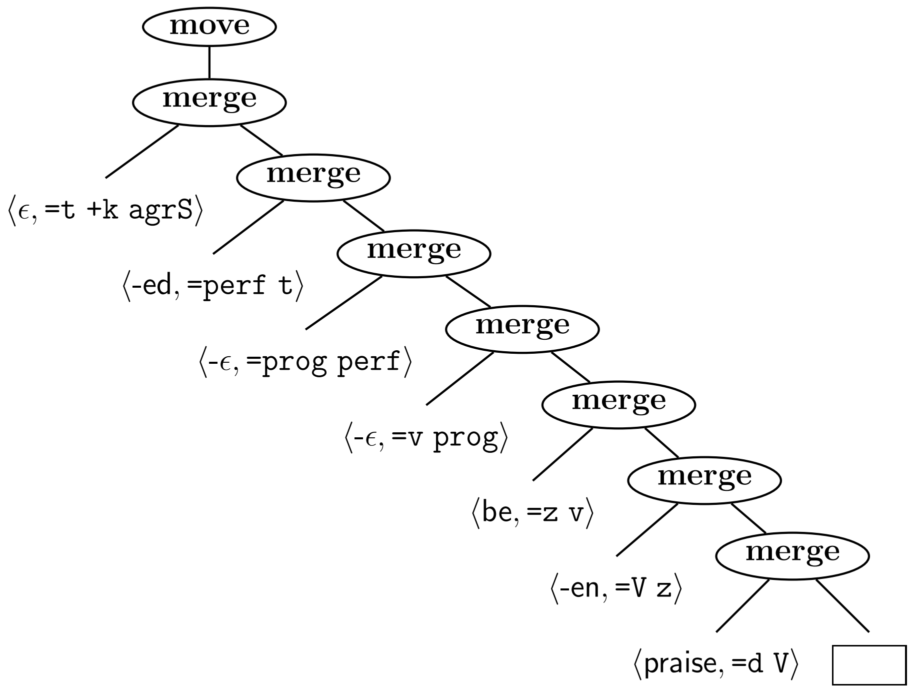

Figure 15: A derivation tree for $\boxed{}$ *was criticized*

weaken the derivational identity hypothesis to allow for identical *contexts* instead of just identical *subtrees* (as shown in figure 15), this would lessen the analytical force of the DIH. Instead, let us take the DIH seriously, and ask the question of what kind of system would give us the desired derivational flexibility to account for voice-mismatches as in 20, as well as passive-passive ellipsis as in 32. This is the topic of the next section.

5.1 Hypothetical Reasoning

The intuition behind the basic extension to the minimalist grammar system that will give us the needed derivational flexibility to satisfy the DIH is that we will be allowed to conduct derivations *as though* other expressions were there. This will allow us to eliminate an asymmetry in the structure of derivations; namely, that all expressions are introduced into derivations in their lowest chain positions. From the perspective of derivation trees, alongside the familiar **merge-move** pair shown on the left in figure 16, we will allow the symmetric **assume-discharge** pair on the right (see also figure 4). How should we interpret these two new operations? The **as-**

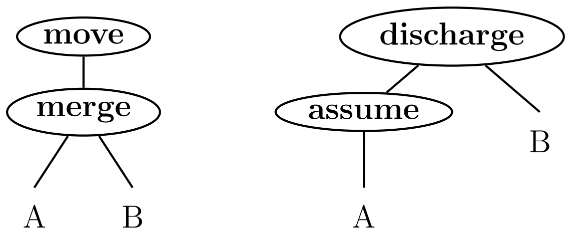

Figure 16: A new symmetry in derivations

sume operation should allow us to 'pretend' we had whatever expression we needed to continue the derivation. In the context of our lexicon in 10, we might have the following derivation.

1″. **assume**(⟨praise, `=d V`⟩)

$$[_{\mathtt{V}}\ \textsf{praise}\ t_d]\ ;\ [_{\mathtt{-k}}\ ]$$

In 1″ above, the grammar hypothesized the existence of an expression with feature bundle '`d -k`', which survives, as it still has a feature to check (`-k`). We continue the derivation, just as we would with a 'real' surviving expression, as follows.

2″. **merge**(⟨-ϵ, `=V +k agrO`⟩,1″)

$$[_{\mathtt{+k\ agrO}}\ \textsf{praise-}\epsilon\ \ [t_V\ \ t_d]]\ ;\ [_{\mathtt{-k}}\ ]$$

At this point, we would like to apply the **move** rule to the above expression; the first feature of the head is the attractor `+k`, and there is a (hypothesized) surviving expression with first feature the matching attractee feature `-k`. As the moving element will have no more unchecked features after this derivational step, it is being put herewith into its surface position. However, it is only a hypothesized element, not a 'real' one, and must be replaced by a 'real' expression before it can become inactive.[23] To do this, we **discharge**

[23]Otherwise, we are in a position where the hypothesis is erroneously sent to spell-out, which would cause the derivation to crash.

the assumption by introducing an expression which 'would have served' at the point the assumption was made.

3″. **discharge**(2″,⟨Jesús, $\texttt{d -k}$⟩)

$$[_{\texttt{agr0}}\ \textsf{Jesús}\ [\textsf{praise-}\epsilon\ \ [t_V\ t_d]]]$$

Note that, in order to discharge the assumption, the grammar must have access to information about what the assumption originally was. To make this maximally transparent, we represent hypotheses using the following notation, where δ is the current feature bundle, and γ the original hypothesized feature bundle.

$$[_\delta\ \gamma]$$

The cases of the **assume** and **discharge** operations are given in figure 17. With this addition to the minimalist grammar framework (for more

$$\textbf{assume}(A) \rightarrow \ A; [_\delta\ \texttt{x}\delta] \quad A \text{ has first feature } \texttt{=x}$$

$$\textbf{discharge}(A; [_{\texttt{-y}\delta}\ \gamma]\,, B) = \begin{cases} A;\ B & B \text{ survives} \\ [_A\ B\ A] & B \text{ has no more features} \\ & \\ \text{In both cases, } A \text{ has first feature } \texttt{+y}, & \\ \text{and } B\text{'s feature bundle begins with } \gamma & \end{cases}$$

Figure 17: Cases of **assume** and **discharge**

details, see Kobele (2007)), we can account for passive-passive verb phrase ellipsis, by delaying the introduction of *both* internal arguments until they are discharged in their case positions. For comparison, the sentence *Jesús was criticized*, derived already in figure 13, has now another derivation, shown in figure 18.

5.2 Antecedent-Contained Deletion

Although introduced to deal with passive-passive ellipsis, hypothetical reasoning allows for solutions to other, well-known elliptical puzzles. Cases of antecedent-contained deletion, as in 34, pose problems for surface-oriented perspectives on ellipsis resolution, as the ellipsis site seems embedded in its antecedent.

(34) Jesús praised every boy that Adam did ~~praise~~.

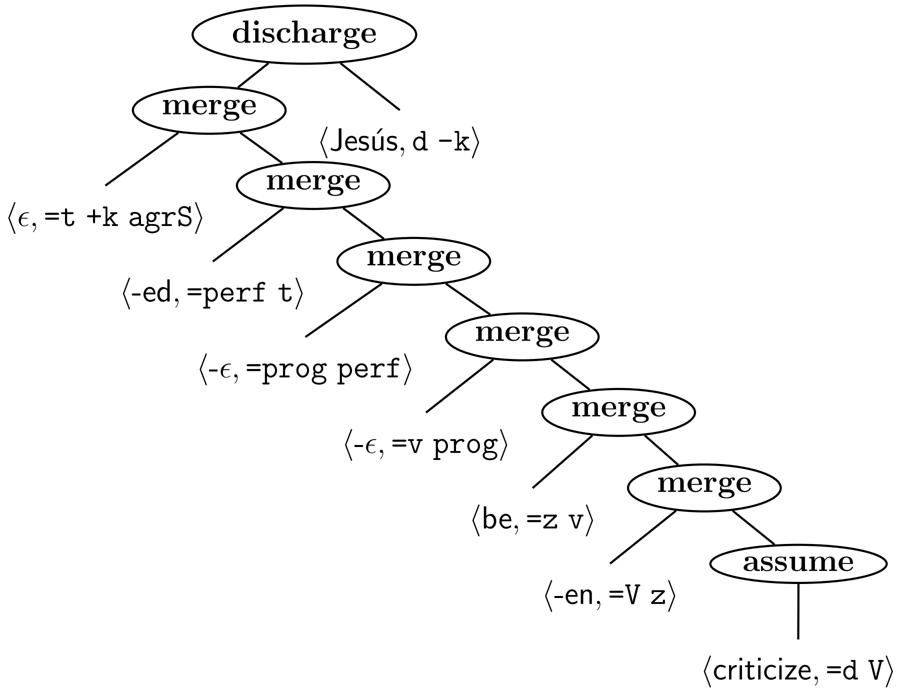

Figure 18: A second derivation tree for *Jesús was criticized*

In particular, for the derivational approach, antecedent-containment is impossible; a (finite) subtree cannot be identical with one of its proper parts. Thus the problem that arises more generally is quite acute in the present setting. The standard remedy to antecedent-containment (which we are forced to by the DIH) is to deny that it exists. In semantic identity theories, this is done in virtue of the fact that the offending DPs are interpreted outside of where they appear on the surface, and in (derived) syntactic identity theories, this is done by moving the offending DP to a point where it is no longer contained in the antecedent. Hypothetical reasoning allows us to do just that, but derivationally. To see this, let us extend our fragment with the lexical items in figure 19, which will allow us to derive relative clauses. These lexical items implement a raising analysis of relative clauses

$\langle$every, `=n d -k`$\rangle$	$\langle$boy, `n`$\rangle$
$\langle$that, `=agrS +rel n`$\rangle$	$\langle\epsilon$, `=n d -k -rel`$\rangle$

Figure 19: Lexical items for relativization

(Kayne, 1994), where the head of the relative clause is formed by merging a noun phrase with a null relative determiner.[24]

$1'''$. **merge**($\langle \epsilon, \texttt{=n d -k -rel}\rangle, \langle \textsf{boy}, \texttt{n}\rangle$)

$$[_{\texttt{d -k -rel}}\ \epsilon\ \textsf{boy}]$$

The resulting DP has an additional feature (`-rel`) which causes it to survive past its case checking position. The `-rel` feature is checked in the specifier of relative *that*, which selects AgrSPs as complements.

$2'''$. **merge**($\langle \textsf{that}, \texttt{=agrS +rel n}\rangle, [_{\texttt{agrS}}\ \textsf{Adam praised}];[_{\texttt{-rel}}\ \textsf{boy}]$)

$$[_{\texttt{+rel n}}\ \textsf{that}\ \ [\textsf{Adam praised}]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]$$

$3'''$. **move**($2'''$)

$$[_{\texttt{n}}\ \textsf{boy}\ \ [\textsf{that}\ \ [\textsf{Adam praised}]]]$$

Note that the expression above has the same feature bundle (`n`), and therefore distribution, as a common noun (and thus can itself serve as the head of another relative clause).

In the grammar fragment developed in this paper (along the general lines of the 'standard' minimalist analysis), DPs have two positions: their base, or θ, position, and their case position. With hypothetical reasoning, we can thus either insert DPs into the derivation in their θ position (using **merge** and **move**), or in their case position (using **assume** and **discharge**). In order to derive sentence 34, both the embedded DP (the relative clause head $[_{\texttt{d -k -rel}}\ \epsilon\ \textsf{boy}]$), as well as the entire relative clause DP ($[_{\texttt{d -k}}\ \textsf{every boy that Adam praised}]$), must be introduced in their case positions. This allows for the VPs of both clauses to be identical, in this case up to the point shown in figure 20.[25] A derivation of sentence 34 is as fol-

[24]In Kobele (2006), the null relative determiner is given a simple conjunctive semantics, representable as the lambda term $\lambda P_{et}.\lambda Q_{et}.\lambda x_e.\ P(x) \wedge Q(x)$. After combining with the head noun *boy* in $1'''$, it denotes the restrictive function on predicate meanings $\lambda Q_{et}.\lambda x_e.\ \mathbf{boy}(x) \wedge Q(x)$. The relative clause in $3'''$ can be shown to denote (as is standard) the predicate below, which holds of an entity just in case it is both a boy, and was praised by Adam.

$$\lambda x_e.\ \mathbf{boy}(x) \wedge \mathbf{praise}(x)(\mathbf{adam})$$

[25]The natural extension of the fragment here to control and raising predicates does not allow for the generation of ACD sentences such as i, where the ellipsis site extends higher than the case position of the object.

i. Jesús wants to eat every jalapeño that Adam does ~~want to eat~~.

This can be resolved by postulating a higher movement position, in *both* clauses, that is above the ellipsis site. It is standard to link this position to QR, that is, to movement for scope (Fiengo and May, 1994). However, it seems that this may not be the whole story (Tanaka, 2005; Yoshida, 2008). Another possibility is to relax the restriction that the **discharge** operation apply before a hypothesis reaches its ultimate landing position. I leave this question to further study.

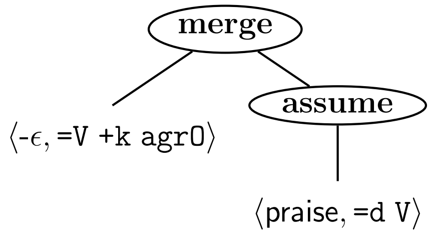

Figure 20: The identical subderivations of the antecedent and ellipsis site in ACD

lows. We begin by simultaneously building up two copies of the verb phrase shown in figure 20.[26]

1. a. **assume**(⟨praise, =d V⟩)
   b. **assume**(⟨praise, =d V⟩)

$$[_{\texttt{V}} \text{ praise } t_d] ; [_{\texttt{-k}} \texttt{ d -k}]$$

2. a. **merge**(⟨-ϵ, =V +k agrO⟩,1a)
   b. **merge**(⟨-ϵ, =V +k agrO⟩,1b)

$$[_{\texttt{+k agrO}} \text{ praise-}\epsilon \ [\ t_V \ t_d]] ; [_{\texttt{-k}} \texttt{ d -k}]$$

Simplifying, we write the twice derived expression above as the below.

$$[_{\texttt{+k agrO}} \text{ praise-}\epsilon] ; [_{\texttt{-k}} \texttt{ d -k}]$$

At this point, we have derived two copies of the above derived phrase, one of which we render phonologically null (deletion under derivational identity with the other copy).

3. **delete**(2a)

$$[_{\texttt{+k agrO}} \ \cancel{\text{praise-}\epsilon}] ; [_{\texttt{-k}} \texttt{ d -k}]$$

We continue building the relative clause with the deleted VP, first discharging the hypothesis $[_{\texttt{-k}} \texttt{ d -k}]$ with the relative clause head *boy* derived in $1'''$, and then merging in the subject, the tense and aspect heads, AgrS, and the relative determiner.

[26]Synchronizing derivations (as outlined in Kobele (2006)) allows for copying without necessitating inspection of the internal structure of the expressions which are being copied. The idea is simple: if you do the same thing to identical objects, the result is the same (i.e. the resulting objects are also identical).

4. **discharge**(3,[$_{\texttt{-k -rel}}$ boy])

$$[_{\texttt{agrO}}\ t_k\ [\cancel{\textsf{praise-}\epsilon}]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]$$

5. **merge**(⟨-ϵ, =agrO =d v⟩,4)

$$[_{\texttt{=d v}}\ \cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\ [t_k\ [t_{agrO}]]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]$$

6. **merge**(5,⟨Adam, d -k⟩)

$$[_{\texttt{v}}\ t_d\ [\cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\ [t_k\ [t_{agrO}]]]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

7. **merge**(⟨-ϵ, =v prog⟩,6)

$$[_{\texttt{prog}}\ \cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\ [t_d\ [t_v\ [t_k\ [t_{agrO}]]]]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

$$\rightsquigarrow\ [_{\texttt{prog}}\ \cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

8. **merge**(⟨-ϵ, =prog perf⟩,7)

$$[_{\texttt{perf}}\ \cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\ [t_{prog}]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

9. **merge**(⟨-ed, =perf t⟩,8)

$$[_{\texttt{t}}\ \cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-ed}\ [t_{perf}\ [t_{prog}]]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

$$\rightsquigarrow\ [_{\texttt{t}}\ \cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-ed}]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

10. **merge**(⟨ϵ, =t +k agrS⟩,9)

$$[_{\texttt{+k agrS}}\ \epsilon\ [\cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-ed}]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]\ ,\ [_{\texttt{-k}}\ \textsf{Adam}]$$

11. **move**(10)

$$[_{\texttt{agrS}}\ \textsf{Adam}\ [\epsilon\ [\cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-ed}]]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]$$

12. **merge**(⟨that, =agrS +rel n⟩,11)

$$[_{\texttt{+rel n}}\ \textsf{that}\ [\textsf{Adam}\ [\epsilon\ [\cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-ed}]]]]\ ;\ [_{\texttt{-rel}}\ \textsf{boy}]$$

13. **move**(12)

$$[_{\texttt{n}}\ \textsf{boy}\ [\textsf{that}\ [\textsf{Adam}\ [\epsilon\ [\cancel{\textsf{praise-}\epsilon}\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-}\epsilon\textsf{-ed}]]]]]$$

The expression derived above is an NP, and we will combine it with every in the next step to form a DP. Because the expression takes up so much space on paper, I will abbreviate it even more severely than usual, relying on an unspecified mechanism to convert the stranded affix *-ed* to *did*.

14. **merge**(⟨every, `=n d -k`⟩,13)

$$[_{\mathtt{d\ -k}}\ \text{every}\ [\text{boy}\ [\text{that}\ [\text{Adam}\ [\epsilon\ [\cancel{\text{praise-}\epsilon}\text{-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-ed}]]]]]]$$

$$\rightsquigarrow [_{\mathtt{d\ -k}}\ \text{every boy that Adam did}]$$

At this point, we use the second of the two synchronously derived expressions as the matrix verb.

15. **discharge**(2b,14)

$$[_{\mathtt{agrO}}\ \text{every boy that Adam did}\ [\text{praise-}\epsilon]]$$

We continue the derivation as before, merging Jesus as the subject.

16. **merge**(⟨-ϵ, `=agrO =d v`⟩,15)

$$[_{\mathtt{=d\ v}}\ \text{praise-}\epsilon\text{-}\epsilon\ [\text{every boy that Adam did}\ [t_{agrO}]]]$$

17. **merge**(16,⟨Jesús, `d -k`⟩)

$$[_{\mathtt{v}}\ t_d\ [\text{praise-}\epsilon\text{-}\epsilon\ \text{every boy that Adam did}\ [t_{agrO}]]]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

18. **merge**(⟨-ϵ, `=v prog`⟩,17)

$$[_{\mathtt{prog}}\ \text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\ [t_d\ [t_v\ [\text{every boy that Adam did}\ [t_{agrO}]]]]]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

$$\rightsquigarrow [_{\mathtt{prog}}\ \text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\ \text{every boy that Adam did}]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

19. **merge**(⟨-ϵ, `=prog perf`⟩,18)

$$[_{\mathtt{perf}}\ \text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-}\epsilon\ [t_{prog}\ \text{every boy that Adam did}]]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

20. **merge**(⟨-ed, `=perf t`⟩,19)

$$[_{\mathtt{t}}\ \text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-ed}\ [t_{perf}\ [t_{prog}\ \text{every boy that Adam did}]]]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

$$\rightsquigarrow [_{\mathtt{t}}\ \text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-ed every boy that Adam did}]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

21. **merge**(⟨ϵ, `=t +k agrS`⟩,20)

$$\big[_{\mathtt{+k\ agrS}}\ \epsilon\ [\text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-ed every boy that Adam did}]\big]\ ;\ [_{\mathtt{-k}}\ \text{Jesús}]$$

22. **move**(21)

$$[_{\mathtt{agrS}}\ \text{Jesús}\ [\epsilon\ [\text{praise-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-}\epsilon\text{-ed every boy that Adam did}]]]$$

6 Conclusion

In derivational grammar formalisms, the derivation itself can be seen as structuring expressions. By formulating the derivational identity hypothesis, the hypothesis that the structure of the ellipsis site is an exact derivational analogue of its antecedent, I have shown that ellipsis can be used as a powerful probe into the underlying mechanisms of grammar. Recent work on ellipsis by Chung (2006) has suggested that we need to make reference to syntactic structure in order to adequately describe the ellipsis data, and has proposed an account based on (partial) identity of numerations. As remarked in footnote 11, in the context of minimalist grammars, numerations and derivation trees are interchangeable. Although we can view Chung's proposal as being related to the one made here, the statement in terms of identity of derivations is general enough to apply independently of grammar formalism, and thus clear enough to interpret its import on data in a meaningful way.

In tandem with the DIH, ellipsis data puts pressure on a grammar formalism to provide rich derivational structures. In order to accommodate the attested antecedence possibilities, we were led to introduce hypothetical reasoning as a dual of movement. Hypothetical reasoning can be thought of as a derivational version of trace deletion (Sauerland, 1998; Fox, 1999), which acts as a syntactic version of the lambda abstraction used by Sag (1976) to resolve the same issues we confronted here, and introduces a symmetry of sorts into the derivational process, allowing dependencies to be satisfied before an expression is introduced into a derivation. The DIH requires that identity be computed over derivational subtrees. Hypothetical reasoning enables us to 'trick' the formal system into allowing us to, in a very restricted way, compute identity over derivational *contexts*. It is of course possible to weaken the DIH from computing identity over subtrees to (sub-)contexts, and obviate thereby the need for hypothetical reasoning.

Many systematic deviations from surface identity between the (hypothesized) ellipsis site and antecedent have been unearthed over the years (see among many others, Ross (1969); Hardt (1993); Fiengo and May (1994); Chung et al. (1995); Potsdam (1997); Merchant (2001)). Adopting the DIH gives us a principled way to view these apparent deviations from identity between ellipsis site and an antecedent; instead of being problematic exceptions to a rule, they provide information not only about which construction types must be derivationally related to which others, but also about what kinds of structure building operations must be part of an adequate theory of grammar. Here I have provided a case study of sorts in one well-known such deviation, presenting by example the general strategy for a response consistent with the DIH. There remain many more. At the very least, the DIH promises a principled way of thinking about ellipsis, and of approaching its analysis.

References

Arregui, A., C. Clifton, Jr., L. Frazier, and K. Moulton (2006). Processing elided verb phrases with flawed antecedents: The recycling hypothesis. *Journal of Memory and Language 55*, 232–246.

Bach, E. W. (1980). In defense of passive. *Linguistics and Philosophy 2*, 297–341.

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

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory 2*(3), 113–124.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, Massachusetts: MIT Press.

Chomsky, N. (1995). *The Minimalist Program*. Cambridge, Massachusetts: MIT Press.

Chung, S. (2006). Sluicing and the lexicon: The point of no return. In R. T. Corver and Y. Kim (Eds.), *BLS 31: General Session on Parasession on Prosodic Variation and Change*, Berkeley, CA, pp. 73–91. Berkeley Linguistics Society.

Chung, S., W. A. Ladusaw, and J. McCloskey (1995). Sluicing and logical form. *Natural Language Semantics 3*(3), 239–282.

Elbourne, P. (2002). *Situations and Individuals*. Ph. D. thesis, Massachusetts Institute of Technology.

Fiengo, R. and R. May (1994). *Indices and Identity*. Cambridge, Massachusetts: MIT Press.

Fox, D. (1999). Reconstruction, binding theory, and the interpretation of chains. *Linguistic Inquiry 30*(2), 157–196.

Frampton, J. and S. Gutmann (2002). Crash-proof syntax. In S. D. Epstein and T. D. Seely (Eds.), *Derivation and Explanation in the Minimalist Program*, pp. 90–105. Oxford: Blackwell.

Frey, W. and H.-M. Gärtner (2002). Scrambling and adjunction in minimalist grammars. In G. Jäger, P. Monachesi, G. Penn, and S. Wintner (Eds.), *Proceedings of Formal Grammar 2002*, pp. 41–52.

Gärtner, H.-M. and J. Michaelis (2007). Some remarks on locality conditions and minimalist grammars. In U. Sauerland and H.-M. Gärtner (Eds.), *Interfaces + Recursion = Language?*, Volume 89 of *Studies in Generative Grammar*, pp. 161–195. Berlin: Mouton de Gruyter.

Hale, J. T. (2003). *Grammar, Uncertainty and Sentence Processing.* Ph. D. thesis, The Johns Hopkins University.

Hale, J. T. and E. P. Stabler (2005). Strict deterministic aspects of minimalist grammars. In P. Blache, E. P. Stabler, J. Busquets, and R. Moot (Eds.), *Logical Aspects of Computational Linguistics*, Volume 3492 of *Lecture Notes in Computer Science*, Chapter 11, pp. 162–176. Springer.

Hardt, D. (1993). *Verb Phrase Ellipsis: Form, Meaning, and Processing.* Ph. D. thesis, University of Pennsylvania.

Harkema, H. (2001). *Parsing Minimalist Languages.* Ph. D. thesis, University of California, Los Angeles.

Kayne, R. (1994). *The Antisymmetry of Syntax.* Cambridge, Massachusetts: MIT Press.

Keenan, E. (1980). Passive is phrasal not (sentential or lexical). In T. Hoekstra, H. van der Hulst, and M. Moortgat (Eds.), *Lexical Grammar*, Volume 3 of *Publications in Language Sciences*, Chapter 7, pp. 181–214. Dordrecht: Foris Publications.

Kobele, G. M. (2002). Formalizing mirror theory. *Grammars 5*(3), 177–221.

Kobele, G. M. (2005). Features moving madly: A formal perspective on feature percolation in the minimalist program. *Research on Language and Computation 3*(4), 391–410.

Kobele, G. M. (2006). *Generating Copies: An investigation into structural identity in language and grammar.* Ph. D. thesis, University of California, Los Angeles.

Kobele, G. M. (2007). A formal foundation for A and A-bar movement in the minimalist program. In M. Kracht, G. Penn, and E. P. Stabler (Eds.), *Mathematics of Language 10.* UCLA.

Kobele, G. M., C. S. Kim, J. T. Hale, and J. T. Runner (2008). A processing model for ungrammatical VP ellipsis. paper presented at the 2008 CUNY Conference on Human Sentence Processing.

Koizumi, M. (1995). *Phrase Structure In Minimalist Syntax.* Ph. D. thesis, Massachusetts Institute of Technology.

Kracht, M. (2007). The emergence of syntactic structure. *Linguistics and Philosophy 30*(1), 47–95.

Lees, R. B. (1960). *The grammar of English nominalizations.* The Hague: Mouton.

Lobeck, A. (1995). *Ellipsis: Functional Heads, Licensing, and Identification.* New York: Oxford University Press.

MacSwan, J. (2000). The architecture of the bilingual language faculty: evidence from intrasentential code switching. *Bilingualism: Language and Cognition 3*(1), 37–54.

Manzini, M. R. and A. Roussou (2000). A minimalist theory of A-movement and control. *Lingua 110*(6), 409–447.

Matushansky, O. (2006). Head movement in linguistic theory. *Linguistic Inquiry 37*(1), 69–109.

Merchant, J. (2001). *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*, Volume 1 of *Oxford Studies in Theoretical Linguistics.* New York: Oxford University Press.

Merchant, J. (2007). Voice and ellipsis. ms., University of Chicago.

Merchant, J. (2008). An asymmetry in voice mismatches in VP-ellipsis and pseudogapping. *Linguistic Inquiry 39*(1), 169–179.

Michaelis, J. (2001). *On Formal Properties of Minimalist Grammars.* Ph. D. thesis, Universität Potsdam.

Murguía, E. (2004). *Syntactic Identity and Locality Restrictions on Verbal Ellipsis.* Ph. D. thesis, University of Maryland.

Parsons, T. (1978). Pronouns as paraphrases. unpublished ms.

Philips, C. (2003). Linear order and constituency. *Linguistic Inquiry 34*(1), 37–90.

Potsdam, E. (1997). English verbal morphology and VP ellipsis. In *Proceedings of the 27th Meeting of the North East Linguistic Society*, Amherst, Massachusetts, pp. 353–368.

Ross, J. R. (1969). Guess who? In R. Binnick, A. Davison, G. Green, and J. Morgan (Eds.), *Papers from the Fifth Regional Meeting of the Chicago Linguistics Society (CLS)*, Volume 5, Chicago, pp. 252–286.

Sag, I. A. (1976). *Deletion and Logical Form.* Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Sauerland, U. (1998). *The Meaning of Chains.* Ph. D. thesis, Massachusetts Institute of Technology.

Stabler, E. P. (1997). Derivational minimalism. In C. Retoré (Ed.), *Logical Aspects of Computational Linguistics*, Volume 1328 of *Lecture Notes in Computer Science*, pp. 68–95. Berlin: Springer-Verlag.

Stabler, E. P. (2001). Recognizing head movement. In P. de Groote, G. F. Morrill, and C. Retoré (Eds.), *Logical Aspects of Computational Linguistics*, Volume 2099 of *Lecture Notes in Artificial Intelligence*, Berlin, pp. 254–260. Springer Verlag.

Stabler, E. P. (2003). Comparing 3 perspectives on head movement. In A. Mahajan (Ed.), *Syntax at Sunset 3: Head Movement and Syntactic Theory*, Volume 10 of *UCLA/Potsdam Working Papers in Linguistics*, pp. 178–198.

Stabler, E. P. (2006). Sidewards without copying. In P. Monachesi, G. Penn, G. Satta, and S. Wintner (Eds.), *Proceedings of the 11th conference on Formal Grammar*, pp. 133–146. CSLI.

Stabler, E. P., T. C. Collier, G. M. Kobele, Y. Lee, Y. Lin, J. Riggle, Y. Yao, and C. E. Taylor (2003). The learning and emergence of mildly context sensitive languages. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, and J. Ziegler (Eds.), *Advances in Artificial Life*, Volume 2801 of *Lecture Notes in Computer Science*, pp. 525–534. Springer.

Steedman, M. (2000). *The Syntactic Process*. MIT Press.

Stroik, T. (2008). *Locality in Minimalist Syntax*. Cambridge, Massachusetts: MIT Press.

Tanaka, H. (2005). Antecedent-contained deletion without quantifier raising. In G. Hicks and G. Walker (Eds.), *York Papers in Linguistics*, Volume 3, pp. 173–184.

Tanenhaus, M. K. and G. N. Carlson (1990). Comprehension of deep and surface verb phrase anaphors. *Language and Cognitive Processes 5*(4), 257–280.

Wadler, P. (1990). Deforestation: Transforming programs to eliminate trees. *Theoretical Computer Science 73*, 231–248.

Yoshida, M. (2008). Antecedent contiained sluicing. ms. Northwestern University.