11 Dependency Grammar

Dependency Grammar is the oldest framework described in this book. Its modern version was developed by the French linguist Lucien Tesnière (1893–1954). His foundational work *Eléments de syntaxe structurale* 'Elements of structural syntax' was basically finished in 1938 only three years after Ajdukiewicz' paper on Categorial Grammar (1935), but the publication was delayed until 1959, five years after his death. Since valence is central in Dependency Grammar, it is sometimes also referred to as Valence Grammar. Tesnière's ideas are wide-spread nowadays. The conceptions of valence and dependency are present in almost all of the current theories (Ágel & Fischer 2010: 262–263, 284).

Although there is some work on English (Anderson 1971; Hudson 1984), Dependency Grammar is most popular in central Europe and especially so in Germany (Engel 1996: 56–57). Ágel & Fischer (2010: 250) identified a possible reason for this: Tesnière's original work was not available in English until very recently (Tesnière 2015), but there has been a German translation for more than 35 years now (Tesnière 1980). Since dependency grammar focuses on dependency relations rather than linearization of constituents, it is often felt to be more appropriate for languages with freer constituent order, which is one reason for its popularity among researchers working on Slavic languages: the New Prague School (Sgall, Hajičová, Panevova) developed Dependency Grammar further, beginning in the 1960s (Hajičová & Sgall 2003). Igor A. Meľčuk, A. K. Žolkovskij worked since the 1960s in the Soviet Union on a model called Meaning–text theory, which was also used in Machine Translation projects (Mel'čuk 1964, 1981, 1988; Kahane 2003). Mel'čuk left the Soviet Union towards Canada in the 1970s and now works in Montréal.

Dependency Grammar is very wide-spread in Germany and among scholars of German linguistics worldwide. It is used very successfully for teaching German as a foreign language (Helbig & Buscha 1969, 1998). Helbig and Buscha, who worked in Leipzig, East Germany, started to compile valence dictionaries (Helbig & Schenkel 1969) and later researchers working at the Institut für Deutsche Sprache (Institute for German Language) in Mannheim began similar lexicographic projects (Schumacher et al. 2004).

The following list provides a probably incomplete list of linguists who are/were based in Germany: Vilmos Ágel (2000), Kassel; Klaus Baumgärtner (1965, 1970), Leipzig later Stuttgart; Ulrich Engel (1977, 2014), IDS Mannheim; Hans-Werner Eroms (1985, 1987, 2000), Passau; Heinz Happ, Tübingen; Peter Hellwig (1978, 2003), Heidelberg; Jürgen Heringer (1996), Augsburg; Jürgen Kunze (1968, 1975), Berlin; Henning Lobin (1993), Gießen; Klaus Schubert (1987), Hildesheim; Heinz Josef Weber (1997), Trier; Klaus Welke (1988, 2011), Humboldt University Berlin; Edeltraud Werner (1993), Halle-Wittenberg.

Although there has been work done in many countries and continuously over the decades since 1959, a periodical international conference was established as late as 2011.[1,2]

From early on, Dependency Grammar was used in computational projects. Meľčuk worked on Machine Translation in the Soviet Union (Mel'čuk 1964) and David G. Hays worked on Machine Translation in the United States (Hays & Ziehe 1960). Jürgen Kunze, based in East Berlin at the German Academy of Sciences, where he had a chair for computational linguistics, also started to work on machine translation in the 1960s. A book that describes the formal background of the linguistic work was published as Kunze (1975). Various researchers worked in the Collaborative Research Center 100 *Electronic linguistic research* (SFB 100, Elektronische Sprachforschung) from 1973–1986 in Saarbrücken. The main topic of this SFB was machine translation as well. There were projects on Russian to German, French to German, English to German, and Esperanto to German translation. For work from Saarbrücken in this context see Klein (1971), Rothkegel (1976), and Weissgerber (1983). Muraki et al. (1985) used Dependency Grammar in a project that analyzed Japanese and generated English. Richard Hudson started to work in a dependency grammar-based framework called Word Grammar in the 1980s (Hudson 1984, 2007) and Sleator and Temperly have been working on Link Grammar since the 1990s (Sleator & Temperley 1991; Grinberg et al. 1995). Fred Karlsson's Constraint Grammars (1990) are developed for many languages (bigger fragments are available for Danish, Portuguese, Spanish, English, Swedish, Norwegian, French, German, Esperanto, Italian, and Dutch) and are used for school teaching, corpus annotation and machine translation. An online demo is available at the project website.[3]

In recent years, Dependency Grammar got more and more popular among computational linguists. The reason for this is that there are many annotated corpora (tree banks) that contain dependency information.[4] Statistical parsers are trained on such tree banks (Yamada & Matsumoto 2003; Attardi 2006; Nivre 2003; Kübler et al. 2009; Bohnet 2010). Many of the parsers work for multiple languages since the general approach is language independent. It is easier to annotate dependencies consistently since there are fewer possibilities to do so. While syntacticians working in constituency-based models may assume binary branching or flat models, high or low attachment of adjuncts, empty elements or no empty elements and argue fiercly about this, it is fairly clear what the dependencies in an utterance are. Therefore it is easy to annotate consistently and train statistical parsers on such annotated data.

Apart from statistical modeling, there are also so-called deep processing systems, that is, systems that rely on a hand-crafted, linguistically motivated grammar. I already mentioned Meľčuk's work in the context of machine translation, Hays & Ziehe (1960) had a parser for Russian, Starosta & Nomura (1986) developed a parser that was used with an English grammar, Jäppinen, Lehtola & Valkonen (1986) developed a parser that was demoed with Finnish, Hellwig (1986, 2003, 2006) implemented grammars of German in the framework of Dependency Unification Grammar, Hudson (1989) developed a Word

[1] http://depling.org/. 10.04.2015.

[2] A conference on Meaning Text Theory is taking place biannually since 2003.

[3] http://beta.visl.sdu.dk/constraint_grammar. 24.07.2015.

[4] According to Kay (2000), the first treebank ever was developed by Hays and did annotate dependencies.

Grammar for English, Covington (1990) developed a parser for Russian and Latin, which can parse discontinuous constituents, and Menzel (1998) implemented a robust parser of a Dependency Grammar of German. Other work on computational parsing to be mentioned is Kettunen (1986); Lehtola (1986); Menzel & Schröder (1998b). The following is a list of languages for which Dependency Grammar fragments exist:

- Danish (Bick 2001)
- English (Muraki et al. 1985; Starosta & Nomura 1986; Lavoie & Rambow 1997; Hudson 1989; Sleator & Temperley 1991; Iordanskaja et al. 1992; Coch 1996)
- Esperanto (Bick 2009)
- Estonian (Müürisep 1999; Müürisep, Puolakainen, Muischnek, Koit, Roosmaa & Uibo 2003)
- Faroese (Trosterud 2009)
- Finnish (Nelimarkka, Jäppinen & Lehtola 1984; Jäppinen, Lehtola & Valkonen 1986)
- French (Iordanskaja et al. 1992; Coch 1996; Bick 2010)
- German (Hellwig 1986; Coch 1996; Heinecke et al. 1998; Menzel & Schröder 1998a; Hellwig 2003, 2006; Gerdes & Kahane 2001)
- Japanese (Muraki, Ichiyama & Fukumochi 1985)
- Latin (Covington 1990)
- Mandarin Chinese (Liu & Huang 2006; Liu 2009)
- Old Icelandic (Maas 1977)
- Portuguese (Bick 2003)
- Russian (Hays & Ziehe 1960; Mel'čuk 1964; Covington 1990)
- Spanish (Coch 1996; Bick 2006)

The Constraint Grammar webpage[5] additionally lists grammars for Basque, Catalan, English, Finnish, German, Irish, Italian, and Norwegian, Sami, Swahili and Swedish.

[5] http://beta.visl.sdu.dk/constraint_grammar_languages.html

11.1 General remarks on the representational format

11.1.1 Valence information, nucleus and satellites

The central concept of Dependency Grammar is valence (see Section 1.6). The central metaphor for this is the formation of stable molecules, which is explained in chemistry with reference to layers of electrons. A difference between chemical compounds and linguistic structures is that the chemical compounding is not directed, that is, it would not make sense to claim that oxygen is more important than hydrogen in forming water. In contrast to this, the verb is more important than the nominal phrases it combines with to form a complete clause. The verb determines the form of its dependents, for instance their case.

One way to depict dependencies is shown in Figure 11.1. The highest node is the verb

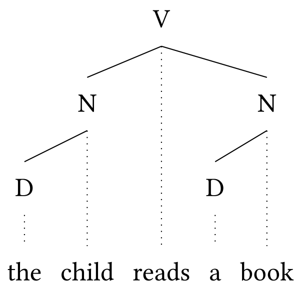

Figure 11.1: Analysis of *The child reads a book.*

reads. Its valence is a nominative NP (the subject) and an accusative NP (an object). This is depicted by the dependency links between the node representing the verb and the nodes representing the respective nouns. The nouns themselves require a determiner, which again is shown by the dependency links to *the* and *a* respectively. Note that the analysis presented here corresponds to the NP analysis that is assumed in HPSG for instance, that is, the noun selects its specifier (see Section 9.6.1). It should be noted though that the discussion whether an NP or a DP analysis is appropriate was also led within the Dependency Grammar framework (Hudson 1984: 90; Van Langendonck 1994; Hudson 2004). See Engel (1977) for an analysis with the N as head and Welke (2011: 31) for an analysis with the determiner as head.

The verb is the head of the clause and the nouns are called *dependents*. Alternative terms for head and dependent are *nucleus* and *satellite*, respectively.

An alternative way to depict the dependencies, which is used in Word Grammar (Hudson 2007), is provided in Figure 11.2 on the next page. This graph displays the grammatical functions rather than information about part of speech, but apart from this it is equivalent to the representation in Figure 11.1. The highest node in Figure 11.1 is labeled with the ROOT arrow in Figure 11.2. Downward links are indicated by the direction of the arrows.

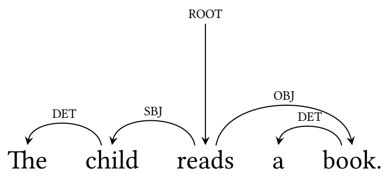

Figure 11.2: Alternative presentation of the analysis of *The child reads a book.*

A third form of representing the same dependencies provided in Figure 11.3 has the tree format again. This tree results if we pull the root node in Figure 11.2 upwards. Since we

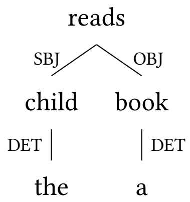

Figure 11.3: Alternative presentation of the analysis of *The child reads a book.*

have a clear visualization of the dependency relation that represents the nucleus above the dependents, we do not need to use arrows to encode this information. However, some variants of Dependency Grammar – for instance Word Grammar– use mutual dependencies. So for instance, some theories assume that *his* depends on *child* and *child* depends on *his* in the analysis of *his child.* If mutual dependencies have to be depicted, either arrows have to be used for all dependencies or some dependencies are represented by downward lines in hierarchical trees and other dependencies by arrows.

Of course part of speech information can be added to the Figures 11.2 and 11.3, grammatical function labels could be added to Figure 11.1, and word order can be added to Figure 11.3.

The above figures depict the dependency relation that holds between a head and the respective dependents. This can be written down more formally as an n-ary rule that is similar to phrase structure rules that were discussed in Chapter 2 (Gaifman 1965: 305; Hays 1964: 513; Baumgärtner 1970: 61; Heringer 1996: Section 4.1). For instance Baumgärtner suggests the rule in (1):

(1) $\chi \rightarrow \varphi_1 \ldots \varphi_i * \varphi_{i+2} \ldots \varphi_n, where\ 0 < i \leq n$

The asterisk in (1) corresponds to the word of the category χ. In our example, χ would be V, the position of the '$*$' would be taken by *reads*, and φ_1 and φ_3 would be N. Together with the rule in (2b) for the determiner-noun combination, the rule in (2a) would license the dependency tree in Figure 11.1.

(2) a. $V \rightarrow N * N$
b. $N \rightarrow D *$

Alternatively, several binary rules can be assumed that combine a head with its subject, direct object, or indirect object (Kahane 2009). Dependency rules will be discussed in more detail in Section 11.7.2, where dependency grammars are compared with phrase structure grammars.

11.1.2 Adjuncts

Another metaphor that was used by Tesnière is the drama metaphor. The core participants of an event are the *actants* and apart from this there is the background, the stage, the general setting. The actants are the arguments in other theories and the stage-describing entities are called *circumstants*. These circumstants are modifiers and usually analyzed as adjuncts in the other theories described in this book. As far as the representation of dependencies is concerned, there is not much of a difference between arguments and adjuncts in Dependency Grammar: Figure 11.4 shows the analysis of (3):

(3) The child often reads the book slowly.

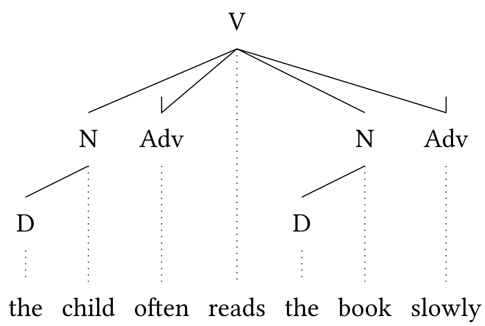

Figure 11.4: Analysis of *The child often reads the book slowly.*

The dependency annotation uses a technical device suggested by Engel (1977) to depict different dependency relations: adjuncts are marked with an additional line upwards from the adjunct node (see also Eroms 2000). An alternative way to specify the argument/adjunct or rather the actant/circumstant distinction is of course an explicit specification of the status as argument or adjunct. So like the grammatical functions in the previous figures, one can use explicit labels to this extent. German grammars and valence dictionaries often use the labels E and A for *Ergänzung* and *Angabe*, respectively.

11.1.3 Linearization

So far we have seen dependency graphs that had connections to words that were linearized in a certain order. However, in principle the order of the dependents is not determined by the dependency and therefore a Dependency Grammar has to contain additional statements that take care of the proper linearization of linguistic objects (stems, morphemes, words). Engel (2014: 50) assumes the dependency graph in Figure 11.5 for the sentences in (4).[6]

(4) a. Gestern war ich bei Tom.
yesterday was I with Tom
'I was with Tom yesterday.'

b. Ich war gestern bei Tom.
I was yesterday with Tom

c. Bei Tom war ich gestern.
with Tom was I yesterday

d. Ich war bei Tom gestern.
I was with Tom yesterday

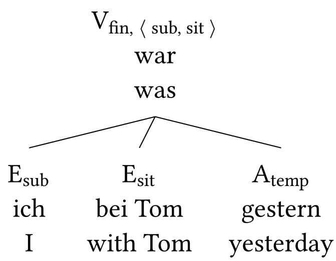

Figure 11.5: Dependency graph for several orders of *ich*, *war*, *bei Tom*, and *gestern* 'I was with Tom yesterday.' according to Engel (2014: 50)

According to Engel (2014: 50), the correct order is enforced by surface syntactic rules as for instance the rule that states that there is always exactly one element in the Vorfeld in declarative main clauses, the finite verb is in second position.[7,8] Furthermore, there are linearization rules that concern pragmatic properties, as for instance given before new information, and weak pronouns have to be placed into the Vorfeld or at the beginning of the Mittelfeld. This conception of linear order is problematic both for empirical and conceptual reasons and we will turn to it again in Section 11.7.1. It should be noted here

[6] Engel uses E_{sub} for the subject and E_{acc}, E_{dat}, and E_{gen} for the objects with respective cases.

[7] "Die korrekte Stellung ergibt sich dann zum Teil aus oberflächensyntaktischen Regeln (zum Beispiel: im Vorfeld des Konstativsatzes steht immer genau ein Element; das finite Verb steht an zweiter Stelle) […]"

[8] Engel (1970: 81) provides counterexamples to the claim that there is exactly one element in the *Vorfeld*. Related examples will be discussed in Section 11.7.1.

that approaches that deal with dependency alone admit discontinuous realizations of heads and their dependents. Without any further constraints, Dependency Grammars would share a problem that was already discussed on page 336 in Section 10.6.3 on Embodied Construction Grammar and in Section 10.6.4.4 with respect to Fluid Construction Grammar: one argument could interrupt another argument as in Figure 11.6. In order to

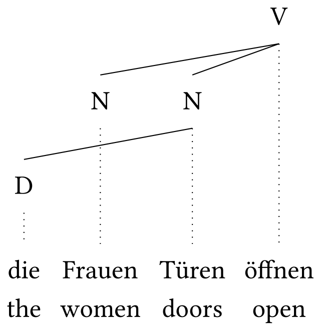

Figure 11.6: Unwanted analysis of *dass die Frauen Türen öffnen* 'that the women open doors'

exclude such linearizations in languages in which they are impossible, it is sometimes assumed that analyses have to be projective, that is crossing branches like those in Figure 11.6 are not allowed. This basically reintroduces the concept of constituency into the framework, since this means that all dependents of a head have to be realized close to the head unless special mechanisms for liberation are used (see for instance Section 11.5 on nonlocal dependencies).[9] Some authors explicitly use a phrase structure component to be able to formulate restrictions on serializations of constituents (Gerdes & Kahane 2001; Hellwig 2003).

11.1.4 Semantics

Tesnière already distinguished the participants of a verb in a way that was later common in theories of semantic roles. He suggested that the first actant is the agent, the second one a patient and the third a benefactive (Tesnière 2015: Chapter 106). Given that Dependency Grammar is a lexical framework, all lexical approaches to argument linking can be adopted. However, argument linking and semantic role assignment is just a small part of the problem that has to be solved when natural language expressions have to be assigned a semantics. Issues regarding the scope of adjuncts and quantifiers have to be

[9] While this results in units that are also assumed in phrase structure grammars, there is a difference: the units have category labels in phrase structure grammars (for instance NP), which is not the case in Dependency Grammars. In Dependency Grammars on just refers to the label of the head (for instance the N that belongs to *child* in Figure 11.4 or one refers to the head word directly (for instance, the word *child* in Figure 11.3). So there are fewer nodes in Dependency Grammar representations (but see the discussion in Section 11.7.2.3).

solved and it is clear that dependency graphs that represent dependencies without taking into account linear order are not sufficient. An unordered dependency graph assigns grammatical functions to a dependent of a head and hence is similar in many respects to an LFG f-structure.[10] For a sentence like (25a) on page 228, repeated here as (5), one gets the f-structure in (25b) on page 228. This f-structure contains a subject (*David*), an object (*a sandwich*), and an adjunct set with two elements (*at noon* and *yesterday*).

(5) David devoured a sandwich at noon yesterday.

This is exactly what is encoded in an unordered dependency graph. Because of this parallel it comes as no surprise that Bröker (2003: 308) suggested to use glue semantics (Dalrymple, Lamping & Saraswat 1993; Dalrymple 2001: Chapter 8) for Dependency Grammar as well. Glue semantics was already introduced in Section 7.1.5.

There are some variants of Dependency Grammar that have explicit treatments of semantics. One example is Meaning Text Theory (Mel'čuk 1988). Word Grammar is another one (Hudson 1991: Chapter 7; 2007: Chapter 5). The notations of these theories cannot be introduced here. It should be noted though that theories like Hudson's Word Grammar are rather rigid about linear order and do not assume that all the sentences in (4) have the same dependency structure (see Section 11.5). Word Grammar is more phrase structure grammar-like and therefore can have a semantics that interacts with constituent order in the way it is known from constituent-based theories.

11.2 Passive

Dependency Grammar is a lexical theory and valence is the central concept. For this reason, it is not surprising that the analysis of the passive is a lexical one. That is, it is assumed that there is a passive participle that has a different valence requirement than the active verb (Hudson 1990: Chapter 12; Eroms 2000: Section 10.3; Engel 2014: 53–54).

Our standard example in (6) is analyzed as shown in Figure 11.7 on the following page.

(6) [dass] der Weltmeister geschlagen wird
that the world.champion beaten is
'that the world champion is (being) beaten'

This figure is an intuitive depiction of what is going on in passive. A formalization of this would probably amount into a lexical rule for the personal passive. See Hellwig (2003: 629–630) for an explicit suggestion of a lexical rule for the analysis of the passive in English.

Note that *der Weltmeister* 'the world champion' is not an argument of the passive auxiliary *wird* 'is' in Engel's analysis. This means that subject verb agreement cannot be determined locally and some elaborated mechanism has to be developed for ensuring agreement.[11] Hudson (1990), Eroms (2000: Section 5.3) and Groß & Osborne (2009)

[10] Tim Osborne (p. c. 2015) reminds me that this is not true in all cases: for instance non predicative prepositions are not reflected in f-structures, but of course they are present in dependency graphs.

[11] This problem would get even more pressing for cases of the so-called remote passive:

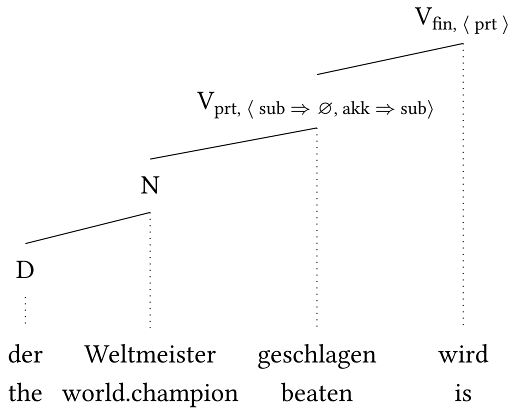

Figure 11.7: Analysis of [*dass*] *der Weltmeister geschlagen wird* 'that the world champion is (being) beaten' parallel to the analyses provided by Engel (2014: 53–54)

assume that subjects depend on auxiliaries rather than on the main verb. This requires some argument transfer as it is common in Categorial Grammar (see Section 8.5.2) and HPSG (Hinrichs & Nakazawa 1994). The adapted analysis that treats the subject of the participle as a subject of the auxiliary is given in Figure 11.8 on the next page.

(i) a. weil der Wagen zu reparieren versucht wurde
because the.NOM car to repair tried was
'because it was tried to repair the car'

b. weil die Wagen zu reparieren versucht wurden
because the.NOM cars to repair tried were
'because it was tried to repair the cars'

Here the object of *zu reparieren*, which is the object of a verb that is two levels embedded, agrees with the auxiliaries *wurde* 'was' and *wurden* 'were'. However, the question how to analyze these remote passives is open in Engel's system anyway and the solution of this problem probably involves the mechanism that was applied in HPSG: the arguments of *zu reparieren* are raised to the governing verb *versucht*, passive applies to this verb and turns the object into a subject which is then raised by the auxiliary, which explains the agreement between the underlying object of *zu reparieren* 'to repair' and *wurde* 'was'. Hudson (1997), working in the framework of Word Grammar, suggests an analysis of verbal complementation in German that involves what he calls *generalized raising*, that is, he assumes that both subjects and complements may be raised to the governing head. Note that such an analysis involving generalized raising would make an analysis of sentences like (i) straight forward, since the object would depend on the same head as the subject, namely on *hat* 'has' and hence can be ordered before the subject.

(ii) Gestern hat sich der Spieler verletzt.
yesterday has self the player injured
'The player injured himself yesterday.'

For a discussion of Groß & Osborne's account of (ii) see page 584.

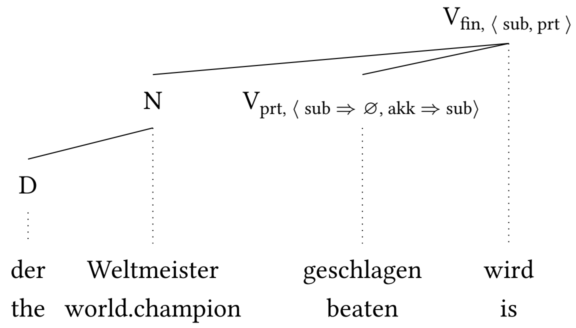

Figure 11.8: Analysis of [*dass*] *der Weltmeister geschlagen wird* 'that the world champion is (being) beaten' with the subject as dependent of the auxiliary

11.3 Verb position

In many Dependency Grammar publications on German, linearization issues are not dealt with and authors just focus on the dependency relations. The dependency relations between a verb and its arguments are basically the same in verb-initial and verb-final sentences. If we compare the dependency graphs of the sentences in (7) given in the Figures 11.9 and 11.10, we see that only the position of the verb is different, but the dependency relations are the same, as it should be.[12]

(7) a. [dass] jeder diesen Mann kennt
that everybody this man knows
'that everybody knows this man'

b. Kennt jeder diesen Mann?
knows everybody this man
'Does everybody know this man?'

The correct linearization of the verb with respect to its arguments and adjuncts is ensured by linearization constraints that refer to the respective topological fields. See Section 11.1.3 and Section 11.7.1 for further details on linearization.

[12] Eroms (2000) uses the part of speech *Pron* for pronouns like *jeder* 'everybody'. If information about part of speech plays a role in selection, this makes necessary a disjunctive specification of all valence frames of heads that govern nominal expressions, since they can either combine with an NP with internal structure or with a pronoun. By assigning pronouns the category N such a disjunctive specification is avoided. A pronoun differs from a noun in its valence (it is fully saturated, while a noun needs a determiner), but not in its part of speech. Eroms & Heringer (2003: 259) use the symbol N_pro for pronouns. If the pro-part is to be understood as a special property of items with the part of speech N, this is compatible with what I have said above: heads could then select for Ns. If N_pro and N are assumed to be distinct, atomic symbols, the problem remains.

Using N rather than Pron as part of speech for pronouns is standard in other versions of Dependency Grammar, as for instance Word Grammar (Hudson 1990: 167; Hudson 2007: 190). See also footnote 2 on page 55 on the distinction of pronouns and NPs in phrase structure grammars.

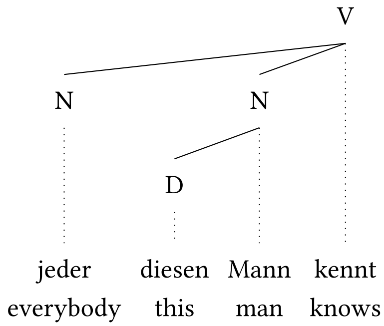

Figure 11.9: Analysis of [*dass*] *jeder diesen Mann kennt* ‘that everybody knows this man’

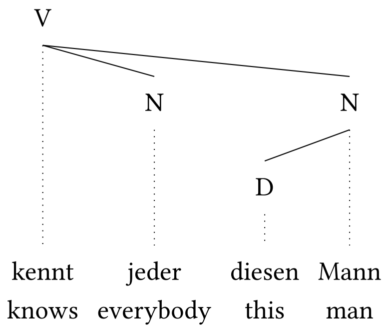

Figure 11.10: Analysis of *Kennt jeder diesen Mann?* ‘Does everybody know this man?’

11.4 Local reordering

The situation regarding local reordering is the same. The dependency relations of the sentence in (8b) are shown in Figure 11.11 on the facing page. The analysis of the sentence with normal order in (8a) was given in Figure 11.9 already.

(8) a. [dass] jeder diesen Mann kennt
that everybody this man knows
‘that everybody knows this man’

b. [dass] diesen Mann jeder kennt
that this man everybody knows
‘that everybody knows this man’

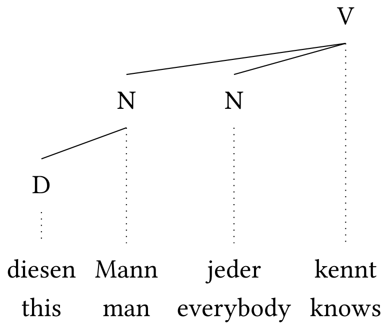

Figure 11.11: Analysis of [*dass*] *diesen Mann jeder kennt* 'that everybody knows this man'

11.5 Long-distance dependencies

There are several possibilities to analyze nonlocal dependencies in Dependency Grammar. The easiest is the one we already saw in the previous sections. One just focuses on the dependency relations and assumes that the order with the verb in second position is just one of the possible linearization variants (Eroms 2000: Section 9.6.2; Groß & Osborne 2009). Figure 11.12 shows the analysis of (9):

(9) [Diesen Mann] kennt jeder.
this man knows everybody
'Everyone knows this man.'

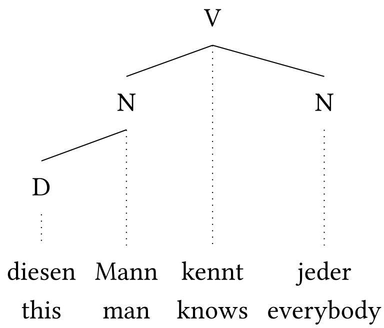

Figure 11.12: Analysis of *Diesen Mann kennt jeder.* 'This man, everybody knows.' without special treatment of fronting

Now, this is the simplest case, so let us look at the example in (10), which really involves a *nonlocal* dependency:

(10) Wen$_i$ glaubst du, daß ich _$_i$ gesehen habe.[13]
who believe you that I seen have
'Who do you think I saw?'

The dependency relations are depicted in Figure 11.13. This graph differs from most

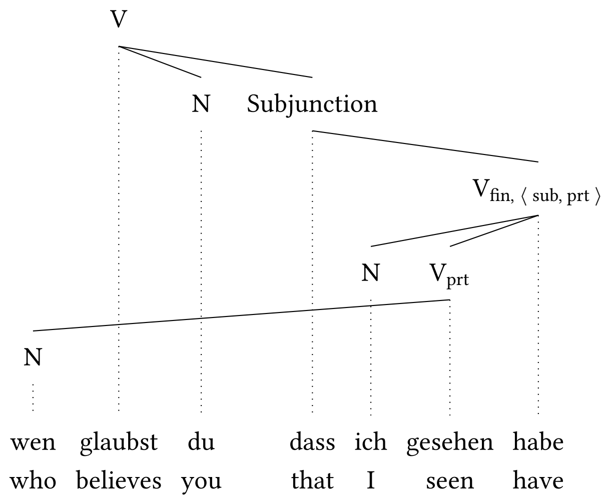

Figure 11.13: Non-projective analysis of *Wen glaubst du, dass ich gesehen habe?* 'Who do you think I saw?'

graphs we have seen before in not being projective, which means that there are crossing lines: the connection between V_{prt} and the N for *wen* 'who' crosses the lines connecting *glaubst* 'believes' and *du* 'you' with their category symbols. Depending on the version of Dependency Grammar assumed, this is seen as a problem or it is not. Let us explore the two options: if discontinuity of the type shown in Figure 11.13 is allowed for as in Heringer's and Eroms' grammars (Heringer 1996: 261; Eroms 2000: Section 9.6.2),[14] there has to be something in the grammar that excludes discontinuities that are ungrammatical. For instance, an analysis of (11) as in Figure 11.14 on the next page should be excluded.

(11) * Wen glaubst ich du, dass gesehen habe?
who.ACC believes I.NOM you.NOM that seen have
Intended: 'Who do you think I saw?'

Note that the order of elements in (11) is perfectly compatible with statements that refer to topological fields as suggested by Engel (2014: 50): there is a *Vorfeld* filled by *wen* 'who', there is a left sentence bracket filled by *glaubst* 'believe', and there is a *Mittelfeld*

[13] Scherpenisse (1986: 84).
[14] However, the authors mention the possibility of raising an extracted element to a higher node. See for instance Eroms & Heringer (2003: 260).

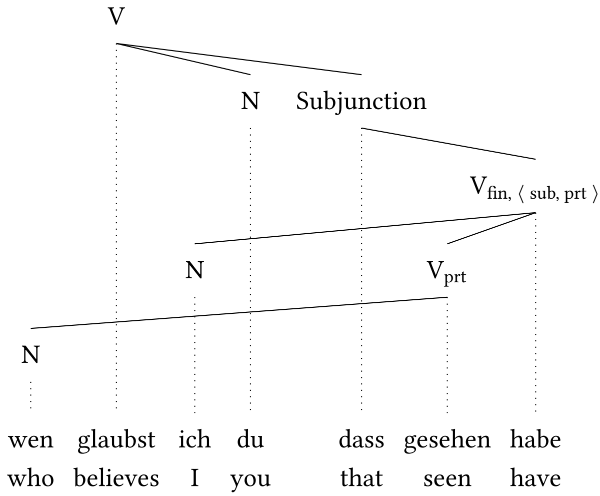

Figure 11.14: Unwanted dependency graph of * *Wen glaubst ich du, dass gesehen habe?* 'Who do you think I saw?'

filled by *ich* 'I', *du* 'you' and the clausal argument. Having pronouns like *ich* and *du* in the *Mittelfeld* is perfectly normal. The problem is that these two pronouns come from different clauses: *du* belongs to the matrix verb *glaubst* 'believe' while *ich* 'I' depends on (*gesehen* 'seen') *habe* 'have'. What has to be covered by a theory is the fact that fronting and extraposition targets the left-most and right-most positions of a clause, respectively. This can be modeled straight-forwardly in constituency-based approaches, as was shown in the previous chapters.

An alternative to assuming discontinuous constituents would be to assume additional mechanisms that promote the dependency of an embedded head to a higher head in the structure. Such an analysis was suggested by Kunze (1968), Hudson (1997, 2000), Kahane (1997), Kahane et al. (1998), and Groß & Osborne (2009). In what follows, I use the analysis by Groß & Osborne (2009) as an example for such analyses. Groß & Osborne depict the reorganized dependencies with a dashed line as in Figure 11.15 on the following page.[15,16] The origin of the dependency (V_{prt}) is marked with a *g* and the dependent is connected to the node to which it has risen (the top-most V) by a dashed line. Instead of realizing the accusative dependent of *gesehen* 'seen' locally, information about the missing element is transferred to a higher node and realized there.

The analysis of Groß & Osborne (2009) is not very precise. There is a *g* and there

[15] Eroms & Heringer (2003: 260) make a similar suggestion but do not provide any formal details.

[16] Note that Groß & Osborne (2009) do not assume a uniform analysis of simple and complex V2 sentences. That is, for cases that can be explained as local reordering they assume an analysis without rising. Their analysis of (9) is the one that is depicted in Figure 11.12. This leads to problems that are discussed in Section 11.7.1.

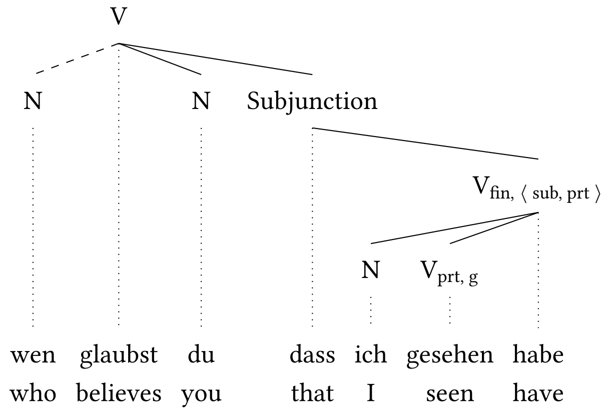

Figure 11.15: Projective analysis of *Wen glaubst du, dass ich gesehen habe?* 'Who do you think I saw?' involving rising

is a dashed line, but sentences may involve multiple nonlocal dependencies. In (12) for instance, there is a nonlocal dependency in the relative clauses *den wir alle begrüßt haben* 'who we all greeted have' and *die noch niemand hier gesehen hat* 'who yet nobody here seen has': the relative pronouns are fronted inside the relative clauses. The phrase *dem Mann, den wir alle kennen* 'the man who we all know' is the fronted dative object of *gegeben* 'given' and *die noch niemand hier gesehen hat* 'who yet nobody here seen has' is extraposed from *Frau* 'woman'.

(12) Dem Mann, den wir alle begrüßt haben, hat die Frau das Buch gegeben, die
the man who we all greeted have has the woman the book given who
noch niemand hier gesehen hat.
yet nobody here seen has
'The woman who nobody ever saw here gave the book to the man, who all of us greeted.'

So this means that the connections (dependencies) between the head and the dislocated element have to be made explicit. This is what Hudson (1997, 2000) does in his Word Grammar analysis of nonlocal dependencies: in addition to dependencies that relate a word to its subject, object and so on, he assumes further dependencies for extracted elements. For example, *wen* 'who' in (10) – repeated here as (13) for convenience – is the object of *gesehen* 'seen' and the extractee of *glaubst* 'believe' and *dass* 'that':

(13) Wen glaubst du, dass ich gesehen habe?
who believe you that I seen have
'Who do you believe that I saw?'

Hudson states that the use of multiple dependencies in Word Grammar corresponds to structure sharing in HPSG (Hudson 1997: 15). Nonlocal dependencies are modeled as a series of local dependencies as it is done in GPSG and HPSG. This is important since it allows one to capture extraction path marking effects (Bouma, Malouf & Sag 2001a: 1–2, Section 3.2): for instance, there are languages that use a special form of the complementizer for sentences from which an element is extracted. Figure 11.16 shows the analysis of (13) in Word Grammar. The links above the words are the usual dependency

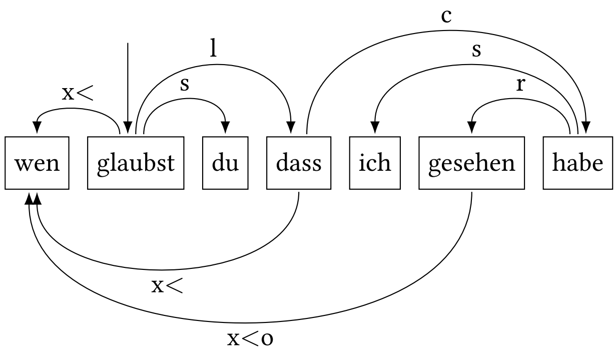

Figure 11.16: Projective analysis of *Wen glaubst du, dass ich gesehen habe?* 'Who do you think I saw?' in Word Grammar involving multiple dependencies

links for subjects (s) and objects (o) and other arguments (r is an abbreviation for *sharer*, which refers to verbal complements, l stands for *clausal complement*) and the links below the words are links for extractees (x<). The link from *gesehen* 'seen' to *wen* 'who' is special since it is both an object link and an extraction link (x<o). This link is an explicit statement that corresponds to the little g and the N that is marked by the dashed line in Figure 11.15. In addition to what is there in Figure 11.15, Figure 11.16 has an extraction link from *dass* 'that' to *wen* 'who'. One could use the graphic representation of Engel, Eroms, and Gross & Osborne to display the Word Grammar dependencies: one would simply add dashed lines from the V_{prt} node and from the Subjunction node to the N node dominating *wen* 'who'.[17]

[17] While this looks simple, I want to add that Word Grammar employs further principles that have to be fulfilled by wellformed structures. In the following I explain the *No-tangling Principle*, the *No-dangling Principle* and the *Sentence-root Principle.*

Principle 7 (The No-tangling Principle) *Dependency arrows must not tangle.*

Principle 8 (The No-dangling Principle) *Every word must have a parent.*

Principle 9 (The Sentence-root Principle) *In every non-compound sentence there is just one word whose parent is not a word but a contextual element.*

The No-tangling Principle ensures that there are no crossing dependency lines, that is, it ensures that structures are projective (Hudson 2000: 23). Since non-local dependency relations are established via the specific dependency mechanism, one wants to rule out the non-projective analysis. This principle also

Note that Hudson (1997: 16) assumes that the element in the *Vorfeld* is extracted even for simple sentences like (9). I will show in Section 11.7.1 why I think that this analysis has to be preferred over analyses that assume that simple sentences like (9) are just order variants of corresponding verb-initial or verb-final sentences.

11.6 New developments and theoretical variants

This section mainly deals with Tesnière's variant of Dependency Grammar. Section 11.6.1 deals with Tesnière's part of speech system and Section 11.6.2 describes the modes of combinations of linguistics objects that Tesnière assumed.

11.6.1 Tesnière's part of speech classification

As mentioned in the introduction, Tesnière is a central figure in the history of Dependency Grammar since he developed the first formal model (Tesnière 1959, 1980, 2015). There are many versions of Dependency Grammar today and most of them use the part of speech labels that are used in other theories as well (N, P, A, V, Adv, Conj, …). Tesnière had a system of four major categories: noun, verb, adjective, and adverb. The labels for these categories were derived from the endings that are used in Esperanto, that is, they are O, I, A, and E, respectively. These categories were defined semantically as specified in Table 11.1.[18] Tesnière assumed these categories to be universal and suggested that there are constraints in which way these categories may depend on others.

According to Tesnière, nouns and adverbs may depend on verbs, adjectives may depend on nouns, and adverbs may depend on adjectives or adverbs. This situation is depicted in the general dependency graph in Figure 11.17 on the facing page. The '*' means

rules out (i.b), where *green* depends on *peas* but is not adjacent to *peas*. Since *on* selects *peas* the arrow from *on* to *peas* would cross the one from *peas* to *green*.

(i) a. He lives on green peas.
   b. * He lives green on peas.

The No-dangling Principle makes sure that there are no isolated word groups that are not connected to the main part of the structure. Without this principle (i.b) could be analyzed with the isolated word *green* (Hudson 2000: 23).

The Sentence-root Principle is needed to rule out structures with more than one highest element. *glaubst* 'believe' is the root in Figure 11.16. There is no other word that dominates it and selects for it. The principle makes sure that there isn't another root. So the principle rules out situations in which all elements in a phrase are roots, since otherwise the No-dangling Principle would loose its force since it could trivially be fulfilled (Hudson 2000: 25).

I added this rather complicated set of principles in a footnote here in order to get a fair comparison with phrase structure-based proposals. If continuity is assumed for phrases in general, the three principles do not have to be stipulated. So, for example, LFG and HPSG do not need these three principles.

[18] As Weber (1997: 77) points out this categorization is not without problems: in what sense is *Angst* 'fear' a substance? Why should *glauben* 'believe' be a concrete process? See also Klein (1971: Section 3.4) for the discussion of *schlagen* 'to beat' and *Schlag* 'the beat' and similar cases. Even if one assumes that *Schlag* is derived from the concrete process *schlag-* by a transfer into the category O, the assumption that such Os stand for concrete substances is questionable.

Table 11.1: Semantically motivated part of speech classification by Tesnière

	substance	process
concrete	noun	verb
abstract	adjective	adverb

that there can be arbitrarily many dependencies between Es. It is of course easy to find

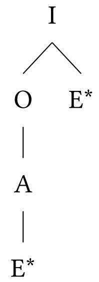

Figure 11.17: Universal configuration for dependencies according to Tesnière (I = verb, O = noun, A = adjective, E = adverb)

examples in which adjectives depend on verbs and sentences (verbs) depend on nouns. Such cases are handled via so-called *transfers* in Tesnière's system. Furthermore, conjunctions, determiners, and prepositions are missing form this set of categories. For the combination of these elements with their dependents Tesnière used special combinatoric relations: junction and transfer. We will deal with these in the following subsection.

11.6.2 Connection, junction, and transfer

Tesnière (1959) suggested three basic relations between nodes: connection, junction, and transfer. Connection is the simple relation between a head and its dependents that we already covered in the previous sections. Junction is a special relation that plays a role in the analysis of coordination and transfer is a tool that allows one to change the category of a lexical item or a phrase.

11.6.2.1 Junction

Figure 11.18 on the next page illustrates the junction relation: the two conjuncts *John* and *Mary* are connected with the conjunction *and.* It is interesting to note that both of the conjuncts are connected to the head *laugh.*

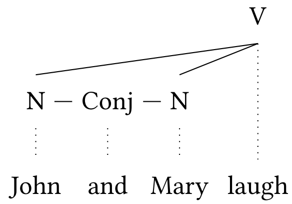

Figure 11.18: Analysis of coordination using the special relation *junction*

In the case of two coordinated nouns we get dependency graphs like the one in Figure 11.19. Both nouns are connected to the dominating verb and both nouns dominate

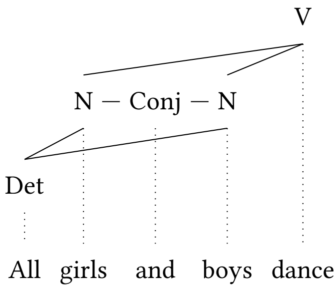

Figure 11.19: Analysis of coordination using the special relation *junction*

the same determiner.

An alternative to such a special treatment of coordination would be to treat the conjunction as the head and the conjuncts as its dependents (I did not use Tesnière's category labels here to spare the reader the work of translating I to V and N to O). The only problem of such a proposal would be the category of the conjunction. It cannot be Conj since the governing verb does not select a Conj, but an N. The trick that could be applied here is basically the same trick as in Categorial Grammar (see Section 21.6.2): the category of the conjunction in Categorial Grammar is (X\X)/X. We have a functor that takes two arguments of the same category and the result of the combination is an object that has the same category as the two arguments. Translating this approach to Dependency Grammar, one would get an analysis as the one depicted in Figure 11.20 on the next page rather than the ones in Figure 11.18 and Figure 11.19. The figure for *all girls and boys* looks rather strange since both the determiner and the two conjuncts depend on the conjunction, but since the two Ns are selecting a Det, the same is true for the result of the coordination. In Categorial Grammar notation, the category of the conjunction would be ((NP\Det)\(NP\Det))/(NP\Det) since X is instantiated by the nouns which would have the category (NP\Det) in an analysis in which the noun is the head and the determiner is the dependent.

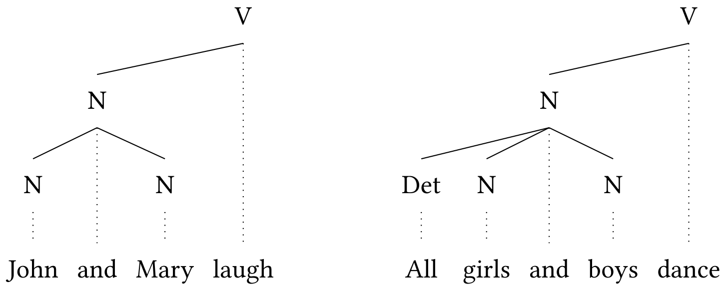

Figure 11.20: Analysis of coordination without *junction* and the conjunction as head

Note that both approaches have to come up with an explanation of subject–verb agreement. Tesnière's original analysis assumes two dependencies between the verb and the individual conjuncts.[19] Since the conjuncts are singular and the verb is plural, agreement cannot be modeled in tandem with dependency relations in this approach. If the second analysis finds ways of specifying the agreement properties of the coordination in the conjunction, the agreement facts can be accounted for without problems.

The alternative to a headed approach that is depicted in Figure 11.20 is an unheaded one. Several authors working in phrase structure-based frameworks suggested analyses of coordination without a head and such analyses are also assumed in Dependency Grammar (Hudson 1988; Kahane 1997). Hudson (1988) and others who make similar assumptions assume a phrase structure component for coordination, that is, the two nouns and the conjunction are combined to form a larger object that has properties that do not correspond to the properties of any of the combined words.

Similarly, the junction-based analysis of coordination poses problems for the interpretation of the representations. If semantic role assignment happens in parallel to dependency relations, there would be a problem with graphs like the one in Figure 11.18, since the semantic role of *laugh* cannot be filled by *John* and *Mary* simultaneously. Rather it is filled by one entity, namely the one that refers to the set that contains John and Mary. This semantic representation would belong to the phrase *John and Mary* and the natural candidate for being the top-most entity in this coordination is the *and*, since it embeds the meaning of *John* and the meaning of *Mary*: *and′*(*John′*, *Mary′*).

Such junctions are also assumed for the coordination of verbs. This however is not without problems, since adjuncts can have scope over the conjunct that is closest to them or over the whole coordination. An example is the following sentence from Levine (2003: 217):

(14) Robin came in, found a chair, sat down, and whipped off her logging boots in exactly thirty seconds flat.

[19] Eroms (2000: 467) notes the agreement problem and describes the facts. In his analysis, he connects the first conjunct to the governing head, although it seems to be more appropriate to assume an internally structured coordination structure and then connect the highest conjunction.

The adjunct *in exactly thirty seconds flat* can refer either to *whipped off her logging boots* as in (15a) or scope over all three conjuncts together as in (15b):

(15) a. Robin came in, found a chair, sat down, and [[pulled off her logging boots] in exactly thirty seconds flat].

b. Robin [[came in, found a chair, sat down, and pulled off her logging boots] in exactly thirty seconds flat].

c. Robin came in in exactly thirty seconds flat and Robin found a chair in exactly thirty seconds flat and Robin pulled off her logging boots in exactly thirty seconds flat.

The Tesnièreian analysis in Figure 11.21 corresponds to (15c), while an analysis that treats the conjunction as the head as in Figure 11.22 on the next page corresponds to (15b).

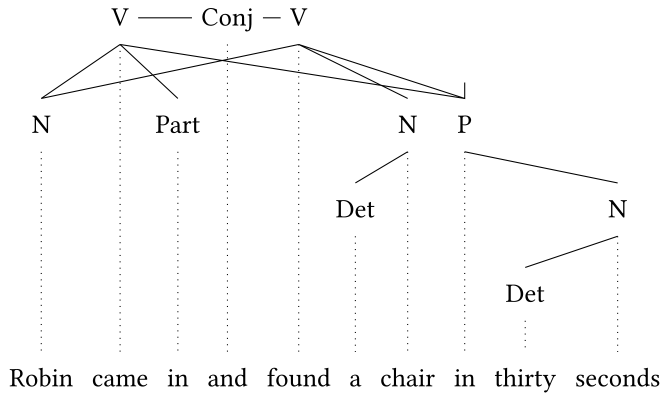

Figure 11.21: Analysis of verb coordination involving the junction relation

Levine (2003: 217) discusses these sentences in connection to the HPSG analysis of extraction by Bouma, Malouf & Sag (2001a). Bouma, Malouf & Sag suggest an analysis in which adjuncts are introduced lexically as dependents of a certain head. Since adjuncts are introduced lexically, the coordination structures basically have the same structure as the ones that are assumed in a Tesnièreian analysis. It may be possible to come up with a way to get the semantic composition right even though the syntax does not correspond to the semantic dependencies (see Chaves (2009) for suggestions), but it is clear that it is simpler to derive the semantics from a syntactic structure that corresponds to what is going on in semantics.

11.6.2.2 Transfer

Transfers are used in Tesnière's system for the combination of words or phrases with a head of one of the major categories (for instance nouns) with words in minor categories

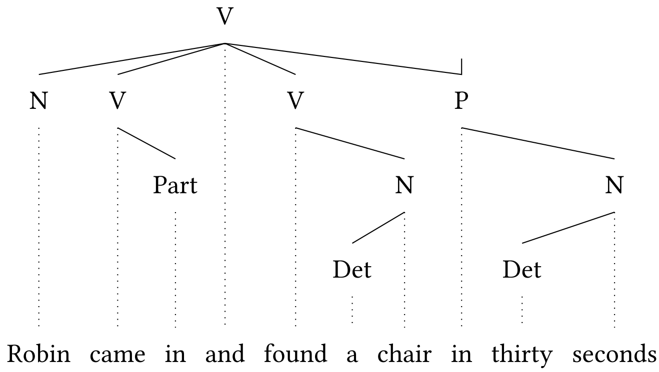

Figure 11.22: Analysis of verb coordination involving the connection relation

(for instance prepositions). In addition, transfers can transfer a word or phrase into another category without any other word participating.

Figure 11.23 shows an example of a transfer. The preposition *in* causes a category

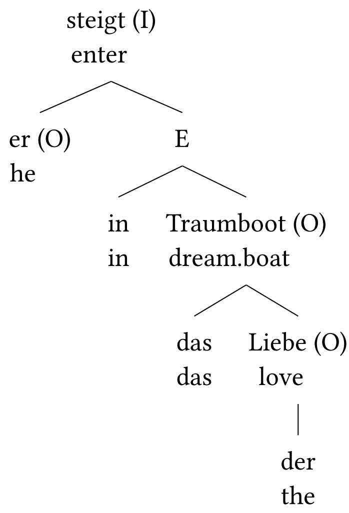

Figure 11.23: Transfer with an example adapted from Weber (1997: 83)

change: while *Traumboot* 'dream boat' is an O (noun), the combination of the preposition and the noun is an E. The example shows that Tesnière used the grammatical category to encode grammatical functions. In theories like HPSG there is a clear distinction: there is information about part of speech on the one hand and the function of elements as modifiers and predicates on the other hand. The modifier function is encoded by the selectional feature MOD, which is independent of the part of speech. It is therefore possible to have modifying and non-modifying adjectives, modifying and non-modifying prepositional phrases, modifying and non-modifying noun phrases and so on. For the example at hand, one would assume a preposition with directional semantics that selects for an NP. The preposition is the head of a PP with a filled MOD value.

Another area in which transfer is used is morphology. For instance, the derivation of *frappant* 'striking' by suffixation of *-ant* to the verb stem *frapp* is shown in Figure 11.24. Such transfers can be subsumed under the general connection relation, if the affix is

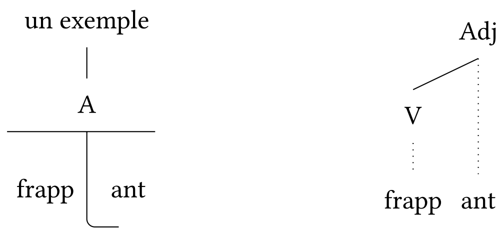

Figure 11.24: Transfer in morphology and its reconceptualization as normal dependency

treated as the head. Morphologists working in realizational morphology and construction morphology argue against such morpheme-based analyses since they involve a lot of empty elements for conversions as for instance the conversion of the verb *play* into the noun *play* (see Figure 11.25). Consequently, lexical rules are assumed for derivations

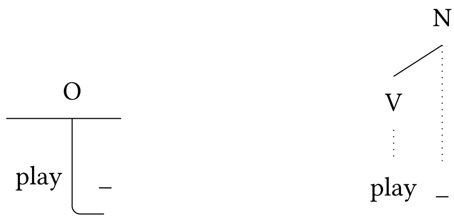

Figure 11.25: Conversion as transfer from I (verb) to O (substantive) and as dependency with an empty element of the category N as head

and conversions in theories like HPSG. HPSG lexical rules are basically equivalent to unary branching rules (see the discussion of (37) on page 283 and Section 19.5). The affixes are integrated into the lexical rules or into realization functions that specify the morphological form of the item that is licensed by the lexical rule.

Concluding it can be said that transfers correspond to

- binary-branching phrase structure rules, if a word or phrase is combined with another word,

- unary phrase structure rules or binary branching phrase structure rules together with an empty head, if a phrase is converted to another category without any additional element present.
- a (unary) lexical rule, if a word or stem is mapped to a word or a stem.

For further discussion of the relation between Tesnière's transfer rules and constituency rules see Osborne & Kahane (2015: Section 4.9.1–4.9.2). Osborne & Kahane point out that transfer rules can be used to model exocentric constructions, that is, constructions in which there is no single part that could be identified as the head. For more on headless constructions see Section 11.7.2.4.

11.6.3 Scope

As Osborne & Kahane (2015: lix) point out, Tesnière uses so-called polygraphs to represent scopal relations. So, since *that you saw yesterday* in (16) refers to *red cars* rather than *cars* alone, this is represented by a line that starts at the connection between *red* and *cars* rather than on one of the individual elements (Tesnière 2015: 150, Stemma 149).

(16) red cars that you saw yesterday

Tesnière's analysis is depicted in the left representation in Figure 11.26. It is worth noting that this representation corresponds to the phrase structure tree on the right of Figure 11.26. The combination B between *red* and *cars* corresponds to the B node in the right-

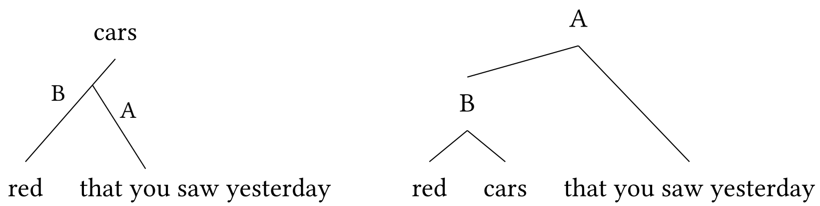

Figure 11.26: Tesnière's way of representing scope and the comparison with phrase structure-based analyses by Osborne & Kahane (2015: lix)

hand figure and the combination A of *red cars* and *that you saw yesterday* corresponds to the A node. So, what is made explicit and is assigned a name in phrase structure grammars remains nameless in Tesnière's analysis, but due to the assumption of polygraphs, it is possible to refer to the combinations. See also the discussion of Figure 11.46, which shows additional nodes that Hudson assumes in order to model semantic relations.

11.7 Summary and classification

Proponents of Dependency Grammar emphasize the point that Dependency Grammar is much simpler than phrase structure grammars, since there are fewer nodes and the general concept is more easy to grasp (see for instance Osborne 2014: Section 3.2, Section 7). This is indeed true: Dependency Grammar is well-suited for teaching grammar in introductory classes. However, as Sternefeld & Richter (2012: 285) pointed out in a rather general discussion, a simple syntax has the price of a complex semantics and vice versa. So, in addition to the dependency structure that is described in Dependency Syntax, one needs other levels. One level is the level of semantics and another one is linearization. As far as linearization is concerned, Dependency Grammar has two options: assuming continuous constituents, that is, projective structures, or allowing for discontinuous constituents. These options will be discussed in the following subsections. Section 11.7.2 compares dependency grammars with phrase structure grammars and shows that projective Dependency Grammars can be translated into phrase structure grammars. It also shows that non-projective structures can be modeled in theories like HPSG. The integration of semantics is dealt with in Section 11.7.2.3 and it will become clear that once other levels are taken into account, Dependency Grammars are not necessarily simpler than phrase structure grammars.

11.7.1 Linearization

We have seen several approaches to linearization in this chapter. Many just assume a dependency graph and some linearization according to the topological fields model. As was argued in Section 11.5, allowing discontinuous serialization of a head and its dependents opens Pandora's box. I discussed the analysis of nonlocal dependencies by Kunze (1968), Hudson (1997, 2000), Kahane, Nasr & Rambow (1998), and Groß & Osborne (2009). With the exception of Hudson those authors assume that dependents of a head rise to a dominating head only in those cases in which a discontinuity would arise otherwise. However, there seems to be a reason to assume that fronting should be treated by special mechanisms even in cases that allow for continuous serialization. For instance, the ambiguity or lack of ambiguity of the examples in (17) cannot be explained in a straightforward way:

(17) a. Oft liest er das Buch nicht.
often reads he the book not
'It is often that he does not read the book.' or 'It is not the case that he reads the book often.'

b. dass er das Buch nicht oft liest
that he the book not often reads
'It is not the case that he reads the book often.'

c. dass er das Buch oft nicht liest
that he the book often not reads
'It is often that he does not read the book.'

The point about the three examples is that only (17a) is ambiguous. Even though (17c) has the same order as far as *oft* 'often' and *nicht* 'not' are concerned, the sentence is not ambiguous. So it is the fronting of an adjunct that is the reason for the ambiguity. The dependency graph for (17a) is shown in Figure 11.27. Of course the dependencies for

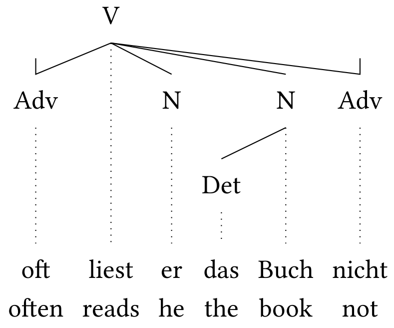

Figure 11.27: Dependency graph for *Oft liest er das Buch nicht.* 'He does not read the book often.'

(17b) and (17c) do not differ, so the graphs would be the same only differing in serialization. Therefore, differences in scope could not be derived from the dependencies and complicated statements like (18) would be necessary:

(18) If a dependent is linearized in the *Vorfeld* it can both scope over and under all other adjuncts of the head it is a dependent of.

Eroms (1985: 320) proposes an analysis of negation in which the negation is treated as the head, that is, the sentence in (19) has the structure in Figure 11.28 on the following page.[20]

(19) Er kommt nicht.
he comes not
'He does not come.'

This analysis is equivalent to analyses in the Minimalist Program that assume a NegP and it has the same problem: the category of the whole object is Adv, but it should be V. This is a problem since higher predicates may select for a V rather than an Adv. See for instance the analysis of embedded sentences like (21) below.

The same is true for constituent negation or other scope bearing elements. For example, the analysis of (20) would have to be the one in Figure 11.29 on the next page.

(20) der angebliche Mörder
the alleged murderer

This structure would have the additional problem of being non-projective. Eroms does treat the determiner differently from what is assumed here, so this type of non-projectivity may not be a problem for him. However, the head analysis of negation would

[20] But see Eroms (2000: Section 11.2.3).

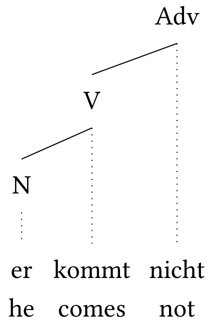

Figure 11.28: Analysis of negation according to Eroms (1985: 320)

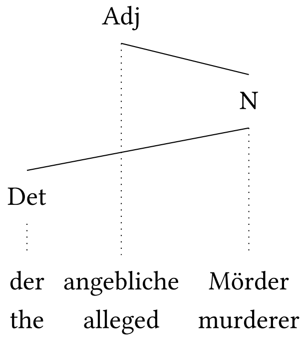

Figure 11.29: Analysis that would result if one considered all scope-bearing adjuncts to be heads

result in non-projectivity in so-called coherent constructions in German. The following sentence has two readings: in the first reading, the negation scopes over *singen* 'sing' and in the second one over *singen darf* 'sing may'.

(21) dass er nicht singen darf
that he not sing may
'that he is not allowed to sing' or 'that he is allowed not to sing'

The reading in which *nicht* 'not' scopes over the whole verbal complex would result in the non-projective structure that is given in Figure 11.30 on the facing page. Eroms also considers an analysis in which the negation is a word part ('Wortteiläquivalent'), but this does not help here since first the negation and the verb are not adjacent in V2 contexts like (17a) and even in verb final contexts like (21). Eroms would have to assume that the object to which the negation attaches is the whole verbal complex *singen darf* 'sing may', that is, a complex object consisting of two words.

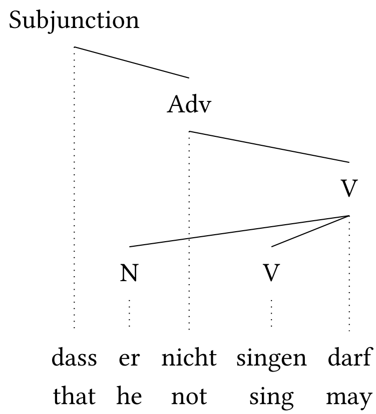

Figure 11.30: Analysis that results if one assumes the negation to be a head

So, this leaves us with the analysis provided in Figure 11.27 and hence with a problem since we have one structure with two possible adjunct realizations that correspond to different readings, which is not predicted by an analysis that treats the two possible linearizations simply as alternative orderings.

Thomas Groß (p. c. 2013) suggested an analysis in which *oft* does not depend on the verb but on the negation. This corresponds to constituent negation in phrase structure approaches. The dependency graph is shown at the left-hand side in Figure 11.31. The

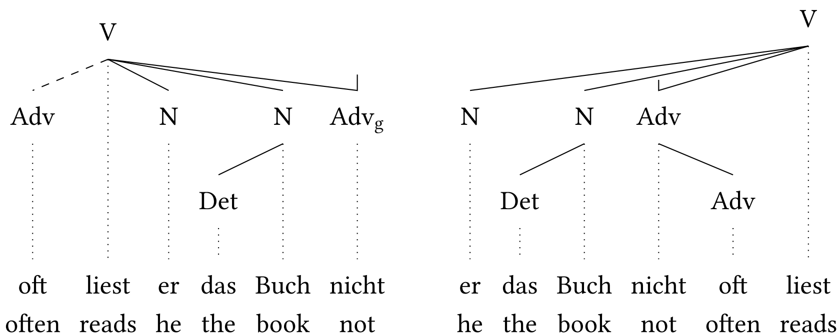

Figure 11.31: Dependency graph for *Oft liest er das Buch nicht.* 'He does not read the book often.' according to Groß and verb-final variant

figure at the right-hand side shows the graph for the corresponding verb-final sentence. The reading that corresponds to constituent negation can be illustrated with contrastive expressions. While in (22a) it is just the *oft* 'often' that is negated, it is *oft gelesen* 'often read' that is in the scope of negation in (22b).

(22) a. Er hat das Buch nicht oft gelesen, sondern selten.
he has the book not often read but seldom
'He did not read the book often, but seldom.'

b. Er hat das Buch nicht oft gelesen, sondern selten gekauft.
he has the book not often read but seldom bought
'He did not read the book often but rather bought it seldom.'

These two readings correspond to the two phrase structure trees in Figure 11.32. Note

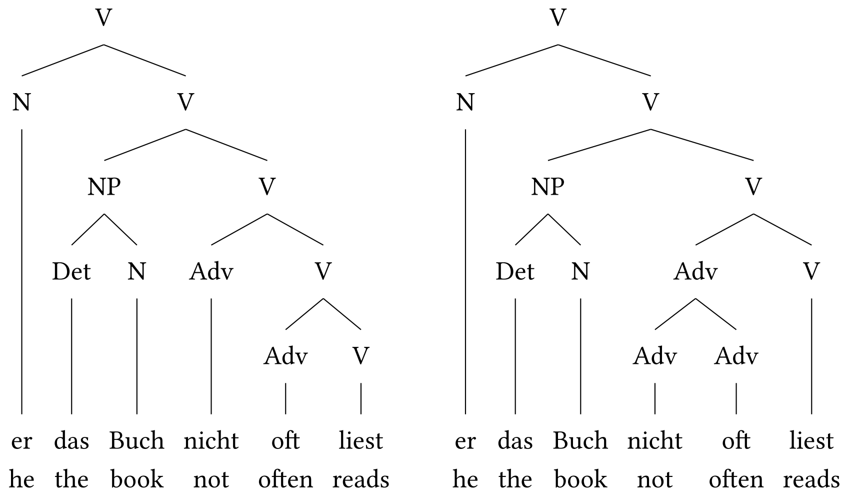

Figure 11.32: Possible syntactic analyses for *er das Buch nicht oft liest* 'He does not read the book often.'

that in an HPSG analysis, the adverb *oft* would be the head of the phrase *nicht oft* 'not often'. This is different from the Dependency Grammar analysis suggested by Groß. Furthermore, the Dependency Grammar analysis has two structures: a flat one with all adverbs depending on the same verb and one in which *oft* depends on the negation. The phrase structure-based analysis has three structures: one with the order *oft* before *nicht*, one with the order *nicht* before *oft* and the one with direct combination of *nicht* and *oft*. The point about the example in (17a) is that one of the first two structures is missing in the Dependency Grammar representations. This probably does not make it impossible to derive the semantics, but it is more difficult than it is in constituent-based approaches.

Furthermore, note that models that directly relate dependency graphs to topological fields will not be able to account for sentences like (23).

(23) Dem Saft eine kräftige Farbe geben Blutorangen.[21]
the juice a strong color give blood.oranges
'Blood oranges give the juice a strong color.'

The dependency graph of this sentence is given in Figure 11.33.

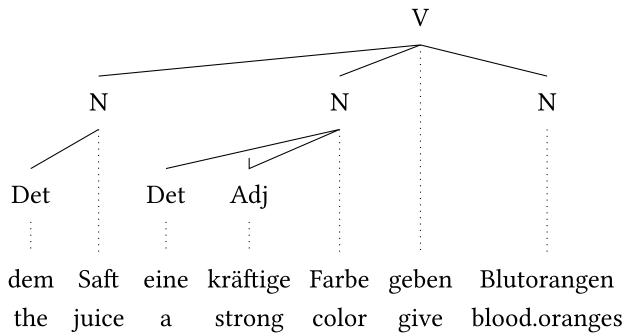

Figure 11.33: Dependency graph for *Dem Saft eine kräftige Farbe geben Blutorangen.* 'Blood oranges give the juice a strong color.'

Such apparent multiple frontings are not restricted to NPs. Various types of dependents can be placed in the *Vorfeld*. An extensive discussion of the data is provided in (Müller 2003a). Additional data have been collected in a research project on multiple frontings and information structure (Bildhauer 2011). Any theory that is based on dependencies alone and that does not allow for empty elements is forced to give up the restriction that is commonly assumed in the analysis of V2 languages, namely that the verb is in second position. In comparison, analyses like GB and those HPSG variants that assume an empty verbal head can assume that a projection of such a verbal head occupies the *Vorfeld*. This explains why the material in the *Vorfeld* behaves like a verbal projection containing a visible verb: such *Vorfelds* are internally structured topologically, they may have a filled *Nachfeld* and even a particle that fills the right sentence bracket. See Müller (2005c, 2015b) for further data, discussion, and a detailed analysis. The equivalent of the analysis in Gross & Osborne's framework (2009) would be something like the graph that is shown in Figure 11.34 on the following page, but note that Groß & Osborne (2009: 73) explicitly reject empty elements, and in any case an empty element that is stipulated just to get the multiple fronting cases right would be entirely ad hoc.[22] It is important to note that the issue is not solved by simply dropping the V2 constraint and allowing

[21] Bildhauer & Cook (2010) found this example in the *Deutsches Referenzkorpus* (DeReKo), hosted at Institut für Deutsche Sprache, Mannheim: http://www.ids-mannheim.de/kl/projekte/korpora

[22] I stipulated such an empty element in a linearization-based variant of HPSG allowing for discontinuous constituents (Müller 2002b), but later modified this analysis so that only continuous constituents are allowed and verb position is treated as head-movement and multiple frontings involve the same empty verbal head as is used in the verb movement analysis (Müller 2005c, 2015b).

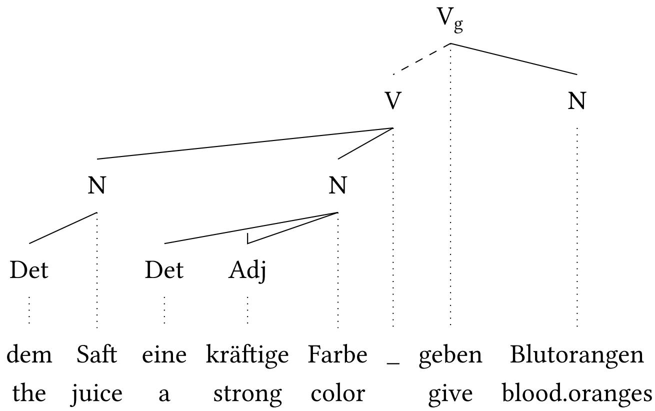

Figure 11.34: Dependency graph for *Dem Saft eine kräftige Farbe geben Blutorangen.* 'Blood oranges give the juice a strong color.' with an empty verbal head for the *Vorfeld*

dependents of the finite verb to be realized to its left, since the fronted constituents do not necessarily depend on the finite verb as the examples in (24) show:

(24) a. [Gezielt] [Mitglieder] [im Seniorenbereich] wollen die Kendoka allerdings nicht werben.[23]
targeted members in.the senior.citizens.sector want.to the Kendoka however not recruit
'However, the Kendoka do not intend to target the senior citizens sector with their member recruitment strategy.'

b. [Kurz] [die Bestzeit] hatte der Berliner Andreas Klöden […] gehalten.[24]
briefly the best.time had the Berliner Andreas Klöden held
'Andreas Klöden from Berlin had briefly held the record time.'

And although the respective structures are marked, such multiple frontings can even cross clause boundaries:

(25) Der Maria einen Ring glaube ich nicht, daß er je schenken wird.[25]
the.NOM Maria a.ACC ring believes I not that he ever give will
'I don't think that he would ever give Maria a ring.'

If such dependencies are permitted it is really difficult to constrain them. The details cannot be discussed here, but the reader is referred to (Müller 2005c, 2015b).

[23] taz, 07.07.1999, p. 18. Quoted from Müller (2002b).
[24] Märkische Oderzeitung, 28./29.07.2001, p. 28.
[25] Fanselow (1993: 67).

Note also that Engel's statement regarding the linear order in German sentences (2014: 50) referring to one element in front of the finite verb (see footnote 7) is very imprecise. One can only guess what is intended by the word *element*. One interpretation is that it is a continuous constituent in the classical sense of constituency-based grammars. An alternative would be that there is a continuous realization of a head and some but not necessarily all of its dependents. This alternative would allow an analysis of extraposition with discontinuous constituents of (26) as it is depicted in Figure 11.35.

(26) Ein junger Kerl stand da, mit langen blonden Haaren, die sein Gesicht
a young guy stood there with long blond hair that his face
einrahmten, […][26]
framed
'A young guy was standing there with long blond hair that framed his face'

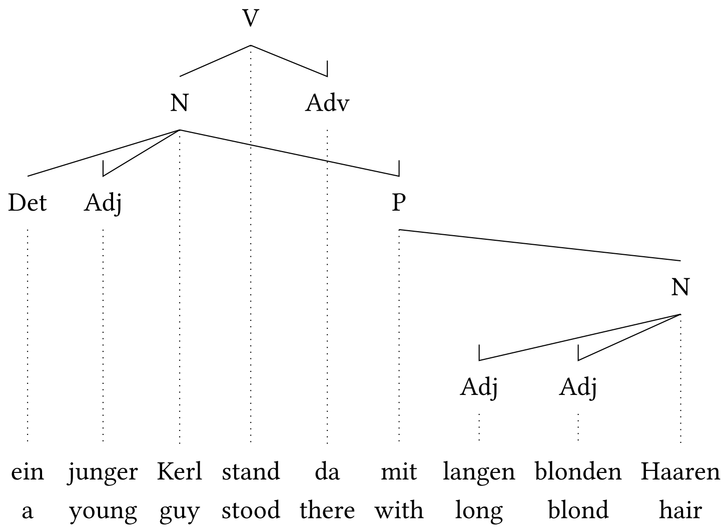

Figure 11.35: Dependency graph for *Ein junger Kerl stand da, mit langen blonden Haaren.* 'A young guy was standing there with long blond hair.' with a discontinuous constituent in the *Vorfeld*

A formalization of such an analysis is not trivial, since one has to be precise about what exactly can be realized discontinuously and which parts of a dependency must be realized continuously. Kathol & Pollard (1995) developed such an analysis of extraposition in the framework of HPSG. See also Müller (1999a: Section 13.3). I discuss the basic mechanisms for such linearization analyses in HPSG in the following section.

[26] Charles Bukowski, *Der Mann mit der Ledertasche.* München: Deutscher Taschenbuch Verlag, 1994, p. 201, translation by Hans Hermann.

11.7.2 Dependency Grammar vs. phrase structure grammar

This section deals with the relation between Dependency Grammars and Phrase Structure Grammars. I first show that projective Dependency Grammars can be translated into Phrase Structure Grammars (Section 11.7.2.1) and then deal with non-projective DGs and show how they can be captured in linearization-based HPSG (Section 11.7.2.2). Section 11.7.2.3 argues for the additional nodes that are assumed in phrase structure-based theories and Section 11.7.2.4 discusses headless constructions, which pose a problem for all Dependency Grammar accounts.

11.7.2.1 Translating projective Dependency Grammars into phrase structure grammars

As noted by Gaifman (1965), Covington (1990: 234), Oliva (2003) and Hellwig (2006: 1093), certain projective headed phrase structure grammars can be turned into Dependency Grammars by moving the head one level up to replace the dominating node. So in an NP structure, the N is shifted into the position of the NP and all other connections remain the same. Figure 11.36 illustrates. Of course this procedure cannot be applied to all phrase

Figure 11.36: Analysis of *a book* in a phrase structure and a Dependency Grammar analysis

structure grammars directly since some involve more elaborate structure. For instance, the rule S → NP, VP cannot be translated into a dependency rule, since NP and VP are both complex categories.

In what follows, I want to show how the dependency graph in Figure 11.1 on page 364 can be recast as headed phrase structure rules that license a similar tree, namely the one in Figure 11.37 on the facing page. I did not use the labels NP and VP to keep the two figures maximally similar. The P part of NP and VP refers to the saturation of a projection and is often ignored in figures. See Chapter 9 on HPSG, for example. The grammar that licenses the tree is given in (27), again ignoring valence information.

(27)	N → D N	N → child	D → the	D → a
	V → N V N	N → book	V → reads	

If one replaces the N and V in the right-hand side of the two left-most rules in (27) with the respective lexical items and then removes the rules that license the words, one arrives at the following lexicalized variant of the grammar:

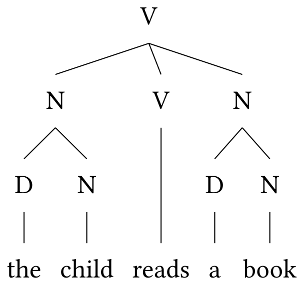

Figure 11.37: Analysis of *The child reads a book.* in a phrase structure with flat rules

(28) N → D book   D → the
N → D child   D → a
V → N reads N

Lexicalized means that every partial tree licensed by a grammar rule contains a lexical element. The grammar in (28) licenses exactly the tree in Figure 11.1.[27]

One important difference between classical phrase structure grammars and Dependency Grammars is that the phrase structure rules impose a certain order on the daughters. That is, the V rule in (28) implies that the first nominal projection, the verb, and the second nominal projection have to appear in the order stated in the rule. Of course this ordering constraint can be relaxed as it is done in GPSG. This would basically permit any order of the daughters at the right hand side of rules. This leaves us with the integration of adjuncts. Since adjuncts depend on the head as well (see Figure 11.4 on page 366), a rule could be assumed that allows arbitrarily many adjuncts in addition to the arguments. So the V rule in (28) would be changed to the one in (29):

(29) V → N reads N Adv*

See page 191 for a similar rule in GPSG and see Kasper (1994) for an HPSG analysis of German that assumes entirely flat structures and integrates an arbitrary number of adjuncts.

Such generalized phrase structures would give us the equivalent of projective Dependency Grammars.[28] However, as we have seen, some researchers allow for crossing

[27] As mentioned on page 365, Gaifman (1965: 305), Hays (1964: 513), Baumgärtner (1970: 57) and Heringer (1996: 37) suggest a general rule format for dependency rules that has a special marker ('*' and '~', respectively) in place of the lexical words in (28). Heringer's rules have the form in (29):

(i) X[Y1, Y2, ~, Y3]

X is the category of the head, Y1, Y2, and Y3 are dependents of the head and '~' is the position into which the head is inserted.

[28] Sylvain Kahane (p. c. 2015) states that binarity is important for Dependency Grammars, since there is one rule for the subject, one for the object and so on (as for instance in Kahane 2009, which is an implementation

edges, that is, for discontinuous constituents. In what follows, I show how such Dependency Grammars can be formalized in HPSG.

11.7.2.2 Non-projective Dependency Grammars and phrase structure grammars with discontinuous constituents

The equivalent to non-projective dependency graphs are discontinuous constituents in phrase structure grammars. In what follows I want to provide one example of a phrase structure-based theory that permitted discontinuous structures. Since, as I will show, discontinuities can be modeled as well, the difference between phrase structure grammars and Dependency Grammars boils down to the question of whether units of words are given a name (for instance NP) or not.

The technique that is used to model discontinuous constituents in frameworks like HPSG goes back to Mike Reape's work on German (1991; 1992; 1994). Reape uses a list called DOMAIN in order to represent the daughters of a sign in the order in which they appear at the surface of an utterance. (30) shows an example in which the DOM value of a headed-phrase is computed from the DOM value of the head and the list of non-head daughters.

(30) $$\textit{headed-phrase} \Rightarrow \begin{bmatrix} \text{HEAD-DTR}|\text{DOM} & \boxed{1} \\ \text{NON-HEAD-DTRS} & \boxed{2} \\ \text{DOM} & \boxed{1} \bigcirc \boxed{2} \end{bmatrix}$$

The symbol '$\bigcirc$' stands for the *shuffle* relation. *shuffle* relates three lists A, B and C, iff C contains all elements from A and B and the order of the elements in A and the order of the elements of B is preserved in C. (31) shows the combination of two sets with two elements each:

(31) $$\begin{aligned} \langle\, a, b \,\rangle \bigcirc \langle\, c, d \,\rangle = & \langle\, a, b, c, d \,\rangle \vee \\ & \langle\, a, c, b, d \,\rangle \vee \\ & \langle\, a, c, d, b \,\rangle \vee \\ & \langle\, c, a, b, d \,\rangle \vee \\ & \langle\, c, a, d, b \,\rangle \vee \\ & \langle\, c, d, a, b \,\rangle \end{aligned}$$

The result is a disjunction of six lists. *a* is ordered before *b* and *c* before *d* in all of these lists, since this is also the case in the two lists $\langle\, a, b \,\rangle$ and $\langle\, c, d \,\rangle$ that have been combined. But apart from this, *b* can be placed before, between or after *c* and *d*. Every word comes with a domain value that is a list that contains the word itself:

of Dependency Grammar in the HPSG formalism). However, I do not see any reason not to allow for flat structures. For instance, Ginzburg & Sag (2000: 364) assumed a flat rule for subject auxiliary inversion in HPSG. In such a flat rule the specifier/subject and the other complements are combined with the verb in one go. This would also work for more than two valence features that correspond to grammatical functions like subject, direct object, indirect object. See also Footnote 27 on flat rules.

(32) Domain contribution of single words, here *gibt* ‘gives’:

$$\boxed{1}\begin{bmatrix}\textsc{phon} & \langle\ \textit{gibt}\ \rangle \\ \textsc{synsem} & \ldots \\ \textsc{dom} & \langle\ \boxed{1}\ \rangle\end{bmatrix}$$

The description in (32) may seem strange at first glance, since it is cyclic, but it can be understood as a statement saying that *gibt* contributes itself to the items that occur in linearization domains.

The constraint in (33) is responsible for the determination of the PHON values of phrases:

(33) $$\textit{phrase} \Rightarrow \begin{bmatrix}\textsc{phon} & \boxed{1} \oplus \ldots \oplus \boxed{n} \\ \textsc{dom} & \left\langle \begin{bmatrix}\textsc{phon} & \boxed{1} \\ \textit{sign}\end{bmatrix}, \ldots, \begin{bmatrix}\textsc{phon} & \boxed{n} \\ \textit{sign}\end{bmatrix} \right\rangle\end{bmatrix}$$

It states that the PHON value of a sign is the concatenation of the PHON values of its DOMAIN elements. Since the order of the DOMAIN elements corresponds to their surface order, this is the obvious way to determine the PHON value of the whole linguistic object.

Figure 11.38 shows how this machinery can be used to license binary branching structures with discontinuous constituents. Words or word sequences that are separated by

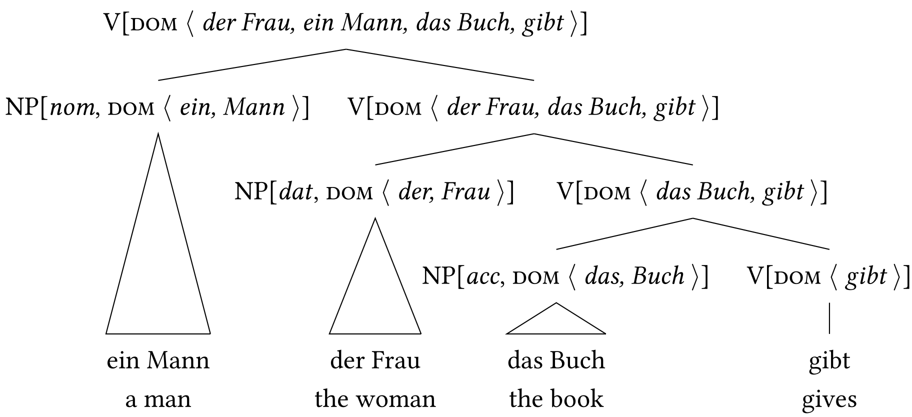

Figure 11.38: Analysis of *dass der Frau ein Mann das Buch gibt* ‘that a man gives the woman the book’ with binary branching structures and discontinuous constituents

commas stand for separate domain objects, that is, ⟨ *das, Buch* ⟩ contains the two objects *das* and *Buch* and ⟨ *das Buch, gibt* ⟩ contains the two objects *das Buch* and *gibt*. The important point to notice here is that the arguments are combined with the head in the order accusative, dative, nominative, although the elements in the constituent order domain are realized in the order dative, nominative, accusative rather than nominative, dative, accusative, as one would expect. This is possible since the formulation of the computation of the DOM value using the shuffle operator allows for discontinuous constituents. The node for *der Frau das Buch gibt* 'the woman the book gives' is discontinuous: *ein Mann* 'a man' is inserted into the domain between *der Frau* 'the woman' and *das Buch* 'the book'. This is more obvious in Figure 11.39, which has a serialization of NPs that corresponds to their order.

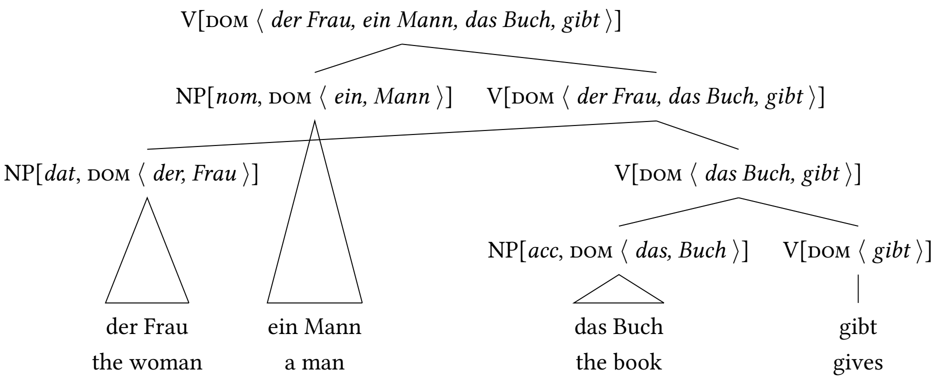

Figure 11.39: Analysis of *dass der Frau ein Mann das Buch gibt* 'that a man gives the woman the book' with binary branching structures and discontinuous constituents showing the discontinuity

Such binary branching structures were assumed for the analysis of German by Kathol (1995, 2000) and Müller (1995, 1996c, 1999a, 2002a), but as we have seen throughout this chapter, Dependency Grammar assumes flat representations (but see Footnote 28 on page 395). Schema 6 licenses structures in which all arguments of a head are realized in one go.[29]

Schema 6 (Head-Argument Schema (flat structure))
head-argument-phrase ⇒

$$\begin{bmatrix} \text{SYNSEM}|\text{LOC}|\text{CAT}|\text{SUBCAT}\ \langle\rangle \\ \text{HEAD-DTR}|\text{SYNSEM}|\text{LOC}|\text{CAT}|\text{SUBCAT}\ \boxed{1} \\ \text{NON-HEAD-DTRS}\ \boxed{1} \end{bmatrix}$$

To keep the presentation simple, I assume that the SUBCAT list contains descriptions of complete signs. Therefore the whole list can be identified with the list of non-head

[29] I assume here that all arguments are contained in the SUBCAT list of a lexical head, but nothing hinges on that. One could also assume several valence features and nevertheless get a flat structure. For instance, Borsley (1989: 339) suggests a schema for auxiliary inversion in English and verb-initial sentences in Welsh that refers to both the valence feature for subjects and for complements and realizes all elements in a flat structure.

daughters.[30] The computation of the DOM value can be constrained in the following way:

(34) $$\textit{headed-phrase} \Rightarrow \begin{bmatrix} \text{HEAD-DTR} & \boxed{1} \\ \text{NON-HEAD-DTRS} & \langle \boxed{2}, \ldots, \boxed{n} \rangle \\ \text{DOM} & \langle \boxed{1} \rangle \bigcirc \langle \boxed{2} \rangle \bigcirc \ldots \bigcirc \langle \boxed{n} \rangle \end{bmatrix}$$

This constraint says that the value of DOM is a list that is the result of shuffling singleton lists that each contain one daughter as elements. The result of such a shuffle operation is a disjunction of all possible permutations of the daughters. This seems to be overkill for something that GPSG already gained by abstracting away from the order of the elements on the right hand side of a phrase structure rule, but note that this machinery can be used to reach even freer orders: by referring to the DOM values of the daughters rather than the daughters themselves, it is possible to insert individual words into the DOM list.

(35) $$\textit{headed-phrase} \Rightarrow \begin{bmatrix} \text{HEAD-DTR}|\text{DOM} & \boxed{1} \\ \text{NON-HEAD-DTRS} & \langle [\ \text{DOM}\ \boxed{2}\ ] \ldots [\ \text{DOM}\ \boxed{n}\ ] \rangle \\ \text{DOM} & \langle \boxed{1} \rangle \bigcirc \langle \boxed{2} \rangle \bigcirc \ldots \bigcirc \langle \boxed{n} \rangle \end{bmatrix}$$

Using this constraint we have DOM values that basically contain all the words in an utterance in any permutation. What we are left with is a pure Dependency Grammar without any constraints on projectivity. With such a grammar we could analyze the non-projecting structure of Figure 11.6 on page 368 and much more. The analysis in terms of domain union is shown in Figure 11.40. It is clear that such discontinuity is

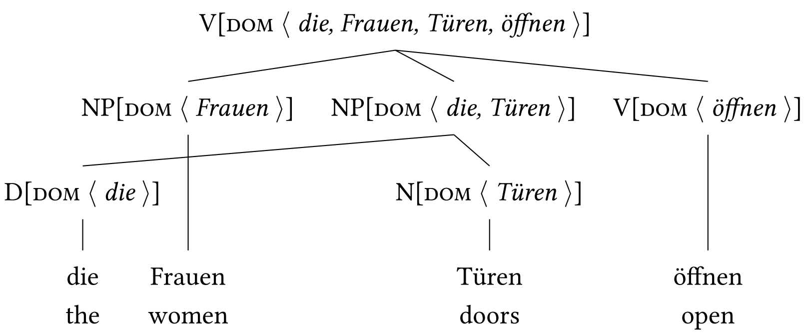

Figure 11.40: Unwanted analysis of *dass die Frauen Türen öffnen* 'that the women open doors' using Reape-style constituent order domains

unwanted and hence one has to have restrictions that enforce continuity. One possible restriction is to require projectivity and hence equivalence to phrase structure grammars in the sense that was discussed above.

30 Without this assumption one would need a relational constraint that maps a list with descriptions of type *synsem* onto a list with descriptions of type *sign*. See Meurers (1999c: 198) for details.

There is some dispute going on about the question whether constituency/dependency is primary/necessary to analyze natural language: while Hudson (1980), and Engel (1996) claim that dependency is sufficient, a claim that is shared by dependency grammarians (according to Engel (1996)), Leiss (2003) claims that it is not. In order to settle the issue, let us take a look at some examples:

(36) Dass Peter kommt, klärt nicht, ob Klaus spielt.
that Peter comes resolves not whether Klaus plays
'That Peter comes does not resolve the question whether Klaus plays.'

If we know the meaning of the utterance, we can assign a dependency graph to it. Let us assume that the meaning of (36) is something like (37):

(37) $\neg$ *resolve*$'$(*that*$'$(*come*$'$(*Peter*$'$)),*whether*$'$(*play*$'$(*Klaus*$'$)))

With this semantic information, we can of course construct a dependency graph for (36). The reason is that the dependency relation is reflected one by one in the semantic representation in (37). The respective graph is given in Figure 11.41. But note that this

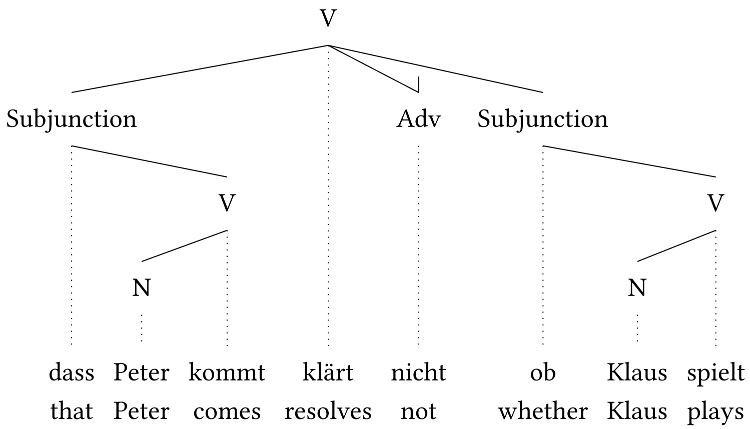

Figure 11.41: The dependency graph of *Dass Peter kommt, klärt nicht, ob Klaus spielt.* 'That Peter comes does not resolve the question whether Klaus plays.' can be derived from the semantic representation.

does not hold in the general case. Take for instance the example in (38):

(38) Dass Peter kommt, klärt nicht, ob Klaus kommt.
that Peter comes resolves not whether Klaus plays
'That Peter comes does not resolve the question whether Klaus comes.'

Here the words *dass* and *kommt* appear twice. Without any notion of constituency or restrictions regarding adjacency, linear order and continuity, we cannot assign a dependency graph unambiguously. For instance, the graph in Figure 11.42 on the next page is perfectly compatible with the meaning that this sentence has: *dass* dominates *kommt* and *kommt* dominates *Peter*, while *ob* dominates *kommt* and *kommt* dominates *Klaus*. I

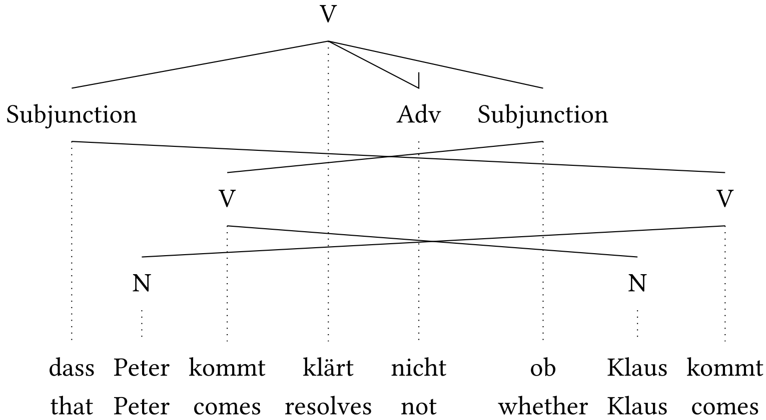

Figure 11.42: The dependency graph of *Dass Peter kommt, klärt nicht, ob Klaus kommt.* 'That Peter comes does not resolve the question whether Klaus comes.' is not unambiguously determined by semantics.

used the wrong *kommt* in the dependency chains, but this is an issue of linearization and is independent of dependency. As soon as one takes linearization information into account, the dependency graph in Figure 11.42 is ruled out since *ob* 'whether' does not precede its verbal dependent *kommt* 'comes'. But this explanation does not work for the example in Figure 11.6 on page 368. Here, all dependents are linearized correctly, it is just the discontinuity of *die* and *Türen* that is inappropriate. If it is required that *die* and *Türen* are continuous, we have basically let constituents back in (see Footnote 9 on page 368).

Similarly, non-projective analyses without any constraints regarding continuity would permit the word salad in (39b):

(39) a. Deshalb klärt, dass Peter kommt, ob Klaus spielt.
therefore resolves that Peter comes whether Klaus plays

b. Deshalb klärt dass ob Peter Klaus kommt spielt.
therefore resolves that whether Peter Klaus comes plays

(39b) is a variant of (39a) in which the elements of the two clausal arguments are in correct order with respect to each other, but both clauses are discontinuous in such a way that the elements of each clause alternate. The dependency graph is shown in Figure 11.43 on the next page. As was explained in Section 10.6.4.4 on the analysis of nonlocal dependencies in Fluid Construction Grammar, a grammar of languages like English and German has to constrain the clauses in such a way that they are continuous with the exception of extractions to the left. A similar statement can be found in

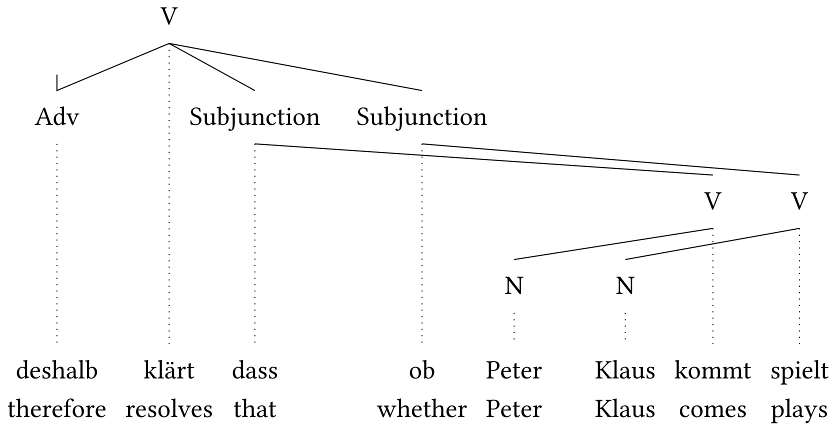

Figure 11.43: The dependency graph of the word salad *Deshalb klärt dass ob Peter Klaus kommt spielt.* 'Therefore resolves that whether Peter Klaus comes plays' which is admitted by non-projective Dependency Grammars that do not restrict discontinuity

Hudson (1980: 192). Hudson also states that an item can be fronted in English, provided all of its dependents are fronted with it (p. 184). This "item with all its dependents" is the constituent in constituent-based grammars. The difference is that this object is not given an explicit name and is not assumed to be a separate entity containing the head and its dependents in most Dependency Grammars. See however Hellwig (2003) for an explicit proposal that assumes that there is a linguistic object that represents the whole constituent rather than just the lexical head.

Summing up what has been covered in this section so far, I have shown what a phrase structure grammar that corresponds to a certain Dependency Grammar looks like. I have also shown how discontinuous constituents can be allowed for. However, there are issues that remained unaddressed so far: not all properties that a certain phrase has are identical to its lexical head and the differences have to be represented somewhere. I will discuss this in the following subsection.

11.7.2.3 Features that are not identical between heads and projections

As Oliva (2003) has pointed out, the equivalence of Dependency Grammar and HPSG only holds up as far as HEAD values are concerned. That is, the node labels in dependency graphs correspond to the HEAD values in an HPSG. There are, however, additional features like CONT for the semantics and SLASH for nonlocal dependencies. These values usually differ between a lexical head and its phrasal projections. For illustration, let us have a look at the phrase *a book*. The semantics of the lexical material and the complete phrase is given in (40) (for lambda expressions see Section 2.3):

(40) a. *a*: $\lambda P \lambda Q \exists x(P(x) \wedge Q(x))$
b. *book*: $\lambda y\ (book'(y)))$
c. *a book*: $\lambda Q \exists x(book'(x) \wedge Q(x))$

Now, the problem for the Dependency Grammar notation is that there is no NP node that could be associated with the semantics of *a book* (see Figure 11.36 on page 394), the only thing present in the tree is a node for the lexical N, that is, the node for *book*.[31] This is not a big problem, however: the lexical properties can be represented as part of the highest node as the value of a separate feature. The N node in a dependency graph would then have a CONT value that corresponds to the semantic contribution of the complete phrase and a LEX-CONT value that corresponds to the contribution of the lexical head of the phrase. So for *a book* we would get the following representation:

(41) $\begin{bmatrix} \textsc{cont} & \lambda Q \exists x(book'(x) \wedge Q(x)) \\ \textsc{lexical-cont} & \lambda y\ (book'(y))) \end{bmatrix}$

With this kind of representation one could maintain analyses in which the semantic contribution of a head together with its dependents is a function of the semantic contribution of the parts.

Now, there are probably further features in which lexical heads differ from their projections. One such feature would be SLASH, which is used for nonlocal dependencies in HPSG and could be used to establish the relation between risen element and the head in an approach á la Groß & Osborne (2009). Of course we can apply the same trick again. We would then have a feature LEXICAL-SLASH. But this could be improved and the features of the lexical item could be grouped under one path. The general skeleton would then be (42):

(42) $\begin{bmatrix} \textsc{cont} & \\ \textsc{slash} & \\ \textsc{lexical} & \begin{bmatrix} \textsc{cont} \\ \textsc{slash} \end{bmatrix} \end{bmatrix}$

But if we rename LEXICAL to HEAD-DTR, we basically get the HPSG representation.

Hellwig (2003: 602) states that his special version of Dependency Grammar, which he calls Dependency Unification Grammar, assumes that governing heads select complete nodes with all their daughters. These nodes may differ in their properties from their head (p. 604). They are in fact constituents. So this very explicit and formalized variant of Dependency Grammar is very close to HPSG, as Hellwig states himself (p. 603).

[31] Hudson (2003: 391–392) is explicit about this: "In dependency analysis, the dependents modify the head word's meaning, so the latter carries the meaning of the whole phrase. For example, in *long books about linguistics*, the word *books* means 'long books about linguistics' thanks to the modifying effect of the dependents." For a concrete implementation of this idea see Figure 11.44 on the following page.

An alternative is to assume different representational levels as in Meaning Text Theory (Mel'čuk 1981). In fact the CONT value in HPSG is also a different representational level. However, this representational level is in sync with the other structure that is build.

Hudson's Word Grammar (2015) is also explicitly worked out and as will be shown it is rather similar to HPSG. The representation in Figure 11.44 is a detailed description of what the abbreviated version in Figure 11.45 stands for. What is shown in the first

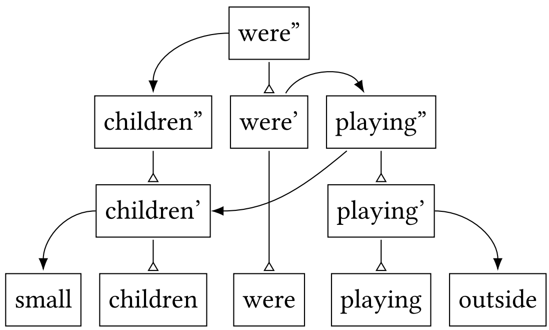

Figure 11.44: Analysis of *Small children are playing outside.* according to Hudson (2015)

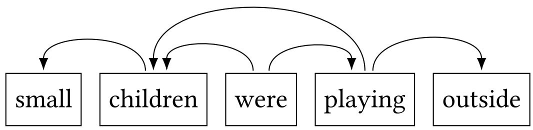

Figure 11.45: Abbreviated analysis of *Small children are playing outside.* according to Hudson (2015)

diagram is that a combination of two nodes results in a new node. For instance, the combination of *playing* and *outside* yields *playing*′, the combination of *small* and *children* yields *children*′, and the combination of *children*′ and *playing*′ yields *playing*′′. The combination of *were* and *playing*′′ results in *were*′ and the combination of *children*′′ and *were*′ yields *were*′′. The only thing left to explain is why there is a node for *children* that is not the result of the combination of two nodes, namely *children*′′. The line with the triangle at the bottom stands for default inheritance. That is, the upper node inherits all properties from the lower node by default. Defaults can be overridden, that is, information at the upper node may differ from information at the dominated node. This makes it possible to handle semantics compositionally: nodes that are the result of the combination of two nodes have a semantics that is the combination of the meaning of the two combined nodes. Turning to *children* again, *children*′ has the property that it must be adjacent to *playing*, but since the structure is a raising structure in which *children* is raised to the subject of *were*, this property is overwritten in a new instance of *children*, namely *children*′′.

The interesting point now is that we get almost a normal phrase structure tree if we replace the words in the diagram in Figure 11.44 by syntactic categories. The result of the replacement is shown in Figure 11.46 on the next page. The only thing unusual in this

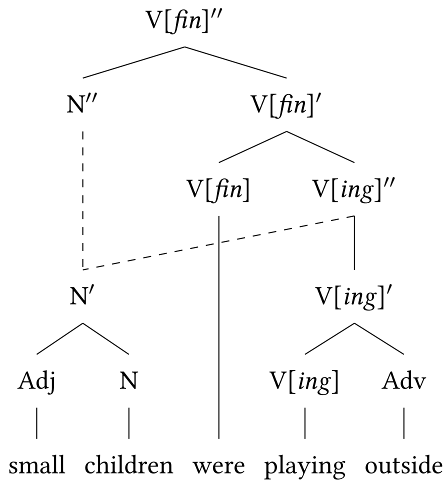

Figure 11.46: Analysis of *Small children are playing outside.* with category symbols

graph (marked by dashed lines) is that N′ is combined with V[*ing*]′ and the mother of N′, namely N″, is combined with V[*fin*]′. As explained above, this is due to the analysis of raising in Word Grammar, which involves multiple dependencies between a raised item and its heads. There are two N nodes (N′ and N″) in Figure 11.46 and two instances of *children* in Figure 11.44. Apart from this, the structure corresponds to what a HPSG would license. The nodes in Hudson's diagram that are connected with lines with triangles at the bottom are related to their children using default inheritance. This too is rather similar to those versions of HPSG that use default inheritance. For instance, Ginzburg & Sag (2000: 33) use a Generalized Head Feature Principle that projects all properties of the head daughter to the mother by default.

So, the conclusion of this section is that the only principled difference between phrase structure grammars and Dependency Grammar is the question of how much intermediate structure is assumed: is there a VP without the subject? Are there intermediate nodes for adjunct attachment? It is difficult to decide these questions in the absence of fully worked out proposals that include semantic representations. Those proposals that are worked out – like Hudson's and Hellwig's – assume intermediate representations, which makes these approaches rather similar to phrase structure-based approaches. If one compares the structures of these fully worked out variants of Dependency Grammar with phrase structure grammars, it becomes clear that the claim that Dependency Grammars are simpler is unwarranted. This claim holds for compacted schematic representations like Figure 11.45 but it does not hold for fully worked out analyses.

11.7.2.4 Non-headed constructions

Hudson (1980: Section 4.E) discusses headless constructions like those in (43):

(43) a. the rich
b. the biggest
c. the longer the stem
d. (with) his hat over his eyes

He argues that the terms *adjective* and *noun* should be accompanied by the term *substantive*, which subsumes both terms. Then he suggests that *if a rule needs to cover the constructions traditionally referred to as noun-phrases, with or without heads, it just refers to 'nouns', and this will automatically allow the constructions to have either substantives or adjectives as heads.* (p. 195) The question that has to be asked here, however, is what the internal dependency structure of substantive phrases like *the rich* would be. The only way to connect the items seems to be to assume that the determiner is dependent on the adjective. But this would allow for two structures of phrases like *the rich man*: one in which the determiner depends on the adjective and one in which it depends on the noun. So underspecification of part of speech does not seem to solve the problem. Of course all problems with non-headed constructions can be solved by assuming empty elements (see Section 2.4.1 for the assumption of an empty head in a phrase structure grammar for noun phrases). This has been done in HPSG in the analysis of relative clauses (Pollard & Sag 1994: Chapter 5). English and German relative clauses consist of a phrase that contains a relative word and a sentence in which the relative phrase is missing. Pollard and Sag assumed an empty relativizer that selects for the relative phrase and the clause with a gap (Pollard & Sag 1994: 216–217). Similar analyses can be found in Dependency Grammar (Eroms 2000: 291).[32] Now, the alternative to empty elements are phrasal constructions (see Chapter 19 on empty elements in general and Subsection 21.10.3 on relative clauses in particular). Sag (1997) working on relative clauses in English suggested a phrasal analysis of relative clauses in which the relative phrase and the clause from which it is extracted form a new phrase. A similar analysis was assumed by Müller (1996c) and is documented in Müller (1999a: Chapter 10). As was discussed in Section 8.6 it is neither plausible to assume the relative pronoun or some other element in the relative phrase to be the head of the entire relative clause, nor is it plausible to assume the verb to be the head of the entire relative clause (pace Sag), since relative clauses modify

[32] The Dependency Grammar representations usually have a *d*- element as the head of the relative clause. However, since the relative pronoun is also present in the clause and since the *d*- is not pronounced twice, assuming an additional *d*- head is basically assuming an empty head.

Another option is to assume that words may have multiple functions: so, a relative pronoun may be both a head and a dependent simultaneously (Tesnière 2015: Chapter 246, §8–11; Osborne & Kahane 2015: xlvi; Kahane 2009: 129–130). At least the analysis of Kahane is an instance of the Categorial Grammar analysis that was discussed in Section 8.6 and it suffers from the same problems: if the relative pronoun is a head that selects for a clause that is missing the relative pronoun, it is not easy to see how this analysis extends to cases of pied-piping like (i) in which the extracted element is a complete phrase containing the relative pronoun rather than just the pronoun itself.

(i) die Frau, von deren Schwester ich ein Bild gesehen habe
the woman of whose sister I a picture seen have
'the woman of whose sister I saw a picture'

$\overline{\text{N}}$s, something that projections of (finite) verbs usually do not do. So assuming an empty head or a phrasal schema seems to be the only option.

Chapter 21 is devoted to the discussion of the question whether certain phenomena should be analyzed as involving phrase structural configurations or whether lexical analyses are better suited in general or for modeling some phenomena. I argue there that all phenomena that interact with valence should be treated lexically. But there are other phenomena as well and Dependency Grammar is forced to assume lexical analyses for all linguistic phenomena. There always has to be some element on which others depend. It has been argued by Jackendoff (2008) that it does not make sense to assume that one of the elements in N-P-N constructions like those in (44) is the head.

(44) a. day by day, paragraph by paragraph, country by country
b. dollar for dollar, student for student, point for point
c. face to face, bumper to bumper
d. term paper after term paper, picture after picture
e. book upon book, argument upon argument

Of course there is a way to model all the phenomena that would be modeled by a phrasal construction in frameworks like GPSG, CxG, HPSG, or Simpler Syntax: an empty head. Figure 11.47 shows the analysis of *student after student*. The lexical item for the empty N

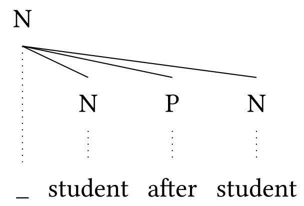

Figure 11.47: Dependency Grammar analysis of the N-P-N Construction with empty head

would be very special, since there are no similar non-empty lexical nouns, that is, there is no noun that selects for two bare Ns and a P.

Bragmann (2015) pointed out an additional aspect of the N-P-N construction, which makes things more complicated. The pattern is not restricted to two nouns. There can be arbitrarily many of them:

(45) Day after day after day went by, but I never found the courage to talk to her.

So rather than an N-P-N pattern Bragmann suggests the pattern in (46), where '+' stands for at least one repetition of a sequence.

(46) N (P N)+

Now, such patterns would be really difficult to model in selection-based approaches, since one would have to assume that an empty head or a noun selects for an arbitrary number of pairs of the same preposition and noun or nominal phrase. Of course one could assume that P and N form some sort of constituent, but still one would have to make sure that the right preposition is used and the noun or nominal projection has the right phonology. Another possibility would be to assume that the second N in N-P-N can be an N-P-N and thereby allow recursion in the pattern. But if one follows this approach it is getting really difficult to check the constraint that the involved Ns should have the same or at least similar phonologies.

One way out of these problems would of course be to assume that there are special combinatorial mechanisms that assign a new category to one or several elements. This would basically be an unheaded phrase structure rule and this is what Tesnière suggested: transfer rules (see Section 11.6.2.2). But this is of course an extension of pure Dependency Grammar towards a mixed model.

See Section 21.10 for the discussion of further cases that are probably problematic for purely selection-based grammars.

Exercises

Provide the dependency graphs for the following three sentences:

(47) a. Ich habe einen Mann getroffen, der blonde Haare hat.
I have a man met who blond hair has
'I met a man who has blond hair.'

b. Einen Mann getroffen, der blonde Haare hat, habe ich noch nie.
a man met who blond hair has have I yet never
'I never met a man who has blond hair.'

c. Dass er morgen kommen wird, freut uns.
that he tomorrow come will pleases us
'That he will come tomorrow pleases us.'

You may use non-projective dependencies. For the analysis of relative clauses authors usually assume an abstract entity that functions as a dependent of the modified noun and as a head of the verb in the relative clause.

Further reading

In the section on further reading in Chapter 3, I referred to the book called *Syntaktische Analyseperspektiven* 'Syntactic perspectives on analyses'. The chapters in this book have been written by proponents of various theories and all analyze the same newspaper article. The book also contains a chapter by Engel (2014), assuming his version of Dependency Grammar, namely *Dependent Verb Grammar*.

Ágel, Eichinger, Eroms, Hellwig, Heringer & Lobin (2003, 2006) published a handbook on dependency and valence that discusses all aspects related to Dependency Grammar in any imaginable way. Many of the papers have been cited in this chapter. Papers comparing Dependency Grammar with other theories are especially relevant in the context of this book: Lobin (2003) compares Dependency Grammar and Categorial Grammar, Oliva (2003) deals with the representation of valence and dependency in HPSG and Bangalore, Joshi & Rambow (2003) describe how valence and dependency are covered in TAG. Hellwig (2006) compares rule-based grammars with Dependency Grammars with special consideration of parsing by computer programs.

Osborne & Groß (2012) compare Dependency Grammar with Construction Grammar and Osborne, Putnam & Groß (2011) argue that certain variants of Minimalism are in fact reinventions of dependency-based analyses.

The original work on Dependency Grammar by Tesnière (1959) is also available in parts in German (Tesnière 1980) and in full in English (Tesnière 2015).