19 Empty elements

This chapter deals with empty elements, I first discuss the general attitude of various research traditions towards empty elements and then show how they can be eliminated from grammars (Section 19.2). Section 19.3 discusses empty elements that have been suggested in order to facilitate semantic interpretation. Section 19.4 discusses possible motivation for empty elements with a special focus on cross-linguistic comparison and the final Section 19.5 shows that certain accounts with transformations, lexical rules, and empty elements can be translated into each other.

19.1 Views on empty elements

One point that is particularly controversial among proponents of the theories discussed in this book is the question of whether one should assume empty elements or not. The discussion of empty elements is quite old: there was already some investigation in 1961 with reference to phrase structure grammars (Bar-Hillel, Perles & Shamir 1961). The discussion of the status of empty elements has carried on ever since (see Löbner 1986; Wunderlich 1987, 1989; von Stechow 1989; Haider 1997a; Sag 2000; Bouma, Malouf & Sag 2001a; Levine & Hukari 2006; Müller 2014c; Arnold & Spencer 2015, for example). There are sometimes empirical differences between analyses that assume empty elements and those that do not (Arnold & Spencer 2015), but often this is not the case. Since empty elements often feature prominently in the argumentation for or against particular theories, I will discuss how they have been used in somewhat more detail here.

In GB theory, empty elements were assumed for traces of movement (verb movement and fronting of phrases) as well as for deleted elements in elliptical constructions. Starting with the analysis of Larson (1988), more and more empty heads have been introduced to ensure uniformity of structures and certain semantic interpretations (binding and scope, see Section 4.1.4 on little *v*). Other examples of an empty element that was introduced in order to maintain particular generalizations are the empty expletives of Coopmans (1989: 734) and Postal (2004: Chapter 1). These fill the subject position in inversion structures in English, where the position preceding the verb is occupied by a PP and not by an overt subject NP. Similarly, Grewendorf (1993: 1311) assumes that the subject position in impersonal passives and passives without subject movement is in fact occupied by an empty expletive. Also, see Newmeyer (2005: 91) and Lohnstein (2014: 180) for this assumption with regard to the passive in German. Sternefeld (2006: Section II.3.3.3) assumes that there is an empty expletive subject in impersonal passives and subjectless sentences such as (1).

(1) a. Mir graut.
me.DAT scares
'I am scared.'
b. Mich dürstet.
me.ACC is.thirsty
'I am thirsty.'

On page 168, we discussed Stabler's proposal for the analysis of sentences with intransitive verbs. Since, following Chomsky (2008: 146), the element that first merges with a head is the complement, intransitive verbs pose a problem for the theory. This problem is solved by Stabler by assuming that intransitive verbs are combined with an empty object (Veenstra 1998: 61, 124). Since these silent elements do not contribute to the meaning of an expression, we are also dealing with empty expletive pronouns.

In other theories, there are researchers that reject empty elements as well as those who assume them. In Categorial Grammar, Steedman suggests an analysis of nonlocal dependencies that does without empty elements (see Section 8.5), but as Pollard (1988) has shown, Steedman's analysis requires various kinds of type raising for NPs or a correspondingly high number of complex lexical items for relative pronouns (see Section 8.5.3). On the other hand, König (1999) uses traces. In GPSG, there is the traceless analysis of extraction by Uszkoreit (1987: 76–77) that we discussed in Section 5.4, but there is also the analysis of Gazdar, Klein, Pullum & Sag (1985: 143) that uses traces. In LFG, there are both analyses with traces (Bresnan 2001: 67) and those without (see Section 7.3 and Section 7.5). Many of the phrasal analyses in HPSG are born out of the wish to avoid empty elements (see Section 21.10). An example for this is the relative clause analysis by Sag (1997) that replaces the empty relativizer in Pollard & Sag (1994) with a corresponding phrasal rule. On the other hand we have Bender (2000) and Sag, Wasow & Bender (2003: 464), who assume a silent copula. Another attempt to eliminate empty elements from HPSG was to handle long-distance dependencies not by traces but rather in the lexicon (Bouma, Malouf & Sag 2001a). As Levine & Hukari (2006) could show, however, theories of extraction that introduce long-distance dependencies lexically have problems with the semantic interpretation of coordinate structures. For a suggestion of how to solve these problems, see Chaves (2009). There are many TAG analyses without silent elements in the lexicon (see Section 12.5 and Kroch (1987), for example), however there are variants of TAG such as that of Kallmeyer (2005: 194), where a trace is assumed for the reordering of constituents in sentences with a verbal complex. Rambow (1994: 10–11) assumes an empty head in every verb phrase (see Section 12.6.2 on V-TAG).[1] In Dependency Grammar, Mel'čuk (1988: 303; 2003: 219), Starosta (1988: 253), Eroms (2000: 471–472), Hudson (2007: Section 3.7; 2010b: 166) and Engel (2014) assume empty elements for determiners, nouns, ellipsis, imperatives, controlled infinitives, and for coordinate structures, but Groß & Osborne (2009: 73) reject empty elements (with

[1] Note that empty elements in TAG are slightly different from empty elements in other theories. In TAG the empty elements are usually part of elementary trees, that is, they are not lexical items that are combined with other material.

the exception of ellipsis, Osborne 2016).

No empty elements are assumed in Construction Grammar (Michaelis & Ruppenhofer 2001: 49–50; Goldberg 2003a: 219; Goldberg 2006: 10), the related Simpler Syntax (Culicover & Jackendoff 2005) as well as in Cognitive Grammar.[2] The argumentation against empty elements runs along the following lines:

1. There is no evidence for invisible objects.
2. There is no innate linguistic knowledge.
3. Therefore, knowledge about empty elements cannot be learned, which is why they cannot be assumed as part of our grammar.

This begs the question of whether all the premises on which the conclusion is based actually hold. If we consider an elliptical construction such as (2), then it is clear that a noun has been omitted:

(2) Ich nehme den roten Ball und du den blauen.
I take the.ACC red.ACC ball and you the.ACC blue.ACC
'I'll take the red ball and you take the blue one.'

Despite there being no noun in *den blauen* 'the blue', this group of words behaves both syntactically and semantically just like a noun phrase. (2) is of course not necessarily evidence for there being empty elements, because one could simply say that *den blauen* is a noun phrase consisting only of an article and an adjective (Wunderlich 1987).

Similar to the fact that it is understood that a noun is missing in (2), speakers of English know that something is missing after *like*:

(3) Bagels, I like.

Every theory of grammar has to somehow account for these facts. It must be represented in some way that *like* in (3) behaves just like a verb phrase that is missing something. One possibility is to use traces. Bar-Hillel, Perles & Shamir (1961: 153, Lemma 4.1) have shown that it is possible to turn phrase structure grammars with empty elements into those without any. In many cases, the same techniques can be applied to the theories presented here and we will therefore discuss the point in more detail in the following section.

19.2 Eliminating empty elements from grammars

It is possible to turn a grammar with empty elements (also called *epsilon*) into a grammar without these by removing all categories that can be rewritten by an epsilon in every rule that uses such categories and then add the respective rules without the empty elements to the grammar. The following example has an epsilon rule for np. One therefore has to replace all rules containing the symbol np with new rules without this np symbol. (5) shows the result of this conversion of the grammar in (4):

[2] However, Fillmore (1988: 51) did not rule them out.

(4) $\overline{\text{v}} \rightarrow$ np, v
$\overline{\text{v}} \rightarrow$ np, pp, v
np $\rightarrow \epsilon$

(5) $\overline{\text{v}} \rightarrow$ np, v
$\overline{\text{v}} \rightarrow$ v
$\overline{\text{v}} \rightarrow$ np, pp, v
$\overline{\text{v}} \rightarrow$ pp, v

This can also lead to cases where all elements on the right-hand side of a rule are removed. Thus, what one has done is actually create a new empty category and then one has to apply the respective replacement processes again. We will see an example of this in a moment. Looking at the pair of grammars in (4)–(5), it is clear that the number of rules has increased in (5) compared to (4) despite the grammars licensing the same sequences of symbols. The fact that an NP argument can be omitted is not expressed directly in (5) but instead is implicitly contained in two rules.

If one applies this procedure to the HPSG grammar in Chapter 9, then the trace does not have a specific category such as NP. The trace simply has to be compatible with a non-head daughter. As the examples in (6) show, adjuncts, arguments and parts of verbal complexes can be extracted.

(6) a. Er$_i$ liest t$_i$ die Berichte.
he reads the reports
b. Oft$_i$ liest er die Berichte t$_i$ nicht.
often reads he the reports not
'Often, he does not read the reports.'
c. Lesen$_i$ wird er die Berichte t$_i$ müssen.
read will he the reports must
'He will have to read the reports.'

The relevant elements are combined with their head in a specific schema (Head-Argument Schema, Head-Adjunct Schema, Predicate Complex Schema). See Chapter 9 for the first two schemata; the Predicate Complex Schema is motivated in detail in Müller (2002a: Chapter 2; 2007b: Chapter 15). If one wishes to do without traces, then one needs further additional schemata for the fronting of adjuncts, of arguments and of parts of predicate complexes. The combination of a head with a trace is given in Figure 19.1 on the facing page. The trace-less analysis is shown in Figure 19.2 on the next page. In Figure 19.1, the element in the SUBCAT list of *kennen* is identified with the SYNSEM value of the trace 4. The lexical entry of the trace prescribes that the LOCAL value of the trace should be identical to the element in the INHER|SLASH list.

The Non-Local Feature Principle (page 295) ensures that the SLASH information is present on the mother node. Since an argument position gets saturated in Head-Argument structures, the accusative object is no longer contained in the SUBCAT list of the mother node.

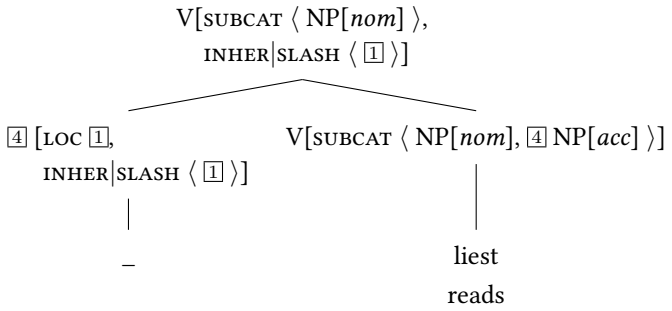

Figure 19.1: Introduction of information about long-distance dependencies with a trace

Figure 19.2 shows the parallel trace-less structure. The effect that one gets by combin-

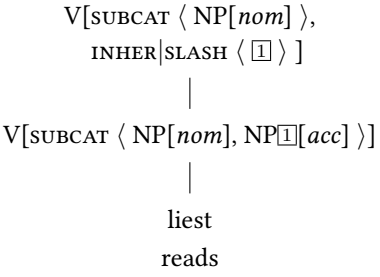

Figure 19.2: Introduction of information about long-distance dependencies using a unary projection

ing a trace in argument position in Head-Argument structures is represented directly on the mother node in Figure 19.2: the LOCAL value of the accusative object was identified with the element in INHER|SLASH on the mother node and the accusative object does not occur in the valence list any more.

The grammar presented in Chapter 9 contains another empty element: a verb trace. This would then also have to be eliminated.

(7) a. Er$_i$ liest$_j$ t$_i$ die Berichte t$_j$.
he reads the reports

b. Oft$_i$ liest$_j$ er die Berichte t$_i$ nicht t$_j$.
often reads he the reports not
'Often, he does not read the reports.'

c. Lesen$_i$ wird$_j$ er die Berichte t$_i$ müssen t$_j$.
read will he the reports must
'He will have to read the reports.'

Figure 19.3 shows the combination of a verb trace with an accusative object. The verb

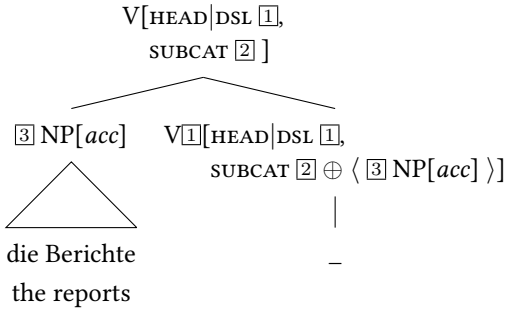

Figure 19.3: Analysis of verb position with verb trace

trace is specified such that the DSL value is identical to the LOCAL value of the trace (see p. 291). Since DSL is a head feature, the corresponding value is also present on the mother node. Figure 19.4 shows the structures that we get by omitting the empty node. This structure may look odd at first sight since a noun phrase is projected to a verb

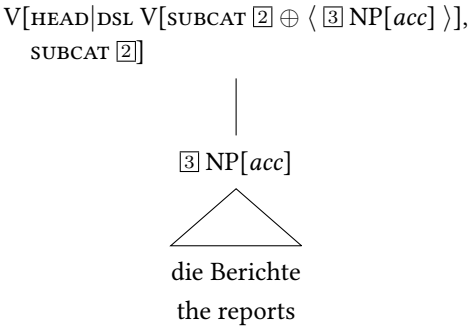

Figure 19.4: Analysis of verb position using a unary projection

(see page 233 for similar verb-less structures in LFG). The information about the fact that a verb is missing in the structure is equally contained in this structure as in the structure with the verb trace. It is the DSL value that is decisive for the contexts in which the structure in Figure 19.4 can appear. This is identical to the value in Figure 19.3 and contains the information that a verb that requires an accusative object is missing in the structure in question. Until now, we have seen that extraction traces can be removed from the grammar by stipulating three additional rules. Similarly, three new rules are needed for the verb trace. Unfortunately, it does not stop here as the traces for extraction and head movement can also interact. For example, the NP in the tree in Figure 19.4 could be an extraction trace. Therefore, the combination of traces can result in more empty

elements that then also have to be eliminated. Since we have three schemata, we will have three new empty elements if we combine the non-head daughter with an extraction trace and the head daughter with a verb trace. (8) shows these cases:

(8) a. Er$_i$ [schläft$_j$ t$_i$ t$_j$]. (Extraction trace (argument) + verb trace)
he sleeps
'He is sleeping.'

b. Jetzt$_i$ [schlaf$_j$ t$_i$ t$_j$]! (Extraction trace (adjunct) + verb trace)
now sleep
'Go to sleep now!'

c. Geschlafen$_i$ [wird$_j$ t$_i$ t$_j$]! (Extraction trace (complex) + verb trace)
slept is
'Now is time to sleep!'

These three new traces can occur as non-head daughters in the Head-Argument Schema and thus one would require three new schemata for Head-Argument structures. Using these schemata, it then becomes possible to analyze the sentences in (8).

Six further schemata are required for the examples in (9) and (10) since the three new traces can each occur as heads in Head-Argument structures (9) and Head-Adjunct structures (10):

(9) a. Den Aufsatz$_i$ liest$_j$ [er t$_i$ t$_j$].
the essay reads he
'He is reading the essay.'

b. Oft$_i$ liest$_j$ er [ihn t$_i$ t$_j$].
often reads he it
'He often reads it.'

c. Lesen$_i$ wird$_j$ er [ihn t$_i$ t$_j$].
read will he it
'He will read it.'

(10) a. Den Aufsatz$_i$ liest$_j$ er [nicht t$_i$ t$_j$].
the essay reads he not
'He isn't reading the essay.'

b. Oft$_i$ liest$_j$ er ihn [nicht t$_i$ t$_j$].
often reads he it not
'He often doesn't read it'

c. Lesen$_i$ wird$_j$ er ihn [nicht t$_i$ t$_j$].
reads will he it not
'He won't read it.'

Eliminating two empty elements therefore comes at the price of twelve new rules. These rules are not particularly transparent and it is not immediately obvious why the mother node describes a linguistic object that follows general grammatical laws. For example, there are no heads in the structures following the pattern in Figure 19.4. Since there is no empirical difference between the theoretical variant with twelve additional schemata and the variant with two empty elements, one should prefer the theory that makes fewer assumptions (Occam's Razor) and that is the theory with two empty elements.

One might think that the problem discussed here is just a problem specific to HPSG not shared by trace-less analyses such as the LFG approach that was discussed in Section 7.5. If we take a closer look at the rule proposed by Dalrymple (2006: Section 2.2), we see that the situation in LFG grammars is entirely parallel. The brackets around the category symbols mark their optionality. The asterisk following the PP means that any number of PPs (zero or more) can occur in this position.

(11) $V' \rightarrow (V)\ (NP)\ PP^*$

This means that (11) is a shorthand for rules such as those in (12):

(12) a. $V' \rightarrow V$
b. $V' \rightarrow V\ NP$
c. $V' \rightarrow V\ NP\ PP$
d. $V' \rightarrow V\ NP\ PP\ PP$
e. ...
f. $V' \rightarrow NP$
g. $V' \rightarrow NP\ PP$
h. $V' \rightarrow NP\ PP\ PP$
i. ...

Since all the elements on the right-hand side of the rule are optional, (11) also stands for (13):

(13) $V' \rightarrow \epsilon$

Thus, one does in fact have an empty element in the grammar although the empty element is not explicitly listed in the lexicon. This follows from the optionality of all elements on the right-hand side of a rule. The rule in (12f) corresponds to the schema licensed by the structure in Figure 19.4. In the licensed LFG structure, there is also no head present. Furthermore, one has a large number of rules that correspond to exactly the schemata that we get when we eliminate empty elements from an HPSG grammar. This fact is, however, hidden in the representational format of the LFG rules. The rule schemata of LFG allow for handy abbreviations of sometimes huge sets of rules (even infinite sets when using '*').

Pollard (1988) has shown that Steedman's trace-less analysis of long-distance dependencies is not without its problems. As discussed in Section 8.5.3, a vast number of recategorization rules or lexical entries for relative pronouns are required.

19.3 Empty elements and semantic interpretation

In this section, I discuss an analysis that assumes empty elements in order to allow for different readings of particular sentences. I then show how one can use so-called underspecification approaches to do without empty elements.

Sentences such as (14) are interesting since they have multiple readings (see Dowty 1979: Section 5.6) and it is not obvious how these can be derived.

(14) dass Max alle Fenster wieder öffnete
that Max all windows again opened
'that Max opened all the windows again'

There is a difference between a repetitive and a restitutive reading: for the repetitive reading of (14), Max has to have opened every window at least once before, whereas the restitutive reading only requires that all windows were open at some point, that is, they could have been opened by someone else.

These different readings are explained by decomposing the predicate *open′* into at least two sub-predicates. Egg (1999) suggests the decomposition into CAUSE and *open′*:

(15) CAUSE(x, *open′*(y))

This means that there is a CAUSE operator that has scope over the relation *open′*. Using this kind of decomposition, it is possible to capture the varying scope of *wieder* 'again': in one of the readings, *wieder* scopes over CAUSE and it scopes over *open′* but below CAUSE in the other. If we assume that *öffnen* has the meaning in (15), then we still have to explain how the adverb can modify elements of a word's meaning, that is, how *wieder* 'again' can refer to *open′*. Von Stechow (1996: 93) developed the analysis in Figure 19.5 on the following page. AgrS and AgrO are functional heads proposed for subject and object agreement in languages like Basque and have been adopted for German (see Section 4.6). Noun phrases have to be moved from the VoiceP into the specifier position of the AgrS and AgrO heads in order to receive case. T stands for Tense and corresponds to Infl in the GB theory (see Section 3.1.5 and Section 4.1.5). What is important is that there is the Voice head and the separate representation of *offen* 'open' as the head of its own phrase. In the figure, everything below Voice′ corresponds to the verb *öffnen*. By assuming a separate Voice head that contributes causative meaning, it becomes possible to derive both readings in syntax: in the reading with narrow scope of *wieder* 'again', the adverb is adjoined to the XP and has scope over open(x). In the reading with wide scope, the adverb attaches to VoiceP or some higher phrase and therefore has scope over CAUSE(BECOME(open(x))).

Jäger & Blutner (2003) point out that this analysis predicts that sentences such as (16) only have the repetitive reading, that is, the reading where *wieder* 'again' has scope over CAUSE.

(16) dass Max wieder alle Fenster öffnete
that Max again all windows opened

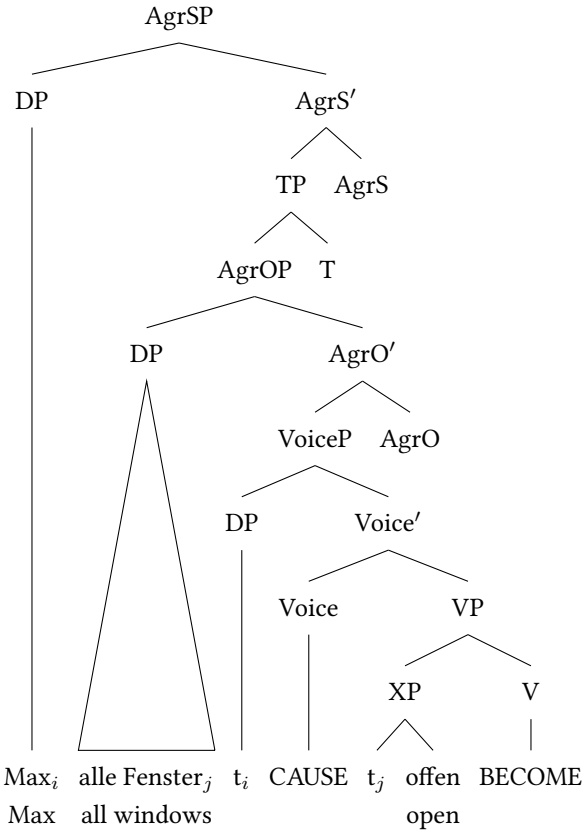

Figure 19.5: Decomposition in syntactic structures

This is because *wieder* precedes *alle Fenster* and therefore all heads that are inside VoiceP. Thus, *wieder* can only be combined with AgrOP or higher phrases and therefore has (too) wide scope. (16) does permit a restitutive reading, however: all windows were open at an earlier point in time and Max reestablishes this state.

Egg (1999) develops an analysis for these *wieder* cases using Constraint Language for Lambda-Structures (CLLS). CLLS is an underspecification formalism, that is, no logical formula are given but instead expressions that describe logical formulae. Using these kind of expressions, it is possible to leave scope relations underspecified. I have already mentioned Minimal Recursion Semantics (MRS) (Copestake, Flickinger, Pollard & Sag 2005) in several chapters of this book. As well as CLLS, MRS together with Underspecified Discourse Representation Theory (Reyle 1993; Frank & Reyle 1995) and Hole Semantics (Bos 1996; Blackburn & Bos 2005) all belong to the class of underspecification formalisms. See Baldridge & Kruijff (2002) for an underspecification analysis in Catego-

rial Grammar and Nerbonne (1993) for an early underspecification analysis in HPSG. In the following, I will reproduce Egg's analysis in an MRS-like notation.

Before we turn to (14) and (16), let us consider the simpler sentence in (17):

(17) dass Max alle Fenster öffnete
that Max all windows opened
'that Max opened all the windows'

This sentence can mean that in a particular situation, it is true of all windows that Max opened them. A less readily accessible reading is the one in which Max causes all of the windows to be open. It is possible to force this reading if one rules out the first reading through contextual information (Egg 1999):

(18) Erst war nur die Hälfte der Fenster im Bus auf, aber dann öffnete Max
first was only the half of.the windows in.the bus open but then opened Max
alle Fenster.
all windows
'At first, only half of the windows in the bus were open, but then Max opened all of the windows.'

Both readings under discussion here differ with regard to the scope of the universal quantifier. The reading where Max opens all the windows himself corresponds to wide scope in (19a). The reading where some windows could have already been open corresponds to (19b):

(19) a. $\forall$ x *window′*(x) $\rightarrow$ CAUSE(*max′*, *open′*(x))
b. CAUSE(*max′*, $\forall$ x *window′*(x) $\rightarrow$ *open′*(x))

Using underspecification, both of these readings can be represented in one dominance graph such as the one given in Figure 19.6 on the next page. Each relation in Figure 19.6 has a name that one can use to refer to the relation or "grasp" it. These names are referred to as *handle*. The dominance graph states that $h0$ dominates both $h1$ and $h6$ and that $h2$ dominates $h4$, $h3$ dominates $h5$, and $h7$ dominates $h5$. The exact scopal relations are underspecified: the universal quantifier can have scope over CAUSE or CAUSE can have scope over the universal quantifier. Figures 19.7 and 19.8 show the variants of the graph with resolved scope. The underspecified graph in Figure 19.6 does not say anything about the relation between h3 and h6. The only thing it says is that h3 somehow has to dominate h5. In Figure 19.7 every (h3) dominates CAUSE (h6) and CAUSE dominates open (h5). So, *every′* dominates *open′* indirectly. In Figure 19.8, CAUSE dominates *every′* and *every′* dominates *open′*. Again the constraints of Figure 19.6 are fulfilled, but h7 dominates h5 only indirectly.

The fact that the quantifier dominates $h4$ is determined by the lexical entry of the quantifier. The fact that the quantifier dominates $h5$ does not have to be made explicit in the analysis since the quantifier binds a variable in the relation belonging to $h5$, namely x. The dominance relation between $h7$ and $h5$ is always determined in the lexicon since CAUSE and *open′* both belong to the semantic contribution of a single lexical entry.

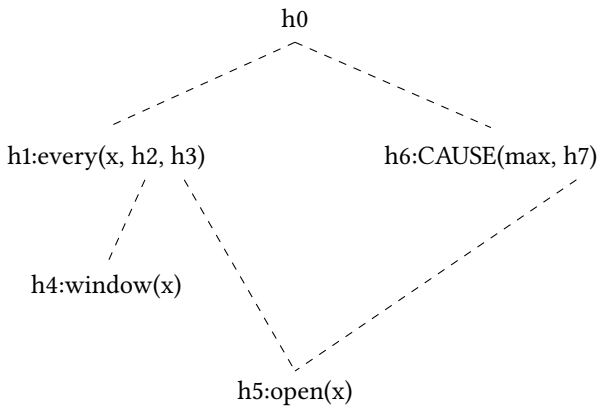

Figure 19.6: Dominance graph for *Max alle Fenster öffnete*

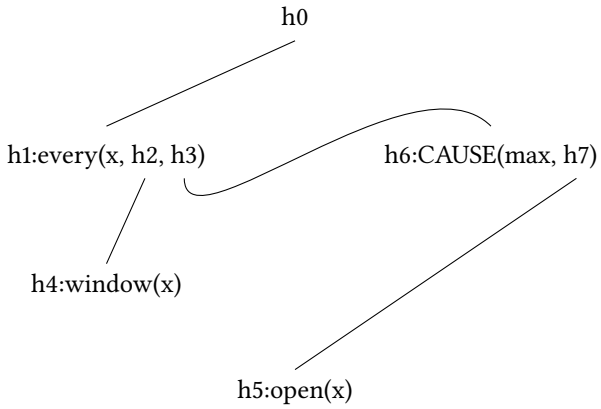

Figure 19.7: Dominance graph for the reading ∀ x window(x) → CAUSE(max,open(x)).

The exact syntactic theory that one adopts for this analysis is, in the end, not of great importance. I have chosen HPSG here. As Figure 19.9 on the facing page shows, the analysis of *alle Fenster öffnet* contains a simple structure with a verb and an object. This structure does not differ from the one that would be assumed for *alle Kinder kennt* 'all children know', involving the semantically simplex verb *kennen* 'to know'. The only difference comes from the meaning of the individual words involved. As shown in Section 9.1.6, relations between individual words are passed on upwards. The same happens with scopal restrictions. These are also represented in lists. HCONS stands for *handle constraints*. $=_q$ in h0 $=_q$ h6 stand for the equality *modulo* quantifier scope.

Egg lists the following readings for the sentence in (16) – repeated here as (20):

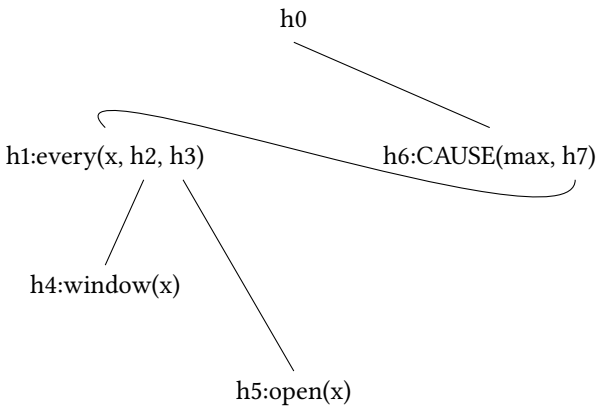

Figure 19.8: Graph for te reading CAUSE(max, ∀ x window(x) → open(x)).

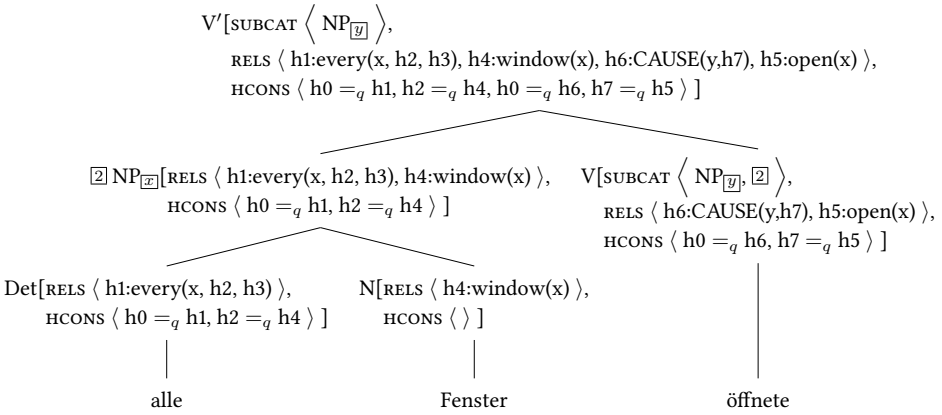

Figure 19.9: MRS analysis of *alle Fenster öffnete*

(20) dass Max wieder alle Fenster öffnete
that Max again all windows opened
'that Max opened all the windows again'

1. Max opened every window and he had already done that at least once for each window (*again′*(∀(CAUSE(open))); repetitive)
2. Max caused every window to be open and he had done that at least once before (*again′*(CAUSE(∀(open))); repetitive)
3. At some earlier point in time, all windows were simultaneously open and Max re-established this state (CAUSE(*again′*(∀(open))); restitutive)

These readings correspond to the dominance graph in Figure 19.10. Figure 19.11 shows

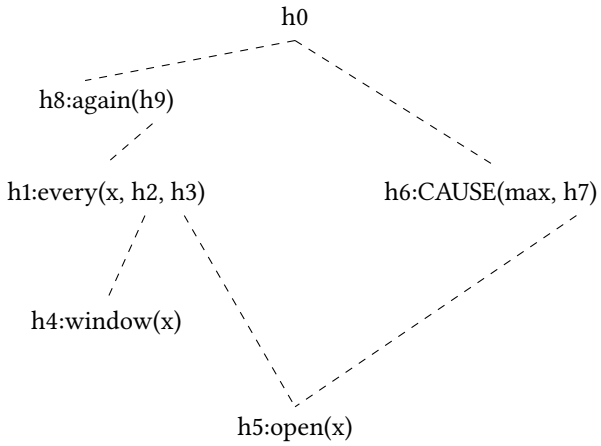

Figure 19.10: Dominance graph for *Max wieder alle Fenster öffnete* 'that Max opened all the windows again'

the graph for (14) – repeated here as (21):

(21) dass Max alle Fenster wieder öffnete
that Max all windows again opened

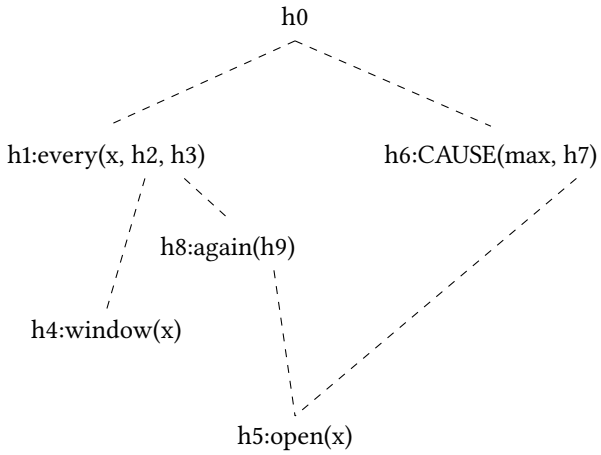

Figure 19.11: Dominance graph for *Max alle Fenster wieder öffnete* 'that Max opened all the windows again'

To derive these dominance graphs from the ones without *wieder* 'again', all one has to do is add the expression h8:again(h9) and the dominance requirements that demand

that $h9$ dominates quantifiers occurring to the right of *wieder* and that it is dominated by quantifiers to the left of *wieder*.

It is therefore unproblematic to derive the relevant readings for modification by *wieder* without empty elements for CAUSE and BECOME. The meaning of the word *öffnen* is decomposed in a similar way but the decomposed meaning is assigned to a single element, the verb. By underspecification of the scopal relations in the lexicon, the relevant readings can then be derived.

19.4 Evidence for empty elements

As previously discussed, grammarians agree that both linguists and speakers notice when there is a constituent missing from a string of words. For cases where it can be shown that analyses with or without traces are indistinguishable empirically, then one can assume empty elements. Nevertheless, the learnability argument put forward by Construction Grammarians has some validity: if one assumes that there is no or little innate linguistic knowledge, then it is not possible to motivate empty elements with data from other languages. This means that just because Basque shows object agreement, this does not mean that one can assume an empty head for object agreement (AgrO) in a grammar of German as for instance von Stechow (1996) and Meinunger (2000) do. Since there is no object agreement in German, there would be no way for the child to learn the fact that there is an AgrO head. Knowledge about AgrO must therefore be innate. Since the assumption of innate linguistic knowledge is controversial (see Chapter 13), any theory that uses cross-linguistic data to motivate the use of empty elements is on shaky ground.

Cross-linguistic considerations can only be drawn upon if there are no empirical differences between multiple alternative analyses compatible and motivated by the language under consideration. In this case, one should follow Occam's Razor and choose the analysis which is compatible with analyses of other languages (see Müller 2015a and Chapter 22.2).

19.5 Transformations, lexical rules, and empty elements

In the discussion of the passive in the framework of TAG, it became clear that lexical rules correspond to particular transformations, namely those which have some relation to a lexical item (lexically governed transformations, Dowty 1978; for the discussion of transformations and lexical rules, see Bresnan (1978) and Bresnan & Kaplan (1982)). In the respective variants of TAG, lexical rules establish a relation between a lexical item for an active tree with a lexical item of a passive tree. Both the active and passive tree can be extended by adjunction.

In theories such as Categorial Grammar, the situation is similar: since the direction in which a functor expects to find its argument is fixed for languages such as English, the lexical item stands for an entire tree. Only the attachment of adjuncts is not yet specified

in lexical items. The positions in the tree where the adjuncts can occur depend on the properties of the adjuncts. In Section 8.4, we have seen suggestions for treatments of languages with free constituent order. If the direction of combination is not fixed in the lexicon, then the lexical item can occur in a number of trees. If we compare lexical rules that can be applied to these kind of lexical items with transformations, we see that lexical rules create relations between different sets of trees.

In HPSG analyses, this works in a similar way: lexical rules relate lexical items with differing valence properties to each other. In HPSG grammars of English, there is normally a schema that licenses a VP containing the verb and all its complements as well as a schema that connects the subject to the VP (Pollard & Sag 1994: 39). In the lexical items for finite verbs, it is already determined what the tree will look like in the end. As in Categorial Grammar, adjuncts in HPSG can be combined with various intermediate projections. Depending on the dominance schemata used in a particular grammar, the lexical item will determine the constituent structure in which it can occur or allow for multiple structures. In the grammar of German proposed in Chapter 9, it is possible to analyze six different sequences with a lexical item for a ditransitive verb, that is, the lexical item can – putting adjuncts aside – occur in six different structures with verb-final order. Two sequences can be analyzed with the passive lexical item, which only has two arguments. As in Categorial Grammar, sets of licensed structures are related to other sets of licensed structures. In HPSG theorizing and also in Construction Grammar, there have been attempts to replace lexical rules with other mechanisms since their "status is dubious and their interaction with other analyses is controversial" (Bouma, Malouf & Sag 2001a: 19). Bouma et al. (2001a) propose an analysis for extraction that, rather than connecting lexical items with differing valence lists, establishes a relation between a subset of a particular list in a lexical item and another list in the same lexical item. The results of the two alternative analyses are shown in (22) and (23), respectively:

(22) a. $\begin{bmatrix} \textsc{subcat} & \langle\ \text{NP}[\textit{nom}], \text{NP}[\textit{acc}]\ \rangle \\ \textsc{slash} & \langle\rangle \end{bmatrix}$

b. $\begin{bmatrix} \textsc{subcat} & \langle\ \text{NP}[\textit{nom}]\ \rangle \\ \textsc{slash} & \langle\ \text{NP}[\textit{acc}]\ \rangle \end{bmatrix}$

In (22), (22a) is the basic entry and (22b) is related to (22a) via a lexical rule. The alternative analysis would only involve specifying the appropriate value of the ARG-ST feature[3] and the SUBCAT and SLASH value is then derived from the ARG-ST value using the relevant constraints. (23) shows two of the licensed lexical items.

(23) a. $\begin{bmatrix} \textsc{arg-st} & \langle\ \text{NP}[\textit{nom}], \text{NP}[\textit{acc}]\ \rangle \\ \textsc{subcat} & \langle\ \text{NP}[\textit{nom}], \text{NP}[\textit{acc}]\ \rangle \\ \textsc{slash} & \langle\rangle \end{bmatrix}$

[3] ARG-ST stands for *Argument Structure*. The value of ARG-ST is a list containing all the arguments of a head. For more on ARG-ST, see Section 9.6.1.

b. $\begin{bmatrix} \textsc{arg-st} & \langle\, \text{NP}[\mathit{nom}], \text{NP}[\mathit{acc}] \,\rangle \\ \textsc{subcat} & \langle\, \text{NP}[\mathit{nom}] \,\rangle \\ \textsc{slash} & \langle\, \text{NP}[\mathit{acc}] \,\rangle \end{bmatrix}$

If we want to eliminate lexical rules entirely in this way, then we would require an additional feature for each change.[4] Since there are many interacting valence-changing processes, things only work out with the stipulation of a large number of auxiliary features. The consequences of assuming such analyses have been discussed in detail in Müller (2007b: Section 7.5.2.2). The problems that arise are parallel for inheritance-based approaches for argument structure-changing processes: they also require auxiliary features since it is not possible to model embedding and multiple changes of valence information with inheritance. See Section 10.2.

Furthermore, the claim that the status of lexical rules is dubious must be rejected: there are worked-out formalizations of lexical rules (Meurers 2001; Copestake & Briscoe 1992; Lascarides & Copestake 1999) and their interaction with other analyses is not controversial. Most HPSG implementations make use of lexical rules and the interaction of a number of rules and constraints can be easily verified by experiments with implemented fragments.

Jackendoff (1975) presents two possible conceptions of lexical rules: in one variant, the lexicon contains all words in a given language and there are just redundancy rules saying something about how certain properties of lexical entries behave with regard to properties of other lexical entries. For example, *les-* 'read-' and *lesbar* 'readable' would both have equal status in the lexicon. In the other way of thinking of lexical rules, there are a few basic lexical entries and the others are derived from these using lexical rules. The stem *les-* 'read-' would be the basic entry and *lesbar* would be derived from it. In HPSG, the second of the two variants is more often assumed. This is equivalent to the assumption of unary rules. In Figure 9.8 on page 287, this has been shown accordingly: the verb *kennt* 'knows' is mapped by a lexical rule to a verb that selects the projection of an empty verbal head. With this conception of lexical rules, it is possible to remove lexical rules from the grammar by assuming binary-branching structures with an empty head rather than unary rules. For example, in HPSG analyses of resultative constructions such as (24), lexical rules have been proposed (Verspoor 1997; Wechsler 1997; Wechsler & Noh 2001; Müller 2002a: Chapter 5).

(24) [dass] Peter den Teich leer fischt
that Peter the pond empty fishes
'that Peter fishes the pond empty'

In my own analysis, a lexical rule connects a verb used intransitively to a verb that selects an accusative object and a predicate. Figure 19.12 on the next page shows the corresponding tree. If we consider what (24) means, then we notice that the fishing act

[4] Alternatively, one could assume a very complex relation that connects ARG-ST and SUBCAT. But this would then have to deliver the result of an interaction of a number of phenomena and the interaction of these phenomena would not be captured in a transparent way.

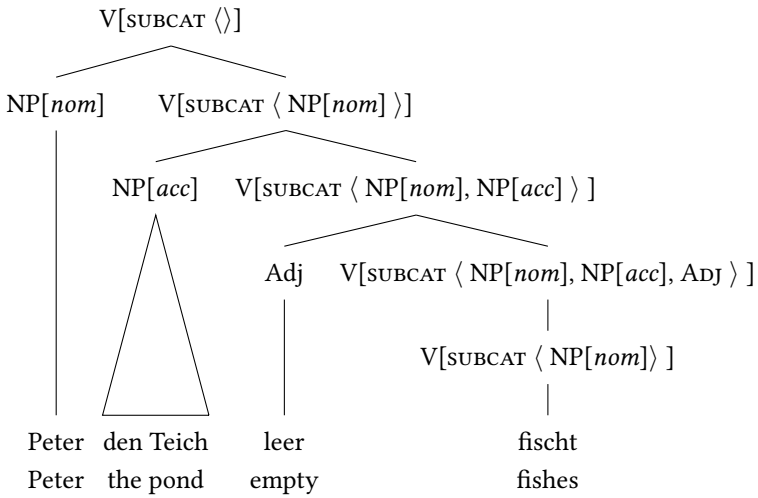

Figure 19.12: Analysis of the resultative construction with a lexical rule

causes the pond to become empty. This causation is not contained in any of the basic lexical items for the words in (24). In order for this information to be present in the semantic representation of the entire expression, it has to be added by means of a lexical rule. The lexical rule says: if a verb is used with an additional predicate and accusative object, then the entire construction has a causative meaning.

Figure 19.13 on the facing page shows how a lexical rule can be replaced by an empty head. The empty head requires the intransitive verb and additionally an adjective, an accusative object and a subject. The subject of *fischt* 'fishes' must of course be identical to the subject that is selected by the combination of *fischt* and the empty head. This is not shown in the figure. It is possible, however, to establish this identity (see Hinrichs & Nakazawa 1994). The causative semantics is contributed by the empty head in this analysis. The trick that is being implemented here is exactly what was done in Section 19.2, just in the opposite direction: in the previous section, binary-branching structures with an empty daughter were replaced by unary-branching structures. In this section, we have replaced unary-branching structures with binary-branching structures with an empty daughter.[5]

[5] Here, we are discussing lexical rules, but this transformation trick can also be applied to other unary rules. Semanticists often use such rules for type shifting. For example, a rule that turns a referential NP such as *a trickster* in (i.a) into a predicative one (i.b) (Partee 1987).

(i) a. A trickster laughs.
b. He is a trickster.

These changes can be achieved by a unary rule that is applied to an NP or with a special empty head that takes an NP as its argument. In current Minimalist approaches, empty heads are used (Ramchand 2005: 370), in Categorial Grammar and HPSG unary-branching rules are more common (Flickinger 2008: 91–92;

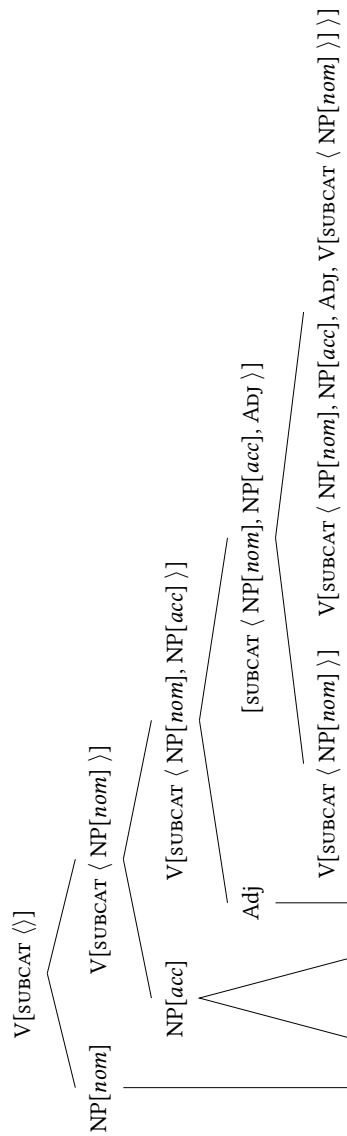

Figure 19.13: Analysis of the resultative construction with an empty head

We have therefore seen that certain transformations can be replaced by lexical rules and also that lexical rules can be replaced by empty heads. The following chapter deals with the question of whether phenomena like extraction, scrambling, and passive should be described with the same tool as in GB/Minimalism or with different tools as in LFG and HPSG.

Müller 2009c, 2012a).