Against strict headedness in syntax

Timm Lichte
University of Tübingen

ABSTRACT

Strict headedness is a common idealization in the structural analysis of linguistic entities, particularly in syntax. This contribution takes a critical look at its premises and applications by demonstrating the surprising sloppiness of both the defining concepts and the test procedures, and by showing how strict headedness is nevertheless implemented as an important axiom into virtually all mainstream grammar formalisms. Subsequently, I present a non-trivial head-agnostic analysis based on Tree Unification & Constraints (TUCO) in order to show that there actually is a choice and that strict headedness can be avoided in principle.

Keywords:
grammar, syntax, head, government, dependency, valency, tree unification, tree constraints

INTRODUCTION 1

Headedness can be seen as a partial relation between several entities (phonemes, morphemes, words, phrases, etc.) in a complex linguistic structure that yields a distinction between heads and non-heads. In a more constrained reading, furthermore, a headedness relation is asymmetric (an entity cannot concurrently be the head and the non-head of another entity), single-headed (every entity has at most one head),[1] complete (every entity has at least one head, except for "lexical" entities), and usually also endocentric (the head of an entity is

[1] Headedness with multiple heads will be treated in Section 3.3.

also a component of the entity). I will call this strict headedness (see Section 2), which seems to be the standard conception of headedness.

At least since stucturalist classics such as Jespersen (1924, p. 96), Bloomfield (1935, p. 195), Harris (1951, §16.5), Tesnière (1959), and Lyons (1968, p. 233), (strict) headedness certainly belongs to the set of core notions in mainstream linguistics and is widely used even across otherwise irreconcilable camps.[2] Furthermore, it seems that headedness has developed into a primitive, underived category in language description and modeling, on a par with morphosyntactic categories like case and part of speech, or more syntactic notions such as linearization and linking patterns. Consequently, the identification of "the head" is unanimously seen as very helpful, or downright indispensable, for language analysis, even at a rather descriptive level. For example, the position of a "head" helps typologists to classify languages as "head-initial" or "head-final" (cf. Hoeksema 1992), and it motivates abstract representations assumed by formal syntacticians, be they dependency-oriented or constituency-oriented. It is therefore not surprising that dealing with heads (in particular to identify them) is an important common cornerstone of introductory textbooks and courses (e.g. Kroeger 2005; Radford 2009).

Given the strong, pervasive belief in the necessity of heads, it should be possible to cleanly operationalize what counts as a head and what does not. Interestingly, this is not always as evident as one wishes – at least with respect to syntactic applications. In this contribution, I will point out the inaccuracy of both the defining concepts and the test procedures, which became particularly evident in the context of the Det-or-N debate (e.g. Zwicky 1985; Hudson 1987, 1993; Zwicky 1993; Van Langendonck 1994; Croft 1996; Beavers 2003; Hudson 2004; Müller 2020b). Despite the criticism that strict headedness has rightfully received over the years, it is a surprising and indeed puzzling fact that the notion is nevertheless widespread in syntactic

[2] Sometimes other names are used in place of "head", for example, "primary", "governor" or "functor". Tesnière, at one point, calls it a "mot principal" translating German "Hauptwort" ('main word'), which he says is sometimes used by German grammarians (Tesnière 1959, p. 103). (In fact, German *Haupt* also has the meaning 'head'.) So this is to be understood modulo superficial terminological differences or specific implementations.

theories. I will trace this resistance back to certain basic properties of the formal machinery that is often used to model syntax. Then, I will present an example for a head-agnostic syntactic model using Tree Unification & Constraints (TUCO) as its framework. With the example of long-distance dependencies, it will be shown that TUCO offers enough flexibility and expressive power to immediately capture a wide range of regularities found in syntactic trees without the detour via heads.

THE NOTION OF STRICT HEADEDNESS 2

In this section, I propose an explication of the notions of head and strict headedness, and then discuss the test procedures that are commonly used to distinguish between heads and non-heads. This will be largely based on an influential article by Arnold Zwicky from the 1980s (Zwicky 1985) and the subsequent replies that it provoked. I am not aware of any more recent overview over the topic that is equally detailed and comprehensive.

Notational preliminaries 2.1

In order to make the explication more uniform and crisp, headedness is thought of as a partial, irreflexive relation $<_H$: $\mathscr{P}(E)^+ \times \mathscr{P}(E)^+$ on linguistic entities $E = \{e_1, e_2, ...\}$, where $\mathscr{P}(E)^+$ is the power set of E without the empty set.[3] Since we are concerned with syntax, I assume that E denotes the set of word tokens in a sentence. Then $\mathscr{P}(E)^+$ is the set of possible constructs, that is, a CONSTRUCT is a set of linguistic entities. Constructs will be written using lower-case letters, for example, c, and the set of constructs that is assumed for a sentence is correspondingly denoted by upper-case letter C. The COMPONENTS of a construct c are the largest constructs in C that are different from c,

[3] Note that there is nothing to be said against defining $<_H$ as a total, reflexive relation. Some of the following definitions, in particular the endocentricity and bijectivity idealizations, would have to be adjusted accordingly.

and c is their union. For reasons of simplicity, I will be assuming that the components of a construct are non-overlapping. Whenever a construct has no components, it is called a LEXICAL CONSTRUCT.

To give an example, the sentence *The student ate an apple* could be analyzed as a set of constructs $C = \{\{the, student, ate, an, apple\}, \{the, student\}, \{ate\}, \{an, apple\}, \{the\}, \{student\}, \{an\}, \{apple\}\}$. Here, {*the, student, ate, an, apple*} would have components {*the, student*}, {*ate*} and {*an, apple*}, of which only {*ate*} would also be a lexical construct.

Following this notation, $c_h <_H c$ means that the construct c_h is a head of the construct c. The transitive closure of $<_H$, $<_H^*$, is called the HEAD PROJECTION. According to standard assumptions, $<_H$ is taken to be endocentric, that is, the constructs that c_h is a head of contain c_h as a component. Idealizations like endocentricity will be addressed in Section 2.4. Following this notational convention, the properties of a construct c are written as P_c.

As it is useful to express the connection between heads and non-heads more directly, avoiding the step "upward" to the level of embedding constructs, I will say that c_h GOVERNS c_i in the construct c, or conversely, that c_i is a DEPENDENT of c_h in c, iff c_i is a component of the construct c and c_h is the head of c (i.e. $c_h <_H c$), and $c_i \neq c_h$. There is thus a government relation $<_G$, which is uniquely defined in terms of $<_H$: $c_h <_G c_i$ iff $c_h <_H c$, and c_i is a component of c and $c_i \neq c_h$. In other words, $<_G$ shares the domain with $<_H$ but not the range.

Going back to our example above with the sentence *The student ate an apple*, one could assume a head relation $<_H = \{(\{ate\}, \{the, student, ate, an, apple\}), (\{student\}, \{the, student\}), (\{apple\}, \{an, apple\})\}$ such that {*ate*} governs {*the, student*} and {*an, apple*}, etc.

2.2 *Popular definitions*

In his seminal article on heads, Zwicky (1985) illustrates the common intuition about heads in the following way:

> The intuition to be captured with the notion head is that in certain syntactic constructs one constituent in some sense 'characterizes' or 'dominates' the whole. (Zwicky 1985, p. 2)

As with every intuition, however, there are “many directions” into which this intuition can evolve once one tries to make it more explicit. Zwicky himself provides five possible definitions of head-related “dominance”; Hudson (1987, Table 4) later lists eight. Let’s go through some of them very briefly.

Morphosyntactic locus 2.2.1

Zwicky’s favored definition is based on the distribution of “morphosyntactic marks”, that is, potentially visible inflectional properties, which determine what Zwicky calls the MORPHOSYNTACTIC LOCUS (Zwicky 1985, §2.1.3). Using the notation above, we can reformulate Zwicky’s definition in the following way:

DEFINITION 1 **Head as morphosyntactic locus** *Given a construct c_h and a construct c with inflectional properties $P^{infl}_{c_h}$ and P^{infl}_{c}, then $c_h <_H c$ iff $P^{infl}_{c_h} = P^{infl}_{c}$.*

Thus, P^{infl}_{c} is the set of inflectional properties of the construct *c* that *might* influence the syntactic relations that *c* can have to other constructs. Zwicky considers two such syntactic relations: agreement and argumenthood. For agreement, he gives the example of the construct *the child*. From Definition 1 it follows that its head is *child,* because it contributes the marking for singular that might participate in an agreement relation with a verb. Conversely, the head of *is controlling those penguins* should be the auxiliary *is,* and the head of *controls those penguins* should be the verb *controls,* since both carry singular marking, which is critical for establishing an agreement relation with the subject.

Turning to the syntactic relation of argumenthood, Zwicky argues that the morphosyntactic locus of a prepositional phrase like *of the news* is the preposition *of* rather than the NP, for the preposition sometimes marks “particular syntactic arguments of the verb” such as in *inform Sandy of the news*. Thus, according to Zwicky, it follows by “analogy”, that the head of *every* instance of an P + NP construct is the preposition, even if it is not participating in a syntactic argument relation in the sentence.

2.2.2 Semantic locus

Another head definition that Zwicky discusses is based on the semantic interpretation of a construct (Zwicky 1985, §2.1.1). Here, the construct is supposed to describe "a kind of the thing" that the head describes.[4] One could accordingly call the head the SEMANTIC LOCUS of the construct, and define this very similarly to the definition of morphosyntactic locus:

DEFINITION 2 **Head as semantic locus** *Given a construct* c_h *and a construct* c *with ontological properties* $P^{ont}_{c_h}$ *and* P^{ont}_{c}*, then* $c_h <_H c$ *iff* $P^{ont}_{c_h} = P^{ont}_{c}$.

Thus, P^{ont}_{c} and $P^{ont}_{c_h}$ are supposed to only include semantic properties that somehow pertain to the ontological type of the construct meaning, that is, what it actually denotes. For example, according to Zwicky, the head of the construct *those penguins* is *penguins*, because it "describes a kind of penguin". Similarly, the head of *will leave*, namely *leave*, contributes the "kind" of the event that is described by the construct. While this is all rather vague, Zwicky tries to reify the relevant semantic properties by means of the functor-argument distinction. But this, of course, only shifts the problem to the question what a functor and an argument are supposed to be, even though Zwicky seems to assume that this is independently assured knowledge.

2.2.3 Subcategorization locus

The third head definition on my list is concerned with the morphosyntactic constraints that a lexical item can impose on its context. Zwicky here employs the notion of subcategorization, defining the head as the "subcategorisand". For instance, the verb *give* must "occur with either NP NP or NP *to* + NP as its sisters", whereas the verb *donate* only cooccurs with NP *to* + NP. Both *give* and *donate* are therefore supposed to

[4] See also the extensive discussion of a semantic definition of heads in Croft (1996, §4, §6). Croft proposes a refinement under the notion of a "Primary Information-bearing Unit (PIBU)", which he claims to be superior to morphosyntactic locus.

be the head of the respective, instantiating constructs.[5] What sets this head definition apart from the first two is that it is introversive: the subcategorization properties do not get projected to the headed construct in the sense that the construct then has the subcategorization properties of the head; instead, the subcategorization properties of the head are a subset of the syntactic properties of the headed construct. Let us frame this using our notational idiom:

DEFINITION 3 **Head as subcategorization locus** *Given a construct* c_h *with subcategorization properties* $P^{subcat}_{c_h}$ *and a construct* c_i *with syntactic properties* $P^{syn}_{c_i}$*, then* $c_h <_H c$ *iff* $P^{subcat}_{c_h} \subseteq P^{syn}_{c_i}$ *and* $c_i \in c$ *and* $c_h \neq c_i$*.*

The problem with this definition, as well as with the underlying notion, is that it is systematically non-functional in the sense that both the verb and the NP sisters can be treated as the subcategorization locus. Zwicky circumvents this to some extent by restricting the subcategorization locus to lexical categories, hence, to the verb in the above cases. But it does not always converge like this. If a construct consists of just two lexical entities, such as determiner-noun constructs, the issue resurfaces. This can be seen from one example in Zwicky (1985, pp. 5–6), namely the construct *each penguin*, where both *each* and *penguin* could be seen to subcategorize for the other: *each* requires a singular count noun, and *penguin* requires *each* rather than *many* or *much* as a determiner. Zwicky avoids this indeterminacy by stipulating, on admittedly "theory-specific" grounds, that *penguin* in *each penguin* is in fact non-lexical in the sense that it is embedded in a phrasal category "Nom". This is a symptom of the general disadvantage of understanding heads as subcategorization loci: due to being introversive, this head notion depends on specifically delimited, nested constructs in order to scale. One telling example is the following *that*-clause, which Zwicky mentions as an instance of a "Comp + S" construct:

(1) that the penguins are flying

Following Zwicky, the head of (1) is the complementizer *that*, because it is lexical and requiring a finite clausal sister. However, the remain-

[5] This view seems to coincide with what is called "Subklassenspezifik" in the German literature. See, for example, Vater (1978), Jacobs (1994, p. 26), Ágel (2000, p. 187), and the summary in Lichte (2015, §2.2.4).

ing words of the clause do not count as lexical here. Instead, they make up an embedded construct (of type S) with a separate head (be it *are* or *flying* or *are flying*). In fact, if the construct in (1) was flat, each of its components would be a head of this construct because it could be argued (e.g., by coming up with adequate minimal pairs) that all of them restrict their morphosyntactic context in some way. This is certainly unwanted if one adheres to the idealization that there is only one head per construct (see Section 2.4). Hence, identifying a head as subcategorization locus presupposes the existence of a very specific structure of constructs in order to be reasonably applied, namely one where only one of the components is a "lexical category".

2.2.4 Government, concord determination, etc.

Zwicky (1985) distinguishes two more allegedly independent head notions that, in my opinion, do not deserve this status. One is "syntactic government", which can be easily included in the definition of subcategorization locus – something Zwicky rejects without adequate justification:

> Syntactic government, speaking rather loosely, is the selection of the morphosyntactic shape of one constituent (the GOVERNED, or SUBORDINATE, constituent) by virtue of its combining with another (the GOVERNOR). Governors are thus easily confused with subcategorisands. Intuitively, the difference is that subcategorization concerns the very possibility of one constituent's combining with some other co-constituent(s), while government concerns the form that a co-constituent has in such a combination. (Zwicky 1985, p. 7)

In other words, Zwicky differentiates between constraints on form (= government) and constraints on existence (= subcategorization) that the head of a construct may impose on non-heads. Yet the examples for subcategorization that Zwicky provides, some of which I mentioned above, are often found together with constraints on form, for example, on the lexical form of a preposition, on the finiteness of the verb, or on the number marking of the noun. So it is not clear why the two notions could not be safely merged.

The other alleged head notion is "determinant of concord", and it looks very similar to government in that it involves constraints on form. But here the scope is narrower, namely on “concord features”, that is, features that are subject to agreement, such as number agreement of subjects and finite verbs. The issue with subject-verb agreement – as well as with any other case of agreement, I suppose – is that it is notoriously unclear which is the “determinant”: the subject, or the verb. Zwicky (1985, p. 9) argues, by looking at Swahili and aiming at typological uniformity, that in English the subject should be seen as the determinant.[6] Be that as it may, this can be seen as a subcase of government where headedness is particularly hard to decide on if one is seeking for a single head. I will come back to this in the next section when dealing with headedness idealizations.

Of course, the literature holds a plethora of further definitions of a syntactic head, but I claim that their essence is covered by Definitions 1–3. Hudson (1987), for example, adds three definitions to Zwicky’s five: head as distributionally equivalent to the construct, head as an obligatory component of the construct, and head as a “ruler”. The first two are indeed treated as “operational criteria” in Zwicky (1985, §2.5) that can be, to some extent, related to the head definitions above. They will be addressed in Section 2.3. The term “ruler”, on the other hand, is used in dependency theory and largely coincides with what Zwicky calls a head, and is similarly vague (Hudson 1984, p. 78, Zwicky 1985, §2.6). Therefore, it is not really helpful when trying to elucidate what a head is.

Popular test procedures 2.3

As for the three information-based head notions discussed in the previous section, corresponding test procedures straightforwardly suggest themselves: just sort out the source of the important morphosyntactic, semantic and subcategorizational properties of a construct, and this will be the head. However, it does not seem obvious how to operationalize these notationally straightforward tests in a uniform and precise way, that is, how to determine what sort of information came from where.

[6] See Müller (2015, §2) for a general criticism of arguments that rely on typological uniformity.

Maybe for this reason, there are a handful of further, more widely used test procedures for headedness that instead rely on grammaticality (or acceptability) judgments. Zwicky (1985, §2.5) confines himself to two of them, admitting that "they appear to be imperfect guides to the heads in syntactic percolation": (i) a test for "distributional equivalence" and (ii) a test for obligatoriness. This is remarkable because it means that there cannot be a one-to-one relationship with the five head definitions that Zwicky mentions (which is also trivially true, but to a lesser extent, for the selection of three distilled above). And this is critical, because it means that there is actually no way to fully test head notions against each other. Moreover, I will argue that even establishing some one-to-one relationship is difficult, which casts much doubt on this entire approach to head identification.

2.3.1 Substitution test for distributional equivalence

The idea that the head ought to be distributionally equivalent to the governed construct goes back to the structuralist literature (see Zwicky 1985, p. 11 for some references). The rationale is "that the head characterizes a construct in the sense that it is the one constituent that belongs to a category with roughly the same distribution as the construct as a whole" (Zwicky 1985, p. 11). Therefore, replacing the construct with the head should retain the grammaticality (or acceptability) of the sentence and the morphosyntactic "category" of the construct while at most reducing the set of its semantic properties. Using our notational idiom, this can be written down in the following way:

TEST 1 **Substitution of the construct** *Given a grammatical sentence S comprising a construct c with a component* c_h*,* c_h *is the head of c if* c_h *can be substituted for c in S such that the resulting sentence remains grammatical and compatible with S in terms of morphosyntactic and semantic properties.*

That is to say, *students* in *the students are waiting for the hungry teacher* passes the substitution test for being the head of *the students* since *students are waiting for the hungry teacher* is grammatical and formally and semantically compatible to the extent that the resulting semantics entails the original one. Contrary to this, *hungry* cannot replace *for the hungry teacher* in a similar way. Despite the grammaticality of the resulting sentence, namely *the students are waiting hungry*, the morphosyntactic and semantic properties clearly diverge.

Looking at Zwicky's above-cited rationale, one is inclined to think that the substitution test relies on the morphosyntactic properties of the head, and that it is therefore a test for the morphosyntactic locus. However, this is somewhat speculative as the morphosyntactic properties are actually never spelled out during the test. The same is true of semantic or subcategorization properties. So it is simply not clear what exactly determines substitutability. Therefore, Zwicky even goes so far as to claim that "the distributional equivalent represents a genuinely new head-like notion" (Zwicky 1985, 13).

Another problem, particularly when subscribing to strict headedness, is – and actually this has long been acknowledged[7] – that the substitution test may work in specific cases, but not in general. One obvious problem is the P-NP construct in English, which generally cannot be replaced by either P or NP. Similarly, Det-N constructs only pass the substitution test if N is not a singular count noun. The substitution test may be inconclusive in that both the determiner and the noun pass it, which may happen with demonstrative determiners, for example. Note that this is easy to reproduce, which raises the question of why linguists would still want to rely on it. The answer is: they do not, at least at the token level. Linguists like Zwicky focus on abstract phrase structure rather than token-instantiated strings, and they apply some sort of preselection based on vague statistical and/or theoretical grounds. For example, Zwicky argues that, since Det-N constructs and their N component have "roughly the same" distribution, the N component should be seen as the head. For the same reason, namely the distribution being "roughly the same", the head of Aux-VP constructs is claimed to be Aux. Similarly, S is given head status in Comp-S constructs.

This sort of cherry-picking weakens the importance of the substitution test considerably. Eventually, one remains free to treat it as one of several pieces of evidence in favor, or against, a certain head/non-head partition. Unfortunately, a similar methodological flaw can be observed in the use of the equally popular omission test, to which we now turn.

---

[7] Zwicky cites a critical discussion in Lyons (1977) regarding the head of Det-N constructs.

2.3.2 Omission test for obligatoriness

The omission test mirrors the substitution test in that the non-heads are now substituted for the construct, albeit with a negative expectation. Zwicky himself notices that "this criterion is closely related to the preceding one, and might be considered to be an extension of it to (some) syntactically exocentric constructions" (Zwicky 1985, p. 13). In fact, the definition of the omission test looks almost identical to the definition of the substitution test:

TEST 2 **Omission of the head** *Given a grammatical sentence S comprising a construct c with a component* c_h*,* c_h *is the head of c if* c_h *cannot be omitted in S such that the resulting sentence remains grammatical and compatible with S in terms of morphosyntactic and semantic properties.*

Again, it is necessary to add the compatibility condition in order to avoid comparing apples and oranges. For example, the omission of *for* in *the students are cooking for the teacher* would be grammatical, but it would also effect a considerable change in the semantics. Note that the omission is performed piecewise, that is, omitting the whole construct *c* is not possible as the head relation is irreflexive (see Section 2.1).

Even though the expectation might be that the omission test is supplementing the substitution test, their results diverge greatly in many cases: while there are no heads in P-NP constructs following the substitution test, the omission test identifies two heads; conversely, when the substitution test identifies two or more heads, there can be no head following the omission test. Only if there is exactly one head following both the substitution test and the omission test, do the two tests converge (see also Section 2.5.1).

Interestingly, Zwicky (1985, p. 14) arrives at a completely different conclusion, namely that both tests are tests "for the same notion", and that they thus greatly converge. How could that happen? As with the substitution test, one trick is to impose some additional, more theory-driven conditions. For example, to rule out the omission of V in V-NP constructs, which is possible in gapping constructions such as *I ate sushi, and Kiyoko a hamburger*, Zwicky requires that omission be restricted to cases of "optionally present" components, excluding "elliptical" ones. This distinction, however, is not at all trivial both theoretically and methodologically, and moreover touches upon a whole new aspect, namely interpretation in context. Dubious as it might be, this distinction helps Zwicky to identify those parts of a construct as heads that are also selected by the (modified) substitution test. To give another example, Zwicky claims that the omission test supports the view that N is the head in Det-N constructs, because he considers the omission of N elliptical, in contrast to the omission of Det, which is supposed to be optional. The second argumentative strategy that Zwicky applies, and which we have already seen above with the substitution test, is to find positive evidence for a token, postulate it for its type, and then to postulate it for other tokens of the type – even if they do not pass the test. A case in point is the head analysis of NP-VP constructs (Zwicky 1985, p. 13). It is argued that the head is VP because (i) omitting NP is ellipsis, and (ii) VP can be standalone in some cases, when forming an imperative sentence. With this sort of argumentation, we arrive at the curious situation that it "follows" from the omission test that the head of *I ate sushi* is *ate sushi*, even though it does not pass the omission test.

Another more conceptual issue of the omission test is that it conflates two notions of obligatoriness that correspond to either heads or non-heads. Obligatoriness can be attributed to the central role of the head in contributing morphosyntactic or semantic properties. But obligatoriness may also hint at a non-head, namely when being the obligatory argument of the head by virtue of subcategorization properties.[8] Therefore, to be fruitfully applied, the results of the omission test must be set against the subcategorization properties of one of the putative heads.

Strict headedness and other popular idealizations 2.4

One central issue when using the test procedures mentioned, be they information-based or grammaticality-based, is that their use often comes with certain strong expectations of how the headedness relation behaves structurally. Unquestionably, the most significant expectation can be referred to as STRICT HEADEDNESS, namely: each non-lexical

[8] Therefore, the omission test is also popular in valency theory as one criterion for distinguishing arguments from adjuncts. See, for example, Somers (1984), Storrer (1992, p. 105), Jacobs (1994), and Mel'čuk (2004, p. 266).

construct contains exactly one head. In fact, this expectation looks so natural and is so firmly implemented in the formal machinery of many grammar models (see below in Section 3) that I rather want to call it an IDEALIZATION in the sense of Stokhof and van Lambalgen (2011). That is to say, strict headedness is not just the result of temporarily neglecting some "parameters", which would amount to what Stokhof and van Lambalgen (2011) call abstraction. It is an indispensable limitation as to how the data are perceived and how the theory is designed. It is an axiom. In this section, I will try to give a more precise characterization of strict headedness and other idealizations that target the head relation, while adhering to the notational conventions laid out above in Section 2.1.

One fundamental idealization is ENDOCENTRICITY, which basically says that the head is contained within a construct:

IDEALIZATION 1 **Endocentricity** *A head relation* $<_H$ *must be* ENDOCENTRIC*, that is: if* $c_h <_H c$*, then* c_h *is a component of c.*

On the other hand, with the definitions presented here, EXOCENTRICITY manifests as constructs *without* head, since the head relation is deemed irreflexive, that is, a construct cannot be the head of itself. From this it also follows that, at least from the perspective of syntax, lexical words are exocentric by definition.

While endocentricity is generally considered the normal case in syntactic theory, exocentric analyses are also discussed for certain phenoma. Zwicky mentions, among others, the notorious example of sentential constructs consisting of a subject and a verbal phrase, hence NP-VP constructs.[9] Here, it is sometimes assumed that the sentential category emerges from the construct as a whole rather than from the NP or VP alone, for example in Lexical Functional Grammar, assuming an exocentric category S (cf., Bresnan *et al.* 2016, §6.3). Following Zwicky, this assumption is fed by the observation that NP-VP constructs have a "unique distribution" (Zwicky 1985, p. 12) different from both NP and VP. Also, saying that the nominative case of the

[9] Following Zwicky (1985, fn. 9), P-NP and Comp-S constructs can be considered exocentric to some extent, too. Moreover, exocentric, that is, "non-headed" analyses have been proposed for coordination constructions and relative clauses – see, for example, Abeillé and Chaves (2021) and Müller (1999).

NP is governed by the VP is "counterintuitive" (Zwicky 1985, fn. 5). On the other hand, what seems to speak for an endocentric treatment is that the morphosyntactic locus of NP-VP constructs rather lies in the VP, as it contributes (at least in some languages, which might be taken as an argument due to the intended typological uniformity; see the case of agreement in Section 2.2.4) various features, for example, tense, aspect, mood, etc. (Zwicky 1985, p. 6). This contrast has been famously addressed in Government and Binding theory with the introduction of an abstract head INFL, which ultimately helps to establish an endocentric phrase structure for NP-VP constructs.[10]

Another important idealization, which extends endocentricity, is BIJECTIVITY. It basically states that every construct is the head of at most one construct, and every construct contains at most one head:[11]

IDEALIZATION 2 **Bijectivity** *A head relation $<_H$ must be* BIJECTIVE, *that is:*

1. *For every construct c_h in the domain of $<_H$, there is exactly one construct c such that $c_h <_H c$.*
2. *For every construct c in the range of $<_H$, there is exactly one construct c_h such that $c_h <_H c$.*

With bijectivity alone, it is still possible to have complex constructs that are lacking a head. This can be prevented by imposing COMPLETENESS on the head relation, in the sense that all constructs that consist of two or more components must have a head.[12] This subset of C (the set of constructs in a sentence) is denoted by $C \setminus L$, where L is the set of lexical constructs with only one component (the lexical construct itself). Note that, usually, lexical constructs are word tokens, but this need not be. Lexical constructs can also be complex constructs comprising more than one word token. Completeness can then be formalized as follows:

[10] See Zwicky (1985, Footnotes 5, 9) for some further remarks on the status and treatment of NP-VP constructs.

[11] Syntactic theories that employ multiple heads in some cases are dicussed in Section 3.3.

[12] The Completeness Idealization corresponds to what Richard Hudson calls the "Non-Dangling Principle", which requires that all words have one governor, or "parent" (Hudson 1994, p. 98, Hudson 1998, (29)).

IDEALIZATION 3 **Completeness** *A head relation $<_H$ on C with lexical constructs L must be* COMPLETE*, that is: For every construct c in $C \setminus L$, there is at least one construct c_h such that $c_h <_H c$.*

With bijectivity and completeness, it is possible to define the idealization of STRICT HEADEDNESS in the following way:

IDEALIZATION 4 **Strict headedness** *A head relation must be bijective and complete, that is:*

1. *For every construct c_h in the domain of $<_H$, there is exactly one construct c such that $c_h <_H c$.*
2. *For every construct c in the range of $<_H$, there is exactly one construct c_h such that $c_h <_H c$.*
3. *The range of $<_H$ is $C \setminus L$.*

In other words, strict headedness amounts to head relations in which every non-lexical construct has exactly one head, and a construct can be the head of at most one other construct. One of the nice consequences of strict headedness is that the inverse of the head relation taken together with the government relation (i.e. the head of a construct "governs" all the other immediate components of the construct) forms a tree-shaped graph on C. However, note that if we draw this dominance tree on top of a sentence, there might be crossing branches due to the fact that constructs might be linearly non-contiguous.

In order to avoid crossing branches (at least under endocentricity), a further idealization can be added, namely LINEAR CONTINUITY, which requires that heads and headed constructs are continuous sequences with respect to the linear order of word tokens, that is, substrings of the sentence (which corresponds to C). In what follows, I will call the linear order of the word tokens of a construct c the LINEARIZATION of c:[13]

IDEALIZATION 5 **Linear continuity** *A head relation must be* LINEARLY CONTINUOUS*, that is: for all constructs c_h and c with $c_h <_H c$, it*

[13] At the same time, there is a body of work that fundamentally questions the linear continuity of heads and constructs. See, among others, the discussion in Wells (1947, §5), Curry (1961, pp. 65–66), McCawley (1982), Zwicky (1986), and, again just as an example, the implementation in Kathol (1995).

holds that the linearizations of c_h and c are substrings of the linearization of C.

Finally, it is also popular additionally to require LINEAR ADJACENCY, which states that the head is linearly adjacent to all the non-heads of the construct:[14]

IDEALIZATION 6 **Linear adjacency** *A head relation must be* LINEARLY ADJACENT*, that is: if $c_h <_H c$, then the linearizations of c_h and each of the components of c form a substring in the linearization of C.*

Of course, one can observe further idealization-like restrictions in the literature that are related to headedness. For example, once we assign categorical labels to constructs, we can impose head-related constraints on the distribution of those labels in terms of projectivity[15] and uniformity.[16] This sort of restriction has received a lot of attention in the Generative Grammar literature (see, e.g., Chomsky 2008, 2013). But unfortunately, this is dealt with in a rather technical way that already presupposes a certain head/non-head distinction – one which seems to be established merely by tradition.

Empirical evidence for strict headedness? 2.5

In the preceding sections, we have come across several notions and tests concerning syntactic heads, and one predominant idealization, strict headedness. The question now is: can we empirically verify strict headedness based on some test for some head notion? In many cases, the answer must clearly be no. To show this, one may look at very simple sentences, like the following one from German:[17]

[14] See, for example, the Adjacency Principle in Hudson (1987, p. 127).

[15] Projectivity here means that the category of a construct follows from the category of the head.

[16] For example, a category-driven uniformity restriction on head relations could state that, if c_h governs c_d in some head relation, then something of the category of c_d can never govern something of the category of c_h in any of the head relations.

[17] German is chosen here because it has a much richer morphology than English. Furthermore, the author is a native speaker of German, which is helpful when evaluating the tests.

Table 1: Results of the headedness tests applied to (2) and the constructs ⓪–②

heads of	⓪	①	②
omission test	–	*der*	–
substitution test	*aß, der Schüler, viele grüne Äpfel*	*der*	*viele, grüne, Äpfel*
morphosyntactic locus	*aß*	*der, Schüler?*	*viele?, Äpfel*
semantic locus	*aß*	*Schüler*	*Äpfel*

(2) [[Der Schüler]① aß [viele grüne Äpfel]②.]⓪
the pupil ate many green apples
'The pupil ate many green apples.'

As can be seen from (2), we have to presuppose some constructs, here labeled ⓪, ①, and ②, on which the tests and definitions can operate. The results are shown in Table 1 for the omission and substitution test, and for morphosyntactic locus and semantic locus. If a test identifies no or more than one head, it contravenes strict headedness, which seems to be the case for all but semantic locus. In order to make the results in Table 1 transparent, I will discuss the individual tests and definitions in more detail below. Subcategorization locus is neglected here due to its inherent problems, discussed in Section 2.2.3.

2.5.1 Omission & substitution test

The omission test (Test 2) is based on the assumption that the head cannot be omitted without making either the semantics incompatible or the sentence ungrammatical. In (2), any component of ⓪ and ② can be omitted while leaving the co-components in place. Thus, (2) could be replaced with *aß* ('ate'), *der Schüler* ('the pupil'), *der Schüler viele grüne Äpfel* ('the pupil ~~ate~~ many green apples') etc. in certain contexts so that the modified sentence would retain a compatible semantics. Those contexts could, for example, consist of questions such as *What did the pupil eat?* or *What did the pupil do?* Note that the omission test does not say anything about the context of the sentence, which can therefore be freely chosen. By contrast, the determiner *der* ('the') of ① *der Schüler* does not appear to be omissible in any context.

The substitution test (Test 1) aims at the distributional equivalence of the construct with its head. The results in Table 1 mirror the results of the omission test: ⓪ and ② can be replaced by more than one of its components, while ① can only be replaced by the determiner *der* (‘the’).

Morphosyntactic locus 2.5.2

The morphosyntactic locus (Definition 1) ideally contains all the inflectional properties of a construct. It is a matter of debate, however, what set of inflectional properties should be taken into account, and it also depends on the language. In what follows, I will only consider the properties for gender (MASC, FEM, NEUT), number (SG, PL), case (NOM, ACC, DAT, GEN), and tense (PRES, PAST).[18]

The first difficulty is to assess the inflectional properties of the entire clausal construct ⓪. Assuming that the number property is SG and the tense property is PAST, because they could be seen to express the finiteness of the clause and therefore to restrict its distribution, the morphosyntactic locus should be assigned to the verb *aß* (‘ate’). The subject *der Schüler* (‘the pupil’), with which *aß* agrees, however, only bears SG.

With the subject construct ①, the determination of the inflectional properties seems to be easier and I will assume, based on the selection above, that these are MASC, NOM, and SG. However, it becomes more difficult when trying to actually assign these properties to one of the components due to inflectional ambiguity. The determiner *der* (‘the’) is ambiguous in that it can be also used with properties FEM, DAT|GEN, SG, or properties FEM|MASC|NEUT, GEN, PL. *Schüler* (‘pupil’), on the other hand, is ambiguous with respect to number and case, only ruling out GEN + SG and DAT + PL. Thus, the only property that is lexically fixed is the MASC property of *Schüler*, and this could be taken to indicate the morphosyntactic locus of ①. However, once the MASC property is set, the determiner *der* actually specifies two other properties, namely NOM and SG, and now *der* seems to act as the morphosyntactic locus of ①. In other words, it is not obvious how to incorporate the morphological ambiguities of constructs while determining the morphological locus.

The situation in the object construct ② is equally inconclusive. ② has the properties MASC, ACC, and PL. *Äpfel* (‘apples’) could be taken

[18] Properties for person are omitted here for the sake of brevity as the examples will only include constructs with third person.

to contribute MASC and PL, while *viele* ('many') contributes PL too and restricts the case property to NOM or ACC (*Äpfel* furthermore allows for GEN). The adjective *grüne* ('green'), on the other hand, is ambiguous between SG and PL. Thus, the noun seems to be just slightly more specific in terms of inflectional properties compared to the determiner.

2.5.3 Semantic locus

The only clear support for strict headedness in (2) so far seems to come from semantic locus – but is it always like that? (3) shows that the noun and semantic locus in the object construct ② can be missing, and it is much less clear which of the two remaining components is now the semantic locus:

(3) [[Der Schüler]① aß [viele grüne]②.]⓪
the pupil ate many green
'The pupil ate many green ones.'

Of course, this can be explained away by ellipsis: that is, one could argue that only the "complete" or "reconstructed" sentence should be tested. Hence, the head of the Det-A construct *viele grüne* in (3) could be claimed to be an invisible N, which inherits the semantic weight from its presumable antecedent *Äpfel*. Note however that, depending on the context, N could also be taken to just refer to *Dinge* ('things'), which comes with very few ontological properties and is certainly less informative in (3) than the quantifier *viele* and the color adjective *grüne*.

But even if we put aside N ellipsis for the moment, there are other sentences that seem to challenge the semantic foundation of strict headedness in different ways. In (4), for example, it is not clear whether the preposition *in* is solely responsible for contributing the ontological type as was claimed above: whether it contributes a location reading or a path reading crucially depends on the case of the determiner:

(4) [[Der Schüler]① sprang [in der/die Schule]②.]⓪
the pupil jumped in the.DAT/the.ACC school
'The pupil jumped in/to school.'

Therefore, in cases like this, the semantic locus seems rather to reside in both the preposition and the determiner (or any other component with case marking) than in the preposition alone.

A similar issue arises with multi-word expressions with an idiomatic meaning, where it is difficult to decide which of the components contributes the semantics. An example from German is shown in (5), which involves the multi-word expression *ins Gras beißen* ('die', lit: 'bite in the grass'):

(5) [[Der Schüler]① biss [ins Gras]②.]⓪
  the pupil bit in.the.ACC grass
  'The pupil died.' (lit. 'The pupil bit in the grass.')

It could be claimed that the general ontological type, namely eventuality, is nevertheless contributed by the verb and not by the PP. But apart from this, dying events and biting events differ considerably (the first one being an accomplishment, the second one an activity). Why does that not count here? In other words, a burning question is where to draw the line between decisive and secondary semantic contributions. There is no easy way to answer this question, as far as I can see.

To avoid such problems, one reviewer suggested determining the semantic locus based on the literal meaning of (5). However, one immediate consequence would be that the idiomatic meaning does not have its own syntactic representation, but adopts that of the literal reading(s). Even though this solution has actually been argued for in psycholinguistic and grammar-theoretical work (cf. Lichte and Kallmeyer 2016), perplexity does not seem a good reason to accept such a severe limitation as a general rule. Moreover, besides the issue of identifying the literal meaning(s), it remains unclear whether this entirely solves the problem as long as the semantic properties that are to determine the semantic locus are not listed.

Interim conclusion 2.5.4

To summarize, none of the test procedures discussed supports strict headedness as is: considerable theory-driven assumptions have to be added in order to make this work out even roughly. Therefore, a last escape hatch is to interpret headedness as multi-factorial notion, that is, to assume that strict headedness arises from a specific combination of those test procedures and the underlying primary notions. In other words, the head could be identified as the construct component that always, or at least most often, occurs in a column in Table 1. Based on this rationale, one could then deduce that the verb is the head of the full sentence, while the head of the subject is the determiner and the head of the object is rather the noun, albeit with a smaller margin. While this actually reflects the state of the infamous Det-or-N debate quite accurately (see Hudson 2004), it is methodologically very questionable, because it is becoming increasingly difficult to reject strict headedness on empirical grounds – and virtually impossible when tests and test results are furthermore non-trivially weighted against each other. In a way, strict headedness then turns into something that is empirically taken for granted.

All this raises the following question: Why do we need strict headedness so badly? My guess is that there are at least three concurrently effective reasons, namely (i) that strict headedness is entrenched by tradition, (ii) that it serves to make syntactic theory more uniform, lean, and hence more elegant, and (iii) that it is enforced by the formal framework. While (i) is certainly the case but without immediate scientific value (though being sociologically important), (ii) is more intricate because it also depends on the choice of the syntactic framework. Therefore, in the rest of the paper, I will concentrate on (iii), and, in the next section, try to show how the formal machinery of syntactic frameworks can point the way toward strict headedness.

3 SYNTACTIC MODELING WITH STRICT HEADEDNESS

When interpreted as algebraic structures, syntactic models have two dimensions that are closely related: (i) the DERIVED STRUCTURE, which is the result of applying operations to elements of the carrier set (i.e. lexical and derived structures), and (ii) the DERIVATION STRUCTURE, which is a record of the operations applied to yield a specific derived structure. Headedness can be reflected in both dimensions, either separately or concurrently, by employing a distinction between heads and non-heads. This distinction can be realized differently, not necessarily leading to strict headedness. In this section, though, I will be concentrating on a representative selection of major syntactic frameworks where this distinction leads to strict headedness. More relaxed but less well-known implementations will be covered in the sections to come. The goal will be not only to show the considerable, paradigmatic commonalities and differences between syntactic models, but also to mark that we actually have a choice. In fact, I claim that the implementation of headedness constitutes another fundamental divide between models of syntax, which is orthogonal to that between generative-enumerative and model-theoretic approaches (Pullum and Scholz 2001), or between lexical versus phrasal approaches (Müller and Wechsler 2014).

Strict headedness in derived structures 3.1

In derived structures, strict headedness appears as the necessity to structurally mark one component of a construct as the head, and the other components as non-heads. For example, given a construct $\{c_1, c_2\}$, either c_1 or c_2 has to act as the head, with the other as non-head. Or, in terms of government, either c_1 governs c_2 or c_2 governs c_1. In other words, the government relation must include all components of a construct and it must be ASYMMETRIC. Note that asymmetric relations in syntactic structures are nothing special per se. One very trivial example for an asymmetric relation is linear precedence, which forms a total order on the word tokens of a sentence. Another one is the subset relation on the set of constructs (C above), which forms a partial order. However, the headedness relation is usually incongruent with the linear precedence relation and, by definition, (properly) embedded in the subset relation.

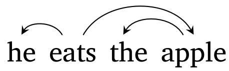
In the remainder of this section, I will give two quite prominent examples from otherwise very distant paradigms that standardly impose strict headedness on their derived structures: dependency structures in Dependency Grammar and phrase structures under the X′-Schema in Generative Grammar.

Dependency structures in Dependency Grammar 3.1.1

Dependency structures are the main theoretical objects in Dependency Grammar (Tesnière 1959; Müller 2020a).[19] While the notion of syn-

[19] Interestingly, the algebraic side of Dependency Grammar, that is, the actual grammar, is usually neglected (but see, for example, Gaifman 1965; Hellwig 2006; Kuhlmann 2010; Müller 2020a, p. 371f).

Figure 1:
Tree-shaped dependency graph



tactic dependency is linguistically congruent with the notion of headedness or government (see van Langendonck 2003), its technical implementation is slightly different: the dependency relation only holds between *lexical* constructs, that is, word tokens. Nevertheless, the dependency relation can be straightforwardly defined based on the headedness relation and the corresponding head projections: given a head relation $<_H$ and word tokens w_1, w_2 and constructs c_1, c_2, w_2 depends on w_1, written as $w_1 \rightarrow w_2$, iff $\{w_1\} <_H^* c_1$ and $\{w_2\} <_H^* c_2$ and $c_1 <_H c_2$. Dependency relations are usually represented as directed graphs where the nodes are word tokens and the edges represent dependencies in such a way that the dependency head dominates (or "points to") the dependent. An example is provided in Figure 1.

As such, dependency graphs do not necessarily implement strict headedness. But they do so with the common set of constraints that are supposed to hold, and which eventually make them dependency *trees* like that shown in Figure 1. Among the most basic, tree-imposing constraints are acyclicity of edges, connectedness of nodes, existence of a unique root, and existence of unique dominance paths (Heringer 1993) – and there is a plethora of further constraints in the literature that build on the more basic ones (e.g. well-nestedness, planarity, projectivity; cf. Maier and Lichte 2011). It should be easy to see that one can deterministically transform a dependency tree into a construct-based head relation which then satisfies strict headedness. Furthermore note that dependency trees are the standard case not only in theoretical work on dependency grammar but also in computational applications that make use of dependencies, for example, in those parsers that follow the guidelines of the Universal Dependencies initiative (de Marneffe *et al.* 2014; Nivre 2015).

That being said, there are actually quite a number of proposals that try to relax the idealization of strict headedness in terms of dependency structures. I will discuss them briefly in Section 3.3.

3.1.2 Phrase structures in Generative Grammar

Looking back over time, one might have doubts that phrase structure is adequately characterized as a derived structure. In the early days of Generative Grammar (Chomsky 1957, 1965), when it was called Transformational Grammar and its formal core was considered to be a string-rewriting system, phrase structure had rather more the status of a derivation structure, that is, it served to record which context-free string-rewriting rules had been applied to derive a specific string of words from some start symbol. Now, however, phrase structure is mainly seen as a derivation-agnostic representation of syntactic structure. There are, in fact, grammar formalisms that treat phrase structure as a derived structure in the first place, for example tree-rewriting formalisms such as Tree-Adjoining Grammar (see Section 3.2.2).

As such, phrase structures lack any references to headedness – they just add balanced and labeled brackets to a given string of words. Heads only come into play when restricting the labeling in ways that unequivocally distinguish heads from non-heads. Consequently, only by knowing the labeling rules can one identify heads and non-heads in a given phrase structure.

In this respect, the most influential set of labeling rules is certainly the X′-SCHEMA (Chomsky 1970; Jackendoff 1977; Kornai and Pullum 1990), which goes back to the following famous quote from Chomsky:

> To introduce a more uniform notation, let us use the symbol $\overline{X}$ for a phrase containing *X* as its *head* [emphasis by author]. Then the base rules introducing *N*, *A*, and *V* will be replaced by a schema (48), where in place of … there appears the full range of structures that serve as complements and *X* can be any one of *N*, *A*, or *V*:
>
> (48) $\overline{X} \rightarrow X \ldots$
>
> Continuing with the same notation, the phrases immediately dominating $\overline{N}$, $\overline{A}$ and $\overline{V}$ will be designated $\overline{\overline{N}}$, $\overline{\overline{A}}$, $\overline{\overline{V}}$ respectively. To introduce further terminological uniformity, let us refer to the phrase associated with $\overline{N}$, $\overline{A}$, $\overline{V}$ in the base structure as the "specifier" of these elements. Then the elements $\overline{N}$, $\overline{A}$, $\overline{V}$ might themselves be introduced in the base component by the schema (49):
>
> (49) $\overline{\overline{X}} \rightarrow [\text{Spec}, X]\ \overline{X}$
>
> (Chomsky 1970, p. 210)

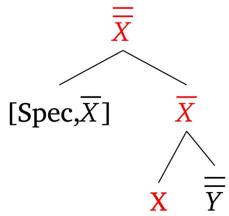


Figure 2:
Schema of the X′-Schema
following (Chomsky 1970, p. 210)

This rough sketch of the X′-Schema can be diagrammed as in Figure 2. What is crucial here is that the described labeling restrictions build on the category label of heads (by definition all preterminals in a CFG): the label of a phrase is composed of the label of its head with an added overline or "bar". This headed phrase then projects the category of its head further to the next embedding phrase, namely by adding another "bar" to its category label. Crucially, for each phrase, there is exactly one head or headed phrase that is used in this way. Therefore, even though many important details remain unexplained at least in this sketch (e.g. the exact interpretation of "head" and the treatment of modifiers), the expressed labeling restrictions already bear direct connection to strict headedness, in that every phrase must be labeled in such a way that it participates in exactly one head projection. In other words, every phrase has exactly one head, which must be reflected in the phrase's label. This remains true in later explications and applications of the X′-Schema – see, for example, the "Lexicality" and "Succession" conditions in Kornai and Pullum (1990). Note that Kornai and Pullum (1990) also prove that the use of the X′-Schema has no consequences regarding the set of string languages that can be generated with such constrained CFGs. But that is not what strict headedness is all about. It rather means that it puts an extra burden on grammar writers, who have to decide for each phrase what label to choose in order to keep consistency with the X′-Schema. As we have seen in Section 2.3 and Section 2.5, there is no general and reliable test procedure for this.

3.2 *Strict headedness in derivation structures*

Similarly to derived structures, strict headedness appears in derivation structures as the necessity to treat components of a construct either as head or as non-head. This basically means that each operation that is used for syntactic composition is to be used either with heads or with non-heads, or that the combinatorial operations have dedicated argument positions, which is to say that they are strictly non-commutative. As a consequence, the head/non-head distinction becomes essential for the mechanics of a grammar formalism.[20]

In this section, I will give two examples of derivationally strictly headed grammar formalisms, Categorial Grammar and Tree-Adjoining Grammar, again trying to cover approaches that are as diverse as possible in other respects.

Categorial Grammar 3.2.1

The basic setup of CATEGORIAL GRAMMAR (CG, Ajdukiewicz 1935; Bar-Hillel 1953; Ades and Steedman 1982) is very simple: lexical words are assigned atomic or complex categories such as np/n which may contain slashes that separate input and output subcategories. For example, with category np/n, np is the output and n the input, and the direction of the slash indicates where the input category is to be found. Note that categories are meant to reflect valency properties, if available. Therefore, transitive verbs are usually assigned a category similar to $s\backslash np/np$, which implies that the first input np is to the right, and then another np to the left acts as the second input. Correspondingly, in order to combine these categories, two combinatorial operations are at hand, forward application and backward application, with reference to the direction of the slashes.[21] Schematic examples of forward and backward application are given in Figure 3, using the common proof-theoretic tableau form.

A complete CG derivation then looks like that shown in Figure 4. Proceeding from top to bottom, first the words in the sentence are

[20] Two caveats are in order here. Firstly, headedness in derivation structure does not need to coincide with headedness in derived structure and vice versa. See the example of Tree-Adjoining Grammar below. Secondly, the operations of derivationally strictly headed grammar formalisms could also be used for the representation of other asymmetric relations, such as linear precedence.

[21] Compared to Categorial Grammar, the set of combinatorial operations is considerably extended in Combinatorial Categorial Grammar (Steedman 2000). Nevertheless, the argument made for Categorial Grammar still applies.

Figure 3:
Schematic examples for forward and backward application. σ and τ stand for arbitrary atomic or complex categories

forward application	backward application
$\dfrac{\sigma/\tau \quad \tau}{\sigma}>$	$\dfrac{\tau \quad \sigma\backslash\tau}{\sigma}<$

Figure 4:
Example of CG derivation

$$\dfrac{\underset{np}{he} \quad \dfrac{\underset{s\backslash np/np}{eats} \quad \dfrac{\underset{np/n}{the} \quad \underset{n}{apple}}{np}>}{s\backslash np}>}{s}<$$

mapped onto their lexically determined categories while retaining the order of the words. Then forward and backward application is used recursively in such a way that only the category *s* remains. As can be seen from this example, the derivation structure is strictly binary (due to the nature of forward and backward application) and the combined categories are consistently treated in a very different way: one being the input of the other. So which is the head? Since categories emerge from valency or subcategorization properties, it seems very natural to identify heads as those categories that "consume" their fellow category in order to satisfy their slashed demand. Thus, *np*/*n* is the head of the construct *np*/*n* *n*, and every forward and backward application creates another head and non-head pair. In this case, strict headedness is clearly unavoidable.

The strict headedness result for CG can be carried over to similar grammar formalisms such as MINIMALIST GRAMMAR (MG, Stabler 2011), a formalization of Minimalism, even though they differ in some formal details. In MG, lexical words introduce ordered lists of features that can be polarized, which then correspond to the slashed part in CG categories. Instead of forward and backward application, there is only one corresponding operation, namely external merge.[22] However, the arguments of external merge are strictly ordered, the

[22] In Minimalist Grammar and in Minimalism, there is another operation called internal merge or move, which is of a very different nature as it allows for copying and deleting syntactic material.

first one being the head and the second the non-head, which again culminates in strictly headed derivation structures.

Another more surprising connection is proposed by Müller (2013), who stresses the similarity of CG and MG with HEAD-DRIVEN PHRASE STRUCTURE GRAMMAR (HPSG, Pollard and Sag 1994; Müller 2013 [2007]), at least as far as the representation of headedness goes. Of course, HPSG could not be more different in formal terms: it is a constraint-based formalism (with typed feature structures as models) lacking an algebraic structure. In other words, there is no such thing as a derivation in HPSG. Still, Müller (2013, p. 938) claims that "the notation for marking the head of a structure [in MG] […] corresponds directly to the HPSG representation of heads". What he means is that the derivation structures of MG (and also of CG) correspond to the structures of certain syntactic features within the feature architecture of HPSG models. For example, in the HPSG version of Ginzburg and Sag (2001), headed phrases in HPSG carry the features HEAD-DTR and DTRS (with the value of HEAD-DTR being a part of the DTRS list), and there is a head feature principle that ensures the projection of head features from DTRS to the head features of the phrase.

Furthermore, one can often see in HPSG textbooks that almost all phrase types are structured in this way, that is, they are of the type *headed-phrase* (e.g., Müller 2013, p. 195). Thus it might seem that HPSG implements strict headedness in the same way as CG and MG. However, this is wrong in a technical sense. For example, other than CG or MG, HPSG allows for a *non-headed-phrase*, too. Non-headed phrases are commonly used in the analysis of relative clauses (Müller 1999, Müller 2013 [2007], §11.2) or idiomatic constructions (Bargmann 2015).[23] Also, analyses with more than one head are possible – see Section 3.3. Yet, regardless of the technical possibilities, HPSG at its core is "head-driven", that is, designed in such a way that it follows strict headedness as far as possible.

Tree-Adjoining Grammar 3.2.2

Strict headedness is not bound to Categorial Grammar and its more or less direct derivatives; it can also be observed in very unlike but

[23] I am grateful to one of the reviewers for pointing this out.

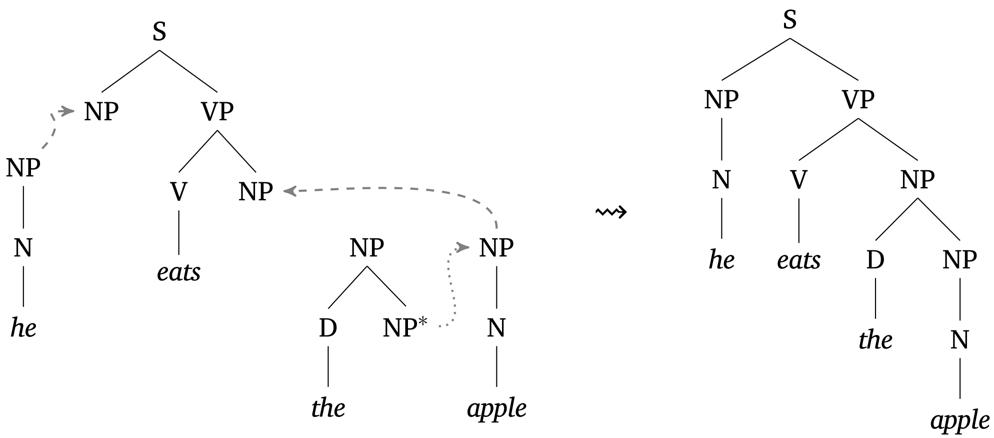


Figure 5: Example of a TAG derivation with the resulting derived tree. The dashed edges indicate substitution, and the dotted edge shows adjunction

still generative-enumerative grammar formalisms too, such as TREE-ADJOINING GRAMMAR (TAG, Joshi and Schabes 1997; Abeillé and Rambow 2000).[24] This is a tree-rewriting formalism where the grammar consists of elementary trees of arbitrary size that can be combined into larger trees using one of two compositional operations, substitution or adjunction: in both cases, a node gets replaced by an elementary tree, but with substitution, it is a leaf node, while with adjunction, it is an internal node. An example is shown in Figure 5, which roughly follows the XTAG Research Group (2001). Thanks to the power of adjunction and the arbitrary size of elementary trees, TAG is said to have an "extended domain of locality" (in contrast to CG and the like) and therefore bears more commonalities with constructionist approaches (Lichte and Kallmeyer 2017).

Despite the constructionist flavor, TAG still adheres to strict headedness, I claim. It is important to note that, for strict headedness, the derived structure on the right side of Figure 5 is not determinative – even though it looks "headed", the derived structure could easily be changed to avoid this impression. What is important, though, is the derivation tree, that is, the nature of substitution and adjunction.

[24] However, despite considerable differences as to formal machinery, TAG is known to have a generative capacity similar to at least some versions of CG and MG (see Joshi *et al.* 1990).

Both operations are non-commutative in the sense that there is a replacing elementary tree and a target elementary tree, and switching those roles leads to different derived trees. Therefore, TAG derivations are usually represented as a derivation tree where the target dominates the substituting or adjoining tree.

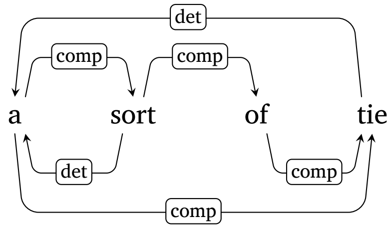
Yet, what is still missing to establish the connection between TAG and strict headedness is a certain interpretation as to what elementary trees represent in syntactic terms. It has long been mainstream to regard elementary trees as realizations of subcategorization properties of the lexical anchor (cf. Abeillé and Rambow 2000; Frank 2002; Lichte 2015, §5.3). That is, the lexical anchor counts as the head of the domain of the anchored elementary tree. Looking back at Figure 5, this is nicely exemplified with the elementary tree of the transitive verb *eats*, which contains NP slots for its subject and object. Moreover, headedness is indicated by choosing a phrase structure that roughly follows the X′-Schema with respect to how the nodes are labeled. It is therefore quite obvious that substitution and adjunction are linguistically interpreted in such a way that they separate heads from non-heads. This culminates in the idea that TAG derivation trees could be homomorphic to canonical dependency trees. Unfortunately, this is not always the case (cf. Rambow *et al.* 1995; Kallmeyer and Kuhlmann 2012).

In a nutshell, with non-commutative operations and the entrenched view of lexical anchors as the heads of the domain of an elementary tree, TAG can be classified as being strictly headed. As a consequence, the grammar writer has to decide for each composition of elementary trees, which to model as the head and which as the non-head – a decision for which, as we have seen in Section 2, there seems to exist no satisfying guidance.

Headedness with multiple heads 3.3

The derived and derivation structures that we have seen so far are *mono-headed,* that is, there is at most one head per construct. In fact, mono-headedness is a precondition for being strictly headed because of the Bijectivity Idealization (see p. 305). Yet there are a number of proposals in various frameworks that (more or less explicitly) make use of multiple heads.

Figure 6:
Multi-headed dependency structure
(from Hudson 2004, p. 39)



One prominent variant of Dependency Grammar that explicitly allows for multi-headed syntactic dependencies is Word Grammar (Hudson 1984, 2007). An example from Hudson (2004, p. 39) is given in Figure 6. In this analysis of an intricate Det-N construct, Hudson (2004) proposes that Det and N should be seen as being mutually dependent on each other, avoiding the difficulty of mono-headed dependency analysis.[25] Hence, multiple heads are an interesting tool, particularly in constructs where there is more than one good candidate for the head, or when a construct can be seen to have more than one governor (which are not necessarily components of the same construct). Other possible cases for multiple heads are therefore relative clauses, raising and control constructions, and coordination constructions. In relative clauses, the relative pronoun agrees with the modified noun, but the relative phrase can be seen to be governed from within the relative clause. In raising constructions such as *Kim seems to sleep*, the raised noun *Kim* can be seen to be governed by the raising verb *seems* with regard to agreement and case properties, and by the embedded infinitive verb *to sleep* in terms of semantics. There are also control constructions such as *Kim tries to sleep*, in which *Kim* acts as a semantic argument of both *tries* and *to sleep*. Lastly, multiple governors can be also assumed for the subject in VP coordination constructions such as *Chapman eats cookies and drinks beer* (Sarkar and Joshi 1997).

However, it is important to note that multi-headedness does not coincide with exocentricity, that is, the lack of heads, in which case the head properties of a construct cannot be attributed to any of the

[25] One reviewer pointed out the similarity to the "mutual selection" approach in HPSG (Pollard and Sag 1994, §9.4), in which Det selects N and N selects Det. Despite "mutual selection", however, the Det-N construct is not treated as multi-headed in HPSG, as N clearly acts as the head, and Det as the non-head.

components.[26] Moreover, a certain kind of multi-headedness, which is not intended here, emerges when representing different types of dependencies (e.g. syntactic and semantic) in one dependency structure. This kind of conflation can be observed, for example, in Meaning-Text Theory (Mel'čuk 1988) and Extensible Dependency Grammar (XDG, Debusmann *et al.* 2004).[27] When considering only syntactic dependencies, multi-headedness is banned in such approaches – even though semantic dependencies are treated more liberally in this respect. In fact, the use of multi-headedness in Word Grammar is quite harshly criticized by Mel'čuk (2009, §1.1) as a "confusion between different types and/or levels of dependency" (p. 68).[28]

Of course, multi-headedness has also been discussed as an option outside Dependency Grammar, for example in TAG (cf. Chen-Main 2006) and HPSG (e.g. Abeillé 2003). Furthermore, movement in Generative Grammar can be generally perceived as an operation (or relation) that helps to express multi-headedness (or more precisely multi-dominance) in phrase structure by means of empty categories.[29]

[26] Note that, in the dependency framework, intermediate or partial structures have been discussed that can be considered headless, such as "connection graphs" (Gerdes and Kahane 2013) or Bubble Trees (Kahane 1997). Thanks to one of the reviewers for pointing this out.

[27] See also the discussion of multi-headedness as a result of different sorts of heads in Zwicky (1993).

[28] Mel'čuk's criticism might look surprising given that, in his Meaning-Text Theory, he assumes two syntactic layers, Surface-Syntactic Structures and Deep-Syntactic Structures, which taken together would impose multi-headedness. It seems, though, that Deep Syntactic Structure is rather seen as an interface level between syntax and semantics. This view is also adopted in recent work within computational linguistics in the context of the Sequoia French Treebank (Candito *et al.* 2014; Michalon *et al.* 2016).

[29] Multi-dominance, or "multi-government", arises in configurations where a construct is governed by more than one head. These heads can be (i) components of one construct (in which case there is also multi-headedness), or (ii) components of overlapping constructs so that they govern the common components. What is meant here is multi-dominance in overlapping constructs. A classical example known from GB Theory is the raising construction *John seems t to like ice cream* (Chomsky 1981, (9iv)), in which *John* is said to be governed by both *seems* (in terms of case marking) and *like* (in terms of θ-role assignment). This is captured in GB Theory, following our terminology, by first base-generating *John*

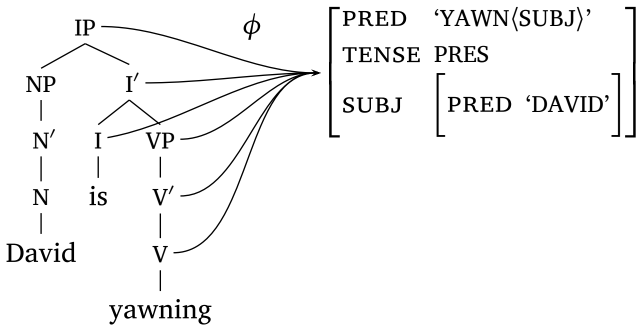


Figure 7: C- and f-structure of an LFG analysis of the sentence *David is yawning* (copied from (11b) in Müller 2020a, p. 226)

Alternatively, Lexical Functional Grammar (LFG, Kaplan and Bresnan 1982; Bresnan *et al.* 2016) adopts a (more relaxed) version of the X′-Schema, but avoids movement by postulating a flexible mapping ϕ from nodes of the phrase structure (c-structure) to components of a structured functional representation (f-structure). With this ϕ mapping, it is possible to associate several c-structure heads with the same f-structure head, for example, extended projections that span a lexicalized VP and a functional IP. An example of this is shown in Figure 7 (copied from (11b) in Müller 2020a, p. 226). It could be argued (as one of the reviewers did) that I and V act as two functional heads of the sentence *David is yawning*. But, again, this view seems to blend different sorts of heads and representations. When considering c-structure and f-structure separately, each is strictly headed: c-structure largely adheres to the principles of the X′-Schema, and even the exocentric category S is usually governed by an "extended head" (Bresnan *et al.* 2016, p. 136) such as I. The f-structure, on the other hand, can be seen to correspond to dependency structures in which dependency relations hold between the PRED features according to their hierarchical order (Przepiórkowski and Patejuk 2020). Note that PRED basically contains a valency list, and the well-formedness conditions of completeness and coherence ensure that the members of the valency list are realized as siblings of PRED (Bresnan *et al.* 2016, §4.7).

Thus, even if multi-headedness seems a widespread alternative to strict headedness (and similarly exocentricity), it does not fundamentally compromise strict headedness. One reason is that multi-

in the construct of *like* and then moving it to the construct of *seems* indicated by the trace *t*.

headedness nevertheless typically shares many idealizations such as the important Completeness Idealization (see Section 2.4, p. 306), which states that every construct bears a head. Another reason is that multi-headedness is also commonly used as a resort for when strict headedness cannot be easily upheld, but it is not used as a general replacement. However, what is intended in this work is precisely the general elimination of strict headedness.

AN ALTERNATIVE LOOK AT HEADS 4

Before presenting an example of a head-agnostic syntactic model, I would like to briefly clarify the conception of heads that underlies it. This conception does not imply strict headedness and the other idealizations mentioned in Section 2.4. Instead, heads are seen as secondary, following from other more fundamental properties of a construct and its alleged heads. In order to capture this, I propose a CONTRIBUTION-BASED conceptualization: something is a head by virtue of contributing some information (i.e. properties) to the embedding construct. In accordance with the head definitions in Section 2.2, properties can be morphological, syntactic, or semantic. Furthermore, the contribution can be made to the embedding construct as a whole, or to another component of the construct. The following explication is kept as simple and general as possible.

Properties are formally treated as unordered flat PROPERTY NAME SETS (PNS) such as {MASC,NOM,HUMAN}.[30] At this level, no distinction is made between properties from different linguistic domains, so that morphological, syntactic and semantic properties are lumped together. Note, however, that PNS can be easily converted into a more ordered format such as a feature-value structure (also with complex values), and vice versa. In order to make the presentation more readable, descriptions are used rather than fully resolved models. Accordingly, PNS may include Boolean operators such as disjunction (|) and negation (¬) with their usual semantics. On top of that, certain

[30] Property name sets are common, for example, in the Lexicon-Grammar framework (Gross 1994; see also Lichte *et al.* 2019).

natural implications are presupposed, for example that {MASC} implies {MASC,¬FEM,¬NEUT}. However, these notational conventions are by no means essential for the argument made here.

Applying the notation of PNS to the German example (2), here repeated as (6), one could decorate the corresponding construct hierarchy with PNS as in Figure 8. Note that the PNS are chosen to serve the example.

(6) [[Der Schüler]① aß [viele grüne
the.NOM.SG pupil ate many.ACC green.ACC
Äpfel]②.]⓪
apples
'The pupil ate many green apples.'

Firstly note that the PNS of the two NPs are simply unifications of the PNS of its components. But this does not have to always be the case. For example, the PNS of the sentence is not constructed by unifying the PNS of its components; it only takes over some properties of the verbal component *aß* ('ate'). These shared or CONTRIBUTED PROPERTIES, which are shown underlined in Figure 8, can then be used to determine the contributional head of a construct, namely by taking the number and kind of contributed properties into consideration. For instance, the determiner in the subject NP *der Schüler* ('the pupil') could be argued to be the head due to the number of contributed properties (similar to, e.g., Zwicky 1985 and Hudson 1987), whereas the noun would be the head when counting only the semantic property HUMAN (which is what Croft (1996) would probably argue for). In other words, under a contribution-based determination of heads, the head status of a component depends on the other components of a construct and the measure involved. Consequently, there can be more than one head per construct across and within different measures.

If PNS are used as in Figure 8, one tricky aspect of measuring property contribution is the "horizontal" or "introversive" contribution from one component to the other, but not to the embedding construct. This is usually the case with subcategorization or valency restrictions (see Section 2.2.3) by which, for example, a finite transitive verb like *aß* is taken to impose the obligatoriness and case of subject and object. To make such contributions visible, a position index is added to the property name indicating to which of the construct's

der Schüler aß viele grüne Äpfel
{EVENT,PAST,SG}

der Schüler
{NOM,MASC,SG,HUMAN,DEF}

der
{NOM,MASC,SG,DEF}|
{DAT|GEN,FEM,SG,DEF}|
{GEN,PL,DEF}

Schüler
{NOM|DAT|ACC,MASC,SG,HUMAN}|
{NOM|GEN|ACC,MASC,PL,HUMAN}

aß
{EVENT,PAST,SG,1.NOM,3.ACC}

viele grüne Äpfel
{ACC,MASC,PL,INDEF,FRUIT}

viele
{NOM|ACC,MASC|FEM|NEUT,PL,INDEF}

grüne
{NOM|ACC,MASC|FEM|NEUT,PL,INDEF}|
{NOM|ACC,FEM,SG,DEF|INDEF}|
{NOM|ACC,MASC|NEUT,SG,DEF}

Äpfel
{NOM|ACC|GEN,MASC,PL,FRUIT}

Figure 8: Construct hierarchy with property name sets for *der Schüler aß viele grüne Äpfel* ('The pupil ate many green apples.'). The underlined properties are contributed to the embedding construct

components it is being contributed. For example, 3.ACC means that the third component of the construct must have the ACC property.

5 HEAD-AGNOSTIC SYNTACTIC MODELING

The PNS-enriched construct hierarchy in Figure 8 can be regarded as the essence of the derived structure of a head-agnostic syntax. The question now is how to arrive at such a derived structure while avoiding the pitfall of headed derivations. Remember that this does not exclude non-commutative operations altogether since we still have to deal with asymmetric relations like linear precedence and the embedding of constructs in larger constructs. But, apart from that, the operation to compose the components of a construct and their PNS must be commutative. In what follows, I will illustrate this sort of grammar formalism while trying to keep the example as general and to the point as possible.

5.1 *An example with TUCO*

The simplest approach I can think of is to use trees to express construct embedding, unification to compose these trees, and tree constraints to guide tree unification and to account for linearization patterns. Accordingly, I will call this sort of syntactic framework TREE UNIFICATION & CONSTRAINTS (TUCO). Despite its simplicity and the absence of strict headedness, the claim will be that this framework nevertheless provides sufficient means to formalize natural language syntax.

5.1.1 Elementary structures

To begin with, the TUCO elementary trees representing the lexical entry of the transitive verb *aß* (‘ate’) are shown in Figure 9. In this example, the valency roles are represented as separate trees, one for the nominative NP and one for the accusative NP. Within these trees, nodes are labeled with PNS that contain contributed as well as “governed” properties, for example, the agreement property SG and the

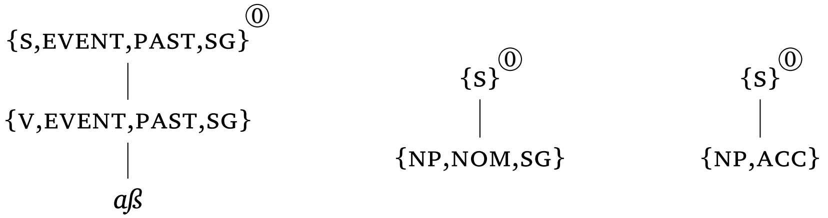


Figure 9: Elementary TUCO trees representing the lexical entry of the transitive verb *aß* ('ate')

case properties NOM and ACC. Moreover, nodes in TUCO trees may carry special marks ⓪, ①, …, which serve two purposes: (i) to constitute a syntax-semantics interface (which is not shown here), and (ii) to express the identity of nodes in the final derived structure. By marking all the root nodes of the trees with ⓪ in Figure 9, it is indicated that these nodes must be combined into one node during the derivation. In other words, even though the valency information of *aß* is distributed over several trees, the mark ⓪ in their respective root nodes ensures that they eventually belong to the identical S node. Note that, as usual, the marks are freshly chosen each time the lexical entry of *aß* is instantiated.

5.1.2 Composition

The composition of TUCO trees uses a sort of TREE UNIFICATION that is akin to the notions in Interaction Grammar (Guillaume and Perrier 2009).[31] A tree is understood as the unique model of a minimal set of tree descriptions, allowing only for descriptions of immediate dominance and immediate precedence relations between nodes. Hence, when unifying trees, one is actually unifying two sets of descriptions, and subsequently compiling all their minimal models – minimal in the sense that no nodes and edges may be added. As for the nodes, tree unification implies the composition of their PNS by set union and the identification of their link marks. To make things easier for now, I will be assuming that nodes with PNS constitute the "non-terminal" nodes, while "terminal" nodes are labeled with word forms written

[31] See also the specific use of tree unification in Popowich (1989), Gerdes (2004), Kahane (2006) and Lichte (2012, 2015). At least the framework presented in the latter work, Synchronous Tree Unification Grammar (STUG), also allows for head-agnostic syntactic modeling.

with italicized font.[32] With this distinction, it is straightforward to impose the usual well-formedness conditions on derived trees, namely that non-terminal nodes must not be leaf nodes, and that terminal nodes are leaf nodes with exactly one non-terminal node immediately dominating them.

5.1.3 Constraints

When applying tree unification to the lexical trees in Figure 9, there will be several derived trees that are not desirable in linguistic terms. For example, nothing so far prevents unifying all the nodes dominated by the root nodes (which must be unified due to the common marker ⓪), thus resulting in a node with the awkward looking PNS {V, NP, EVENT, PAST, SG, ACC, NOM}. In order to achieve only reasonable solutions, one has to furthermore specify TREE CONSTRAINTS, that is, conditional statements with tree descriptions on the left- and right-hand sides. To give a very simple example, the tree constraint {A} $\Rightarrow$ {A, B} imposes the following: if there is a node with property A, then it also has property B. Conversly, {A, B} $\Rightarrow \bot$ indicates that A and B are incompatible and there is no solution whenever a PNS contains both. Thus stating that {V, NP} $\Rightarrow \bot$ will prevent nodes from carrying the categories V and NP at the same time. Such constraints pertaining to single PNS, as well as more complex tree constraints, are shown in Figure 10, guiding the unification of the lexical trees in Figure 9 in the desired way, that is, unification of nodes is only possible when all tree constraints are fulfilled.

The more complex tree constraints have the same basic shape and semantics as the PNS constraints above, but they describe the configuration of the nodes in a well-formed tree in terms of dominance ($\rightarrow$) and linear precedence ($\prec$). The tree constraint at the bottom of Figure 10 basically states: if there is an S node with three separate daughters with categories NP, V, and NP, then the two NP nodes have to surround the V node. The convention will be that nodes in the description are treated as separate nodes in the model as long as they are not explicitly identified in a description.

[32] One could also generalize PNS to terminal nodes.

Intermediate derived tree:

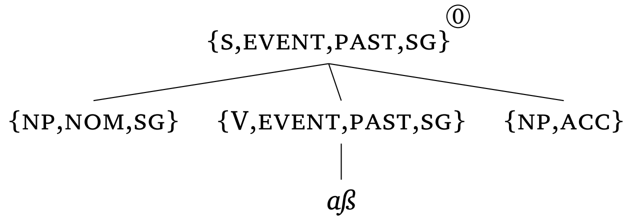
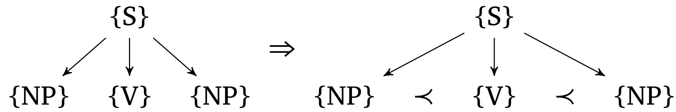


Figure 10:
Intermediate derived tree using the elementary TUCO trees in Figure 9
and conforming to the tree constraints shown below

Tree constraints:

$\{\text{V}, \text{NP}\} \Rightarrow \bot$,
$\{\text{S}, \text{NP}\} \Rightarrow \bot$,
$\{\text{S}, \text{V}\} \Rightarrow \bot$,
$\{\text{ACC}, \text{NOM}\} \Rightarrow \bot$,



5.1.4 The full parse

We now have everything to derive the sentence *der Schüler aß viele grüne Äpfel* in the way depicted in Figure 11, while respecting the tree constraints in Figure 10 and Figure 12.[33] The first two tree constraints in Figure 12 guarantee that there is at most one nominative NP and at most one accusative NP under an S node, whereas the last two constraints determine the internal structure of an NP.[34] The resulting well-formed derived tree is shown in Figure 13.

[33] As the constraints in Figure 10 are not specified for case, the alternative OVS word order

(i) Viele grüne Äpfel aß der Schüler.
many.ACC green.ACC apples ate the.NOM.SG pupil
('Many green apples, the pupil ate.')

would also be licensed.

[34] Note that I do not claim this to be a valid generalization about German. Rather, the aim is to demonstrate that it is possible to express this kind of constraint. There are perfectly grammatical sentences in German that contain more than one noun phrase with the same case as immediate components. Thanks to one of the reviewers for pointing this out.

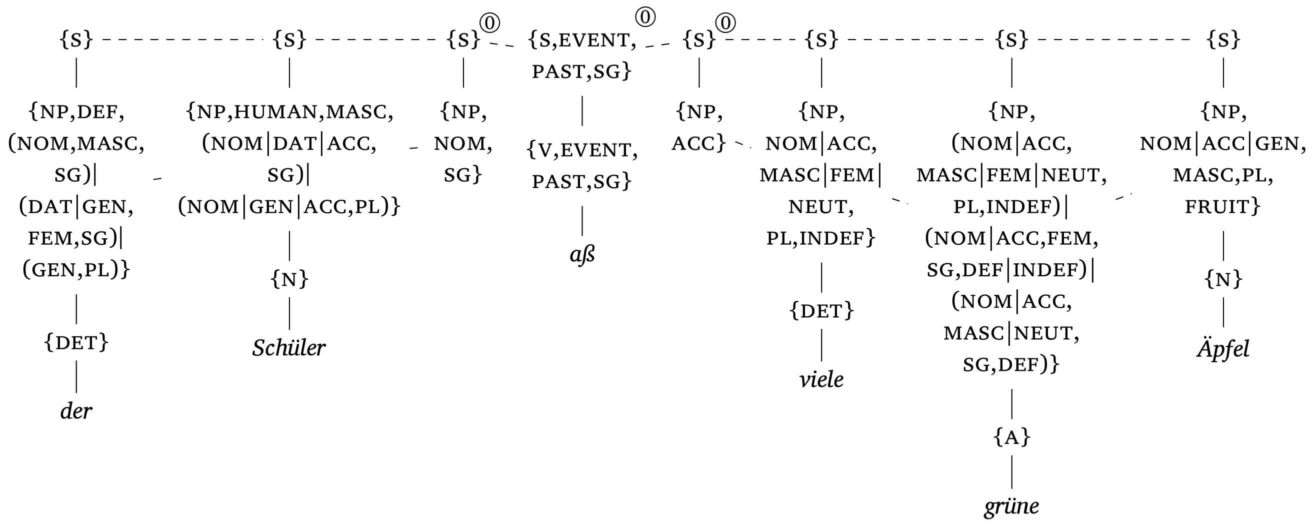


Figure 11: Head-agnostic TUCO derivation of *der Schüler aß viele grüne Äpfel* using, among others, the elementary trees in Figure 9 and complying with the constraints in Figure 10 and Figure 12

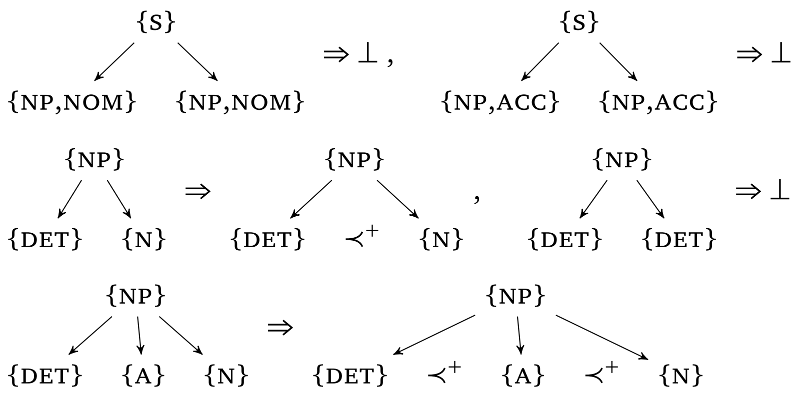


Figure 12: Further TUCO constraints used in the derivation in Figure 11

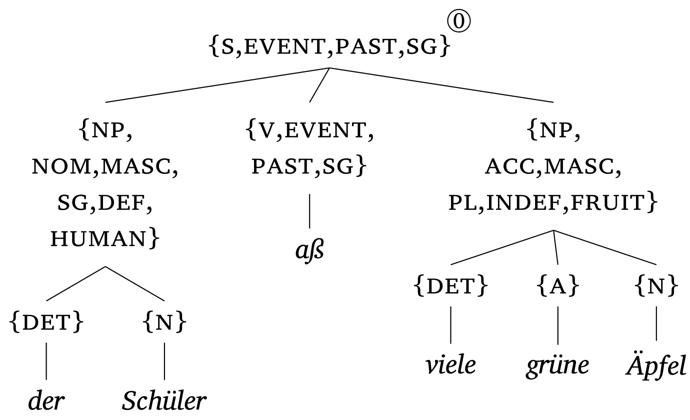


Figure 13: Resulting derived tree of the derivation in Figure 11

Note that both the derived and derivation structure do not presuppose a distinction between heads and non-heads, even though the analysis implements strong lexicalization, including the representation of valency. So it might seem that headedness has slipped in again, since the valency contribution of the verb *aß* contains nodes for the governed nominative and accusative NPs, and it might seem that the derivation hinges on their presence. But this is an illusion. Since valency structures are unified, the same result could be achieved when omitting those nodes in the contribution of *aß*. Moreover, one could *concurrently* assume an entry for subjects that in turn contributes to the preterminal node of *aß*, namely by specifying (or "checking") agreement properties. In other words, there is not necessarily a single head-like component on which the derivation depends; rather, the components of a construct may contribute equally to the syntactic structure of the construct.

5.2 *Notational enhancements*

The example of a head-agnostic analysis presented here uses only the basic elements of the TUCO framework, in particular tree constraints built from an explicit tree description language. While the bare expressive power seems sufficient, the number of tree constraints in a TUCO grammar might very soon become confusingly large. For example, there is one constraint in Figure 12 explicitly prohibiting the occurrence of two DET nodes within an NP, another constraint to determine the order of siblings DET and N, and a third constraint for the correct linearization of DET, A and N. Instead, it would be desirable to have just one concise constraint and description of a well-formed NP.

Fortunately, there are numerous ways in which higher-level abstractions that help to increase the conciseness and readability of a TUCO grammar can be defined from elementary tree constraints. In this section, I will briefly discuss one such abstraction that makes use of bracketing and two-dimensional regular expressions – two-dimensional in the sense that they can refer to both linear precedence and dominance in trees.

Bracketing and regular expressions of linear precedence 5.2.1

Trees can be written as bracketed expressions with the parent immediately after the opening bracket and the children following it. For example, in [{A} {B} [{C} {D}]], {A} is the parent of {B} and {C}, and {C} is the parent of {D}.[35] The linearization and composition of nodes is usually taken literally, that is, the denotation of this expression exactly contains one tree with nodes {A}, {B}, {C}, {D}, and the dominance relations and linear order indicated in the expression.

In the following, however, I will deviate from this convention in two respects: (i) bracketed expressions will be interpreted as constraints by marking the antecedent, that is, the nodes whose existence is presupposed in order for the consequent to apply, with the hash symbol, #; (ii) the string following the parent will be interpreted as a regular expression of children nodes, which fully characterizes the possible strings of children nodes of the given parent. With this adaptation, the structure of NPs in our example can be captured with just one contraint in (7):[36]

(7) [{NP}# {DET}? {A}* {N}]

In prose, the constraint means the following: if a node is an NP, it must immediately and only dominate at most one DET node followed by an arbitrary number (including zero) of A nodes, followed by one N node. To capture this with basic TUCO constraints would require the set in Figure 14. The left constraint states that an NP node may only dominate nodes with properties DET, A or N. The right constraint states that an NP node must dominate one N node at least. The two constraints below furthermore impose that DET must be the first node and N must be the last node under NP. Note that the meaning of the constraint in (7) would change considerably if the antecedent marker # was shifted to DET, for example, or assigned to several nodes at the same time.

[35] The bracket notation of phrase structure trees is widespread in linguistics and goes back at least to Chomsky (1972, p. 130).

[36] This is reminiscent of the use of regular expressions in LFG's c-structure constraints (Kaplan and Bresnan 1982, p. 277), which have the shape of context-free rules. TUCO is more expressive, however, as TUCO constraints may vertically extend beyond the parent and children of a node – see Section 5.2.2 below.

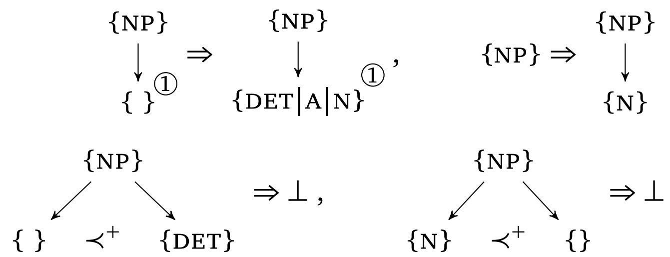


Figure 14: The set of basic TUCO constraints that is equivalent to the single constraint in enhanced bracket notation shown in (7)

5.2.2 Regular expressions of dominance

The Kleene star operator can also be used to describe non-immediate dominance between nodes, similarly to the description of non-immediate linear precedence in (7). The main difference in the notation is that, while in (7) the Kleene star was attached to a child node, it is now attached to a parent as indicated in (8):

(8)
[{S,¬SLASH} 1{ }
{ }*
[{S,SLASH}* [{S,SLASH} { }*
[1{ } {TRACE}#]
{ }*]]
{ }*]

The part with [{S,SLASH}* roughly means that there is an arbitrarily long dominance path consisting only of S nodes with a SLASH property. The boxed number 1 is a PNS label that indicates the equality of the two labeled PNS. The equivalent basic TUCO constraint is given in Figure 15. Note that the right-hand side of the constraint spans the entire conjunction including the two implications. This is necessary in order to consistently ensure the presence of properties S and SLASH in the dominance path.

The example obviously hints at filler-trace analyses of long-distance dependencies, as in *what did Kim claim Sandy ate t*. In transformational terms, the constituent *what* is usually said to be base-generated at the position of the trace *t* and later moved to some sentence initial position (see, e.g., Chomsky 1986, Chapter 6). However, TUCO does not have transformations. The SLASH property is therefore named after the slash mechanism in GPSG (Gazdar 1981; Gazdar *et al.* 1985, Chapter 7) and HPSG (Pollard and Sag 1994, Chapter 4), which

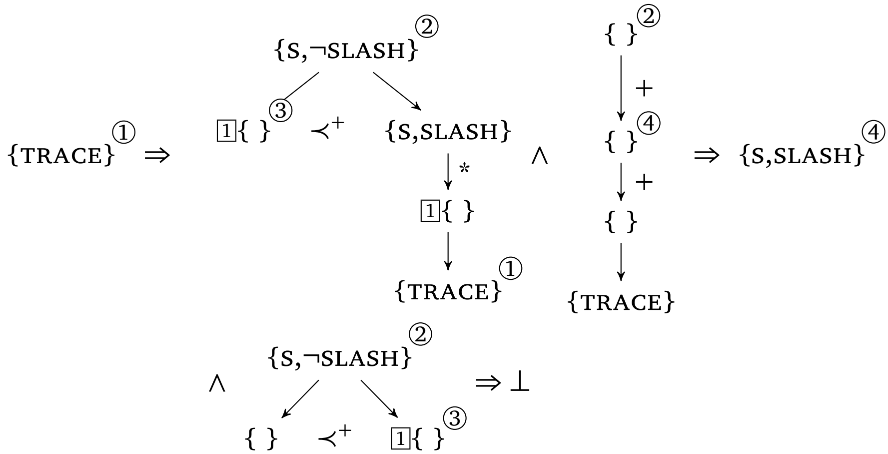


Figure 15: A basic TUCO constraint that is equivalent to the constraint in enhanced bracket notation shown in (8)

copes with these "unbounded dependency constructions" in a non-transformational manner. Similarly, in the constraint above, the first child of {S, ¬SLASH} acts as the filler of the trace under {S, SLASH}.

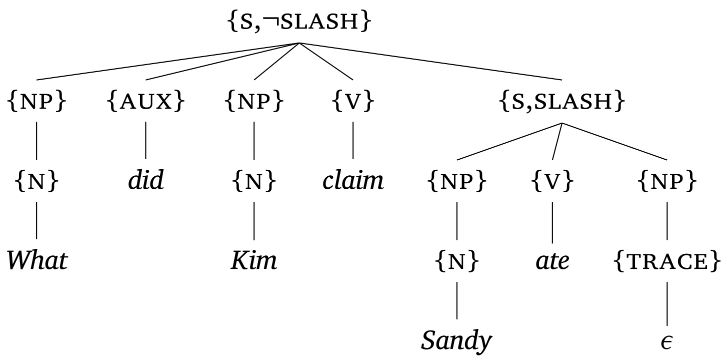
At least two further constraints that come with filler-trace analyses of this sort are not yet integrated in Figure 15. One is that, under S nodes with a SLASH marking, exactly one child must either contain a trace and carry the SLASH property, or be the trace.[37] Secondly, there has to be exactly one filler for each trace. All this can be captured by adding the single constraint in (9):

(9) [{S,SLASH}# {¬SLASH}*
({S,SLASH} | [{ } {TRACE}])
{¬SLASH}*]

Note that {¬SLASH} cannot dominate a trace without a filler because every TRACE property initiates the projection of the SLASH property up to the filler due to the constraint in (8). Both constraints together license the derived tree for *What did Kim claim Sandy ate* in Figure 16. The underlying elementary trees, from which the tree is assembled

[37] This is probably too strict, but serves the example. Pollard and Sag (1994, Chapter 4) also discuss the possibility of multiple traces in connection with tough movement and parasitic gaps. Gazdar (1981, §3) shows that his slash mechanism is also able to capture Across-the-Board extractions with multiple traces.

Figure 16: Tree for *What did Kim claim Sandy ate* satisfying the two constraints in (8) and (9)



via tree unification, are not shown here. There are plenty of ways in which this could be done, so I will leave that to the readers.

What should be obvious, though, is that at no point do we need strict headedness in order to arrive at concise theories or intuitive representations. As shown in this section, higher-level abstractions such as two-dimensional regular expressions can be defined based on regular TUCO constraints. They offer enough flexibility and expressive power to immediately capture a wide range of regularities found in syntactic trees without any detour via heads.

6 CONCLUSIONS

In this article, I have tried to delineate one of the most dominant idealizations found in mainstream syntactic theory: strict headedness. The idea that each construct should be partitioned into heads and non-heads is unanimously taken for granted, or so it seems. At the same time, however, the considerable empirical issues as to the operationalizability of the competing head notions are well-known. I hypothesized that this puzzling paradox can be at least partly explained with the formal mechanics of the predominant syntactic models, which presuppose strictly headed derived or derivation structures. Subsequently, I presented a non-trivial head-agnostic grammar formalism based on TUCO in order to show that there actually is a choice that we should be aware of.

Another explanation for the resilience of strict headedness could be the term *head* itself, which gives rise to certain anthropomorphic expectations, as beautifully proven by the following quote from Croft (1996, p. 57):

> For some people, myself included, the two-headed model for phrases and clauses implied by the profile-determinant semantic definition of head is an unsatisfactory conclusion. The idea of a two-headed phrase sounds about as natural as a two-headed baby.

Since no linguist would be cruel enough to deliberately create a "two-headed baby",[38] the goal of producing a one-headed baby is immediately understandable from a psychological point of view. So, maybe, we should not use the word *head* in connection with syntactic constructs to spare the linguist from ethical dilemmas. Just imagine what would happen if we replaced *head* by *leg* or *tentacle*.

In searching for better evidence of strict headedness, one should perhaps also take into account results from NLP and psycholinguistics. But it is beyond doubt that any reasonable evaluation of strict headedness presupposes a serious exploration of head-agnostic alternatives. This work is intended to be a first step in that direction.

ACKNOWLEDGMENTS

The idea for this article came from participating in a workshop on "Köpfigkeit und/oder grammatische Anarchie" ('Headedness and/or grammatical anarchy'), beautifully organized by Horst Simon and Ulrike Freywald, which took place at the Freie Universität in Berlin in May 2017. I am also very grateful to the reviewers of the Journal

[38] Let alone "monstres animaux" as formulated by Tesnière (1959, p. 347): "Les phrases bifides [e.g. coordination constructions] sont comparables aux monstres animaux, qui ont deux têtes ou deux extrémités inférieures." (Translation from Tesnière (2015, p. 351): "The bifid sentences are comparable to monsters that have two heads or two bodies.")

of Language Modelling for helpful comments & criticism, to the editors for their patience, and to Carmela Chateau-Smith who did a great job proof-reading the manuscript. Last but not least, the article has benefited enormously from discussions with Laura Kallmeyer, Rainer Osswald, Simon Petitjean, Agata Savary, and Christian Wurm. This work was partially financed by the CRC 991 "The Structure of Representations in Language, Cognition, and Science" at the University of Düsseldorf.

REFERENCES

Anne ABEILLÉ (2003), A Lexicon- and Construction-Based Approach to Coordinations, in Stefan MÜLLER, editor, *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar, Michigan State University*, pp. 5–25, CSLI Publications, Stanford, CA, https://web.stanford.edu/group/cslipublications/cslipublications/HPSG/2003/abeille.pdf.

Anne ABEILLÉ and Rui P. CHAVES (2021), Coordination, in Stefan MÜLLER, Anne ABEILLÉ, Robert D. BORSLEY, and Jean-Pierre KOENIG, editors, *Head-Driven Phrase Structure Grammar: The Handbook*, number 9 in Empirically Oriented Theoretical Morphology and Syntax, pp. 725–776, Language Science Press, Berlin, doi:10.5281/zenodo.5599848.

Anne ABEILLÉ and Owen RAMBOW (2000), Tree Adjoining Grammar: An Overview, in Anne ABEILLÉ and Owen RAMBOW, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analyses and Processing*, number 107 in CSLI Lecture Notes, pp. 1–68, CSLI Publications, Stanford, CA.

Anthony E. ADES and Mark J. STEEDMAN (1982), On the Order of Words, *Linguistics and Philosophy*, 4(4):517–558.

Vilmos ÁGEL (2000), *Valenztheorie*, Gunter Narr Verlag, Tübingen.

Kazimierz AJDUKIEWICZ (1935), Die syntaktische Konnexität, *Studia Philosophica*, 1:1–27.

Yehoshua BAR-HILLEL (1953), A Quasi-Arithmetical Notation for Syntactic Description, *Language*, 29(1):47–58.

Sascha BARGMANN (2015), Syntactically Flexible VP-Idioms and the N-after-N Construction, Poster presented at the 5th General Meeting, PARSEME, 23–24 September 2015, Iaşi, https://typo.uni-konstanz.de/parseme/index.php/2-general/138-admitted-posters-iasi-23-24-september-2015.

John BEAVERS (2003), More Heads and Less Categories: A New Look at Noun Phrase Structure, in Stefan MÜLLER, editor, *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar, Michigan State University*, pp. 47–67, CSLI Publications, Stanford, CA, `http://web.stanford.edu/group/cslipublications/cslipublications/HPSG/2003/beavers.pdf`.

Leonard BLOOMFIELD (1935), *Language*, George Allen & Unwin, London.

Joan BRESNAN, Ash ASUDEH, Ida TOIVONEN, and Stephen WECHSLER (2016), *Lexical-Functional Syntax*, Wiley Blackwell, Chichester, UK, 2nd edition.

Marie CANDITO, Guy PERRIER, Bruno GUILLAUME, Corentin RIBEYRE, Karën FORT, Djamé SEDDAH, and Eric De La CLERGERIE (2014), Deep Syntax Annotation of the Sequoia French Treebank, in Nicoletta Calzolari (Conference CHAIR), Khalid CHOUKRI, Thierry DECLERCK, Hrafn LOFTSSON, Bente MAEGAARD, Joseph MARIANI, Asuncion MORENO, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, European Language Resources Association (ELRA), Reykjavik, `http://www.lrec-conf.org/proceedings/lrec2014/pdf/494_Paper.pdf`.

Joan CHEN-MAIN (2006), *On the Generation and Linearization of Multi-Dominance Structures*, Ph.D. thesis, Johns Hopkins University, Baltimore, MD.

Noam CHOMSKY (1957), *Syntactic Structures*, Mouton, Den Haag.

Noam CHOMSKY (1965), *Aspects of the Theory of Syntax*, MIT Press, Cambridge, MA.

Noam CHOMSKY (1970), Remarks on Nominalization, in Roderick JACOBS and Peter ROSENBAUM, editors, *Readings in English Transformational Grammar*, pp. 184–221, Ginn, Waltham, MA.

Noam CHOMSKY (1972), The Formal Nature of Language, in *Language and Mind*, pp. 115–160, Harcourt Brace Jovanovich, New York, 2nd edition.

Noam CHOMSKY (1981), *Lectures on Government and Binding*, Foris, Dordrecht.

Noam CHOMSKY (1986), *Barriers*, number 13 in Linguistic Inquiry Monographs, MIT Press, Cambridge, MA.

Noam CHOMSKY (2008), On Phases, in Robert FREIDIN, Carlos P. OTERO, and Maria Luisa ZUBIZARRETA, editors, *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, pp. 133–66, MIT Press, Cambridge, MA.

Noam CHOMSKY (2013), Problems of Projection, *Lingua*, 130:33–49.

William CROFT (1996), What's a Head?, in Johan ROORYCK and Laurie ZARING, editors, *Phrase Structure and the Lexicon*, pp. 35–75, Springer, Dordrecht, NL, doi:10.1007/978-94-015-8617-7_3.

Haskell B. CURRY (1961), Some Logical Aspects of Grammatical Structure, in Roman JAKOBSON, editor, *Structure of Language and its Mathematical Aspects: Proceedings of the Twelfth Symposium in Applied Mathematics held in New York City, April 14–15, 1960*, pp. 56–68.

Marie-Catherine DE MARNEFFE, Timothy DOZAT, Natalia SILVEIRA, Katri HAVERINEN, Filip GINTER, Joakim NIVRE, and Christopher D. MANNING (2014), Universal Stanford Dependencies: A Cross-linguistic Typology, in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 4585–4592, European Language Resources Association (ELRA), Reykjavik, `http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062_Paper.pdf`.

Ralph DEBUSMANN, Denys DUCHIER, and Geert-Jan M. KRUIJFF (2004), Extensible Dependency Grammar: A New Methodology, in Geert-Jan M. KRUIJFF and Denys DUCHIER, editors, *Proceedings of the COLING 2004 Workshop on Recent Advances in Dependency Grammar*, pp. 70–76, Geneva, `https://aclanthology.org/W04-1510`.

Robert FRANK (2002), *Phrase Structure Composition and Syntactic Dependencies*, MIT Press, Cambridge, MA.

Haim GAIFMAN (1965), Dependency Systems and Phrase-structure Systems, *Information and Control*, 8(3):304–337, doi:10.1016/S0019-9958(65)90232-9, also published as report p-2315, RAND Corp. Santa Monica (CA) in 1961.

Gerald GAZDAR (1981), Unbounded Dependencies and Coordinated Structure, *Linguistic Inquiry*, 12:155–184.

Gerald GAZDAR, Ewan KLEIN, Geoffrey K. PULLUM, and Ivan A. SAG (1985), *Generalized Phrase Structure Grammar*, Harvard University Press, Cambridge, MA.

Kim GERDES (2004), Tree Unification Grammar: Problems and Proposals for Topology, TAG, and German, *Electronic Notes in Theoretical Computer Science*, 53:108–136, doi:10.1016/S1571-0661(05)82578-2.

Kim GERDES and Sylvain KAHANE (2013), Defining Dependencies (and Constituents), in Kim GERDES, Eva HAJIČOVÁ, and Leo WANNER, editors, *Computational Dependency Theory*, number 258 in Frontiers in Artificial Intelligence and Applications, pp. 1–25, IOS Press, Amsterdam.

Jonathan GINZBURG and Ivan A. SAG (2001), *Interrogative Investigations: The Form, Meaning, and Use of English Interrogatives*, number 123 in CSLI Lecture Notes, CSLI Publications, Stanford, CA.

Maurice GROSS (1994), Constructing Lexicon-Grammars, in Beryl T. Sue ATKINS and Anotonio ZAMPOLLI, editors, *Computational Approaches to the Lexicon*, pp. 213–263, Oxford University Press, Oxford.

Bruno GUILLAUME and Guy PERRIER (2009), Interaction Grammars, *Research on Language and Computation*, 7(2–4):171–208.

Zellig S. HARRIS (1951), *Methods in Structural Linguistics*, The University of Chicago Press.

Peter HELLWIG (2006), Parsing with Dependency Grammars, in Vilmos ÁGEL, Ludwig M. EICHINGER, Hans-Werner EROMS, Hans Jürgen HERINGER, and Henning LOBIN, editors, *Dependenz und Valenz/Dependency and Valency. Ein internationales Handbuch der zeitgenössischen Forschung / An International Handbook of Contemporary Research*, volume 25.2 of *Handbücher zur Sprach- und Kommunikationswissenschaft/Handbooks of Linguistics and Communication Science (HSK)*, pp. 1081–1109, Walter de Gruyter, Berlin.

Hans Jürgen HERINGER (1993), Basic Ideas and the Classical Model, in Joachim JACOBS, Arnim VON STECHOW, Wolfgang STERNEFELD, and Theo VENNEMANN, editors, *Syntax. Ein internationales Handbuch zeitgenössischer Forschung / An International Handbook of Contemporary Research*, volume 9.1 of *Handbücher zur Sprach- und Kommunikationswissenschaft / Handbooks of Linguistics and Communication Science (HSK)*, pp. 298–316, Walter de Gruyter, Berlin.

Jack HOEKSEMA (1992), The Head Parameter in Morphology and Syntax, *Language and Cognition*, 2:119–132.

Richard A. HUDSON (1984), *Word Grammar*, Blackwell, Oxford.

Richard A. HUDSON (1987), Zwicky on Heads, *Journal of Linguistics*, 23(1):109–132.

Richard A. HUDSON (1993), Do We Have Heads in Our Minds?, in Greville G. CORBETT, Norman M. FRASER, and Scott MCGLASHAN, editors, *Heads in Grammatical Theory*, pp. 266–291, Cambridge University Press, Cambridge, UK, doi:10.1017/CBO9780511659454.012.

Richard A. HUDSON (1994), Discontinuous Phrases in Dependency Grammar, *UCL Working Papers in Linguistics*, 6:89–120.

Richard A. HUDSON (1998), Adjunct Preposing, Wh-interrogatives and Dependency Competition, *UCL Working Papers in Linguistics*, 10:441–466.

Richard A. HUDSON (2004), Are Determiners Heads?, *Functions of Language*, 11(1):7–42.

Richard A. HUDSON (2007), *Language Networks: The New Word Grammar*, Oxford University Press, Oxford.

Ray S. JACKENDOFF (1977), *$\overline{X}$ Syntax: A Study of Phrase Structure*, MIT Press, Cambridge, MA.

Joachim JACOBS (1994), *Kontra Valenz*, Wissenschaftlicher Verlag Trier, Trier.

Otto JESPERSEN (1924), *The Philosophy of Grammar*, George Allen & Unwin, London.

Aravind K. JOSHI and Yves SCHABES (1997), Tree-Adjoining Grammars, in Grzegorz ROZENBERG and Arto SALOMAA, editors, *Handbook of Formal Languages*, volume 3, pp. 69–124, Springer, Berlin.

Aravind K. JOSHI, K. Vijay SHANKER, and David WEIR (1990), The Convergence of Mildly Context-sensitive Grammar Formalisms, CIS Technical Report MS-CIS-90-01, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, http://repository.upenn.edu/cis_reports/539/.

Sylvain KAHANE (1997), Bubble Trees and Syntactic Representations, in Tilman BECKER and Hans-Ulrich KRIEGER, editors, *Proceedings of the 5th Meeting on Mathematics of Language (MOL5)*, pp. 70–76, DFKI, Saarbrücken.

Sylvain KAHANE (2006), Polarized Unification Grammars, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 137–144, Sydney, http://www.aclweb.org/anthology/P/P06/P06-1018.

Laura KALLMEYER and Marco KUHLMANN (2012), A Formal Model for Plausible Dependencies in Lexicalized Tree Adjoining Grammar, in Chung-Hye HAN and Giorgio SATTA, editors, *Proceedings of the 11th International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+11), 26–28 September 2012, Paris, France*, pp. 108–116, https://aclanthology.org/W12-4613.pdf.

Ron M. KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Joan BRESNAN, editor, *The Mental Representation of Grammatical Relations*, pp. 173–281, MIT Press, Cambridge, MA.

Andreas KATHOL (1995), *Linearization-Based German Syntax*, Doctoral thesis, Ohio State University, Columbus, OH.

András KORNAI and Geoffrey K. PULLUM (1990), The X-Bar Theory of Phrase Structure, *Language*, 66(1):24–50, http://www.jstor.org/stable/415278.

Paul R. KROEGER (2005), *Analyzing Grammar: An Introduction*, Cambridge University Press, Cambridge.

Marco KUHLMANN (2010), *Dependency Structures and Lexicalized Grammars: An Algebraic Approach*, number 6270 in Lecture Notes in Computer Science, Springer, Berlin.

Timm LICHTE (2012), Synchronous Tree Unification Grammar, in Chung-Hye HAN and Giorgio SATTA, editors, *Proceedings of the 11th International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+11), 26–28 September 2012, Paris, France*, pp. 46–54, http://www.aclweb.org/anthology-new/W/W12/W12-4606.pdf.

Timm LICHTE (2015), *Syntax und Valenz: Zur Modellierung kohärenter und elliptischer Strukturen mit Baumadjunktionsgrammatiken*, number 1 in Empirically Oriented Theoretical Morphology and Syntax, Language Science Press, Berlin, doi:10.17169/langsci.b20.102.

Timm LICHTE and Laura KALLMEYER (2016), Same Syntax, Different Semantics: A Compositional Approach to Idiomaticity in Multi-word Expressions, in Christopher PIÑÓN, editor, *Empirical Issues in Syntax and Semantics 11*, pp. 111–140, http://www.cssp.cnrs.fr/eiss11/eiss11_lichte-and-kallmeyer.pdf.

Timm LICHTE and Laura KALLMEYER (2017), Tree-Adjoining Grammar: A Tree-based Constructionist Grammar Framework for Natural Language Understanding, in Luc STEELS and Jerome FELDMAN, editors, *The AAAI 2017 Spring Symposium on Computational Construction Grammar and Natural Language Understanding, March 27–29, 2017, Stanford University (Technical Report SS-17-02)*, pp. 205–212, http://aaai.org/ocs/index.php/SSS/SSS17/paper/view/15330/14536.

Timm LICHTE, Simon PETITJEAN, Agata SAVARY, and Jakub WASZCZUK (2019), Lexical Encoding Formats for Multi-word Expressions: The Challenge of "Irregular" Regularities, in Yannick PARMENTIER and Jakub WASZCZUK, editors, *Representation and Parsing of Multiword Expressions: Current Trends*, number 1 in Phraseology and Multiword Expressions, pp. 1–33, Language Science Press, Berlin, doi:10.5281/zenodo.2579033.

John LYONS (1968), *Introduction to Theoretical Linguistics*, Cambridge University Press, Cambridge.

John LYONS (1977), *Semantics*, Cambridge University Press, Cambridge.

Wolfgang MAIER and Timm LICHTE (2011), Characterizing Discontinuity in Constituent Treebanks, in Philippe DE GROOTE, Markus EGG, and Laura KALLMEYER, editors, *Formal Grammar. 14th International Conference, FG 2009. Bordeaux, France, July 25-26, 2009. Revised Selected Papers*, number 5591 in Lecture Notes in Artificial Intelligence, pp. 167–182, Springer, Berlin.

James D. MCCAWLEY (1982), Parentheticals and Discontinuous Constituent Structure, *Linguistic Inquiry*, 13(1):91–106.

Igor A. MEL'ČUK (1988), *Dependency Syntax: Theory and Practice*, State University of New York Press, Albany.

Igor A. MEL'ČUK (2004), Actants in Semantics and Syntax II: Actants in Syntax, *Linguistics*, 42(2):247–291, doi:https://doi.org/10.1515/ling.2004.009.

Igor A. MEL'ČUK (2009), Dependency in Natural Language, in Alain POLGUÈRE and Igor A. MEL'ČUK, editors, *Dependency in Linguistic Description*, number 111 in Studies in Language Companion Series, pp. 1–110, John Benjamins, Amsterdam, doi:10.1075/slcs.111.03mel.

Olivier MICHALON, Corentin RIBEYRE, Marie CANDITO, and Alexis NASR (2016), Deeper Syntax for Better Semantic Parsing, in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, December 11–16, 2016, Osaka, Japan*, pp. 409–420, https://aclanthology.org/C16-1040.pdf.

Stefan MÜLLER (1999), An HPSG-Analysis for Free Relative Clauses in German, *Grammars*, 2:53–105, doi:10.1023/A:1004564801304.

Stefan MÜLLER (2013), Unifying Everything: Some Remarks on Simpler Syntax, Construction Grammar, Minimalism and HPSG, *Language*, 89(4):920–950, https://hpsg.hu-berlin.de/~stefan/Pub/unifying-everything.html.

Stefan MÜLLER (2013 [2007]), *Head-Driven Phrase Structure Grammar: Eine Einführung*, number 17 in Stauffenburg Einführungen, Stauffenburg Verlag, Tübingen, dritte überarbeitete Auflage edition, http://hpsg.fu-berlin.de/~stefan/Pub/hpsg-lehrbuch.html.

Stefan MÜLLER (2015), The CoreGram Project: Theoretical Linguistics, Theory Development, and Verification, *Journal of Language Modelling*, 3(1):21–86, doi:10.15398/jlm.v3i1.91.

Stefan MÜLLER (2020a), *Grammatical Theory: From Transformational Grammar to Constraint-Based Approaches*, number 1 in Textbooks in Language Sciences, Language Science Press, Berlin, fourth revised and extended edition edition, doi:10.5281/zenodo.3992307, https://langsci-press.org/catalog/book/287.

Stefan MÜLLER (2020b), Headless in Berlin: Headless Structures in Head-Driven Phrase Structure Grammar, in Ulrike FREYWALD and Horst SIMON, editors, *In Headedness and/or Grammatical Anarchy?*, pp. 1–32, Language Science Press, Berlin, https://hpsg.hu-berlin.de/~stefan/Pub/headless.html.

Stefan MÜLLER and Stephen M. WECHSLER (2014), Lexical Approaches to Argument Structure, *Theoretical Linguistics*, 40(1–2):1–76, http://hpsg.fu-berlin.de/~stefan/Pub/arg-st.html.

Joakim NIVRE (2015), Towards a Universal Grammar for Natural Language Processing, in Alexander GELBUKH, editor, *Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015, Cairo, Egypt, April 14–20, 2015, Proceedings, Part I*, pp. 3–16, Springer, Cham, doi:10.1007/978-3-319-18111-0_1.

Carl POLLARD and Ivan A. SAG (1994), *Head-Driven Phrase Structure Grammar*, University of Chicago Press, Chicago, IL.

Fred POPOWICH (1989), Tree Unification Grammar, in *27th Annual Meeting of the Association for Computational Linguistics*, pp. 228–236, Association for Computational Linguistics, Vancouver, British Columbia, Canada, doi:10.3115/981623.981651, https://aclanthology.org/P89-1028.pdf.

Adam PRZEPIÓRKOWSKI and Agnieszka PATEJUK (2020), From Lexical Functional Grammar to Enhanced Universal Dependencies: The UD-LFG Treebank of Polish, *Language Resources and Evaluation*, 54:185–221, doi:10.1007/s10579-018-9433-z.

Geoffrey K. PULLUM and Barbara C. SCHOLZ (2001), On the Distinction Between Model-theoretic and Generative-enumerative Syntactic Frameworks, in Philippe DE GROOTE, Glyn MORILL, and Christian RETORÉ, editors, *Logical Aspects of Computational Linguistics: 4th International Conference, LACL 2001 Le Croisic, France, June 27–29, 2001*, pp. 17–43, Springer, Berlin, doi:10.1007/3-540-48199-0_2.

Andrew RADFORD (2009), *An Introduction to English Sentence Structure*, Cambridge University Press, Cambridge.

Owen RAMBOW, K. VIJAY-SHANKER, and David WEIR (1995), D-Tree Grammars, in *Proceedings of the 33rd Annual Conference of the Association for Computational Linguistics*, pp. 151–158, Association for Computational Linguistics, Cambridge, MA, doi:10.3115/981658.981679, https://www.aclweb.org/anthology/P95-1021.pdf.

Anoop SARKAR and Aravind K. JOSHI (1997), Handling Coordination in a Tree Adjoining Grammar, http://www.cs.sfu.ca/~anoop/papers/pdf/tag-coordination.pdf, longer version of paper in Proceedings of COLING 1996. Draft of August 19, 1997.

Harold L. SOMERS (1984), On the Validity of the Complement-adjunct Distinction in Valency Grammar, *Linguistics*, (22):507–530, doi:10.1515/ling.1984.22.4.507.

Edward P. STABLER (2011), Computational Perspectives on Minimalism, in Cedric BOECKX, editor, *The Oxford Handbook of Linguistic Minimalism*, Oxford Handbooks in Linguistics, chapter 27, pp. 616–641, Oxford University Press, Oxford.

Mark STEEDMAN (2000), *The Syntactic Process*, MIT Press, Cambridge, MA.

Martin STOKHOF and Michiel VAN LAMBALGEN (2011), Abstractions and Idealisations: The Construction of Modern Linguistics, *Theoretical Linguistics*, 37:1–26, doi:10.1515/THLI.2011.001.

Angelika STORRER (1992), *Verbvalenz. Theoretische und methodische Grundlagen ihrer Beschreibung in Grammatikographie und Lexikographie*, number 126 in Reihe Germanistische Linguistik, Niemeyer, Tübingen.

Lucien TESNIÈRE (1959), *Eléments de syntaxe structurale*, Klincksieck, Paris.

Lucien TESNIÈRE (2015), *Elements of structural syntax*, John Benjamins, Amsterdam, translated by Timothy Osborne and Sylvain Kahane.

Willy VAN LANGENDONCK (1994), Determiners as Heads?, *Cognitive Linguistics*, 5(3):243–260, doi:10.1515/cogl.1994.5.3.243.

Willy VAN LANGENDONCK (2003), The Dependency Concept and its Foundations, in Vilmos ÁGEL, Ludwig M. EICHINGER, Hans-Werner EROMS, Hans Jürgen HERINGER, and Henning LOBIN, editors, *Dependenz und Valenz/Dependency and Valency. Ein internationales Handbuch der zeitgenössischen Forschung/An International Handbook of Contemporary Research*, volume 25.1 of *Handbücher zur Sprach- und Kommunikationswissenschaft/Handbooks of Linguistics and Communication Science (HSK)*, pp. 170–187, Walter de Gruyter, Berlin.

Heinz VATER (1978), Probleme der Verbvalenz, *Leuvense Bijdragen*, 67:267–308.

Rulon S. WELLS (1947), Immediate Constituents, *Language*, 23:81–117.

XTAG RESEARCH GROUP (2001), A Lexicalized Tree Adjoining Grammar for English, Technical report, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA.

Arnold M. ZWICKY (1985), Heads, *Journal of Linguistics*, 21:1–29, doi:10.1017/S0022226700010008.

Arnold M. ZWICKY (1986), Concatenation and Liberation, in *Papers from the General Session at the Twenty-Second Regional Meeting of the Chicago Linguistic Society*, pp. 65–74, Chicago Linguistic Society, Chicago, IL.

Arnold M. ZWICKY (1993), Heads, Bases and Functors, in Greville G. CORBETT, Norman M. FRASER, and Scott MCGLASHAN, editors, *Heads in Grammatical Theory*, pp. 292–315, Cambridge University Press, Cambridge, UK.

Timm Lichte
0000-0002-9202-8320
timm.lichte@uni-tuebingen.de

University of Tübingen, Germany

Timm Lichte (2021), *Against strict headedness in syntax*, Journal of Language Modelling, 9(2):291–348
https://dx.doi.org/10.15398/jlm.v9i2.249