What sort of cognitive hypothesis is a derivational theory of grammar?

Tim Hunter

Two kinds of theories of natural language syntax can be distinguished: representational theories and derivational theories. A representational theory posits some set of constraints, and defines a well-formed syntactic object to be one that satisfies all of the constraints. A derivational theory instead takes the form of a nondeterministic mechanical procedure, for example a symbol-rewriting procedure or a procedure that builds larger objects out of smaller ones, and defines a well-formed syntactic object to be one that is generated by this procedure.

The mentalistic claims of a representational theory are relatively clear: it is generally understood that when a speaker comprehends or produces a sentence, a representational theory predicts that a corresponding well-formed syntactic object (say, a tree structure with the sentence's words at its leaves) is grasped in the speaker's mind. With a representational theory, nothing is said about how a speaker might go about constructing (a representation of) this syntactic object, and the linguist's everyday use of the theory also does not involve any descriptions of procedures that construct syntactic objects.

The situation for a derivational theory, however, is slightly less straightforward. Consider for example the mainstream contemporary derivational theories deriving from Chomsky (1995) and subsequent work. It is natural to assume that a speaker grasps the syntactic object that is the end product of the derivational process corresponding to the sentence being comprehended/produced, i.e. the tree structures that are routinely used to illustrate proposals in this literature. But if that is the extent of a derivational theory's mental commitments, what is the scientific role of the derivational procedure? If we have an existing derivational theory T_1, and an alternative theory T_2 proposes a derivational procedure that differs from that of T_1 but yields the same set of well-formed syntactic objects, then is there any clear sense in which we should understand the two theories to be different? If they are not different — i.e. if the procedural component of a derivational theory does not contribute to its empirical bottom line — then why bother with the derivational procedures at all? If they are different, then *how* are they different, i.e. *how* does the procedural component of a derivational theory contribute to the theory's empirical bottom line?

Answering this last question is the main goal of this paper: I will lay out a way of understanding

derivational theories according to which the derivational process itself, in addition to the end result of this process, plays a part in determining the empirical predictions of a theory. For concreteness, I will illustrate by showing at the end of the paper — in entirely artificial, and artificially small-scale, case studies — how derivational operations play a part in determining predictions about sentence comprehension difficulty, and predictions about which grammar a learner will choose in response to some collection of input. The main point I want to stress, however, is not the particulars of either of these case studies, but rather the general understanding of derivational frameworks which makes a derivational *process* a first-class theoretical object which can underpin empirical predictions just as naturally as the static objects in a representational theory can. The key idea is that we can identify an atemporal structured object — typically, a *derivation tree* — that encapsulates the derivational process and yet is static in the same sense that syntactic objects in representational theories are.

This point is an attempt to address the issues raised by some who have questioned the role of derivational processes in modern generative syntax (Sag and Wasow, 2011; Jackendoff, 2011; Ferreira, 2005; Phillips and Lewis, 2013). This criticism appears to stem largely from the fact that, in practice, descriptions of how a particular theory accounts for some relevant data rarely requires making reference to the derivational operations[1] posited by the theory; very often, the final constructed syntactic object is all we need to consider when working out the empirical predictions of a theory, and it seems that any number of different ways of describing how that object is constructed would leave the account intact.

I will suggest that this impression is due to a perhaps unfortunate quirk in modern generative syntax: the fact that the end product of a derivational process very often encodes a large amount (or nearly all) of the derivational process itself, for example in the form of co-indexed traces or copies. A clearer understanding of how derivational grammars in general can constitute hypotheses about mental phenomena can be achieved by considering other derivational systems that do not have this quirk, and where it is therefore easy to see the role of the derivational process itself (because this role is not duplicated by representational devices). With more light thus shed on the kind of mental significance a derivational process can in principle take on, we will be better placed to ask (i) what it would mean to ascribe this same kind of mental significance to the derivational processes typically invoked in modern generative syntax, and (ii) whether doing so is consistent with the standard ways in which linguists already work with these theories. In answer to the second question, I will argue that it is not only consistent with standard practices but furthermore is, given the way our theories have developed over the decades, a very natural understanding of syntactic

[1]This is not to deny that we sometimes talk about, for example, a certain sentence being unacceptable "because this movement step violates such-and-such constraint". But due to the presence of devices like traces or copies, this appeal to the processes themselves is often dispensable. Much more on this point below.

derivations.

In Section 1 I will review in more detail the distinguishing features of representational and derivational theories of grammar, and the questions that are sometimes raised about the mental significance of derivational processes. In doing so I will discuss the abovementioned quirk of modern generative syntax, and the way the questions are clarified by considering other systems that do not share this quirk. I will then turn to minimalist syntax more specifically in Section 2. The goal here will be to identify the static representations (namely, derivation trees) that encapsulate this kind of theory's derivational processes. I will do this for two subtly different variants of minimalist theory: one which takes merge and move to be distinct primitive operations, and one which unifies them into a single operation. These two variants that I will introduce differ *only* in their derivational processes, not in the syntactic objects that they construct. In Section 3 I will then present case studies where the two variants nonetheless make distinct predictions in two empirical domains: sentence comprehension difficulty, and grammar selection by a learner. Since the two variants' differences concern only their derivational processes, this serves to demonstrate that the procedural component of a derivational system contributes to a theory's empirical bottom line. Section 4 summarizes and concludes.

1 Representations, derivations and derived expressions

1.1 Purely representational and purely derivational systems

Caricaturing at least slightly, Figure 1 illustrates one possible conception of the relationship between a representational system and a derivational system. On the left is the static syntactic structure assigned to the sentence 'Kim gives Sandy Fido' in HPSG, one of the more widely-known representational theories of grammar (Pollard and Sag, 1994, p.33). This syntactic object is well-formed by virtue of satisfying the relevant array of constraints. As mentioned above, the mental commitments of this kind of theory are relatively clear: in comprehending or producing this sentence, a representation of this syntactic object is grasped[2] in the speaker's mind. By virtue of the fact that this grasped syntactic object is well-formed, the theory predicts that this sentence will be judged to be acceptable. And perhaps there are other predictions that one might make on the basis of other properties of this syntactic object: to take an overly simplistic example, one might predict that the time taken to comprehend this sentence will be some function of the size of this object.

[2]I will assume that the intended meaning of this term, while difficult to spell out explicitly, is sufficiently clear. Since the questions I aim to address here largely centre on the difficulties that come with adopting derivational as opposed to representational grammars, I am taking as my concrete goal to show that there are no such *additional* difficulties. Fleshing out the notion that I am calling "grasping" is a difficulty that will affect derivational and representational theories equally.

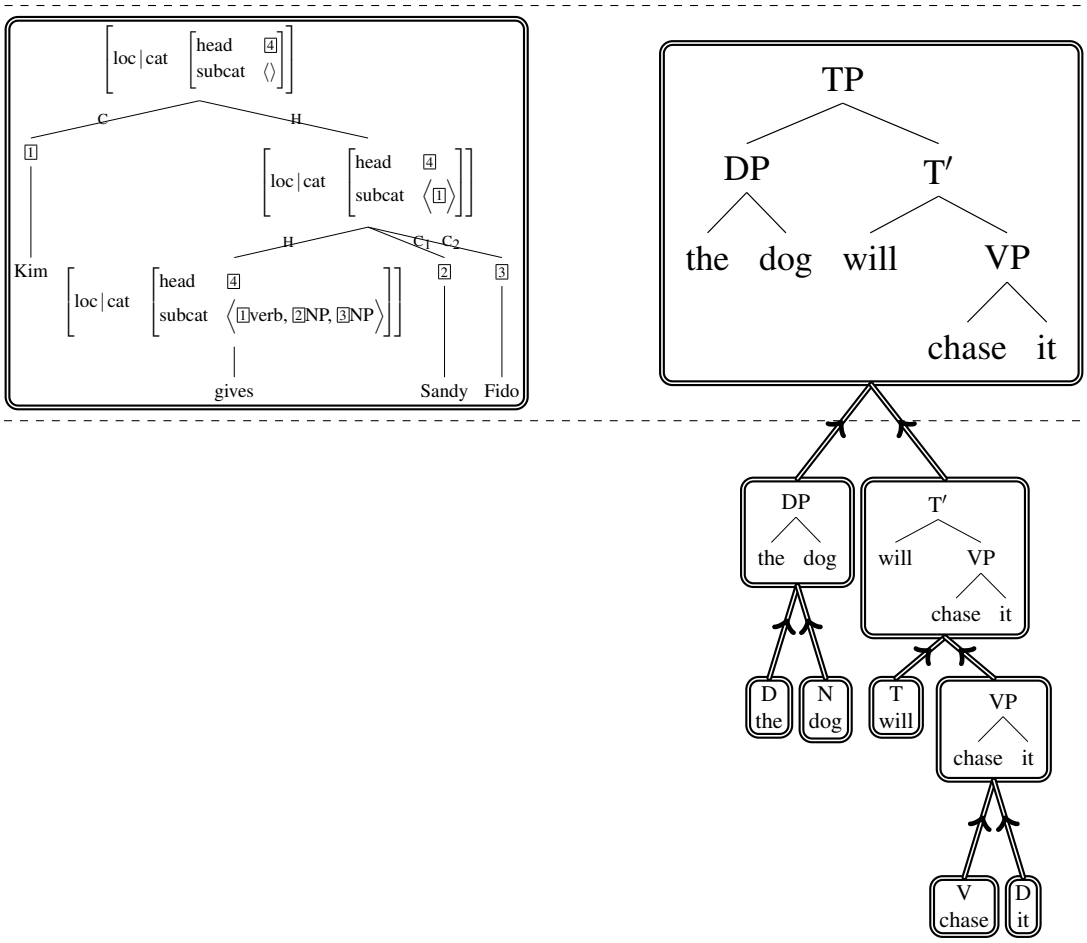


Figure 1: A view that I will argue against: only the end product of a derivational process is given the easily-understandable empirical status corresponding to that of a static representation in a non-derivational theory.

On the right of Figure 1, for comparison, is a sketch of how a derivation on modern minimalist grammar might be thought of. There is a final derived expression of the familiar sort, the tree with yield 'the dog will chase it' shown at the top. One possible thought — although I will argue against this — is that *this* tree is the thing in this derivational system that best corresponds, as indicated by the horizontal dashed lines, to HPSG's static syntactic object on the left. Since this is a derivational framework, however, there is more to the picture than just this: there is also a derivational process which is taken to have given rise to this derived expression, as shown underneath. (For reasons that should become clear, I am writing them *underneath* the derived expression, despite the usual idea that these other pieces of the picture *precede* the derived expression. This usual notion of precedence is reflected in the arrows.) The layout of the diagram is intended to emphasize the way this perception of a derivational system makes the derivational process seem like something "extra", and perhaps even something superfluous, in comparison with a representational system:

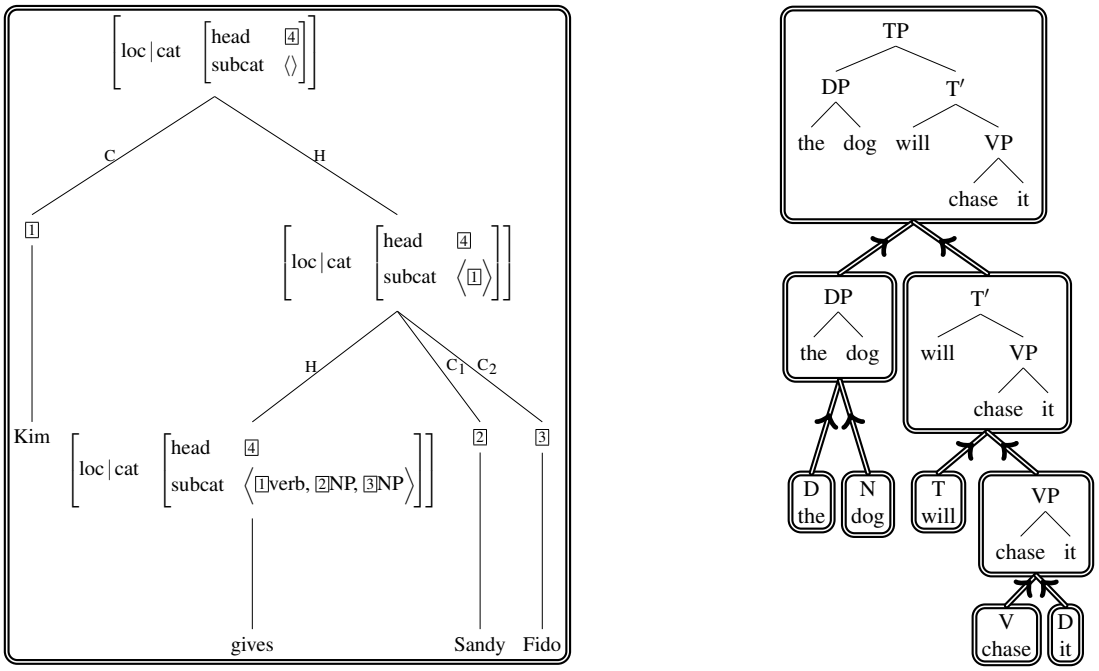


Figure 2: The view that I will argue for: the derivational process itself, in its entirety, is the relevant object.

there is nothing in the illustration of the representational system on the left which corresponds to this derivational process. So what is it there for?

I will argue that instead of this view, we should consider the derivational process as a whole (including, but not limited to, the final derived expression) to be the analog of the static representation in a representational system. This shift in perspective is reflected in the shift from Figure 1 to Figure 2. The arrows that are usually thought of (and can still be, harmlessly) as indicating a kind of precedence are now simply part of the object that a speaker must grasp; the formal *relationships amongst expressions* that they express are part of the information that a speaker must recover.[3]

It will be useful to establish some terminology for what follows. I will use the term *expression*

[3]The difference between Figure 1 and Figure 2 perhaps corresponds to the difference between what Phillips and Lewis (2013) call the "extensionalist" view of derivations and the "formalist" view, respectively. The formalist view can be seen as an intermediate position between two extremes: the extensionalist view, according to which individual derivational steps are not understood to be making any mentalistic commitments at all, and the "literalist" view, according to which individual derivational steps are interpreted very directly as hypothesized real-time mental operations. From the perspective in Figure 2 that I aim to elucidate here, linking hypotheses can be formulated that expose derivational operations to empirical scrutiny (unlike the extensionalist view), but these linking hypotheses do not include the straightforward one that makes immediate and direct predictions about real-time mental operations (as is the case on the literalist view).

Phillips and Lewis mention the intermediate formalist option only relatively briefly, and focus mainly on the literalist and extensionalist extremes, without going into much detail about what a fleshed-out formalist position would look like. But the notion of a static atemporal derivation tree, mentioned above, corresponds closely to the collection of formally related structures that Phillips and Lewis mention.

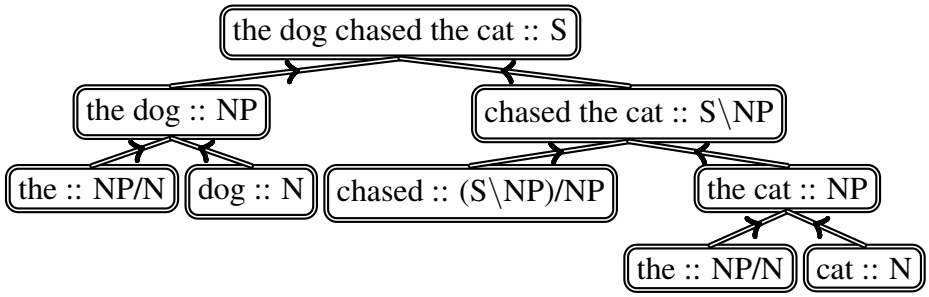
for an object of the sort that might be manipulated or inspected by a grammar: either checked for consistency with some representational constraint, or used as input to or produced as output from some derivational operation. I will show expressions inside rounded double boxes throughout. I will use *object* as a much more general term for any kind of structured representation that a mind might grasp. Expressions are objects, but not all objects are expressions. In a representational setting, there are no relevant objects to consider besides expressions themselves, and so the object to be grasped upon encountering the sentence 'Kim gives Sandy Fido' is simply the expression itself that appears on the left of Figure 1 and Figure 2. The difference between these two figures is that Figure 1 expresses a view where, in the derivational system, the object to be grasped is the single expression shown within the horizontal dashed lines; whereas Figure 2 expresses the view that the object to be grasped is an object of a different sort, an object encoding certain relations among expressions. This object is a derivation (and can be represented on paper by a derivation tree).

A clear illustration of the perspective presented in Figure 2 is provided by the various kinds of categorial grammar. In this framework, the categories into which lexical items are classified can be complex, and a small number of very general combinatory rules apply in a manner that is guided by these potentially complex categories. For example, using the lexical items shown in (1), the two general rules of forwards and backwards function application can be applied recursively to construct the sentence 'the dog chased the cat'. This is typically illustrated using a format like (2), but an equivalent representation that follows the conventions I adopt throughout this paper is the one in (3).

(1) the :: NP/N
dog :: N
cat :: N
chased :: (S\NP)/NP

(2)
$$\dfrac{\dfrac{\dfrac{\text{the}}{\text{NP/N}}\quad\dfrac{\text{dog}}{\text{N}}}{\text{NP}}\qquad\dfrac{\dfrac{\text{chased}}{\text{(S\NP)/NP}}\qquad\dfrac{\dfrac{\text{the}}{\text{NP/N}}\quad\dfrac{\text{cat}}{\text{N}}}{\text{NP}}}{\text{S\NP}}}{\text{S}}$$

(3)



A distinctive feature of this kind of grammar is that the expressions being manipulated are essentially unstructured: they are things like 'the dog :: NP', i.e. a string[4] paired with a category, where the category dictates how the expression can be used by any subsequent operations. So the derivational process indicated in (2) and (3) is one which works with the "ingredients" shown in (1), and produces as a result the expression 'the dog chased the cat :: S'. Notice that the final derived expression is an object of the same sort as the ingredient expressions in (1), i.e. a string with a category. As Jacobson (2007) describes this kind of system, "there is no room to state constraints on structured representations. For 'structure' is not something that the grammar ever gets to see". In the terminology introduced above, this is to say that the expressions here, the things that the grammar can "see" — inspect, manipulate, whatever — have no structure; the only structured object is the derivation.

The crucial point here is that it would make little sense to suppose that the object that is "grasped" by a speaker upon encountering the sentence 'the dog chased the cat' is simply the one *constructed by* this derivational process, namely the expression 'the dog chased the cat :: S'. For the theory to be doing any work at all, there must at the very least be some difference between what the speaker does upon encountering 'the dog chased the cat' and what he/she does upon encountering 'cat dog the the chased'. But this is not a difference between 'the dog chased the cat :: S' being well-formed and 'cat dog the the chased :: S' being ill-formed relative to some constraints on representations — there are no such constraints. Rather, the difference is that in the case of 'the dog chased the cat', *there is some derivational process* that produces the expression 'the dog chased the cat :: S', whereas in the case of 'cat dog the the chased' there is no derivational process that produces the expression 'cat dog the the chased :: S'. So what is grasped by a speaker encountering 'the dog chased the cat' is some representation like (3): something that encodes the relationships between the ingredients like 'the :: NP/N' and 'dog :: N' and the things that are built from them like 'the dog :: NP'. It is clear, then, that in this kind of system the derivational process is doing some real work, in such a way that it makes sense to construe the derivational process itself as the object that corresponds to the representations to be grasped in the setting of a representational theory; see Figure 3.

What makes the importance of the derivation so clear in categorial grammars is the fact that, as emphasized above, the expressions constructed by these derivations are just strings (with categories) that have no significant structure. Thus there is, roughly speaking, "nothing but the deriva-

[4]Of course there is also a semantic representation that accompanies each such expression, so they are really *triples* comprising a string, a meaning and a category. I will leave out the meaning component only for simplicity (an omission that is perhaps particularly egregious given that a distinguishing property of categorial grammars is the manner in which the composition of strings and the composition of meanings take place in sync with each other). The crucial point remains that these objects do not have any syntactic structure, in contrast to the case of transformational grammars.

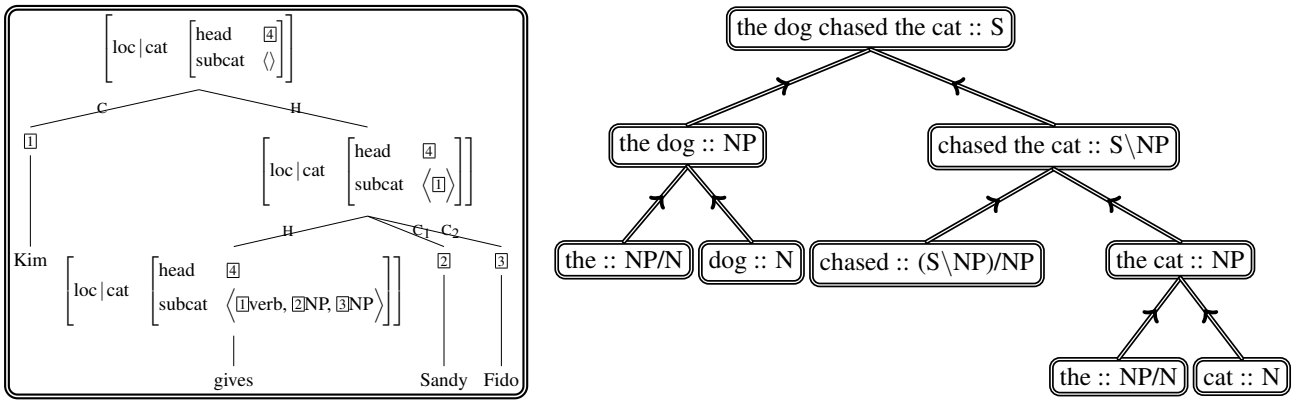


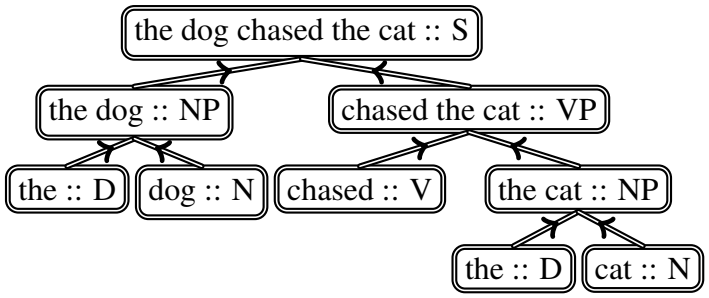
Figure 3

tion", and so when it comes to asking what the theory says about (what a speaker will do upon encountering) a particular sentence, the derivation itself is the only thing to look to. But the general point can be carried over to systems where the derived expressions have more structure, for example, if they are trees rather than strings: in such systems, it is less obvious that it is *necessary* to treat the derivation with the significance indicated in Figure 2 and Figure 3, but there is no obstacle to doing so if it is useful. My goal in this paper is roughly to show that doing so in the context of modern generative syntax is both useful and, implicitly at least, even standard.

As another example, note that a familiar context-free grammar (CFG) can be understood as a device that generates unstructured objects much like the way categorial grammars do. Specifically, the CFG in (4) can be understood as a collection of statements that allow the expression 'the dog chased the cat :: S' to be generated by the derivational process illustrated in (5). As above, this is to be understood as a record of the fact that 'the :: D' combined with 'cat :: N' to produce the expression 'the cat :: NP', which in turn combined with 'chased :: V' and so on.

(4) $S \rightarrow NP\ VP$
$NP \rightarrow D\ N$
$VP \rightarrow V\ NP$
$D \rightarrow$ the
$N \rightarrow$ dog
$N \rightarrow$ cat
$V \rightarrow$ chased

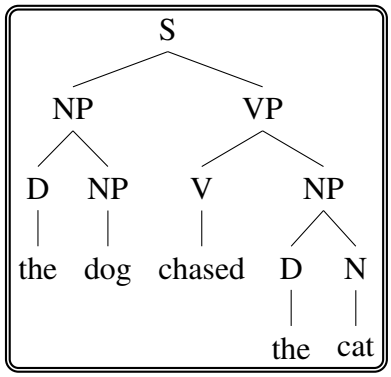
(5)



From this point of view, a rule such as 'NP → D N' is a statement about what can be combined with what (to produce what), and a CFG does not derive structured expressions any more than a categorial grammar does. Accordingly, in order for a CFG understood this way to serve as a model of linguistic competence, it is natural to take the derivation itself to be what is "grasped" by speakers, just as it is with categorial grammars.

This entirely derivational approach is not the only construal of CFGs, however, and perhaps is not even the most common one. Moving to the other extreme, we can instead consider an entirely representational construal. On this view the rules in (4) are understood not as statements specifying allowable derivational operations, but as well-formedness conditions on static expressions (McCawley, 1968), just like in HPSG. Expressions of the sort dealt with in (3) and (5) — namely things like 'the dog :: NP' and 'chased the cat :: VP' — don't contain enough information for these well-formedness conditions to take their intended effect, and so on this construal we must take the expressions that the grammar works with to be trees. One such tree that is well-formed according to the grammar in (4) is shown in (6).

(6)



On this view, the rule 'NP → D N' is not a statement about what can be combined with what, or about anything that an abstract derivational procedure can or cannot do. It is a statement that says, of a static tree-shaped expression, "If a node is labeled NP and has two daughters, the left of which is labeled D and the right of which is labeled N, then that node is well-formed". If all the parts of a tree are well-formed according to this interpretation of the grammar, then the tree is well-formed. A representation of this static tree is what is taken to be grasped by a speaker upon encountering

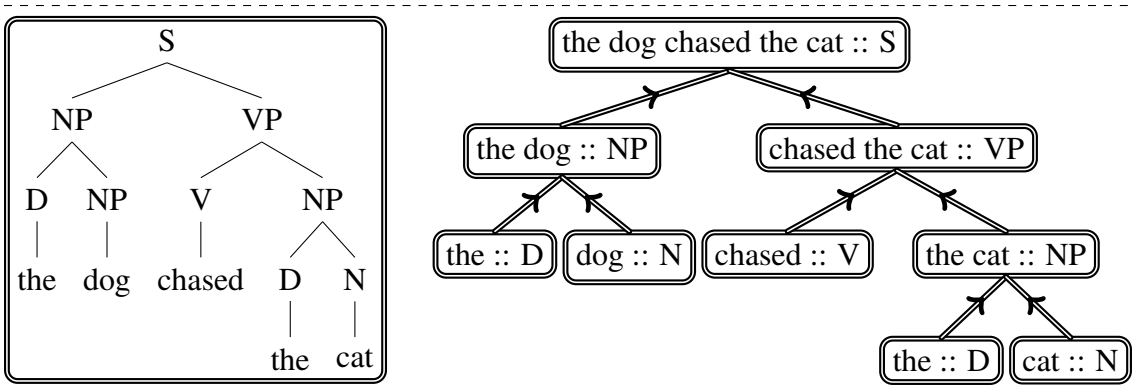


Figure 4

the sentence 'the dog chased the cat', in just the same way that the static HPSG representation on the left-hand side of Figure 1 and Figure 2 is.

Of course, these two construals of the CFG in (4) are barely distinguishable (if at all) as cognitive hypotheses. This feeling that they are one and the same is based on the assumption that the derivational process in (5) can serve as an object to be grasped, so that the two construals stand in the trivial relationship to each other illustrated in Figure 4. Denying this role to derivations themselves would force us to conclude that the construal illustrated in (5) differed significantly from the construal illustrated in (6): we would end up with the relationship between the two illustrated in Figure 5, where only the representational construal in (6) is sensible because, as discussed in relation to the categorial grammar example, it makes no sense to suppose that the object to be grasped is simply 'the dog chased the cat :: S'. I will argue that we should reject the view of modern minimalist derivations illustrated in Figure 1 for essentially the same reason that we reject the view of CFGs illustrated on the right of Figure 5.

1.2 Mixed systems

Recall from above that when a grammar deals with expressions that have more structure than strings — for example, trees — it is less obviously *necessary* that the derivation must take on the significance indicated in Figure 2 and Figure 3, but nonetheless still possible. The question is whether the possibility of doing so is useful. As an illustration of what doing so looks like when it is *not* useful, we can consider (somewhat perversely) a construal of the CFG in (4) according to which it specifies a *derivational process* (like in (5), but unlike in (6)) that works with *structured expressions* (like in (6), but unlike in (5)). A derivation in this unwieldy and redundant system is shown in (7).

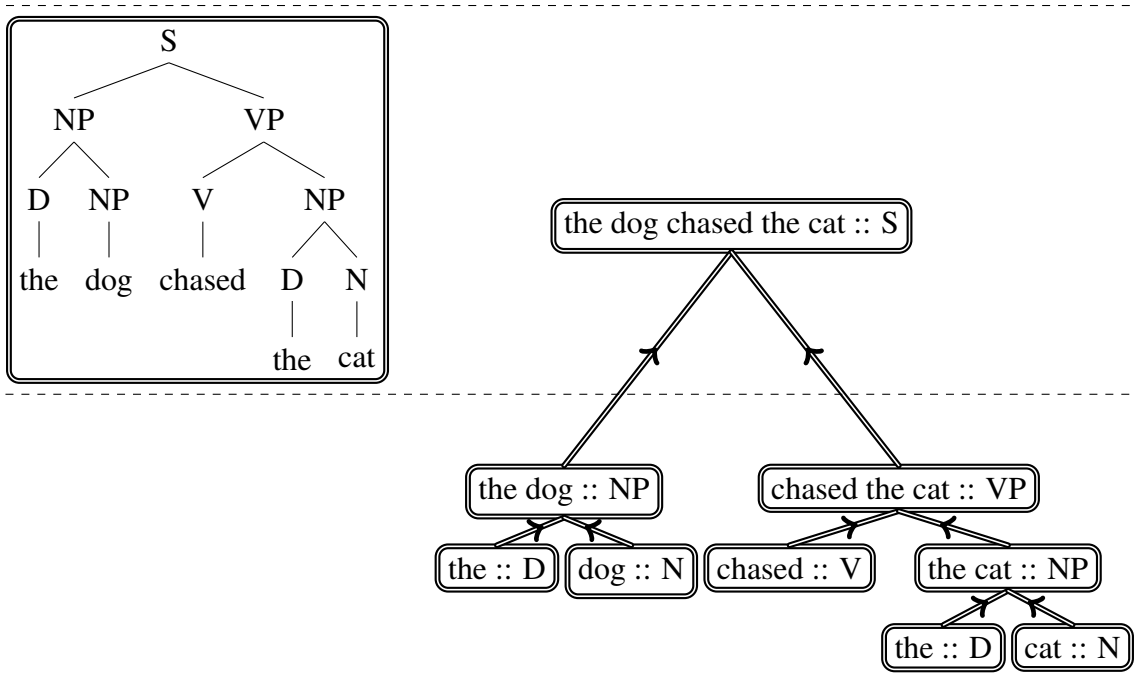
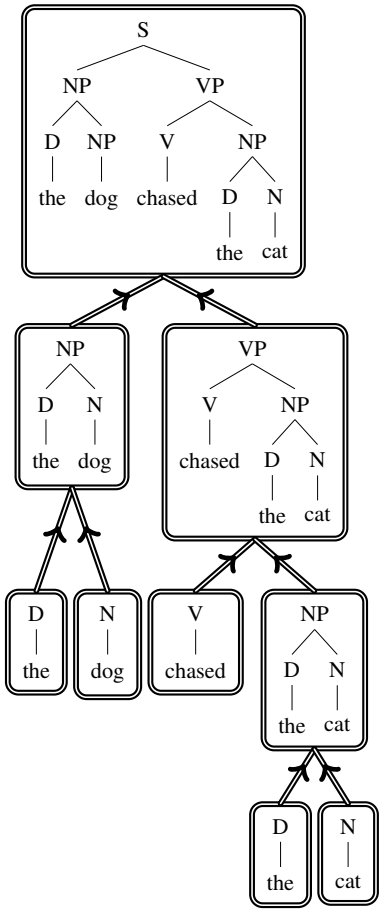

S
NP
VP
D
NP
V
NP
the
dog
chased
D
N
the
cat
the dog chased the cat :: S
the dog :: NP
chased the cat :: VP
the :: D
dog :: N
chased :: V
the cat :: NP
the :: D
cat :: N

Figure 5

(7)


S
NP
VP
D
NP
V
NP
the
dog
chased
D
N
the
cat
NP
D
N
the
dog
VP
V
NP
chased
D
N
the
cat
D
N
V
NP
the
dog
chased
D
N
the
cat
D
N
the
cat

Now the rule 'NP → D N', for example, says two mutually redundant things. First, with regard to the derivational process, it says that it is possible to put together a tree with a root node labeled D and a tree with a root node labeled N, to form a tree with a new root node labeled NP. Second, with regard to the structured expressions that are derived, it says that a node labeled NP is well-formed if it has two daughters, a left daughter labeled D and a right daughter labeled N. These "two things" that the rule says are redundant since, of course, they are really just the one thing said in two different ways. So the redundancy stems from the fact that a CFG really only has one thing to say, and while that one thing can be expressed and enforced either derivationally as in (5) or representationally as in (6), having the rule enforce it in *both* ways is redundant.

The crucial point to note is that because of this redundancy, it is plausible to take the final expression derived by the entire derivational process, namely the tree at the root node in (7), as the object that is grasped by a speaker, since — in contrast to the situation illustrated earlier in (5) and Figure 5 — grasping the entire derivational process provides no additional information beyond what is provided by grasping the final derived expression. If all derivational systems that worked with structured expressions were redundant in this way, then the conception I began with in Figure 1 would be reasonable. But my aim here is to show that this is not the case.

I will use the term *purely derivational* for systems like (3) and (5), where "everything the grammar says" is expressed derivationally; and *purely representational* for systems like (6), where everything is expressed representationally. The system illustrated in (7) is neither purely derivational nor purely representational — it is what I will call a *mixed* system, since the grammar makes both representational and derivational statements.[5] So to repeat the crucial point, although this first example of a mixed system has the property that the representational and derivational aspects are mutually redundant, there are other mixed systems that are not redundant in this way — instead, some parts of the important work are accomplished derivationally, and other parts are accomplished representationally.

One example of a mixed but non-redundant system is the framework of early transformational grammars in Miller and Chomsky (1963) and Chomsky (1965). A clear illustration of this comes from the famous comparison between the two sentences in (8) and (9) (see Miller and Chomsky, 1963, pp.476–480).

(8) John is easy to please.

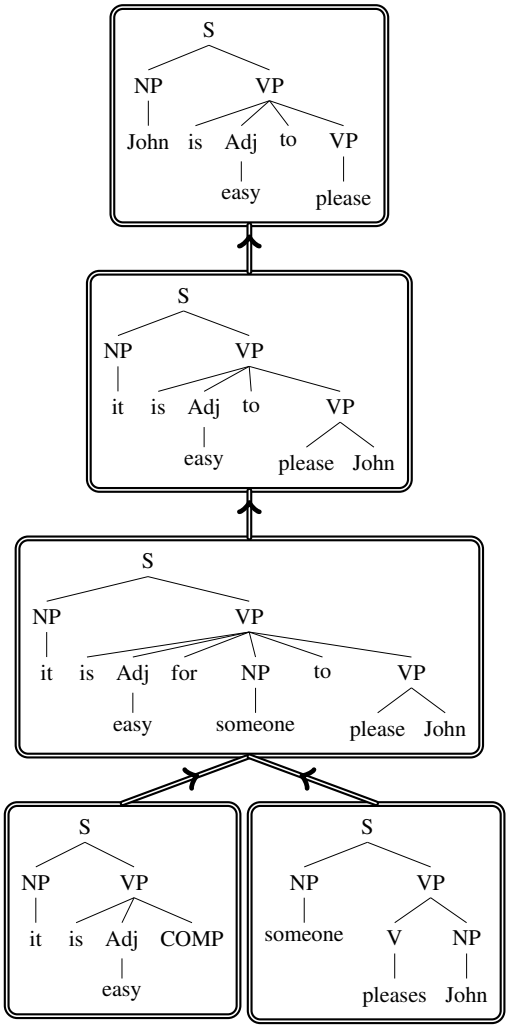
(9) John is eager to please.

Each of these sentences is derived by base-generating two monoclausal "underlying P-markers", and then manipulating and combining these P-markers (these are the expressions that this system

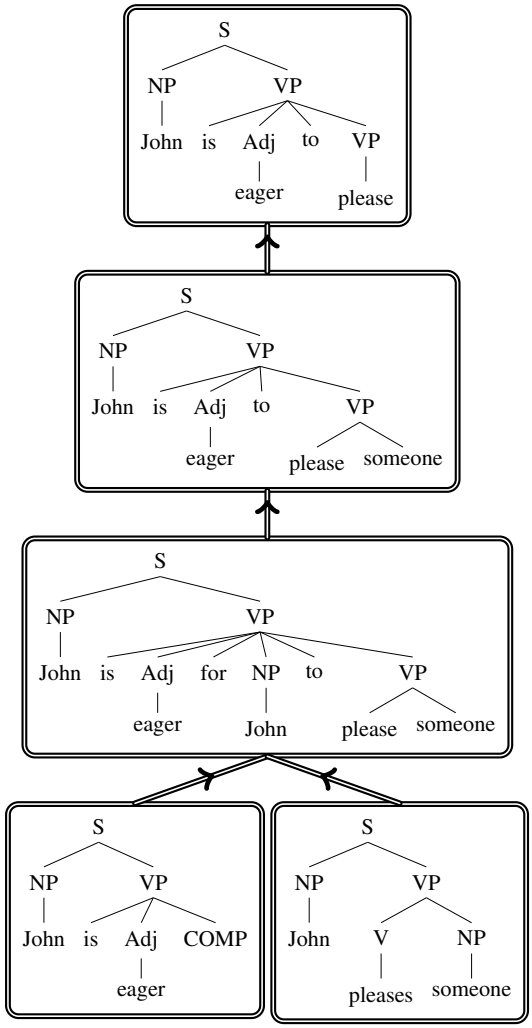
[5]Note that it does not make sense to ask whether the set of rules in (4) itself is purely derivational or purely representational (or mixed). It depends on whether those rules are interpreted as well-formedness conditions on static representations, or as statements about what can be derived from what.

works with) to arrive at a single "derived P-marker", as illustrated in (10) and (11).

(10)



(11)



Like the understanding of CFGs illustrated in (7), this is a derivational system that works with structured expressions (specifically, trees, rather than strings), so this is a mixed system. The grammar licenses certain derivational steps that relate P-markers to one another — specifically, certain transformations, such as the transformation that combines two S-rooted trees and the transformation that fronts an NP from an embedded object position to overwrite 'it' — and also imposes certain representational constraints ("surface filters"[6]) on the eventual derived expression. But unlike the first example of a mixed system in (7), this is not redundant: in this case, the work that is performed derivationally and the work that is performed representationally are separate, and accordingly grasping the entire derivational process provides more information than does grasping the final derived expression alone.

[6]This is not strictly true of the transformational grammars of the 1960s: filters as we currently know them were not introduced until the 1970s . . . but they were introduced into a system that still routinely "destroyed" information as the derivation progressed, so the crucial point remains.

Furthermore, it is clear that the intended interpretation of these early transformational grammars *did* involve the idea that a speaker encountering (8) or (9) grasped the entire derivational process illustrated in (10) or (11). This pair of sentences provides a dramatic illustration of this: the interesting point about this pair is that speakers understand them to have different structures in some important sense — as evidenced by the fact that speakers understand 'John' to be the pleasee in (8) but the pleaser in (9), and the fact that speakers know there is an expletive-'it' variant of (8) but not (9), etc. The crucial point to note is that the theory would not provide any account of these differences if one supposed that the object grasped by speakers were simply the eventual derived structures, because these two structures are identical (modulo the alternation of 'easy'/'eager' itself), as (10) and (11) make clear. In order to provide any explanation for the different ways in which speakers treat these two sentences, the derivational processes posited by the theory, i.e. the entire tree structures shown in (10) and (11), must be the objects thought to be grasped by speakers.

This point is not only logically necessary in hindsight, but was clearly the intended interpretation at the time:

> . . . we see that the grammatical relations of 'John' and 'please' in [(8)] and [(9)] are represented in the intuitively correct way in the structural descriptions provided by a transformational grammar. The structural description of [(8)] consists of the two underlying P-markers [at the bottom of (10)] and the derived P-marker [at the top of (10)] (as well as a record of the transformational history T_1, T_4, T_5). The structural description of [(9)] consists of the two underlying P-markers [at the bottom of (11)] and the derived P-marker [at the top of (11)] (along with the transformational history T_1, T_2, T_3). Thus the structural description of [(8)] contains the information that 'John' in [(8)] is the object of 'please' in the underlying P-marker [at the bottom right of (10)]; and the structural description of [(9)] contains the information that 'John' in [(9)] is the subject of 'please' in the underlying P-marker [at the bottom right of (11)].
>
> . . . [one component of the perceptual model] will utilize the full resources of the transformational grammar to provide a structural description, consisting of a set of P-markers and a transformational history, in which deeper grammatical relations and other structural information are represented.

Miller and Chomsky (1963, pp.479–480)

This kind of transformational grammar is therefore a mixed system where the final derived expression does not provide all of the grammatically relevant information — in essentially the same way as was noted earlier with respect to the purely derivational systems in (3) and (5). So having trees instead of strings as the derived expressions does not automatically make the derivational process redundant.

1.3 What kind of system is modern transformational syntax?

Against the backdrop of these distinctions — between purely derivational/representational and mixed systems, and between redundant and non-redundant mixed systems — we can now ask what kind of system contemporary versions of transformational grammar are. Clearly they are mixed systems of some sort, since they are specified derivationally and work with structured expressions, so the question is whether the derivational process that derives a structured expression provides additional information that is not encoded in the derived expression itself, like in (10) and (11), or is redundant, like in (7). To the extent that the derivational process provides additional information that theories appeal to, the background assumption that researchers are working with must be the view outlined in Figure 2 (because the view in Figure 1 would put this information "out of bounds").

It is at this point that we must contend with the "unfortunate quirk" of modern transformational grammars mentioned in the introduction. Over the decades a number of representational devices have been introduced that encode in the final derived expression information that previously was encoded only in underlying phrase markers, such as traces/copies and co-indexed silent elements like PRO. This has created a situation where, in very many cases, the eventual derived expression *does* uniquely identify the history of transformational operations (essentially, merge and move steps) that derived it. In such cases, recovering the derivational process itself is redundant, in much the same way as it is in (7); and the prevalence of such cases might create the impression that researchers are working with the view in Figure 1. But I will argue that this does not seem to be the case in general: even in the minimalist era, there are clear instances of proposals that only "make sense" under the view that the entire derivation is relevant (Figure 2), in ways that are formally analogous to the 'easy to please'/'eager to please' analysis discussed above. I discuss some of these in Section 1.3.1. A natural and important question to ask, admittedly, is why such cases have become so rare, i.e. why derivational information so frequently ends up "duplicated" via representational devices. I will argue in Section 1.3.2 that this is simply the result of historical accidents that have led to a theoretical architecture that makes thinking about these questions unnecessarily difficult.

1.3.1 Sensitivity to derivational history

The most obviously relevant development since the system illustrated in (10)/(11) is the introduction of traces. One possibility is that the introduction of traces coincided with a wholesale adjustment away from the perspective where complete derivations are the relevant objects, towards a view where only the derived structure matters. Two indications that this was not the case can be seen in arguments motivating the Strict Cycle Condition (Freidin, 1978, 1999) and in the proposal

by Lebeaux (1988, 2000) to account for anti-reconstruction effects by allowing late adjunction. More recently still, movement has generally been taken to leave behind not just a trace of the moved constituent, but rather a full copy of it. This has the consequence of duplicating even more information between the derivational process and its output, but even this development does not seem to have coincided with a switch to a view where only the derived structure matters. Appeals to information that cannot be gleaned from derived structure can be seen in, for example, Lasnik (1999) and McCloskey (2002).

As a first example, consider the argument for the Strict Cycle Condition based on the unacceptability of (12) (Chomsky, 1973). Freidin (1978, p.524) (see also Freidin 1999, p.100) points out that in the context of the assumption that intermediate traces of successive cyclic movement can be deleted, one needs both subjacency *and* the Strict Cycle Condition to rule out such a sentence.

(12) * What did he wonder who ate?

There are two relevant derivations to consider:

(13) he wondered [who ate what]
he wondered [who$_i$ t_i ate what]
what$_j$ he wondered [who$_i$ t_i ate t_j]

(14) he wondered [who ate what]
he wondered [what$_j$ who ate t_j]
what$_j$ he wondered [t_j who ate t_j]
what$_j$ he wondered [who$_i$ t_i ate t_j]

In (13), first 'who' moves to the embedded SpecCP position, and then 'what' is forced to moved in a single step to the matrix SpecCP position, violating subjacency. But on the assumption that intermediate A-bar traces can be deleted/overwritten, the fact that the derivation in (13) violates subjacency is not sufficient to rule out the sentence, because the derivation in (14) provides a way around subjacency: move 'what' to the matrix SpecCP in two subjacency-obeying steps, and then move 'who' into the embedded SpecCP position (overwriting the trace of 'what'). The additional constraint that is needed is the Strict Cyclic Condition, which prevents the order of operations in (14).

The important point for our purposes is that the two derivations in (13) and (14) produce the same final derived expression, as the last lines of each make clear. Thus it would make no sense to point out that, without the Strict Cycle Condition, (13) would be ruled out as desired but (14) would not, *unless* the things being ruled out and ruled in were derivations. Put differently: if we suppose that the expression on the last line of (13) is what the theory says is grasped by a speaker upon encountering the string in (12), then it would make no sense to say that although this expression is correctly classified as ungrammatical, something more must be added to our theory to rule out (the

expression on the last line of) (14). As in the 'easy'/'eager' example, the final derived expression *underdetermines* the entities that the theory is evidently taken to actually care about — namely, the derivations themselves.[7]

As another example of this kind of situation, consider the analysis of the contrast in (15) proposed by Lebeaux (1988, 2000).

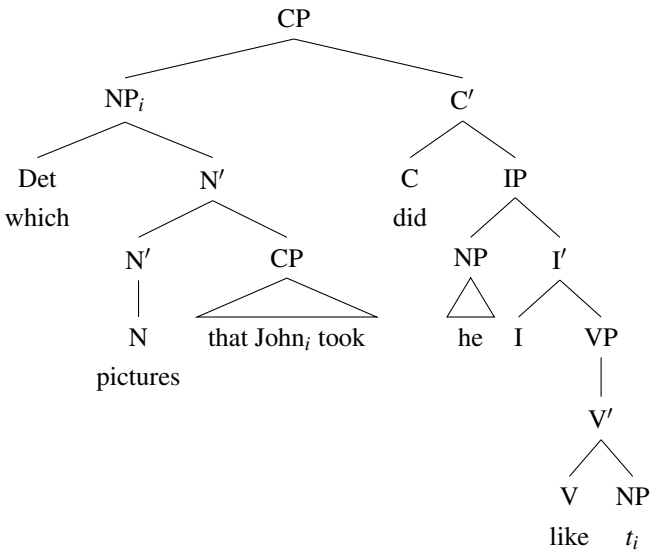
(15) a. Which pictures [that John$_i$ took] did he$_i$ like?

b. * Whose claim [that John$_i$ took pictures] did he$_i$ deny?

Lebeaux's influential account of this contrast involved supposing that the relative clause in (15a) could be added after the wh-movement transformation that fronts 'which report', since the relative clause is not required to be present in d-structure. The bracketed clause in (15b), however, being a complement rather than an adjunct, does not have this flexibility, and therefore has no way to avoid the Condition C violation induced by the co-indexed matrix subject 'he' at d-structure.

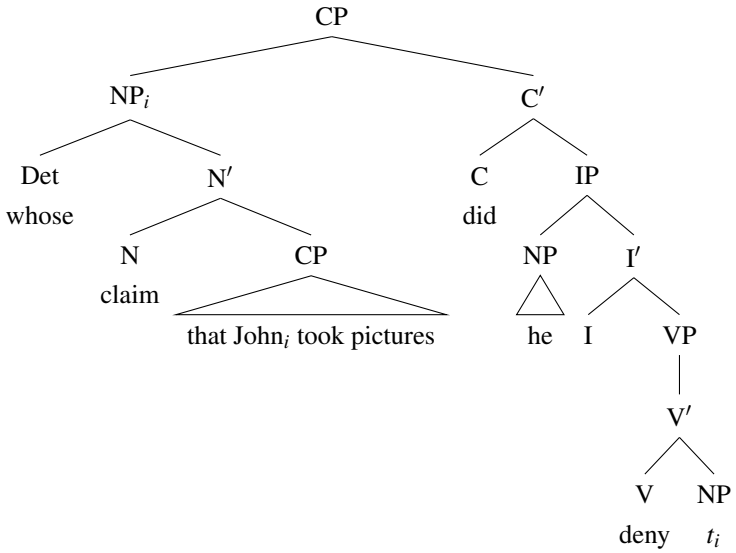
The crucial point for our purposes here is that this distinction between the derivation of (15a) that circumvents Condition C and the derivation of (15b) that violates it was not encoded in the s-structure phrase markers that were assumed at the time. The two trees shown in (16) do not *themselves* differ in any respect that is relevant to compliance with Condition C.[8]

(16) a.



[7]Freidin (1978) notes that a ban on the deletion of traces achieves the same result. But this does not change the fact that the argument for the Strict Cycle sketched here, which relies on the view that derivations are the objects being ruled in and ruled out, was taken as a valid pattern of reasoning regarding the consequences of not having such a ban. Freidin in fact argues for the approach that disallows trace deletion rather than the one that enforces the Strict Cycle Condition. The shift from deletable traces to non-deletable traces can be seen as part of the broader trend towards more and more "substantive" residues of movement, ensuring that more and more derivational history is encoded in the final derived object, culminating with the full-fledged copy theory of movement.

b.



The theory would not provide any account for the fact that speakers' judgements of (15a) differ from their judgements of (15b) if it were assumed that speakers grasped only the trees in (16). This is analogous to the way the early transformational grammars would not provide any account for speakers' differing judgements regarding the 'easy'/'eager' contrast if it were assumed that speakers grasped only the derived P-markers shown at the top of (10) and (11). Instead, the theory must be interpreted as claiming that speakers grasp the entire derivational process, including specifications of what the d-structure phrase marker looked like and whether or not the relative clause was adjoined via a transformation that followed the wh-movement of 'which claim': picture representations along the lines of (10) and (11) with d-structures at the bottom and the trees in (16) at the top, where one of the crucial transformations involves *adding* the relative clause to a post-wh-movement structure to produce (16a).

Note that the important point here is independent of whether Lebeaux's analysis is correct. What is significant is that there does not appear to have been any objection to the analysis based on the idea that *since* the crucial distinction is not encoded in the trees in (16), the theory cannot account for the contrast in judgements in (15). If it were standardly assumed that final derived expressions[9] were the objects that were grasped by speakers, then one would expect this objection to be raised.

[8]Lebeaux (2000, pp.107–108) is quite explicit about this: "There are two possible derivations for [(15a)]. In one, Adjoin-α applies prior to Move-α. ... In this derivation ... Condition C will apply to the intermediate structure, ruling it out. ... There is, however, another derivation [in which] Move-α ... applies before Adjoin-α. *This derivation gives rise to the appropriate s-structure as well*." (emphasis added).

[9]Perhaps the point becomes even clearer in light of the fact that there is no single "final derived expression" in the GB system assumed by Lebeaux (1988). Grasping only the s-structure phrase marker would provide no encoding of, for example, the scope of covertly-moving expressions; and grasping only the LF phrase marker would provide no distinction between, for example, the wh-phrases that are pronounced in fronted positions and the wh-phrases that are pronounced in their base positions in English multiple wh-questions. The only reasonable interpretation of GB-style

This kind of situation, where a derived expression underdetermines its derivational history, becomes less likely in the context of the recent shift to adopt copies rather than traces as the residues of movement. In particular, the two examples just discussed would no longer provide clear evidence that derivations are the relevant objects if we update them in accord with these more modern assumptions: the derived structure for (15a) would encode the crucial distinction between early and late adjunction by having a copy of the wh-phrase with or without the adjunct in the low position; and the assumption that residues of movement can be deleted/overwritten is (at best) difficult to reconcile with the idea that these residues are full-fledged copies, undermining the argument based on cyclicity. So we might ask whether, even if the derivational mindset that was clear in Miller and Chomsky (1963) was not abandoned with the introduction of traces in the 1970s, perhaps it was abandoned later with the introduction of copies.

But this also does not seem to be the case. For example, Lasnik (1999) raises the possibility that A-movement may not leave a copy, as an explanation for the fact that A-movement does not show reconstruction effects. As a consequence, the final derived expression would not encode the base positions of A-moved elements, and therefore could not be used to determine which theta roles they are assigned (or even whether they had been assigned theta roles, as the Theta Criterion or equivalent would require). Lasnik suggests instead that "θ-roles are 'checked' in the course of a derivation" (p.207) — for this to amount to any account of why speakers interpret sentences involving A-movement in the ways that they do, the background assumption must be that they grasp a complete derivational history. This is certainly not an uncontroversial proposal, and Lasnik notes that it departs from Chomsky's (1995) assumption that θ-roles are "determined specifically at the LF level" (i.e. in the final derived expression), but there is no sign that this departure requires an adjustment to the fundamental question of which objects are grasped by speakers. In fact Lasnik notes (p.208) that his proposal can be seen as a direct descendent of the way the Standard Theory takes θ-assignment to be a "base property", as illustrated with the 'easy'/'eager' example above.

Another example can be seen in McCloskey's (2002) account of certain facts involving complementizers in Irish. This proposal's "core claim is that the morpho-syntactic make-up of a head is influenced not by the syntactic material with which it is in a local relation, but rather by the mode of introduction of that material" (p.202). Specifically, a C head is pronounced as 'aL' if its specifier was filled by an application of move, and as 'aN' if its specifier was filled by an application of merge (and as 'go' if its specifier is not filled). So inspecting the contents of the C head's projection in the final derived expression will not suffice to determine which of these pronunciations is applicable for a given structure.[10] McCloskey notes that the proposal is unusual in this respect, but again there is no sign that he takes this novelty to involve an adjustment to our understanding

theories is to suppose that speakers grasp representations at all four levels — d-structure, s-structure, PF and LF — along with a history of the transformations that relate these four to each other.

of the fundamentals of what a syntactic derivation is.

1.3.2 The rise of representational devices

Let us suppose, then, that the implicit assumption in contemporary minimalist syntax is still that speakers grasp entire derivations, in the manner that is straightforwardly necessary for the 'easy'/'eager' contrast in (10) and (11). Why then have we seen such an increase in representational devices (which make this implicit assumption easier to overlook)? If the reason for these developments was not a shift to a point of view more in line with Figure 1, where the final derived object encodes all grammatically relevant information by design, then what was the reason? To the extent that no other reason can be identified, the argument I made in the previous subsection would be weakened.

To be concrete: why is it that contemporary theories would assign derived structures something like those in (17) to the 'easy'/'eager' sentences, rather than those shown at the top of (10) and (11)?

(17) a. John$_i$ is easy [t_i PRO$_{\text{arb}}$ to please t_i]

b. John$_i$ is eager [PRO$_i$ to please]

Note that it is no answer to simply say that "We need a co-indexed PRO there in order to represent the fact that 'John' receives the subject theta role from 'please' " — this begs the question, since we have seen that there are alternative, derivational ways of representing this information.

A problem that soon arise for the "purely deep structure" encoding of thematic information as in (10) and (11) was the fact that some movement operations feed semantic interpretation. For example, the quantifier 'every boy' can bind the variable in 'his' in (18a), but not in (18b).

(18) a. [Every boy]$_i$ seems to his$_i$ mother to be intelligent

b. * It seems to his$_i$ mother that [every boy]$_i$ is intelligent

Since the D-structures of these two sentences are equivalent in all relevant respects, it is not possible to maintain the simple Standard Theory (Chomsky, 1965) view that deep structures were the only objects relevant to semantic interpretation. Somehow the theory needed to allow semantic interpretation to be dependent on both D-structure, where thematic relations were encoded, and S-structure, where the scope of quantifiers and other operators was encoded. (See for example van Riemsdijk and Williams (1986, p.186) for discussion.)

[10]This is perhaps not as clear a case as some of the earlier examples, because even though inspecting the C head's projection will not provide the relevant information, inspecting the entire derived phrase marker will: if other copies of the phrase that fills the SpecCP position are present lower in the structure, then it will follow that the SpecCP position was filled by move. But McCloskey makes no mention of this and states the relevant criterion in terms of the derivational operations themselves. This would be surprising if he was working under the assumption that this information could only by recovered via inspection of copies.

Logically speaking, there are two different ways in which this could be achieved.

The first possibility is to take the input to semantics to be *derivational histories* rather than D-structures. The idea here would be to take each derivational operation to be associated with some particular semantic compositional rule, in the style of Montague (1974) and much subsequent work; in modern terminology this kind of approach is sometimes described as "directly compositional" (Barker and Jacobson, 2007). Specifically, the raising transformation that applies in the derivation of (18a) would affect the syntactic and semantic computations in parallel: it would displace the phrase 'every boy' into its matrix clause position on the syntactic side, and widen the scope of this phrase's interpretation on the semantic side.[11]

As it happened, however, this is not the way things proceeded. A second option was made possible by the introduction of traces: the input to semantic interpretation was taken to be not D-structure as in the Standard Theory, nor the derivation as a whole as in the first option just outlined, but rather S-structure. The thematic information that was not available at S-structure in the Standard Theory could now be retrieved at that level via the traces left by transformations that moved things out of their thematic positions. This in turn subsequently developed into the idea that the input to semantic interpretation is an "even later" level of representation, namely LF, with traces (or copies) still encoding all positions that constituents had moved through in earlier stages of the derivation.

But importantly, the possibility of this approach to semantic interpretation was a *by-product* of the presence of traces, not a *motivation for* introducing traces:

> The principal motivation for traces comes from the parallelism between movement structures and antecedent-anaphor relations. ...
> Essentially, movement must always be to a c-commanding position and an anaphor must always be c-commanded by its antecedent. ...
> we might say that a trace has anaphor properties and that the moved phrase has antecedent properties.
>
> van Riemsdijk and Williams (1986, p.141–142)

The posited connections between movement dependencies and antecedent-anaphor dependencies were developed further, to the point where by the mid-1980s, the distribution of PRO, A-traces and $\overline{\text{A}}$-traces was accounted for to some large extent by the Binding Principles. And the Binding Principles seemed to be very naturally understood as representational constraints, since their canonical and original purpose was to describe the distribution of various kinds of NPs which were

[11] This is exactly analogous to the way, in a categorial grammar, a single application of the function-application operation has parallel effects on the syntactic and semantic side: when this operation applies to "John :: NP" and "ran :: S\NP" the effect is (i) to concatenate the two strings to form 'John ran' on the syntactic side, and (ii) to apply the verb's meaning, say **ran**, to the subject's meaning, say **j**, to form **ran**(**j**).

taken at the time to be base-generated (rather than by-products of certain transformations). So when similarities were noted between the configurations in which reflexives could appear and the configurations in which raising was licit, the natural way to bring them under the same umbrella was to *suppose that there is an A-trace at s-structure* that is subject to the existing, representational constraint known as Principle A.

To the extent that this kind of logic drove the shift towards enriched representations, the shift had nothing to do with a preference for representational encodings of locality conditions, nor a preference for having a particular representational level as the input to semantic interpretation, nor with a preference for systems where one does not have to posit that speakers grasp derivational histories; rather it was simply an attempt to unify various primitives that had been empirically discovered to pattern together. If two things (reflexives and raising) behave alike and one (reflexives) is taken for granted to be constrained representationally, then it is natural to *create a representational reflex of the other* in order to bring the two into line.

Furthermore, the half of this scenario that was taken to be uncontroversially representational in the 1980s is arguably no longer thought to be so. The general trend in minimalist syntax has been to try to derive the effects of earlier *representational* constraints, such as Principle A, from *derivational* constraints on merge and/or move, such as some version of a Shortest Move condition (e.g. Hornstein, 1999, 2001; Kayne, 2002). Very broadly speaking: a derivational explanation of the ungrammaticality of (19), based on the fact that it involves a movement step that goes too far, would most likely be more in keeping with contemporary thinking than would a representational explanation of the ungrammaticality of (20), based on the fact that it involves a trace/copy that is not appropriately bound/licensed.

(19) * John$_i$ thinks that Mary likes himself$_i$

(20) * John is likely that it seems to be tall

To the extent that the locality constraints on raising, for example, are nowadays explained via minimality-style limits on the applicability of movement transformations, nothing would be lost by reverting to a system roughly along the lines of 1960s transformational grammars, with no traces or copies left by raising: the traces/copies are no longer of any relevance to Principle A. (And recall that "we need a co-indexed silent element there in order to encode the theta role that 'John' received in its base position" only begs the question.) In other words, we have roughly reverted to using derivational mechanisms to constrain both antecedent-anaphor dependencies and movement dependencies, as was the case *before* the introduction of traces; but despite the fact that traces were introduced with the aim of representationalizing those constraints (a purpose that they no longer serve), we have maintained the assumption that movement leaves some kind of representational residue.

So not only was the representational shift driven by empirical practicalities rather than architectural preferences, but the assumptions that made the shift practical in the 1970s and 1980s arguably no longer hold. If this analysis is correct, then it suggests that many of the representational encodings of derivational history in modern syntax — for example, the unpronounced copy left in the lower clause of a raising construction — are unnecessary. Put differently, we have a mixed system where there is a certain amount of redundancy, but it is arguably the *representational* aspects that are redundant, not the derivational ones.[12]

1.4 Interim summary

This section has had two aims. The first aim was to establish what it looks like for a theory of grammar to suppose that the objects being grasped by a speaker are derivations. This comes out most clearly in the case of systems like categorial grammar, but importantly there are also mixed systems that derive structured expressions (for example, trees) and yet also require this same derivational interpretation. The second aim was to demonstrate that although it is no longer as clearly the case as it was in the early days of the 1960s, generative grammar has never ceased to be a system of the mixed kind that takes derivations themselves to be the objects to be grasped by speakers. The instances where this can be seen have become rarer over the years for unrelated empirical reasons (which arguably are less relevant now than they were in the GB era), but certain specific well-known points in the literature make it clear that this is still the intended interpretation.

If we accept this conclusion about the cognitive commitments of modern generative grammar, then we should expect that in principle there will be ways to empirically distinguish theories on the basis of their derivational claims — not the claims they make about derived expressions (for example, which ones are well-formed and which ones have which particular interpretations), but the claims they make about the derivational processes of which those expressions are the end result. My goal in the rest of this paper is to show one way of cashing out these claims. To sharpen the issue, I will consider two versions of the theory that differ *only* in their derivational processes: the set of expressions derivable by the two are identical (as are their classifications of which expressions are grammatical, and which have certain interpretations, etc.).

[12] Brody (2002) observes the same redundancy, but argues to eliminate it in the opposite way: switching to a completely representational theory, where the idea would be that both (20) and (19) should be ruled out by representational constraints in roughly the manner of GB systems. The purpose of this paper is only to explore the option that retains derivations, since this seems like a less drastic departure from contemporary mainstream thinking, so I leave aside proper consideration of the relative merits of this option versus the alternative that Brody proposes.

2 A derivational theory of minimalist syntax (or two)

In this section I will present two minimally-different versions of minimalist syntax. The two versions agree entirely on the set of derivable final expressions, and differ only in the derivational processes that are taken to construct those expressions. Specifically, in one version merge and move are two distinct primitive structure-building operations, and in the other the structure-building functionality of merge and move is abstracted out into a single primitive operation.

If the final derived expressions are all that play a part in the cognitive claims of a grammatical theory, then these two versions of the theory will obviously be empirically indistinguishable. I have shown in the previous section that, in occasional cases, the derivational properties of a theory (i.e. the fact that speakers grasp complete derivations) are relied upon in an account of acceptability facts of the sort standardly used in syntactic research — and therefore, that is reasonable (and indeed necessary, if existing arguments in the syntactic literature are to be taken seriously) to suppose that speakers grasp complete derivations. Here I hope to show that other empirical measures can also be sensitive to the derivational properties of a theory. In other words, the contributions of derivational processes to accounts of acceptability facts are not an artifact of some peculiarities of the ways in which grammars relate to acceptability judgements; they are a part of the quite general claims that are made by positing a generative grammar as a component of a speaker's mind. As I will show in Section 3, the two versions of minimalist syntax that I introduce here can (in combination with reasonable linking hypotheses), make distinct empirical predictions about sentence comprehension difficulty phenomena and about the choices a learner will make between candidate grammars.

2.1 Merge and move as distinct primitives

I will start by presenting a relatively standard version of minimalist syntax in this section, from a perspective that emphasizes the place of derivations in the sense outlined in Section 1. In Section 2.2 I will present the derivationally-distinct alternative by highlighting the ways it differs from the system introduced here.

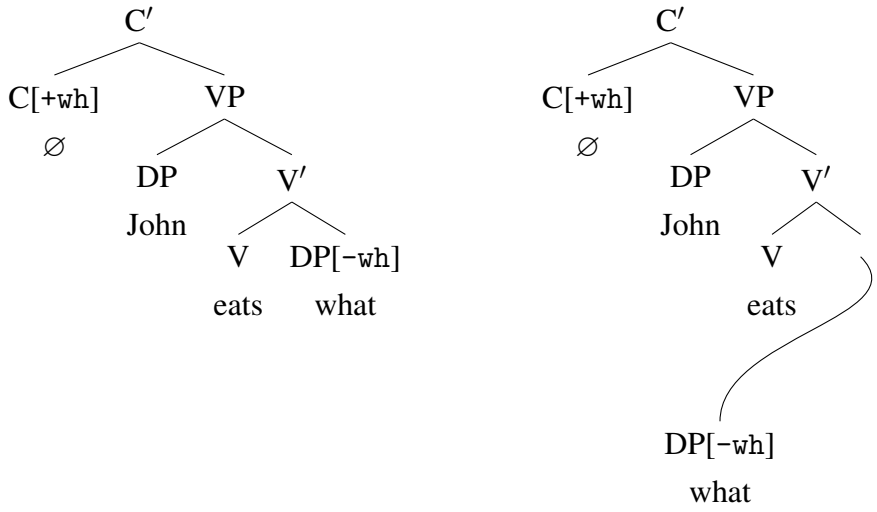
As an example, consider the derivational process underlying (21). Here and throughout this section I will ignore head movement: for simplicity, I will suppose that this is a simple wh-question in a language just like English, but lacking auxiliary inversion (or alternatively, an embedded question in English). I will also make a number of simplifying assumptions about the particulars of clause structure (e.g. ignoring the TP layer).

(21) what John eats

In particular, consider the expression that has been derived immediately preceding the final wh-

movement step: the wh-phrase is in the direct object position, but has an unchecked feature indicating that it must move to another position for the derivation to be valid. I will represent this by labeling the phrase DP[-wh] (as opposed to simply DP). In addition, to highlight the way phrases with unchecked features have "unfinished business" that needs to be completed by some subsequent derivational step (in contrast to the way the DP 'John', for example, has done everything it needs to do), I will adopt a notation for tree structures where phrases with these unfulfilled requirements stand out visually, as shown on the right in (22). It bears emphasizing that this unusual graphical convention says nothing more than what was already said by annotating the 'what' node with an unchecked -wh feature in the more conventional diagram on the left in (22). It is no departure from standard minimalist assumptions. I adopt it here only because it will help to clarify the relationship between the two subtly different derivational implementations of movement that I am introducing in this section.

(22)



Adopting the notational conventions from Section 1, I will represent the full derivation of 'what John eats' as shown in Figure 6. The tree in Figure 6 should be thought of as an encapsulation of a derivational history in much the same way as (10) and (11) above (on page 13).[13] A few specific points are worth noting.

First, notice that as soon as 'eats' and 'what' are merged, we have a structure where one component has "unfinished business" in the sense introduced above, and therefore the tree structure containing only these two words already shows 'what' set aside in the manner introduced in (22).

Second, notice that the step that combines 'eats what' with 'John' is shown with the former on the left and the subject on the right in the derivation structure. This has nothing to do with

[13] One difference is that the trees in (10) and (11) showed only the transformational part of the derivations of the relevant sentences, ignoring the base component's construction of the underlying P-markers, and therefore show tree structures at the leaves of the derivation. The tree in Figure 6, in contrast, shows primitive lexical items at the leaves, since all structure-building is performed by generalized transformations in this system.

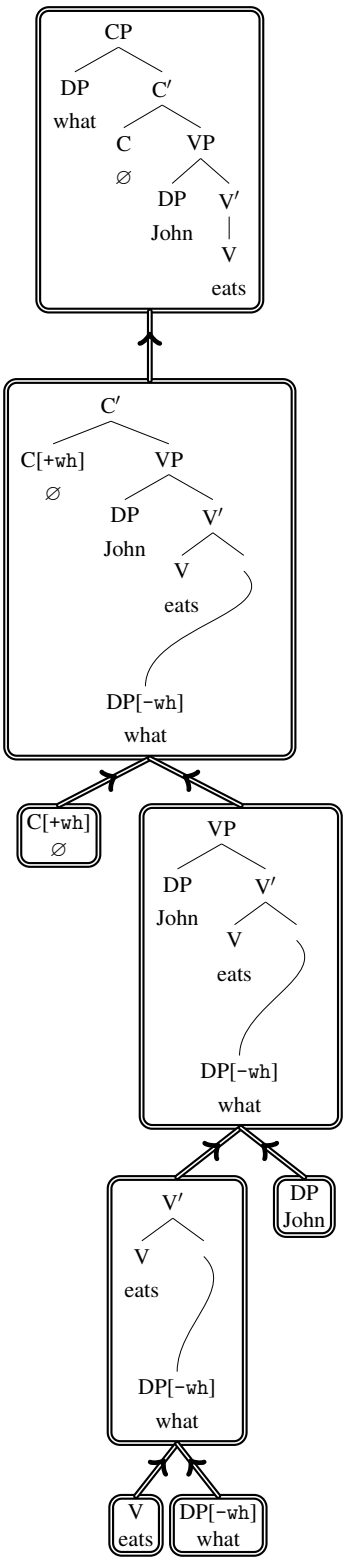

CP
DP
what
C′
C
∅
VP
DP
John
V′
V
eats
C′
C[+wh]
∅
VP
DP
John
V′
V
eats
DP[-wh]
what
C[+wh]
∅
VP
DP
John
V′
V
eats
DP[-wh]
what
V′
V
eats
DP[-wh]
what
DP
John
V
eats
DP[-wh]
what

Figure 6

the eventual linear order of these two constituents, nor with the order in which they appear in the resulting derived VP structure, shown immediately above. It simply records the fact that 'eats what' is the "selector" (here, more specifically, theta-assigner) and 'John' is the "selectee" (here, more specifically, theta-assignee), and as the rest of the tree makes clear, merge steps are recorded with the selector, or the element which projects, on the left (e.g. 'eats' and the null C head) and the selectee on the right (e.g. 'what' and the completed VP). Although no ambiguity would arise if this convention were not maintained, I will do things this way in order to bring out the distinction between the structure of the derivation (what combines with what, indicated with double lines and arrows) and the structure of the derived expressions.

Third, the final step of the derivation is a move step. This is a unary operation, which takes one derived expression as its input to produce a new derived expression — in contrast to a binary operation such as merge, which takes two derived expressions as input — much like the unary transformation that fronts 'John' in the final step of the derivation shown in (10). After this move step, 'what' has no remaining unchecked features and therefore is shown having settled fully into its final position in the derived tree. For expository purposes, I am assuming, somewhat unconventionally, that no copy or trace is left in object position. Nothing significant would change if a copy were shown in the final derived tree, but my choice here is intended as a reminder that the final derived expression does not *in general* uniquely determine a history of derivational operations (even if it would in most theories' analyses of this particular simple sentence). If we do things this way, then the fact that 'what' bears the object theta role is encoded by the earlier derivational steps rather than in the final derived expression, just as was the case for the fact that 'John' is the underlying object in (10); and in addition, to take but one example, strong crossover will need to be stated as a derivational constraint on wh-movement rather than by supposing that the residue of wh-movement is an R-expression constrained by Condition C. Whatever the explanatory virtues of assimilating strong crossover to Condition C by including some residue of 'what' in the final derived expression in Figure 6, however, the motivation behind my choice is, to repeat, simply to provide a reminder that derivational histories are sometimes relied upon as the sole record of certain pieces of information. Recall, for example, the suggestion of Lasnik (1999) and Fox (1999) that A-movement leaves no copies, which, whatever its pros and cons, is not taken to make it impossible to enforce the requirement that all DPs receive theta roles. My choice to not show copies is simply a reminder of those sorts of possibilities.

2.2 A single structure-building operation

I will turn now to the alternative derivational procedure that generates the same range of derived expressions as the system just outlined in Section 2.1.

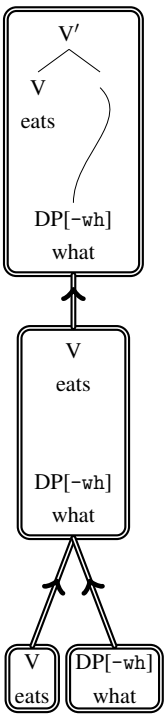
Recall that when subconstituents of a tree have unchecked features that require future movement operations, I have drawn these subconstituents below the rest of the tree, as illustrated in (22) — intuitively, one can think of them as waiting in a kind of buffer or "holding zone" for the opportunity to fulfill their remaining combinatory requirements. It is this holding zone, naturally enough, that movement operations draw on when, for example, a C′ constituent has been constructed whose head bears a `+wh` feature and can therefore check the `-wh` feature of a waiting phrase, as shown in the last step of Figure 6.

The idea behind the unification of merge and move into a single structure-building operation, as I will implement it here, is to suppose that not only move but also merge draws on this same "holding zone". So it will not only hold phrases that are waiting to move into certain structural positions, but also phrases that are waiting to *merge* into certain structural positions. It is the shared use of this holding zone that unifies all instances of structure-building in this system, which departs somewhat from standard intuitions regarding the unification of merge and move (the latter being an instance of the former, perhaps in combination with copy); see Hunter (2011) for much discussion, drawing on Stabler (2006). But for present purposes all that is important is that it provides a minimally-different conception of the derivational processes that produce the same range of derived expressions as the more standard system introduced above — and one that has been formulated explicitly enough to (i) allow us to be certain that the two systems do indeed generate the same range of derived expressions, and (ii) allow us to integrate both systems into models of parsing and learning to conduct the kind of tests that will follow in Section 3.

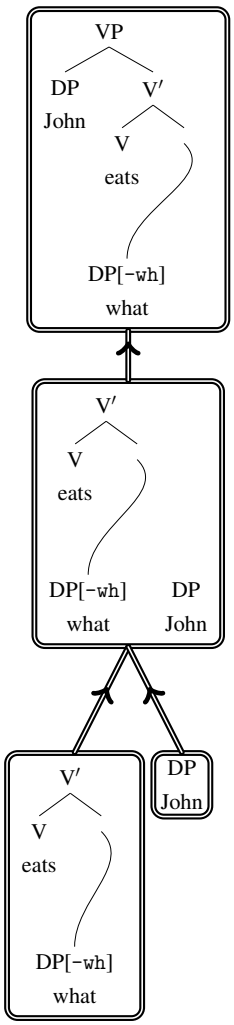
To illustrate, consider the derivational steps that combine the verb 'eats' with its two arguments. For each argument, the effects of what was previously the merge step that introduced it and combined it with (a projection of) 'eats' are now achieved by two distinct derivational operations in succession: the first of which I will call *insert*, and the second of which I will call *build*. The build operation is the one that is a generalized version of both merge and move from above; the insert operation takes up the slack of the extra book-keeping that is created by this unification. What insert does is introduce new material into an expression without fulfilling any of this new material's requirements; for example, (23) shows a small part of a derivation, the first step of which is an insert step that introduces 'what' into the derivation without putting it into a position that fulfills its requirement of a theta role. Instead, it is simply placed in the "holding zone" shown at the bottom of these double-boxed expressions. This corresponds to the fact that at this point 'what' certainly has unfinished business, in fact two kinds of unfinished business (because it is not even "started business"): both the establishment of a theta role and the requirement to move into an operator position remain to be completed. The second step shown in (23) is a build step. The build operation is essentially identical to move operation as presented in Section 2.1: it draws on material waiting in the holding zone, to establish dependencies required by the "main part" of the tree,

shown at the top of the double-boxed expression. The sense in which this system unifies merge and move is that both "first merge" and "re-merge" involve the build operation, drawing something from this holding zone. After the build step establishes its thematic dependency with 'eat', 'what' still has unfinished business in just the same sense that was discussed earlier, namely the requirement encoded by the `-wh` feature, so it remains held out, waiting for an opportunity to fulfill this final requirement. The holding zone contains elements that have *one or more* as-yet-unfulfilled requirements.

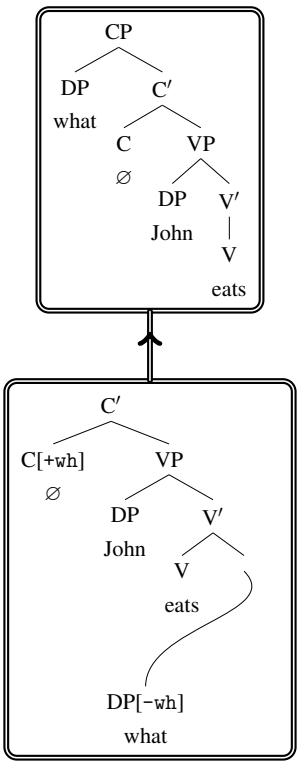
(23)



(24)



(25)



The next two derivational steps are shown in (24). These two steps are another insert-build "pair" that together have the effect of what was a single merge step in the system of Section 2.1. Here it is 'John' that is first added, without having any dependencies established, by an insert step, and then subsequently drawn on by build to establish the necessary external theta role dependency. In the intermediate derived expression in (24), 'John' is shown in the holding zone just as 'what' was in the intermediate derived expression in (23). But unlike 'what', 'John' has no further busi-

ness to conduct beyond thematic requirements, and so after the build step in (24) it is shown fully settled into its final position.

After a third insert-build pair of steps (corresponding to the third merge step of the derivation shown in Figure 6) that combines the C head with its completed VP complement, the derivation of 'what John eats' ends with the build step shown in (25). Note that although this build step corresponds to what was previously a move step, it is not formally different from any of the previous build steps that corresponded to (parts of) merge steps: it establishes a dependency between an element waiting on the holding zone and the head of the main tree. If the waiting element has further requirements that remain unfulfilled after a build step, then it remains in the holding zone to await a future build step that will satisfy the next of its further requirements, as in the build step at the top of (23); if it does not, then it will not need to participate in any further derivational operations and is fully integrated into the tree, as in the build steps at the top of (24) and (25).

I will call the system introduced here *Insertion Minimalist Grammars* (IMGs), in contrast to the more standard system in Section 2.1 which I will call *Minimalist Grammars* (MGs). This terminology follows the technical literature where more details of these two systems and the relationship between them can be found; see for example Stabler (2011, 2006); Hunter (2011).

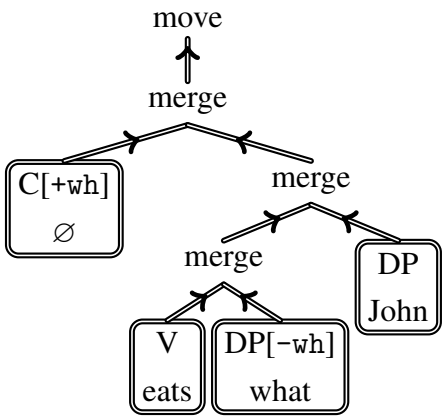
2.3 Interim summary

Notice that the final expression derived by the IMG in (25) is identical to the final expression derived by the MG in Figure 6. It should be intuitively clear that any expression that can be derived by one of these systems can be derived by the other, but also that the structure of the two corresponding *derivations* will be slightly different. Given the conclusions reached in Section 1, this means that the two theories make distinct claims about the objects that are grasped by speakers — they are therefore just as distinct as two representational theories that posit structurally different representations.

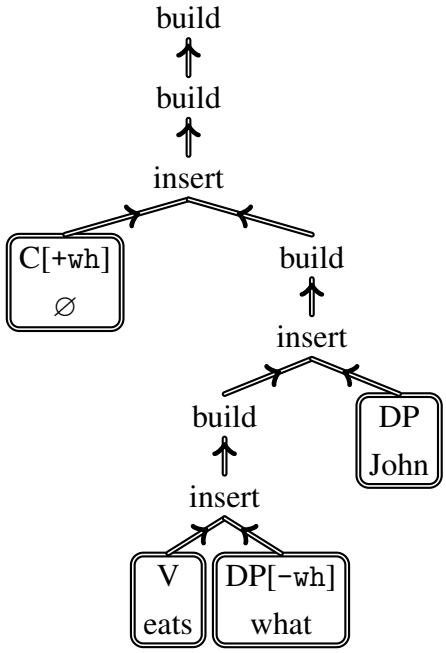
Full derivation tree structures of the sort shown in Figure 6 for an MG are unwieldy, and even moreso for IMGs. An alternative notation that allows for a more direct comparison between the two systems labels the internal nodes of derivation trees simply with the name of the operation that applies at the corresponding derivational step. This loses no information, because all of the derivational operations we are considering here are functions: if we know that merge applied to 'eat' and 'what', for example, then we have all the information we need to work out what the resulting derived expressions is. So to save space, we can represent the MG derivation from Figure 6 much more compactly as shown in (26); for comparison, the corresponding IMG derivation (previously shown only partially and piecemeal in (23), (24) and (25)) can be represented as shown in (27). These are analogous to the "T-markers" of early transformation theory; see e.g. Chomsky 1965,

p.130.

(26)



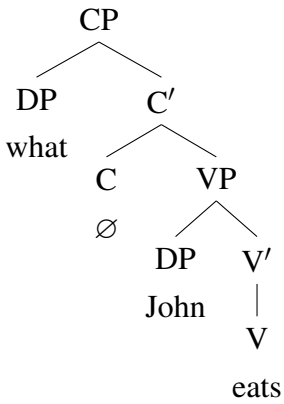
(27)



3 The empirical reflexes of derivational processes

I now turn to the task of demonstrating that, in two simple case studies, there are empirical consequences to the choice between (i) supposing that the expression shown in (28) is derived by the combination of merge and move steps shown in (26), as the MG theory would have it, and (ii) supposing that it is derived by the combination of insert and build steps shown in (27), as the IMG theory would have it.

(28)



Specifically, I will show that these two proposals make distinct predictions — holding all other factors fixed — with regard to sentence comprehension difficulty and with regard to the choices a

learner will make among a given range of grammars.

If it is shown that these two theories make empirically-distinguishable cognitive claims, then it should be clear that the "empirical payload" of a grammatical theory should not in general exclude the derivational properties of that theory: in other words, Figure 2, rather than Figure 1, more accurately characterizes the cognitive claims of a derivational theory.

This general point, of course, has nothing specifically to do with, for example, the two particular derivational systems I am considering here, or with the virtues of the idea of unifying merge and move, or with the degree to which the approach I have taken to this unification is in line with other proposals in the syntax literature. These two systems just provide a simple setting for tackling the abstract problem of relating derivational processes to various kinds of empirical predictions. Similarly, the details of the two case studies that follow — for example, the use of surprisal as a complexity metric, or the use of maximum likelihood estimation on the part of a learner — are also largely independent of the main concerns here. Some such assumptions, and some collection of linking hypotheses, must be chosen in order to make the questions concrete; replacing the choices I have made with others would no doubt change the empirical predictions that I derive, but would leave unaffected the broader point that such empirical predictions do follow.

3.1 A probability model

In many of the cases where grammars play a part in some cognitive model, they do so by being supplemented with probabilities. The two empirical domains that I will deal with in the case studies below are both instance of this: information-theoretic complexity metrics such as surprisal are computed as some function of a distribution over sentences, and probabilistic learning models often involve calculating the likelihood of the observed input relative to a certain hypothesized grammar in order to assess the fit of the grammar to these observations. One way for the predictions of such models to be sensitive to the distinction between MGs and IMGs, then, is for the probability distributions definable over the common set of generated expressions to be sensitive to this distinction. This is the approach that I adopt.

In this section I will give a very brief overview of how to supplement grammars of the sort introduced in Section 2 with probabilities, in such a way that produces different results depending on whether one adopts an MG or an IMG that produces the same set of derived expressions. Readers who are ready to assume that this can be done can safely skip to Section 3.2; readers who would like more information on the technical details than is provided here should consult Hunter and Dyer (2013).

A conventional, non-probabilistic grammar can be thought of as defining a space of probabilistic grammars, each of which defines a particular probability distribution over the objects generated

by the original grammar. To add probabilities to a grammar — whether a simple context-free phrase structure grammar, or an MG, or an IMG — is therefore to choose from the space of associated probabilistic grammars, and in particular this is usually done by choosing values for some collection of real-valued parameters. In the case of a CFG such as the one in (29), we can think of there being one parameter for each rule; these parameters are the λ_1, λ_2, etc. shown in (29). Then the task of choosing one of the many probabilistic versions of this grammar is the task of choosing values for the parameters λ_1, λ_2, etc.

(29)
λ_1	S $\rightarrow$ NP VP
λ_2	NP $\rightarrow$ John
λ_3	NP $\rightarrow$ Mary
λ_4	NP $\rightarrow$ D N
λ_5	VP $\rightarrow$ ran
λ_6	VP $\rightarrow$ walked
λ_7	D $\rightarrow$ the
λ_8	N $\rightarrow$ dog
λ_9	N $\rightarrow$ cat

How does fixing values for these *parameters* have the effect of attaching *probabilities* to the grammar's rules (and, as a result, to its derivations)? There are many possibilities, but on the standard way of doing things the probability of NP rewriting as 'John', for example, is the value $\frac{\lambda_2}{\lambda_2+\lambda_3+\lambda_4}$; the probability of NP rewriting as 'Mary' is $\frac{\lambda_3}{\lambda_2+\lambda_3+\lambda_4}$; and the probability of VP rewriting as 'ran' is $\frac{\lambda_5}{\lambda_5+\lambda_6}$; and so on. And the probability of a particular derivation in this grammar is the product of the probabilities of the rules that are used in the derivation.

Typically what we would like to do is to supplement an existing grammar G with probabilities (i.e. choose values for the parameters) in the manner that maximizes its degree of fit with some body of training data D. What does it mean to maximize degree of fit? Again there are many possibilities, but a common and simple answer is that we would like to choose values for the vector of parameters $\boldsymbol{\lambda}$ that maximizes the likelihood of the data D according to the probabilistic grammar $G_{\boldsymbol{\lambda}}$ (i.e. the grammar supplemented with probabilities according to $\boldsymbol{\lambda}$). This is known as "maximum-likelihood training" since the quantity it maximizes is the likelihood $P(D|G_{\boldsymbol{\lambda}})$. In the case of a CFG, this is a relatively simple process: if one sets the parameter λ_2 to be the number of times that NP rewrites as 'John' in the training corpus, and sets the parameter λ_5 to be the number of times that VP rewrites as 'ran' in the training corpus, etc., then one arrives at the values of these parameters that maximize this likelihood.[14]

[14] And this has the effect, of course, that the probability of VP rewriting as 'ran' will be the number of times that VP rewrote as 'ran' in the training corpus (i.e. λ_5), divided by the number of times VP rewrote at all in the training corpus (i.e. $\lambda_5+\lambda_6$).

Due to a formal property of MGs established by Michaelis (2001), it turns out that a broadly similar strategy to the one just outlined can be adopted for supplementing MGs (and also IMGs) with probabilities. The range of possible derivations in these grammars can be characterized via a branching process that has exactly the same structure as a context-free grammar; the relationship between this branching process and the surface strings (and indeed the surface tree structures) generated by the grammar is more complex and less transparent in the case of MGs than it is for CFGs, but this difference is irrelevant to the task of defining a probability distribution over the objects that a grammar generates. Hale (2006) made use of this fact to supplement MGs with probabilities.

What this underlyingly context-free branching process provides is a characterization of what the "choice points" are that a machine would encounter when carrying out possible derivations licensed by the grammar, and what the competing candidate options are at each point: this corresponds to knowing, in the case of the CFG above, that there are points where you need to decide between 'John' and 'Mary' and 'D N' as the expansion of NP, and there are points where you need to decide between 'ran' and 'walked' as the expansion of VP, etc. In a CFG, any particular grammatical rule only ever enters into consideration at one such choice point: if a rule's left-hand side is NP, then it enters into consideration at the choice point corresponding to deciding how to rewrite the symbol NP, and no others. But in an MG or an IMG, the relationship between the *choice points* and the *grammatical rules* is more complex. There are various ways in which one might flesh out the notion of a "grammatical rule" in these systems. Following Hunter and Dyer (2013), I will suppose that grammatical rules are roughly things like "merge to assign a theta role" or "move to check a `-wh` feature" (in an MG) or "build to check a `-wh` feature" (in an IMG) — for other reasonable choices, the fact remains that the relationship between choice points and grammatical rules is complex and many-to-many.

Given this assumption about what we take to be the rules that the grammar is trafficking in, it is natural to design a probability model where the *parameters* have interpretations that relate to notions like merge steps, move steps, theta roles, wh-features, build steps and insert steps. Broadly speaking, the parameter λ_2 in (29) is a measure of "how much NP gets rewritten as 'John' ", λ_5 is a measure of "how much VP gets rewritten as 'ran' ", etc.; and accordingly training data is taken to provide information about "how much NP gets rewritten as 'John' ", etc. So the model proposed for MGs by Hunter and Dyer (2013) includes parameters that are measures of "how much merge happens", "how much `wh` features get checked", "how much move happens", etc.; and for IMGs, there are measures of "how much build happens", "how much insert happens", etc. The relationship between these *parameters* and the *probabilities* that are multiplied together to determine the probability of entire derivation is complex (more complex than it is for CFGs), for precisely the reason that the relationship between the *grammatical rules* and the *choice points* is

overlapping and complex.

These complications aside, the model has the same form as the one explicated for CFGs above in that (i) it involves a choice of parameter values, which in turn determine probabilities, and (ii) one can use a training corpus to choose the parameter values $\boldsymbol{\lambda}$ that maximize the value $P(D|G_{\boldsymbol{\lambda}})$. What differs is that in the case of an MG, choosing parameter values can be interpreted as answering questions about "how much merge happens" (for the parameter λ_{merge}) and "how much move happens" (for the parameter λ_{move}), whereas in the CFG the questions being answered include "how much NP rewrites as 'John' " (for the parameter λ_2, which could also have been called $\lambda_{\text{NP} \rightarrow \text{John}}$). And, crucially, in the case of an IMG the parameter values being chosen during training are different again: they correspond to answering questions about "how much build happens" (λ_{build}), "how much insert happens" (λ_{insert}), etc. The answers that a given body of training data provides to the MG-based questions about merge and move steps will in general be different from the answers that this same training data provides to the IMG-based questions about build and insert steps.

3.2 Case study: Sentence comprehension difficulty and surprisal

Surprisal is an information-theoretic complexity metric that has been hypothesized to predict human sentence comprehension difficulty (Hale, 2001; Levy, 2008). Given a probability distribution over sentences, and a particular sentence whose processing we are interested in, a surprisal value is defined for each word in the sentence. This sequence of values is taken to represent the difficulty of integrating the information provided by each word as the sentence is read or heard incrementally.

Specifically, given the sentence $w_1 w_2 \ldots w_n$, the surprisal at word w_i is

$$-\log P(W_i = w_i | W_1 = w_1, W_2 = w_2, \ldots, W_{i-1} = w_{i-1})$$

The probability here is simply the probability of encountering the word w_i in that position, given all the preceding context. The negative logarithm is a monotonic decreasing function, and therefore has the effect of converting high probabilities to low surprisal values, and converting low probabilities into high surprisal values.

In the concrete case that I am presenting here, I will work with grammars generating all and only the sentences shown in (30):

(30) boys will shave
boys will shave themselves
who will shave
who will shave themselves

some boys will shave
some boys will shave themselves

I make relatively obvious (i.e. English-like) assumptions about the structures of these sentences. The one somewhat unusual aspect of the analyses I adopt is that reflexives are generated via a doubling-style movement theory: in 'boys will shave themselves', for example, it is 'boys themselves' that combines as the object of 'shave', and 'boys' then moves up to the SpecTP position. (This is in order to maximize the number of "merge versus move" choices while keep the derivations as small as possible overall.)

Given a common lexicon where the words that appear in these sentences are annotated with appropriate features, I will consider the relationship between the MG that generates (30) and the IMG that generates this very same set of sentences. Notice that in addition to generating the same set of *strings*, these grammars implement the same *analyses* of these strings. By this I mean that the two grammars make all the same assumptions about "what goes where" in the course of the derivation — they differ only in whether these same interactions among words and phrases are effected by merge and move steps or by insert and build steps.

Suppose we adopt the following (artificial, and entirely arbitrary) "corpus" as the training data that will provide the basis for choosing probabilistic versions of our two grammars. The number at the beginning of each line is the frequency of the sentence in the training data.

(31)
10	boys will shave
2	boys will shave themselves
3	who will shave
1	who will shave themselves
5	some boys will shave

For the reasons outlined above, this training data will be "interpreted" differently depending on whether one is using it to train the MG or the IMG. As a training corpus for adding probabilities to the MG that generates the sentences in (30), it is a collection of merge and move events that provide a basis for estimating the parameters λ_{merge} and λ_{move} (as well as the others that relate to specific features). Choosing a value for each of these MG-based parameters picks out a particular probabilistic MG, which in turn defines a particular probability distribution over the set of sentences in (30). The values of these MG-based parameters that this training corpus leads to pick out a probabilistic MG that defines the following distribution:

(32)
0.35478	boys will shave
0.35478	some boys will shave
0.14801	who will shave
0.05022	boys will shave themselves
0.05022	some boys will shave themselves
0.04199	who will shave themselves

From the perspective of the corresponding IMG, however, the training corpus provides a basis for estimating the parameters λ_{insert} and λ_{build} (as well as the others that relate to specific features). The probabilistic IMG that is picked out by using the same training corpus to estimate the values of these parameters defines the following, distinct, distribution over the same set of sentences:

(33)
0.35721	boys will shave
0.35721	some boys will shave
0.095	who will shave
0.095	who will shave themselves
0.04779	boys will shave themselves
0.04779	some boys will shave themselves

Note that even the one sentence that was not in the training corpus, 'some boys will shave themselves', is assigned different probabilities by the two grammars. Although the two grammars assign the same analyses to all six sentences, the information provided by the common training corpus bears on the probability of this unseen sentence differently depending on whether one adopts the MG-based or IMG-based probability model.

From here it is a simple final step to complete the picture: calculations of surprisal values for 'who will shave themselves' derived from the MG-based distribution are shown in (34), and the corresponding calculations using the IMG-based distribution are shown in (35). (I have chosen this sentence because it shows a relatively striking difference, but the same point could be made with any of the other sentences.) The surprisal values, and therefore the predicted degrees of sentence

comprehension difficulty, differ.

(34)
$$\begin{aligned}
\text{surprisal at `who'} &= -\log P(W_1 = \text{who}) \\
&= -\log(0.15 + 0.04) \\
&= -\log 0.19 \\
&= 2.4 \\
\text{surprisal at `themselves'} &= -\log P(W_4 = \text{themselves} \mid W_1 = \text{who}, \dots) \\
&= -\log \frac{0.04}{0.15 + 0.04} \\
&= -\log 0.21 \\
&= 2.2
\end{aligned}$$

(35)
$$\begin{aligned}
\text{surprisal at `who'} &= -\log P(W_1 = \text{who}) \\
&= -\log(0.10 + 0.10) \\
&= -\log 0.2 \\
&= 2.3 \\
\text{surprisal at `themselves'} &= -\log P(W_4 = \text{themselves} \mid W_1 = \text{who}, \dots) \\
&= -\log \frac{0.10}{0.10 + 0.10} \\
&= -\log 0.5 \\
&= 1
\end{aligned}$$

To recap: I took it as given that the language of the speaker(s) of interest consists of precisely the set of sentences in (30), and moreover held fixed a particular analysis of each of those sentences (e.g. the assumption that reflexives are created by movement, that 'who' moves to SpecCP, etc.). Against the backdrop of these fixed assumptions, we would like to know whether the mental grammar of the speaker(s) of interest is the MG that expresses those analyses in terms of merge and move steps, or the corresponding IMG that expresses those same analyses in insert and build steps. What the calculations above show is that (in combination with plausible independent linking assumptions) an experiment where we measure the difficulty that our speaker of interest encounters in incrementally reading a sentence can help us to answer this question. Concretely, if we suppose that the speaker's probabilistic knowledge of language is informed by the pattern in (31) — based on, for example, the fact that this is data we collected from newspaper articles that are representative of the speaker's linguistic experience — then we predict roughly equal comprehen-

sion difficulty at the first and last words of 'who will shave themselves' if the speaker's mental grammar is the MG, but significantly less comprehension difficulty at the last word than at the first if the speaker's mental grammar is the corresponding IMG.

3.3 Case study: Grammar selection

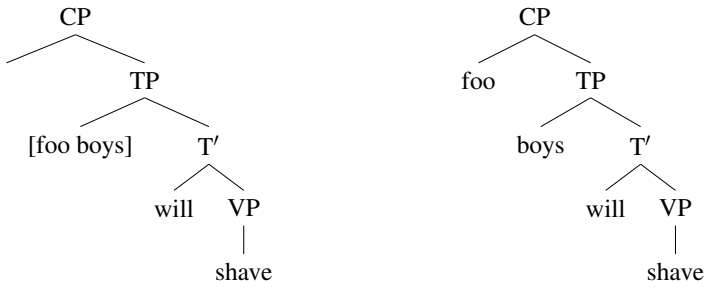
In this section, I will consider a very simple model of grammar selection by a learner. It will be useful to begin with the specifics of the (artificial) learning problem that the model learner will confront, before returning to the details of the forms of particular grammars and the differences between MGs and IMGs.

The learner I will consider must choose between two grammars, G^{det} and G^{wh}. Both grammars generate the same set of surface strings, although they assign different structures to some of these strings. The common set of surface strings is shown in (36).

(36) boys will shave
boys will shave themselves
who will shave
who will shave themselves
foo boys will shave
foo boys will shave themselves

Where the two grammars differ is in their treatment of the word 'foo'. In G^{det}, this word is a determiner, as shown in the tree on the left in (37). (G^{det} corresponds to what was used in the previous case study, with the string 'foo' in place of 'some'.) In G^{wh}, this word is a wh-phrase base-generated in SpecCP (in line with certain proposals about words like 'why' and 'how'), as shown on the right in (37).

(37)



The learner will be provided with some training data, on the basis of which to decide between these two analyses. The training data will be some collection of tokens of the sentences in (36), all of which are generated by both grammars. In order to decide whether G^{det} or G^{wh} best fits the data, the learner will have to consider which grammar can best capture the statistical properties of the training corpus.

One way to tackle this problem builds directly on the kind of training that was used in the previous case study. Let us suppose that supplementing grammar G^{det} with probabilities requires choosing values for parameters $\boldsymbol{\lambda}$, and that supplementing grammar G^{wh} with probabilities requires choosing values for parameters $\boldsymbol{\mu}$. We know from above that the learner can discover which of the various probabilistic versions of G^{det} best fits the data by choosing $\boldsymbol{\lambda}$ so as to maximize $P(D|G^{\text{det}}_{\boldsymbol{\lambda}})$; and similarly, the learner can choose a probabilistic version of G^{wh} by choosing $\boldsymbol{\mu}$ so as to maximize $P(D|G^{\text{wh}}_{\boldsymbol{\mu}})$. Having thus identified the "winner" $G^{\text{det}}_{\boldsymbol{\lambda}}$ amongst all the versions of G^{det} and the "winner" $G^{\text{wh}}_{\boldsymbol{\mu}}$ amongst all the versions of G^{wh}, the learner can pit these two winners against each other in a grand final by comparing $P(D|G^{\text{det}}_{\boldsymbol{\lambda}})$ with $P(D|G^{\text{wh}}_{\boldsymbol{\mu}})$: this is comparing the best that any version of G^{det} can do with the best that any version of G^{wh} can do. If $P(D|G^{\text{det}}_{\boldsymbol{\lambda}}) > P(D|G^{\text{wh}}_{\boldsymbol{\mu}})$, then our learner will choose G^{det} over G^{wh}.

Notice now that I have not said anything so far about whether G^{det} over G^{wh} are MGs or IMGs. So we can consider two different instantiations of the learning scenario that has just been introduced: one where a learner must choose between two MGs, MG^{det} and MG^{wh}, and one where a learner must choose between two IMGs, IMG^{det} and IMG^{wh}. These two learners are "doing the same thing" — deciding whether to analyze 'foo' as a determiner or as a wh-phrase — but one is using the MG system to do this and the other is using the IMG system instead. But I will show that these two learners can reach different conclusions about how to analyze this unknown word, even while the training data is held constant.

As a first concrete example, consider the (artificial and arbitrary) "training corpus" in (38). As before, the numbers at the beginning of each line are token frequencies.

(38)
5 boys will shave
5 boys will shave themselves
5 who will shave
5 who will shave themselves
5 foo boys will shave

We have seen in the previous case study that the best-fitting probabilistic version of MG^{det} can differ from the best-fitting probabilistic version of IMG^{det}, since the former is determined by setting values of parameters including λ_{merge} and λ_{move} but whereas the latter has parameters including λ_{build} and λ_{insert}. For just the same reasons, the best-fitting probabilistic version of MG^{wh} can differ from the best-fitting probabilistic version of IMG^{wh} — where the former has parameters μ_{merge} and μ_{move}, the latter has μ_{build} and μ_{insert}. These two divergences mean that the determiner-versus-wh competition taking place in the MG setting may look very different from the determiner-versus-wh competition taking place in the IMG setting. Specifically, it turns out that for the MG-based learner, the best likelihood attainable by some probabilistic version of MG^{det} is 75 times higher than the best likelihood attainable by some probabilistic version of MG^{wh}; whereas for

the IMG-based learner confronted with the same choice, the best likelihood attainable under the determiner analysis is only 13.7 times higher than the best likelihood attainable under the wh-phrase analysis.

$$\text{preference factor for determiner analysis with MGs} = \frac{P(D|MG^{\text{det}})}{P(D|MG^{\text{wh}})} = \frac{\texttt{3.36e-18}}{\texttt{4.48e-20}} = 75.0$$

$$\text{preference factor for determiner analysis with IMGs} = \frac{P(D|IMG^{\text{det}})}{P(D|IMG^{\text{wh}})} = \frac{\texttt{3.36e-18}}{\texttt{2.45e-19}} = 13.7$$

The training corpus in (38) therefore provides much stronger evidence for the determiner analysis if the two competing analyses are seen through the lens of the MG framework, than it does if the competition is seen through the lens of the IMG framework. In the context of a more elaborate learning model (for example in combination with certain Bayesian priors), this means that it is possible that the training corpus in (38) could provide evidence that tips the scale in favour of the determiner analysis for an MG-based learner, but not for an IMG-based learner. This despite the fact that the decision in each case is the decision between the two tree structures in (37) — all that differs is whether these trees are taken to be constructed by the derivational operations merge and move, or the derivational operations build and insert.

A more dramatic result is provided by the (equally arbitrary and artificial) training corpus in (39).

(39)
8	boys will shave
1	boys will shave themselves
12	who will shave
1	who will shave themselves
4	foo boys will shave

In this case, the consequences for the MG-based learner and the IMG-based learner differ not in degree (of preference for the determiner analysis), but in direction: MG^{det} beats MG^{wh} in the MG-based determiner-versus-wh competition, but IMG^{wh} beats IMG^{det} in the IMG-based determiner-versus-wh competition. In the MG-based scenario, the best likelihood attainable under the determiner analysis is 64900 times higher than the best attainable under the wh-phrase analysis; in the IMG-based scenario, however, this ratio is only 0.749.

$$\text{preference factor for determiner analysis with MGs} = \frac{P(D|MG^{\text{det}})}{P(D|MG^{\text{wh}})} = \frac{\texttt{2.83e-15}}{\texttt{4.36e-20}} = 64900$$

$$\text{preference factor for determiner analysis with IMGs} = \frac{P(D|IMG^{\text{det}})}{P(D|IMG^{\text{wh}})} = \frac{\texttt{1.31e-17}}{\texttt{1.75e-17}} = 0.749$$

So even in the absence of other surrounding assumptions (e.g. Bayesian priors) to interact with,

this training corpus will favour the determiner analysis for the MG-based learner, but favour the wh-phrase analysis for the IMG-based learner.

To recap: what has been demonstrated is that the choice between the two analyses of 'foo' shown in (37) can have a different outcome, depending on whether those analyses are expressed in MGs, with merge and move as distinct primitive operations, or in IMGs, with build as the single unified structure-building operation — all while holding fixed the training corpus.

4 Conclusion

My aim here has been to answer a question posed in the introduction: how (if at all) does the procedural component of a derivational theory contribute to the theory's empirical bottom line? The central idea is that in derivational systems we can identify a procedure that constructs a complex expression with a static, structured representation of the relations among certain expressions — much as we can identify the procedure "add x to y *and then* multiply the result by z" with the structured representation $z \times (x+y)$. This idea lets us formulate the hypothesis that this structured object, the derivation tree, is the object that is grasped by a speaker when using the corresponding sentence. This eliminates what sometimes appears to be almost a kind of category mismatch that arises when considering how derivational theories are to be cashed out as cognitive hypotheses, particularly as compared to representational theories (recall Figure 1 and Figure 2).

It is not immediately obvious, however, that it is sensible to interpret modern generative grammar in such a way that a structured representation of the derivation itself has this primary status: it is tempting to suspect that the derivational operations that *lead up to* a particular derived expression are redundant extra trimmings, and therefore that debates over these derivational operations themselves are debates without empirical grounding. I have argued that this is an illusion based on historical trends (towards representational encodings of syntactic generalizations) which (i) were not motivated by, and therefore should be taken independently of, the discussions about the mental status of derivational operations, and (ii) are arguably beginning to reverse anyway. In support of the claim that this effect is illusory, I highlighted cases where standard syntactic practice is clearly incompatible with assuming that only final derived expressions are grasped.

It therefore follows that the choice of derivational operations posited by a theory has an effect on the structured object that a speaker is taken to grasp or retrieve upon using a sentence. This in turn means that two theories that differ in their choices of derivational operations — even if the two systems generate the same set of grammatical derived structures — will make distinct claims about speakers' mental representations that can lead to distinct empirical predictions for models of speaker behaviour that include grammatical systems as one of their components. There are many ways that this could be done. I have illustrated the effect by taking the probabilistic enrichment of

a grammar to be one locus of sensitivity to the entire object grasped (i.e. the entire derivation tree), since probabilities are a part of many common models of behavioural tasks. Specifically, in the context of surprisal-based models of incremental sentence comprehension difficulty and of a simple maximum-likelihood-based learner, I showed that two grammars differing only in the derivational operations taken to be responsible for constructing a common set of grammatical structures make distinct empirical predictions.

In narrow terms, this serves as a demonstration that if we are confronted with two theories that differ only in their derivational claims, there are ways to go beyond the standard methodology of acceptability judgements in order to gather evidence that will distinguish them empirically. But in principle we need not wait until we are confronted by the need for such a tie-breaker before attempting to flesh out the empirical consequences of the derivational components of theories: the derivational aspects of a theory can be treated as a first-class component of its empirical payload just as much as all aspects of the representation on the left of Figure 2 are. For psycholinguists, this perspective has the potential to promote more direct engagement with syntactic theory; for syntacticians, it promotes clarified ways of understanding the relationship between derivational and representational ways to express generalizations.

References

Barker, C. and Jacobson, P., editors (2007). *Direct Compositionality*. Oxford University Press, Oxford.

Brody, M. (2002). On the status of representations and derivations. In Epstein and Seely (2002), pages 19–41.

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

Chomsky, N. (1973). Conditions on transformations. In Anderson, S. R. and Kiparsky, P., editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York.

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

Epstein, S. D. and Hornstein, N., editors (1999). *Working Minimalism*. MIT Press, Cambridge, MA.

Epstein, S. D. and Seely, T. D., editors (2002). *Derivation and Explanation in the Minimalist Program*. Blackwell, Oxford.

Ferreira, F. (2005). Psycholinguistics, formal grammars, and cognitive science. *The Linguistic Review*, 22:365–380.

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

Freidin, R. (1978). Cyclicity and the theory of grammar. *Linguistic Inquiry*, 9(4):519–549.

Freidin, R. (1999). Cyclicity and minimalism. In Epstein and Hornstein (1999), pages 95–126.

Hale, J. T. (2001). A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Hale, J. T. (2006). Uncertainty about the rest of the sentence. *Cognitive Science*, 30:643–672.

Hornstein, N. (1999). Movement and control. *Linguistic Inquiry*, 30(1):69–96.

Hornstein, N. (2001). *Move! A minimalist theory of construal*. Blackwell, Oxford.

Hunter, T. (2011). Insertion Minimalist Grammars: Eliminating redundancies between merge and move. In Kanazawa, M., Kornai, A., Kracht, M., and Seki, H., editors, *The Mathematics of Language (MOL 12 Proceedings)*, volume 6878 of *LNCS*, pages 90–107, Berlin Heidelberg. Springer.

Hunter, T. and Dyer, C. (2013). Distributions on minimalist grammar derivations. In *Proceedings of the 13th Meeting on the Mathematics of Language*.

Jackendoff, R. (2011). What is the human language faculty?: Two views. *Language*, 87(3):586–624.

Jacobson, P. (2007). Direct compositionality and variable-free semantics. In Barker, C. and Jacobson, P., editors, *Direct Compositionality*, pages 191–236. Oxford University Press, Oxford.

Kayne, R. (2002). Pronouns and their antecedents. In Epstein and Seely (2002), pages 133–166.

Lasnik, H. (1999). Chains of arguments. In Epstein and Hornstein (1999), pages 189–215.

Lebeaux, D. (1988). *Language acquisition and the form of the grammar*. PhD thesis, University of Massachussetts, Amherst.

Lebeaux, D. (2000). *Language acquisition and the form of the grammar*. John Benjamins, Philadelphia.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

McCawley, J. D. (1968). Concerning the base component of a transformational grammar. *Foundations of Language*, 4:243–269.

McCloskey, J. (2002). Resumption, successive cyclicity, and the locality of operations. In Epstein and Seely (2002), pages 184–226.

Michaelis, J. (2001). Derivational minimalism is mildly context-sensitive. In Moortgat, M., editor, *Logical Aspects of Computational Linguistics*, volume 2014 of *LNCS*, pages 179–198. Springer, Berlin Heidelberg.

Miller, G. A. and Chomsky, N. (1963). Finitary models of language users. In Luce, R. D., Bush, R. R., and Galanter, E., editors, *Handbook of Mathematical Psychology*, volume 2. Wiley and Sons, New York.

Montague, R. (1974). *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, CT. Edited and with an introduction by Richmond H. Thomason.

Phillips, C. and Lewis, S. (2013). Derivational order in syntax: evidence and architectural consequences. *Studies in Linguistics*, 6:11–47.

Pollard, C. and Sag, I. A. (1994). *Head-driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Sag, I. A. and Wasow, T. (2011). Performance-compatible competence grammar. In Borsley, R. and Borjars, K., editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*. Wiley-Blackwell.

Stabler, E. P. (2006). Sidewards without copying. In Wintner, S., editor, *Proceedings of The 11th Conference on Formal Grammar*, pages 157–170, Stanford, CA. CSLI Publications.

Stabler, E. P. (2011). Computational perspectives on minimalism. In Boeckx, C., editor, *The Oxford Handbook of Linguistic Minimalism*. Oxford University Press, Oxford.

van Riemsdijk, H. and Williams, E. (1986). *Introduction to the Theory of Grammar*. The MIT Press, Cambridge, MA.