Handbook of Lexical Functional Grammar

Edited by

Mary Dalrymple

Empirically Oriented Theoretical Morphology and Syntax

Empirically Oriented Theoretical Morphology and Syntax

In this series:

1. Lichte, Timm. Syntax und Valenz: Zur Modellierung kohärenter und elliptischer Strukturen mit Baumadjunktionsgrammatiken.
2. Bîlbîie, Gabriela. Grammaire des constructions elliptiques: Une étude comparative des phrases sans verbe en roumain et en français.
3. Bowern, Claire, Laurence Horn & Raffaella Zanuttini (eds.). On looking into words (and beyond): Structures, Relations, Analyses.
4. Bonami, Olivier, Gilles Boyé, Georgette Dal, Hélène Giraudo & Fiammetta Namer. The lexeme in descriptive and theoretical morphology.
5. Guzmán Naranjo, Matías. Analogical classification in formal grammar.
6. Flick, Johanna. Die Entwicklung des Definitartikels im Althochdeutschen: Eine kognitiv-linguistische Korpusuntersuchung.
7. Crysmann, Berthold & Manfred Sailer (eds.). One-to-many relations in morphology, syntax, and semantics.
8. Zinova, Yulia. Russian verbal prefixation: A frame semantic analysis.
9. Müller, Stefan, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig. Head-Driven Phrase Structure Grammar: The handbook.
10. Diewald, Gabriele & Katja Politt (eds.). Paradigms regained: Theoretical and empirical arguments for the reassessment of the notion of paradigm.

ISSN: 2366-3529

Handbook of Lexical Functional Grammar

Edited by

Mary Dalrymple

Mary Dalrymple (ed.). 2022. *Handbook of Lexical Functional Grammar* (Empirically Oriented Theoretical Morphology and Syntax). Berlin: Language Science Press.

ISBN: no digital ISBN
no print ISBNs!
ISSN: 2366-3529
no DOI
Source code available from www.github.com/langsci/312
Collaborative reading: paperhive.org/documents/remote?type=langsci&id=312

Cover and concept of design: Ulrike Harbort
Fonts: Libertinus, Arimo, DejaVu Sans Mono, Source Han Serif ZH, Source Han Serif JA
Typesetting software: XƎLATEX

Language Science Press
xHain
Grünberger Str. 16
10243 Berlin, Germany
langsci-press.org

Storage and cataloguing done by FU Berlin

Contents

Chapter 1

Negation

Oliver Bond

University of Surrey

Negation is one of the few grammatical features observed in all languages. While typically thought of as a property of predicates, it can be manifested in a wide range of structural positions associated with verbs (typically, V, I or Î or as a verbal adjunct, represented as NEG), but is also observed on other parts of speech (e.g. D/N, C, P and CONJ) and is sometimes expressed across two or more nodes within c-structure (e.g. Butt et al. 1999, Alsharif & Sadler 2009, Laczkó 2014, Bond 2016, Alruwaili & Sadler 2018).

In the most straightforward cases there is one representation of negation at f-structure, with a binary feature indicating the presence or absence of this value. However, distributional differences between superficially similar negators, and evidence from structures with multiple negative forms within a single clause, suggest that more than one feature may be necessary to account for the syntactic and semantic effects observed in negative contexts. For instance, when a negation scopes over a sub-constituent in c-structure (so-called *constituent negation* or cneg) which is part of a finite syntactic structure which is also negated (known as *eventuality negation* or eneg) two representations of negation appear to be required within the same f-structure (Przepiórkowski & Patejuk 2015). The distribution of Negative Concord Items (NCIs), Negative Polarity Items (NPIs) and case-forms licenced by negation also suggests that multiple features must also play an important role in accounting for restrictions on the occurrence of certain forms in antiveridical contexts (Sells 2000, Camilleri & Sadler 2017).

1 Introduction

No theoretical model of language is complete without a way to represent negation or the range of grammatical effects that it induces in linguistic structures.

Oliver Bond. Forthcoming. Negation. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Superficially, this is necessary because negation is one of the few grammatical categories that is uncontroversially universal in nature. Yet, as we will see, this does not mean that negation is especially uniform across languages: the cross-linguistic manifestations of negation are diverse and the structural consequences associated with the presence of negation are manifold and varied.[1]

For the purposes of the current chapter, I take negation to be the formal manifestation of a semantic operator $\neg$ that combines with an argument A to form a complex semantic expression $\neg A$. In propositional logic, negation combines with a propositional argument P to form $\neg P$. The presence of negation indicates that the conditions under which the proposition P is true are not satisfied at reference time.

Consider the proposition P given in (1):

(1) P: Eva is an experienced astronaut.

The truth conditions for the proposition P in (1) are not met if Eva is considered to an inexperienced astronaut, or if she isn't an astronaut at all. In such cases we can say that P is false, and express this using negation.[2] An important logical property of negation, is that if P is false, $\neg P$ must be true. Similarly, if $\neg P$ is true, P must be false. $\neg P$ can also be paraphrased as "it is not the case that P", as shown for (1) in (2). The ability to form this paraphrase has been proposed as a rough semantic test for what Jackendoff (1969) calls *sentential negation*.

(2) $\neg P$: It is not the case that Eva is an experienced astronaut.

Jackendoff's concept of sentential negation is associated with a wide-scope reading of negation. Negation is maximally wide-scoping when the whole proposition – including the subject – is in the scope of negation.[3] In practice, in natural languages, the subject is usually an established discourse referent outside the

[1] For instance, negation is frequently seen as an important diagnostic tool for discriminating between different lexical categories (e.g. Stassen 1997) or structures (e.g. Brown & Sells 2016), where differential behaviour under negation is used to support linguistic argumentation. At the same time, what we intuitively think of as negation is, itself, commonly subject to diagnostics, which attempt to distinguish negatives from affirmatives, or to distinguish different subtypes of the phenomena (e.g. Jespersen 1917; Klima 1964; Jackendoff 1969; de Haan 1997; Zanuttini 2001; Giannakidou 2006; Przepiórkowski & Patejuk 2015).

[2] In this chapter, I discuss only contradictory negation. See Horn (2020) for a recent discussion.

[3] In strictly semantic terms, the scope of negation describes its operational domain. It is said to be wide, rather than narrow, when other semantic operations occur *before* negation applies. Negation with propositional scope is also commonly referred to as *external negation* because the negative operation is external to the proposition.

scope of negation (Keenan 1976; Givón 1979). Consequently, the negative structures that are typically reported in grammars and general discussions of negation are examples of PREDICATE NEGATION, where negation is an evaluation of the relationship between the subject and the predicate.[4] What sentential negation proper and predicate negation share in common is that the *main predicate* is within the scope of negation, and the negative operator scopes over other predicate level operators (see Payne 1985; Acquaviva 1997, De Clercq 2020).

Some examples of clauses in which the predicate is negated can be seen in (3)-(5) from Polish, Modern Standard Arabic and Eleme (Niger Congo, Ogonoid; Nigeria). In the Polish example in (3) negation is marked with a negative particle *nie* (see Section 2.1). In (4), from Modern Standard Arabic, negation is expressed by a negative auxiliary *laysuu* (see Section 2.3). In the Eleme example in (5), negation is signalled through morphological means, and the affirmative verb form is quite distinct from the form employed in the negative (see Section 2.2).

(3) Polish (Przepiórkowski & Patejuk 2015: 324; own data)

a. Janek lubi Marię.
Janek.NOM likes Maria.ACC
'Janek likes Maria.'

b. Janek nie lubi Marii.
Janek.NOM NEG likes Maria.GEN
'Janek doesn't like Maria.'

(4) Modern Standard Arabic (Alsharif & Sadler 2009: 23; own data)

a. al-awlad-u ya-ktub-uu-n
the-boys-NOM 3M-study.IPFV-3MP-IND
The boys write/are writing.

b. al-awlad-u lays-uu ya-ktub-uu-n
the-boys-NOM NEG-3MP 3M-write.IPFV-3MP-IND
The boys do not write/are not writing.

(5) Eleme (Bond 2016: 283; own data)

a. òsáro è-dé-a òfí́
Osaro 3[SG]-eat-HAB mango
'Osaro (usually) eats mango.'

[4] cf. Jespersen's (1917) *nexal negation*, Klima's (1964) *sentence negation*, and Payne's (1985) *standard negation.*

b. òsáro è-dé~dè òfɪ́
Osaro 3[SG]-NEG~eat[HAB] mango
'Osaro doesn't (usually) eat mango.'

As well as having means to negate the main predicate of the clause, languages frequently have negators with distinct behavioural properties that do not have scope over the finite predicate and hence can be said to have low(er) negative scope (De Clercq 2020). Negators of this type are typically bundled together in descriptions as examples of *constituent negation*. The term 'constituent negation' has its origins in the work of Klima 1964, who formulated a range of now famous tests to distinguish it from negation with scope over the predicate (see Payne 1985, de Haan 1997 and De Clercq 2020 for discussion). An example of constituent negation in English can be seen in (6). Here a verbless secondary predication modifying a noun is in the scope of negation, but not the main predicate. Such negators are said to have narrow scope.

(6) Dora found a job [not far away].
(cf. Dora found a job that is not far away.)

It is common to find that negators used to negate predicates may also be used in narrow scope negation (De Clercq 2020). The following Hungarian data from Laczkó (2014: 306-7) illustrate predicate negation (7a) and narrow-scope negation over the object referent (7b). Small caps indicate focussed elements. In (7a) negation scopes over the predicate, or put another way, the truth conditions for the relationship between the predicate and its subject are not met. In (7b), narrow scope negation indicates that it is the relationship between the object referent and the rest of the assertion that is relevant.

(7) Hungarian (Laczkó 2014: 306-307)

a. Péter nem hívta fel a barátjá-t.
Peter.NOM not called up the friend.his-ACC
'Peter didn't call up his friend.'

b. Péter NEM A BARÁTJÁ-T hívta fel
Peter.NOM not the friend.his-ACC called up
'It wasn't his friend that Peter called up.'

In (8) these two strategies are combined within the same clause, providing evidence for the need to be able to simultaneously distinguish these types of negation within formal models (see Section 3 for discussion).

(8) Hungarian (Laczkó 2014: 306-7)

Péter	NEM	A	BARÁTJÁ-T	nem	hívta	fel.
Peter.NOM	not	the	friend.his-ACC	not	called	up

‘It wasn’t his friend that Peter didn’t call up.’

Cross-linguistically, narrow-scope negation is formally distinguished from wider-scoping predicate negation by a variety of means, including differences in syntax, the use of different negators or prosodic alternations, etc.

Other examples that are described as constituent negation involve negative quantifiers modifying a noun, as in (9). In such cases the negation of the predicate is achieved through a more complex process of logical implication:

(9) Dora found no [reason to worry].
(cf. Dora didn’t find a reason to worry.)

Informally, we can say of (9) that if Dora found no reason to worry, the reasons to worry equal zero, therefore Dora didn’t find any (i.e. > 0) reason to worry. Quantifiers interact with negation in a number of complex ways and the literature on this topic is extensive (see Krifka 1995; de Swart 2009; Penka 2010 amongst others). While negation and quantification have been subject to some discussion in the LFG literature (Fry 1999; Dalrymple 2001: 291-295; 309-311), I will leave aside this topic here.

While syntax and semantics often align, the scope of negation should really be considered to be a semantic phenomenon (see Penka 2016 for an overview of negation in formal semantics), and must be analysable within the semantic component of grammar in parallel to considering how this is played out in syntax and prosody. In practice, when authors talk about scope, they often treat it as a syntactic phenomenon because of differences in the syntactic domain in which the effects of negation can be observed (see Reinhart 1979; Szabolcsi 2012). Because of this, the term scope is typically also used to refer to the syntactic domain in which the effects of negation are observed. However, it is useful to untangle these two properties of negative clauses. This is – in theory – easy to do in a model like LFG because syntax and semantics are dealt with in separate, yet parallel modules of grammar. Establishing the extent to which the two are independent is one of the major goals of investigating the syntax-semantics interface.

It should be clear from this brief overview that an adequate discussion of the topic necessitates not only an exploration of the formal devices used to express negation (and the domains in which the effects of negation are observed), but also how this relates to the semantic interpretation of the utterance.

Most analyses of negation in LFG to date have focussed on the syntactic properties of negation constructions by examining the role of negation in c-structure and f-structure, most notably Sells (2000) on Swedish, Alsharif & Sadler (2009) on Arabic, Przepiórkowski & Patejuk (2015) on Polish and Camilleri & Sadler (2017) on Maltese. Despite a growing body of work in this domain (some of which is briefly outlined in Dalrymple et al. 2019: 67-69), negation has remained focussed on the syntactic properties and effects of negation. A rare exception is Dalrymple & Nikolaeva (2011) who briefly discuss the semantic contribution of negation within the context of information structure, while Bond (2016) examines issues related to the morphological expression of negation (Section 2.2).

Negation is manifested using a variety of formal devices which differ according to the extent to which this affects (i) syntactic constituency of negative clauses and (ii) the domains in which operations sensitive to negation occur. In what follows, we first look at the arguments that support possible representations of syntactic components of grammar (Section 2) before exploring the representation of negation in a component of grammar unique to LFG, namely f-structure (Section 3).

2 Representations of negation as a formative

Negation of verbal predicates can be manifested in a wide variety of ways, most commonly by (adverbial) particles (Section 2.1), changes in verbal morphology (Section 2.2) or through the use of a negative auxiliary (Section 2.3). A combination of these strategies is also widely attested (Section 4).

2.1 Negative particles

A large body of cross-linguistic work (Dahl 1979; Dryer 1989; Payne 1985; Miestamo 2005; Dryer 2013) indicates that the most common way in which the world's languages express the negation of propositions about (epistemically unmodified) dynamic events (i.e 'standard negation' (Payne 1985 Miestamo 2005) or 'clausal negation in declarative sentences' (Dryer 2013)) is through the use of a uninflecting negative particle. This is observed in at least 44% (n=502) of Dryer's 2013 sample of 1157 languages. Further languages in his sample including a particle as part of a more complex strategy consisting of multiple formatives (n=119), and others still classified as unclear with respect to whether they are particles or uninflecting negative auxiliary verbs (n=73).[5] Given their isomorphic nature,

[5] The numbers from the World Atlas of Language Structures reported here are those from Dryer (2013); those presented in the earliest editions were lower due to a programming error.

Bond (2013) takes the expression of negation through the use of particles to be a property of canonical negation.

In typological work on negation, the term particle is used as a general term for an independent word whose distribution is not better characterised in reference to a larger class of items, and includes negators described as negative adverbs. The syntactic status of negative particles (in this typological sense) has been one of considerable attention within the theoretical syntactic literature (see Pollock 1989; Haegeman 1995; Zanuttini 1997; Rowlett 1998 among others), including LFG (see Butt et al. 1999; Przepiórkowski & Patejuk 2015). This is in part motivated by the fact that the negative particle in English (and similar forms in related languages) are usually described as adverbs. While they frequently share some of the properties of adverbs in the language in which they are found, they also tend to have special syntactic characteristics that make them distinct. These characteristics, such as restrictions on their syntactic position, or the inability to be modified, make them unlike regular phrasal heads (e.g. Butt et al. 1999: 141-2). Crucially, these properties differ even among closely related languages, demonstrating that adopting the category 'particle' in broadscale typological work presents a convenient opportunity to be vague rather than explicit about the syntactic properties of any given negative formative. For instance, taking a minimalist approach, Repp 2009 argues that while both are described as adverbs in their respective descriptive traditions, German *nicht* and English *not* have different syntactic behaviour. The former is proposed to be a simple adverbial adjoining to the verb phrase (VP) while the latter is a functional head projecting a NegP. Butt et al. (1999: 141-2) conclude that *nicht* and *not* both belong to a special category NEG that distinguishes them from other adverbs, with the differences in their distribution encoded in c-structure rules.

In many Chomskian treatments of negation in English, *not* is the specifier of NegP, a separate negative projection (see Pollock 1989; Repp 2009; amongst others). Even if the validity of NegP approach seems appropriate in some analyses, the existence of such a functional head for all instances of negation would not be consistent with the lexicalist approach to syntax. Negation is commonly expressed through morphological alternations that suggest this is a considerably less useful tool for accounting for negation in languages where the category is expressed through non-concatenative morphology (Section 2.2).

This leads to us to the first problem of determining how negative particles should be represented in the X-bar theory employed to represent c-structure in most LFG work. Given that negation can be associated with almost any part of speech, and a functional projection in LFG is not required for the purposes of movement, is a NegP motivated within a declarative theory of syntax at all?

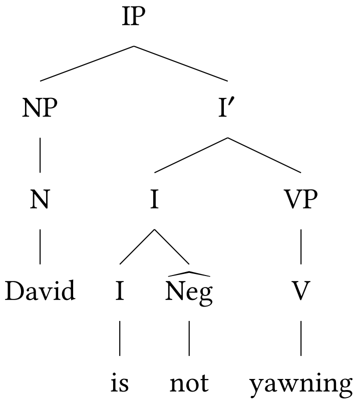
There are several possibilities with respect to dealing with this issue: first, that a node in constituent structure is required that has the properties of a regular phrasal head (e.g. AdvP), second that a special functional head is required (i.e. NegP), or third that the negative particle occupies a non-projecting phrase (in the sense of Toivonen 2003).

The first major paper dedicated to tackling negation with the LFG framework is Sells (2000), who proposes an account of negation in Swedish. Therein, he considers whether a NegP is required to account for the distribution of the negative adverb *inte*. He reviews the evidence in favour of positioning the Swedish negative adverb *inte* inside or outside the VP, concluding that neither the negation adverb *inte* nor negative quantifiers can appear within the VP. Sells observes that the unmarked position for negation is to the left of VP, though positions higher up in IP and CP are also possible. He concludes that *inte* occupies a special NEG node in c-structure, but argues against the view that a NegP is required to account for its syntactic properties.

As with Swedish *inte*, English *not* is usually described as an adverb, but they have different distributions. Since *not* must be preceded by a tensed auxiliary verb when expressing sentential negation, as in (10), Dalrymple (2001: 61) assumes that it is adjoined to the tensed verb in I, as illustrated in (11). A similar structure is proposed in Bresnan (2001a).

(10) David is not yawning.

(11) English non-projecting negative particle *not* (based on Dalrymple 2001: 61)



While brief, Dalrymple's (2001: 61) analysis captures an observation that some negative particles are non-projecting categories that are not heads of phrases, but adjoin to heads. Toivonen (2003) proposes that non-projecting categories have distinct characteristics that make then unlike regular phrases:

- They are independent words which do not project a phrase.
- They must adjoin to X^0 (i.e. at the lexical level).
- They cannot take complements or modifiers.

In (11), *Neg* is not a NegP, but a non-projecting word adjoined to I.

A similar analysis of negative particles is proposed by Alsharif & Sadler (2009) and Alsharif (2014), who examine negation in Modern Standard Arabic (MSA). MSA has three negative particles used with imperfective predicates *laa*, *lam* and *lan*. The particles differ according to the grammatical categories with which they combine. Each occurs with a verbal element as the main predicate: *laa* occurs with the indicative imperfective, *lam* with the jussive imperfective expressing negation in the past, and *lan* with the subjunctive imperfective, expressing negation in the future (Alsharif & Sadler 2009: 8). Regardless of combinatorial potential, their default syntactic distribution is the same – immediately before the auxiliary – as illustrated in (12).[6]

(12) MSA (Benmamoun 2000: 95 cited in Alsharif & Sadler 2009: 7-8)

a. t-tullaab-u laa ya-drus-uu-n
the-students-NOM NEG 3M-study.IPFV-3MPL-IND
'The students do not study/are not studying.'

b. t-tullaab-u lan ya-dhab-u
the-students-NOM NEG.FUT 3M-go.IPFV-MPL.SBJV
'The students will not go.'

c. t-tullaab-u lam ya-dhab-uu
the-students-NOM NEG.PST 3M-go.IPFV-MPL.JUSS
'The students did not go.'

Given strong adjacency restrictions between the particle and the following auxiliary verb, Alsharif & Sadler (2009) propose they are non-projecting categories adjoined to I. The c-structure representation for (13) (without the time adverbial) is given in (14). Syntactically, the particle *laa* occupies a node $\widehat{\mathrm{I}}$ that is defined according to that on which it is structurally dependent, I.

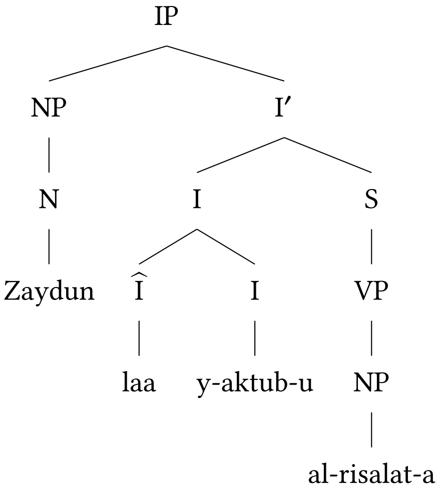
[6] I have adjusted the glosses in these examples to correct segmentation issues in the original examples.

(13) MSA (Alsharif & Sadler 2009: 7)

Zayd-un	laa	y-aktub-u	al-yawm-a	al-risalat-a
Zayd-NOM	NEG	3M-write.IPFV-3MS.IND	the-day-ACC	the-letter-ACC

Zayd is not writing the letter today.

(14) Arabic non-projecting negative particle *laa* (Alsharif & Sadler 2009: 14)



The lexical entry for *laa* is given in (15).

(15) *laa* Î (↑ TENSE PAST) ≠ +
(↑ POL) = NEG
(Alsharif & Sadler 2009: 16)

It specifies that its f-structure has the polarity value NEG , but also that it cannot have the tense value PAST. Equations of this type can be used to account for the distribution of different negative forms within the same language, as indicated by the lexical entries in (16) and (17) for *lam* and *lan*.

(16) *lam* Î (↑ TENSE PAST) = +
(↑ POL) = NEG
(↑ MOOD) $=_c$ JUSS
(Alsharif & Sadler 2009: 16)

(17) *lan* Î (↑ TENSE FUT) = +
(↑ POL) = NEG
(↑ MOOD) $=_c$ SBJV
(Alsharif & Sadler 2009: 16)

The possibility within LFG to formulate different lexical entries for different negators provides an additional opportunity to account for differences in their behavioural distribution and the features with which they are compatible.

2.2 Negative verbal morphology

Negation is indicated by verbal morphology in at least 36% of the word's languages (Dryer 2013).[7] There is a slight preference for prefixation of negative affixes over suffixation (Dryer 2013), which reflects a general cross-linguistic preference for negators to precede the verb (Dryer 1989).

In a lexicalist approach to syntax like LFG, it is notionally straightforward for negation to be expressed morphologically, but there is little consensus about how morphology itself should be modelled. The main issue is that affixes are often presented as having lexical entries that are distinct from their hosts. This suggests that an incremental model of morphology has been used in which morphosyntactic information gets added incrementally as morphemes are added to a stem (see Camilleri & Sadler (2017: 158) on the lexical entries for Polish *nie* discussed in Section 3.3). However, there are strong arguments for adopting a realizational approach in accounting for morphology, whereby a word's association with certain morphosyntactic properties licenses morphological operations. Under an approach of this kind, having distinct lexical entries for negative morphemes is highly questionable.

The first detailed LFG analysis of negation expressed through morphological means is provided in Bond (2016), who examines the expression of negation through tone and reduplication within Eleme (Niger-Congo, Cross River, Ogonoid) spoken in Rivers State, Nigeria. Like many other languages across Africa, Eleme has a multitude of means for expressing negation, many of which involve negation morphology. Negation in Eleme is distinctive from a cross-linguistic perspective in that in addition to affixation, negation of verbal predicates is also indicated though other morphological means, notably tonal alternations and stem reduplication. Two of the basic alternations, between perfectives and habituals are shown in (18) and (19).

Negation of perfectives is realised using a set of prefixes with the shape *rV́*-. The quality of the vowel is dependent on several factors: (i) the person and number of the subject, (ii) vowel harmony with the initial segment of the verbs stem

[7] This is a conservative figure calculated from the addition of two categories in Dryer's sample of 1157 languages: *negative affix* (n = 395) and *variation between negative word and affix* (n =21).

(Bond 2016: 280).[8] The negative prefix is obligatorily realised on Negative Perfective verb forms.[9] It is the only clear exponent of negation in (18b). However, in certain discourse contexts, prefixation is accompanied by pre-reduplication of the initial mora of the verb stem – shown in parentheses in (18b). This results in full reduplication of monomoraic stems and partial reduplication of bimoraic stems (see Bond 2016: 281 for examples).

(18) a. ǹ-sí
1SG-go
'I went.'

b. rĩ́-(si)~sí
NEG.1SG-(NEG)~go
'I didn't go.'
(Bond 2016: 281)

Habitual predicates in Eleme are distinguished by the presence of a Habitual suffix *-a* on the lexical verb stem, as in (19a). Negative Habituals are formed through the obligatory pre-reduplication of the first mora of the verb stem, as in (19b). The presence of the Habitual suffix *-a* is not attested in Negative Habituals, giving rise to an asymmetric pattern of negation in the sense of Miestamo (2005). Negative Habituals do not have a negative prefix. In (19b), negation is expressed though stem reduplication and tone.

(19) Eleme (Bond 2016: 278)

a. ǹ-sí-a
1SG-go-HAB
'I (usually) go.'

b. ǹ-sí~sì
1SG-NEG~go
'I don't (usually) go.'

Some examples of transitive constructions are given in (20).

(20) Eleme (Bond 2016: 283; own data)

[8] There is also intra-speaker variation in the realisation of the initial consonant, which varies between an alveolar nasal and alveolar approximant.

[9] Perfectivity is a default category in Eleme and is not overtly realised on verb stems by segmental morphology.

a. òsáro ré-de~dé òfí́
Osaro NEG.3SG-(NEG)~eat[HAB] mango
“‘Osaro didn’t eat (any) mango.’

b. òsáro è-dé~dè òfí́
Osaro 3[SG]-NEG~eat[PFV] mango
‘Osaro doesn’t (usually) eat mango.’

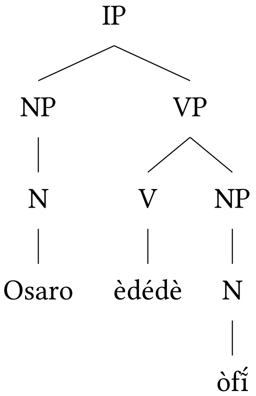
The examples show that there is no single affix that can be picked out for accounting for negation in Eleme, rather a number of different morphological processes are responsible for deriving negative verb stems (and a distinct theory of morphology is required to account for that because LFG does not yet have its own established native approach). In languages like Eleme, the feature responsible for contributing negation to the f-structure for clauses of this type comes directly from the lexical entry for the verb. Lexical entries for these verb forms are given in (21) and (22):

(21) *rédedé* V (↑ PRED) = ‘EAT⟨SUBJ,OBJ⟩’
(↑ POL) = NEG
(↑ ASP) = PFV
(↑ SUBJ PERS) = 3
(↑ SUBJ NUM) = SG

(22) *èdédè* V (↑ PRED) = ‘EAT⟨SUBJ,OBJ⟩’
(↑ POL) = NEG
(↑ ASP) = HAB
(↑ SUBJ PERS) = 3
(↑ SUBJ NUM) = SG

The c-structure for (20b) is provided in (23).

(23) C-structure containing an Eleme Negative Habitual verb *èdédè*



The central claim about negative verbs of this kind, whether negation is expressed by affixation, stem modification, reduplication, tone or any other morphological means, is that morphological negators do not occupy a syntactic node distinct from the element of which they are part, and any morphological exponent that can be identified as marking negation should be understood to be a property of a verb form (i.e. part of a paradigm) rather than having its own distinct lexical entry.

HPSG analyses of the morphological expression of negation (e.g. Borsley & Krer 2012, Kim 2000, Kim 2021) likewise propose that morphological exponence is dealt within the lexical component of grammar and, therefore, individual morphological exponents have no syntactic status distinct from the word of which they are part. Kim (2000) proposes that negation marked by affixation is achieved by a lexical rule (see Kim 2021 for a summary). The view of morphology proposed in Bond (2016) is a more complex one, chosen to deal with non-concatenative exponents as well as more straightforward instances of affixation. However, the basic underlying assumption is the same; morphology is governed by autonomous, non-syntactic principles (Bresnan & Mchombo 1995).

In derivational theories of syntax in which morphology is considered to be a post-syntactic process, there is no divide between the construction of words and sentences. In Distributed Morphology (DM), for instance, words are formed through syntactic operations like Merge and Move. Negative affixes – like other affixes expressing inflectional information – are realisations of abstract morphemes that are merged with roots. No such motivation for morphological operations needs to be justified within a lexicalist theory like LFG. There are two main approaches to accounting for reduplication in DM (see Frampton 2009, Haugen 2011 for discussion). Reduplication is proposed either to result from a readjustment operation on some stem triggered by a (typically null) affix, or through the insertion of a special type of affix which is inserted into a syntactic node in order to discharge some morphosyntactic feature(s), but which receives its own phonological content, distinct from its base. Recent proposals concerning the analysis of tone expressing grammatical categories can be found in Rolle (2018) and Pak (2019). See Chung (2007) on negation and suppletive forms in DM. A combination of these approaches would be required to account for morphologically complex expressions of negation like those seen here.

2.3 Negative auxiliaries

Negative auxiliaries are widely attested in the world's languages. Alongside the negative particles discussed in Section 2.1, MSA also has a negative auxiliary *laysa* employed in negative imperfectives.

Alsharif & Sadler (2009) argue that *laysa* is a fully projecting I, taking a range of complements. Unlike the particles discussed in Section 2.1, it is not subject to verb-adjacency restrictions, as illustrated in (24).[10] If the negative auxiliary verb is preceded by its subject, it agrees with it in gender and number. If the subject follows the auxiliary, number agreement is defective, and a default singular form is used.

(24) Modern Standard Arabic (Alsharif & Sadler 2009: 23)

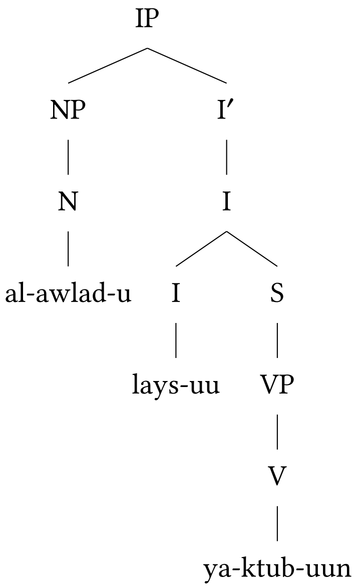
a. al-awlad-u lays-uu ya-ktub-uun
the-boys-NOM NEG-3MP 3M-write.IPFV-3MP-IND
The boys do not write/are not writing.

b. lays-a al-awlad-u ya-ktub-uun
NEG-3MS the-boys-NOM 3M-write.IPFV-3MP-IND
The boys do not write/are not writing.

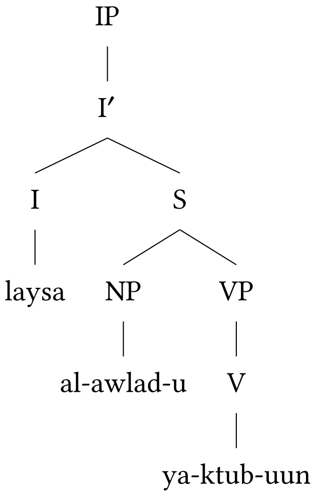
The corresponding c-structures for the examples in (24) are given in (25) and (26).

(25) Arabic negative auxiliary *laysa* in S-AUX order (Alsharif & Sadler 2009: 23)

[10] The gloss in (24b) has been corrected from the original source to show that number on the negative auxiliary is defective when it precedes the subject (Alsharif & Sadler 2009: 7).



(26) Arabic negative auxiliary *laysa* in AUX-S order (Alsharif & Sadler 2009: 23)



The differences between the behaviour of the negative particles (see discussion in Section 2.1) and the negative auxiliary in MSA are captured by differences in their lexical entries. The lexical entry for *laysa* is provided in (27).

(27) *laysa* I (↑ TENSE PAST) = –
(↑ TENSE FUT) = –
(↑ POL) = NEG
(↑ SUBJ PERS) = 3
(↑ SUBJ GEND) = MASC
V ∈ CAT(↑) ⇒ (↑ ASP) $=_c$ PROG
(Alsharif & Sadler 2009: 24)

Dalrymple & Nikolaeva (2011: 87) propose that English *didn't* also occupies the I node in c-structure (cf. *not* as a non-projecting head adjoined to I in Section 2.1).

3 Representations of negation as a feature

Negation is usually thought of as a property of a predicate, closely associated with verbal elements within the clause. Within f-structure representations, negation is typically represented in one of three distinct ways: as a feature-value pair (Section 3.1), as an adjunct with a negative value (Section 3.2), or by recognising that negation may be represented by multiple features within the same f-structure (Section 3.3)

3.1 Single feature-value pair

The majority of LFG analyses of negation treat negation as a value of predicate-level feature POL(arity). Like other attributes in the f-structure such as [TENSE] and [ASP], the [POL] specification has more than one possible value, either represented as a binary feature (i.e. = ±POL or ±NEG), or a feature with multiple values, e.g. [AFF] and [NEG]. The former approach is used by King (1995), Niño (1997), Butt et al. (1999), Bresnan (2001b: 183) and Dalrymple & Nikolaeva (2011: 87) , while Alsharif & Sadler (2009) and Bond (2016) employ the multiple value approach (i.e. POL: NEG). Falk (2001: 12, 149) uses NEG+ and POL: NEG within the same book.

In each case, it is always possible to identify an inherently negative element; this element always contributes the specification [POL –], [NEG +] or [POL NEG] to f-structure. They are all used to represent exactly the same thing, using different notation systems. In the lexical entries so far, I have used the attribute POL, with the value NEG, to account for sentential negation.

In the illustrations of the different proposals that follow, I use the representation system proposed in the original analysis.

Let's start by considering the English example in (28) from Dalrymple & Nikolaeva (2011: 87), with the f-structure in (29).

(28) 'John didn't love Rosa.'

(29)
$$\left[\begin{array}{ll} \textsc{pol} & - \\ \textsc{pred} & \text{`}\textsc{love}\langle\textsc{subj}, \textsc{obj}\rangle\text{'} \\ \textsc{subj} & \left[\textsc{pred}\ \ \text{`}\textsc{John}\text{'}\right] \\ \textsc{obj} & \left[\textsc{pred}\ \ \text{`}\textsc{Rosa}\text{'}\right] \end{array}\right]$$

Here, the only representation of negation in the f-structure is with the feature POL (Dalrymple & Nikolaeva 2011: 87). The '–' specification indicates that it does not have affirmative polarity.

3.2 Adjunct value

In contrast to introducing negation through a binary feature (e.g. NEG), in some LFG analyses, negation is introduced as an appropriate element of the ADJ(unct) feature, as illustrated in (31) for (30), discussed in Przepiórkowski & Patejuk 2015: 323-4.[11]

(30) 'John doesn't like Mary.'

(31)
$$\left[\begin{array}{ll} \textsc{pred} & \text{`}\textsc{like}\langle\textsc{subj}, \textsc{obj}\rangle\text{'} \\ \textsc{subj} & \left[\textsc{pred}\ \ \text{`}\textsc{John}\text{'}\right] \\ \textsc{obj} & \left[\textsc{pred}\ \ \text{`}\textsc{Rosa}\text{'}\right] \\ \textsc{adj} & \left\{\left[\begin{array}{ll} \textsc{pred} & \text{`}\textsc{not}\text{'} \\ \textsc{adj-type} & \textsc{neg} \end{array}\right]\right\} \end{array}\right]$$

Here, the ADJ(unct)-TYPE feature enables the syntactic properties of negative adjuncts to be distinguished from other adjuncts.[12] One rationale for adopting this approach is that it makes it easy to represent multiple negation (via multiple negative elements of the ADJ set). This is the approach taken by Laczkó (2014) in his account of negation in Hungarian, where both predicate negation and narrow-scope negation are treated as adjuncts because they can co-occur, as in (32) repeated from (8).

[11] Przepiórkowski & Patejuk (2015) state that, within PARGRAM, the majority of XLE implementations of negation to date take this approach, but this is not reflected in the LFG literature, in which verbal negation is nearly always represented by a feature in works that predate their paper (e.g. Sells 2000, Alsharif & Sadler 2009).

[12] An anonymous reviewer points out that there are added complications associated with this model in accounting for the presence of *do* in English negatives if *not* is added as an adjunct.

(32)

Péter	NEM	A	BARÁTJÁ-T	nem	hívta	fel.
Peter.NOM	not	the	friend.his-ACC	not	called	up

'It wasn't his friend that Peter didn't call up.'
(Laczkó 2014: 307)

Importantly, both instances of *nem* occur in the same clause, although not in the same f-structure (cf. the bi-clausal translation in English). The simplified f-structure in (33), representing (32), is consistent with the essence of Laczko's (2014) analysis of similar sentences.[13]

(33)

$$\left[\begin{array}{ll}
\text{PRED} & \text{`CALL.UP}\langle\text{SUBJ, OBJ}\rangle\text{'} \\
\text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`PETER'} \\ \text{CASE} & \text{NOM}\end{array}\right] \\
\text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`HIS FRIEND'} \\ \text{CASE} & \text{ACC} \\ \text{ADJ} & \left\{\left[\begin{array}{ll}\text{PRED} & \text{`NOT'} \\ \text{ADJ-TYPE} & \text{NEG}\end{array}\right]\right\}\end{array}\right] \\
\text{ADJ} & \left\{\left[\begin{array}{ll}\text{PRED} & \text{`NOT'} \\ \text{ADJ-TYPE} & \text{NEG}\end{array}\right]\right\}
\end{array}\right]$$

One of the major issues with this approach concerns how to limit the number of instances of the adjunct with clauses. Przepiórkowski & Patejuk (2015) report that in a later presentation, Laczkó (2015) revises his account, suggesting that two binary features may be necessary to account for the negation in Hungarian. He proposes distinguishing between ±POL and ±NEG, where each is a different feature (rather than different ways of notating the same feature).

3.3 Multiple feature-value pairs

Building on the observations made by Laczkó (2015) for Hungarian, Przepiórkowski & Patejuk (2015) propose that two different types of binary-valued attributes are required to account for negation in Polish. This distinction is motivated by (i) the distinctive behaviour of two sets of negative constructions in which the negator *nie* exhibits different degrees of syntactic independence, and (ii) the possibility that two instances of negation can occur within the same clause. This

[13] Laczko's (2014) formalisations are somewhat idiosyncratic in that his f-structure representations deviate from those typically seen in the LFG literature. While he does not actually provide an f-structure containing two instances of *nem*, there is much more analysis included in the paper than can be discussed here, and readers are directed to his paper for an extensive discussion of negation in Hungarian.

leads them to propose two distinct features known as eventuality negation (eneg) and constituent negation (cneg).

While typically represented orthographically as a separate word, manifestations of *nie* can be broadly distinguished as 'bound' and 'independent'. Bound *nie* has a strong adjacency requirement with its host, and is described as a prefix that forms a prosodic unit with the stem to which it attaches (Kupść & Przepiórkowski 2002; Przepiórkowski & Patejuk 2015: 324). Negation expressed by prefixal *nie* cannot scope over co-ordinands, demonstrating that its semantic effects are bounded. It triggers a range of syntactic effects: first, it requires that otherwise accusative arguments of the element that is negated occur in the genitive case (the so-called 'genitive of negation'), seen in (34a), and second, it licences a syntactic domain in which negative indefinites occur, shown in (34b).

(34) Polish (Przepiórkowski & Patejuk 2015: 324)

a. Janek nie lubi Marii.
Janek.nom neg likes Maria.gen
'Janek doesn't like Maria.'

b. Nikt nie lubi nikogo.
nobody.nw.nom neg likes nobody.nw.gen
'Nobody likes anybody.'

Bound *nie* is associated with eventuality negation, so called because it is used to negate eventualities (i.e. events and states). The syntactic properties associated with eneg are observed when *nie* is realised on verbs, adjectives and deadjectival adverbs, and it is for this reason that they favour the adoption of the term eventuality negation over sentential negation or predicate negation (Przepiórkowski & Patejuk (2015: 324-6) for discussion of this). Negative indefinite pronouns (see Section 5.2) are also licensed by the preposition *bez* 'without', leading Przepiórkowski & Patejuk (2015: 326) to suggest that this also introduces a value for the eneg feature.

In contrast to the bound realisation, independent *nie* may be separated from the constituent over which it scopes (Przepiórkowski & Patejuk 2015: 329), indicating that it is not a morphological exponent of negation. This structural difference is reflected in a number of associated effects. Unlike the bound negator, it can scope over co-ordinands, and it does not licence negative case alternations or negative indefinites, as shown by the ungrammaticality either of the genitive object *Marii* or an negative indefinite pronoun object, in (35).

(35) Nie Janek lubi Marię \ *Marii \ *nikogo (lecz
NEG Janek.NOM likes Maria.ACC Maria.GEN nobody.NW.ACC/GEN but
Tomek).
Tomek.NOM
‘It is not Janek that likes Maria (but Tomek).’
(Przepiórkowski & Patejuk 2015: 326)

Crucially, the two different types of negation are sometimes attested in superficially similar environments, as seen with infinitival clauses. In (36), in which the infinitival clause, but not the head of the main predicate is within the scope of negation, the genitive of negation is not permitted. This is an example of CNEG. In (37), where the negated infinitival clause functions as the post-verbal subject, only genitive case is permitted: this is an example of ENEG. Similar effects are observed with the licensing of negative indefinites (Przepiórkowski & Patejuk 2015: 327).

(36) Ma skakać, a nie pisać wiersze \ *wierszy.
has jump.INF and NEG write.INF poems.ACC poems.GEN
‘He is to jump, and not to write poems.’ [of a sportsman]
(Przepiórkowski & Patejuk 2015: 326)

(37) Poetyckim marzeniem Karpowicza było: nie pisać wierszy \
poetic.INS dream.INS Karpowicz.GEN was NEG write.INF poems.GEN
*wiersze.
poems.ACC
‘The poetic dream of Karpowicz was not to write poems.’
(Przepiórkowski & Patejuk 2015: 327)

Building on these observations, Camilleri & Sadler (2017: 158) propose the following (basic) lexical entries for the two types of negation, in order to provide an explicit characterisation of their differences:

(38) nie: ENEG (↑ ENEG) = +
(Camilleri & Sadler 2017: 158)

(39) nie: CNEG (↑ CNEG) = +
(Camilleri & Sadler 2017: 158)

In their formalisation, the lexical entries are identical other than the feature they introduce. However, since *nie* is a prefix when introducing the ENEG value, and is therefore part of the morphology of the verb, this should not be considered to have a lexical entry that is distinct from that of the verb form of which it is part (cf. Bond's 2016 analysis of negative verbs forms in Eleme, discussed in Section 2.2). A minimal lexical entry for *niepisác* is provided in (40).

(40) *niepisác* V (↑ PRED) = ‘WRITE⟨SUBJ, OBJ⟩’
(↑ ENEG) = +

These two different features are required to account for the fact that both types of negation may occur in the same clause, as shown in (41) (cf. ‘The Catholic Church not cannot…). Przepiórkowski & Patejuk (2015: 327) do not distinguish the two types of negation in their glossing.

(41)

Kościół	katolicki	nie	nie	portrafi,	ale	nie	chce.
church.NOM	catholic.NOM	NEG	NEG	can	but	ENEG	want

‘It's not that the Catholic Church cannot, but rather that it doesn't want to.’
(Przepiórkowski & Patejuk 2015: 327)

Przepiórkowski & Patejuk (2015: 327) propose the following f-structure to account for the first part of (41):

(42)
$$\left[\begin{array}{ll} \text{ENEG} & + \\ \text{CNEG} & + \\ \text{PRED} & \text{‘CAN}\langle\text{SUBJ, XCOMP’}\rangle \\ \text{SUBJ} & \left[\text{PRED ‘Catholic Church’}\right] \\ \text{XCOMP} & \left[\ldots\right] \end{array}\right]$$

Other scholars have also observed that more than one negation may be required within a clause (e.g. Butt et al. 1999, Sells 2000, Laczkó 2014). We now explore this subject in Section 4 in relation to bipartite negation, and in Section 5 in relation to antiveridical contexts.

4 Multipartite negation

In many languages, negation is reflected in the formal properties of multiple elements with the clause. For instance, Standard (Written) French usually requires the use of preverbal *ne* and post-verbal *pas* in the formation of negative clauses.[14]

[14] This is not true of colloquial varieties of French, in which *pas* is usually used without *ne*.

In a very brief analysis, Butt et al. (1999: 142-3) propose that both elements should be represented in f-structure, with the initial component *ne* contributing a NEG feature, and *pas* contributing a related feature NEG-FORM, as illustrated for (43) in (44) from Dalrymple et al. (2019: 67).

(43) David n' a pas mangé de soupe.
David NEG have POSTNEG eaten of soup
'David did not eat any soup.'
(adapted from Butt et al. 1999: 143, following Dalrymple et al. 2019: 67)

(44)
$$\begin{bmatrix} \textsc{neg} & + \\ \textsc{neg-form} & \textsc{pas} \\ \textsc{pred} & \text{`}\textsc{eat}\langle\textsc{subj},\ \textsc{obj}\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`}\textsc{David}\text{'} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`}\textsc{soup}\text{'} \end{bmatrix} \end{bmatrix}$$

In the analysis of Butt et al. (1999: 142-3), the marker providing the NEG + feature at f-structure may only appear if the NEG-FORM feature, contributed by the other negative particle, is present.

Their proposal aims to capture the view that (i) two distinct manifestations of negation are required to negate a clause, (ii) that there is an asymmetry between the roles of the negators in terms of their featural specification, and (iii) that the presence of *ne* is dependent on the presence of some other negative formative. This helps to account for the distribution of *ne* in clauses like (45), where it co-occurs with the adverb *jamais* 'never'.[15] However, their analysis does not deal with the use of *pas* as the only negator of a clause, as typically found in spoken French varieties. In such cases, *pas* must either be treated as separate negative item that contributes a NEG feature without *ne*, or a more serious revision to this analysis is required.

(45) David ne mange jamais de soupe.
David NEG eat POSTNEG.NEVER of soup
'David never eats soup.'
(adapted from Butt et al. 1999: 143)

Working with HPSG, Kim (2000, 2021) takes a different approach to analysing the distribution of *ne* and *pas* in spoken French, proposing that *ne*-*pas* are part

[15] However, *jamais* only has this interpretation within the context of negation, meaning 'ever' in non-negative contexts. If their analysis is correct, a separate lexical entry must exist for *jamais* when it is not negative, or this proposal requires revision in some other way.

of a single lexical entry, and in this sense parallel the type lexical entry for *not* in English.

Expression of negation by multiple negative formatives is extremely common in the Niger-Congo languages of Africa. For instance, this is the case in Ewe (Niger-Congo, Kwa; Ghana), where negation is simultaneously expressed by a negative particle *mé* that precedes the VP and a post-VP particle *o*, that follows objects and adverbial elements within the VP, as illustrated in (46).[16] Both NEG1 and NEG2 are obligatory.[17]

(46) Ewe

a. Kofi mé-ɖu nú o
Kofi NEG1-eat thing NEG2
'Kofi didn't eat.' (Collins et al. 2018: 333)

b. nye-mé-ƒo nu kplé Kofí o
1SG-NEG1-hit mouth with Kofi NEG2
'I didn't speak with Kofi.' (Collins et al. 2018: 334)

c. Kofí mé-wɔ-a é-ƒé aƒéme-dɔ́ gbeɖé o
Kofi NEG1-do-HAB 3SG-POSS home-work ever NEG2
'Kofi never does his homework.' (Collins et al. 2018: 361)

When an auxiliary is present, it hosts the negative marker, as in (47) with the future auxiliary *-á* and in (48) with the 'not yet' auxiliary *kpɔ́*:

(47) Ewe (Collins et al. 2018: 360)
nye-mé-á yi China gbeɖé o
1SG-NEG1-FUT go China ever NEG2
'I will never go to China.'

(48) Ewe (Ameka 1991: 50)[18]

[16] Although Collins et al. (2018) adopt an orthographic convention in which *mé* is written as a prefix, their description, taken together with discussion in Ameka (1991: 64-9) and Aboh (2010: 64-9), suggests that *mé* occupies a node in syntax distinct from its host. Ameka (1991: 64-9) notes that *mé* usually encliticises to the verb.

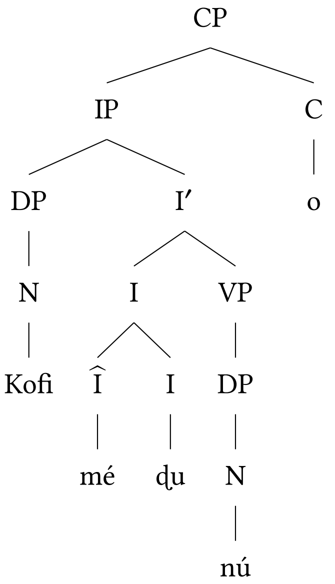
[17] This is unlike typical examples of negative concord, in which so called N-words are licensed only in the presence of sentential negation, and can be the answer to a sentence fragment question (see Section 5.2). Most fragment answers obligatorily require the presence of *o*, but this is because it occurs together with an NPI, not an N-word (Collins et al. (2018: 350-354)).

[18] The glosses have been adjusted slightly to reflect the conventions in Collins et al. (2018), but the text line remains unchanged.

nye-mé kpɔ́ wɔ dɔ lá o
1SG-NEG MOD do work DEF NEG2
'I have not had the opportunity to do the work'

Collins et al. (2018) analyse sentences such as those in (46) as having a structure in which NEG1 and NEG2 are not part of the same inflectional phrase. In their analysis, NEG2 occupies a syntactic position outside the TP (this would be an IP in a typical LFG analysis), in the specifier position of a C (see Collins et al. 2018: 293 for the structure). The c-structure in (49) reflects the principal aspects of their descriptive analysis, although the NEG1 particle *mé* is analysed as adjoined to I (rather than as the specifier of T), in a similar way to the analysis from Alsharif & Sadler (2009) discussed in Section 2.1. Assuming that *o* takes an IP complement, NEG2 is rendered here as C (rather than in the specifier position of an empty C).

(49) Ewe bipartite negation based on Collins et al. (2018: 293)



As with French, the question arises as to whether these two manifestations of negation should be represented in f-structure by multiple features, or whether a single feature is sufficient. I propose that it is the latter that is true; despite having multiple attestations within the clause, only one f-structural representation of negation is required, as illustrated by the f-structure in (50).[19]

This corresponds to the f-structure in (50).

[19] cf. the representation of clitic doubling in Dalrymple (2001: 79-81).

(50)
$$\begin{bmatrix} \textsc{eneg} & + \\ \textsc{compform} & \textsc{neg} \\ \textsc{pred} & \text{`}\textsc{eat}\langle\textsc{subj}, \textsc{obj}\rangle\text{'} \\ \textsc{subj} & \left[\textsc{pred}\ \text{`}\textsc{Kofi}\text{'}\right] \\ \textsc{obj} & \left[\textsc{pred}\ \text{`}\textsc{thing}\text{'}\right] \end{bmatrix}$$

Crucially, both negative elements are obligatory, but, in the analysis I propose for Ewe in (50), the negative particles constrain a single attribute-value pair. This type of analysis is commonly encountered when dealing with features in LFG – for instance when featural specifications of a GF (e.g. SUBJ) are specified by both the predicate and its subject noun phrase (see Dalrymple 2001: 100-4 for an introduction). Because it is possible and indeed common for two f-structure descriptions to constrain the same attribute value pairs, it should not be particularly strange that negation can also behave in this way. In other languages, where the value of the POL feature must be contributed by a single form, and where multiple contributions are consequently disallowed, then an instantiated symbol can be used as the value of the POL attribute. See Section 5.2 for an example of the usage of this symbol.

In order to ensure that both elements are present in a well formed negative sentence, a constraining equation needs to be specified to impose an additional requirement on the minimal solution obtained from the defining equations in the f-description. A complete analysis of these structures requires that the presence of *o* is constrained (since it is obligatory here). Without a very detailed examination of the Ewe negation system, it is difficult to say exactly what type of constraint might be most appropriate. However analyses of other languages with bipartite negation have involved the addition of a special feature in f-structure, NEG-FORM, which must be contributed by the second negative formative (see Section 3.1).

5 Negative Sensitive Items

Much of the theoretical literature on the syntax of negation examines the distribution of so-called Negative Sensitive Items (NSIs), that is, words whose distribution is sensitive to the presence of negation within a clause. Here we consider three types of sensitivity. The first, which I will refer to as Polarity Sensitive Cases (PSCs) is discussed in Section 5.1. Two further mains types of NSIs are distinguished in the literature: Negative Concord Items (NCIs), introduced in Section 5.2, and Negative Polarity Items (NPIs), discussed in Section 5.3.

5.1 Polarity Sensitive Case

Polarity Sensitive Cases are observed when the case-marking of an argument is sensitive to the polarity of its clause. The most well-known example of this is seen in the genitive of negation in Slavic languages (e.g. Neidle 1988, Brown 1999). The basic contrast in case assignment is illustrated by (51) and (52) from Patejuk & Przepiórkowski (2014) using Polish examples from the Polish National Corpus.

(51) Polish (Patejuk & Przepiórkowski 2014: 431)
Poczytam książkę.
read.1SG book.ACC
'I'll read a book.'

(52) Polish (Patejuk & Przepiórkowski 2014: 431)
Nie poczytają książki czy gazety.
read.3PL NEG book.GEN or newspaper.GEN
'They won't read a book or a newspaper.'

Patejuk & Przepiórkowski (2014) propose that structural case assignment generalisations of this type could be formalised using constraints placed in the lexical entries of verbs that follow this pattern.

The STRCASE constraint in (53) indicates that verbs that follow structural case assignment rules follow different disjunctive constraints, labelled as AFFIRMATIVE and NEGATIVE. Note that in Patejuk and Przepiórkowski's (2014) analysis, negation is assumed to be a binary feature represented by the attribute NEG in f-structure.

(53) $\textsc{strcase} \equiv [\textsc{affirmative} \vee \textsc{negative}]$

(54) $\textsc{affirmative} \equiv [\neg(\uparrow \textsc{neg}) \wedge (\uparrow \textsc{obj case}) =_c \textsc{acc}]$

(55) $\textsc{negative} \equiv [(\uparrow \textsc{neg}) =_c + \wedge (\uparrow \textsc{obj case}) =_c \textsc{gen}]$

The AFFIRMATIVE constraint in (54) ensures that when there is no negation in the f-structure of the head ($\neg(\uparrow$ NEG$)$) , the object is marked for accusative case: $(\uparrow$ OBJ CASE$) =_c$ ACC. The NEGATIVE constraint in (55) ensures that when the f-structure of the head is negative ($(\uparrow$ NEG$) =_c +$), the object is marked for genitive case: $(\uparrow$ OBJ CASE$) =_c$ GEN.

Patejuk & Przepiórkowski (2014) demonstrate that while such constraints can account for simple cases of structural case assignment, case assignment in constructions with control or raising verbs combining with (open) infinitival arguments (i.e. XCOMPs) do not follow these constraints. Consider (56). In this example, the verb *chcesz* 'want' takes an infinitival complement whose subject is controlled by the subject of the higher verb.

The verb subcategorising for the object (i.e. the infinitival verb *poczytać* 'read') is not negative, yet the genitive of negation is still required because *chcesz* 'want' is negative. Negation is present in (56), but it is 'non-local' to the infinitival clause of the verb subcategorising for the object.

(56) Polish (Patejuk & Przepiórkowski 2014: 432)

Nie	chcesz	poczytać	Kodeksu.
NEG	want.2SG	read.INF	Code.GEN

'You don't want to read the Code.'

While the genitive of negation is possible when negation is non-local, they observe that there appears to be some variation as to whether the lower object should occur in the accusative or in the genitive, citing semantic and structural or linear distance factors as potentially important.

For instance, in (57), the object is marked for accusative case (*książkę* 'book'), even though there is (non-local) verbal negation present higher in the structure of the sentence (at the main verb *chce* 'wants'). This illustrates that the presence of negation in a higher clause is not sufficient to ensure that the genitive of negation occurs.

(57) Polish (Patejuk & Przepiórkowski 2014: 432)

Mama	nie	chce	iść	poczytać	książkę.
mum	NEG	want.3SG	go.INF	read.INF	book.ACC

'Mum doesn't want to go and read a book.'

To account for this difference in case-marking, they propose that the constraints in (53)–(55) could be rewritten as (58)–(60).

(58) $\textsc{strcase} \equiv [\textsc{affirmative} \vee \textsc{negative}]$

(59) $\textsc{affirmative} \equiv [\neg(\uparrow \textsc{neg}) \wedge (\uparrow \textsc{obj case}) =_c \textsc{acc}]$

(60) $\textsc{negative} \equiv [((\textsc{xcomp}^{*} \uparrow)\ \textsc{neg}) =_c + \wedge (\uparrow \textsc{obj case}) =_c \textsc{gen}]$

The constraint in (59) states that accusative case is necessary whenever there is no local negation, while (60) indicates that genitive case is possible whenever sentential negation is available somewhere in the verb chain, locally or non-locally. Specifically, this is achieved by using an inside-out path ((xcomp* ↑) neg) $=_c$ + which makes it possible to reach into any number of successive higher predicates subcategorising for an infinitival complement (i.e. an xcomp), and check if any of these predicates is negated.

5.2 Negative Concord Items

In many languages negation may be expressed through the use of negative indefinite pronouns such as English *nothing* and Polish *nikt* 'nobody'. Haspelmath (1997) argues that there are three main subtypes of construction involving negative indefinite pronouns. First, in some languages there are negative indefinites that always co-occur with verbal negation, e.g. the Polish *ni-* series, as in (61).

(61) Polish (Haspelmath 1997: 194)

a. Nikt nie przyszedł.
nobody neg come.pst.3sg
'Nobody came.'

b. Nie widziałam nikogo.
neg saw nobody
'I saw nobody.'

The second type of negative indefinites do not usually co-occur with verbal negation, e.g. the Standard British English *no*-series: *Nobody came* and *I saw nobody*. If they do co-occur, they are rejected by speakers, or are interpreted as having a 'double negative' reading cf. *Nobody didn't come* (=*Everybody came*).[20]

His third type of negative indefinites sometimes co-occur with verbal negation and sometimes do not, e.g. the Spanish *n*-series, exemplified in (62).[21]

(62) Spanish (Haspelmath 1997: 201)

a. Nadie vino.
nobody came
'Nobody came.'

[20] Negative indefinites in the *no*-series in some other varieties of English do not behave in this manner, and thus they belong to one of the other types.

[21] The fact that the languages used to exemplify these types all come from European languages indicates the prevalence of indefinite pronouns in this area. It is largely unknown to what extent indefinite pronouns might be restricted by areal or genetic factors.

b. No vi a nadie.
NEG I.saw ACC nobody
'I saw nobody.'

The role that a negative pronoun plays in negating a clause depends on its ability to appear independently of another negation strategy. Negative pronouns like those in Polish which do not appear without an expression of negation are known as Negative Concord Items (NCIs), sometimes known as *n*-words. By definition, NCIs never occur outside of negative contexts, and when they combine with other expressions of negation, they contribute to a single semantic negation (Labov 1972). NCIs must combine with sentential negation as in (34b) and (61) with Polish *nikt* 'nobody'. NCIs are important tools for investigating the domains in which negation has structural affects. The following definition, based on Giannakidou (2006: 328), is adopted by Camilleri & Sadler (2017: 150):

(63) An *n*-word or NCI is understood to be an expression α that can be used in structures containing sentential negation or another α -expression to yield a reading equivalent to one logical negation, and which can provide a negative fragment answer.

Because NCIs in Polish always occur with another negator, the lexical entries for *n*-words such as *nikt* 'nobody.NOM' and *nikogo* 'nobody.ACC/GEN' must include a constraining equation that ensures their f-structure is specified for eventuality negation (Przepiórkowski & Patejuk 2015: 331):

(64) *nikt* N $(\uparrow \textsc{case}) = \textsc{nom}$
$((\textsc{xcomp}^{*}\ \textsc{gf}+\ \uparrow)\ \textsc{eneg}) =_{c} +$

(65) *nikogo* N $(\uparrow \textsc{case}) \in \{\textsc{acc}, \textsc{gen}\}$
$((\textsc{xcomp}^{*}\ \textsc{gf}+\ \uparrow)\ \textsc{eneg}) =_{c} +$

There is much more to say about how differences in the distribution of NCIs cross-linguistically could be modelled in LFG, but I leave this aside as a topic for further investigation.

5.3 Negative Polarity Items

Negative Polarity Items (NPIs) are a set of elements that, while not inherently negative, are licensed within a set of restricted contexts including negative ones. Examples from English include the indefinite quantifier *any* and the adverb *yet*, as illustrated in (66).

(66) a. Isaac wouldn't give her any/*Isaac would give her any.
b. Eva hasn't finished yet/*Eva has finished yet.

Since NPIs are also observed in a range of other syntactic contexts, such as comparatives, modal and conditional contexts and polar interrogatives, as in (67), they are not inherently negative, and the term, attributed to Baker (1970) by Haspelmath (1997), is somewhat misleading.

(67) a. Would Isaac give her any?
b. Has Eva has finished yet?

However, assuming that all items described as NPIs can be minimally licensed in negative contexts, they can be further divided into two main types, that may exist within one and the same language:

- Weak Negative Polarity Items: NPIs that exhibit a range of non-negative contexts of use. These are sometimes referred to as *Affected Polarity Items* (API) (Giannakidou 1998).

- Strong Negative Polarity Items: NPIs that are only licensed in antiveridical contexts (Giannakidou 1998), i.e. sentential negation and '*without*' clauses (cf. eventuality negation).

For Weak Polarity Items, such as those in (66) and (67), negation is a sufficient, but not necessary condition for the licensing. For Strong Negative Polarity items, the context must be antiveridical (see Zwarts 1995 and Giannakidou 1998).

Consider the technical definition in (68) from Giannakidou (2002), who treats verdicality as a propositional operator:

(68) A propositional operator F is veridical iff Fp entails p: Fp → p; otherwise F is nonveridical. Additionally, a nonveridical operator F is antiveridical iff Fp entails not p: Fp → ¬p.

A veridical context is one in which the semantic or grammatical assertion about the truth of an utterance is made. The presence of a veridicality entails that the truth conditions for the underlying proposition are met, while non-veridical expressions do not entail that the truth-conditions for the underlying proposition have been met. Though (69a) is veridical, with or without the auxiliary, (69b-69c) are both nonveridical.

(69) a. 'I (do) like her.'

b. ‘I might like her’

c. ‘I don’t like her.’

Nonveridical operators are antiveridical if (and only if) the truth conditions for the underlying proposition are not met, as in (69c). Strong NPIs are sensitive to such environments.

These differences in behaviour raise important questions about how best to account for the distribution of NSIs and in which structures of grammar – essentially – to what extent can and should the distribution of NCIs and NPIs be accounted for through c-structure and f-structure representations. Problems of this kind have been addressed by Sells (2000) in relation to Swedish, and Camilleri & Sadler (2017) with respect to Maltese.

Camilleri & Sadler (2017) examine the relationship between sentential negation in Maltese and the set of negative sensitive items (NSIs). They demonstrate that the N-series of negative indefinites in Maltese exhibit mixed behaviour with respect to the environments in which they occur (Camilleri & Sadler 2017: 154–156). The majority of items can occur in a range of non-veridical contexts, and are not limited to antiveridical ones, exemplifying properties consistent with being classified as weak NPIs. However two NSIs show a more limited distribution: the determiner *ebda* is strictly limited to antiveridical contexts (and thus is a Strong NPI), while *ħadd* is largely restricted outside of antiveridical contexts, showing less categorical behaviour.

In finite verbal predicates in Maltese, negation is expressed through the use of the particle *ma* together with a verbal form inflected with the suffix *-x*, as illustrated in (70) and (71).[22]

(70) Maltese (Camilleri & Sadler 2017: 147)
Ma qraj-t-x il-ktieb.
NEG read.PFV-1SG-NVM DEF-book.
‘I didn’t read the book.’

(71) Maltese (Camilleri & Sadler 2017: 147)
Ma n-iekol-x ħafna.
NEG 1-eat.IPFV.SG-NVM a.lot.
‘I don’t eat a lot.’

[22] I have adjusted the glosses in these examples so that *-x* is glossed as NVM rather than NEG, to reflect the final analysis proposed by Camilleri & Sadler (2017). .

Imperfectives can also be negated using a different strategy otherwise associated with non-verbal predicates and non-finite forms. In (72) , *m(a)-* is prefixed to a form identical to a nominative pronominal, which, like the verbs in (70) and (71), is suffixed with *-x*. This pronominal may occur in a default third person singular masculine form, or vary according to the features of the subject, as shown here.

(72) Maltese (Camilleri & Sadler 2017: 148)
Mhux ~ minix n-iekol ħafna.
NEG.3SGM.NVM ~ NEG.1SG.NVM 1-eat.IPFV.SG a.lot.
'I am not eating a lot.'

Although the formation of negative indicative clauses of the type in (70) and (71) involves both the particle *ma* and the suffix *-x*, verb forms only inflected with *-x* cannot license a domain in which items from the N-series or any NSI are permitted. Rather, such items are in complementary distribution with *-x*. Camilleri & Sadler (2017: 150) consequently propose that *ma* expresses eventuality negation (ENEG), that introduces a syntactic requirement for a further element, which they call a non-veridical marker (NVM). In examples like (70) and (71), the presence of *-x* on the verb satisfies this requirement, while in examples like (73) it is satisfied by the presence of an NCI, such as *xejn* 'nothing'.

(73) Maltese (Camilleri & Sadler 2017: 159)
Ma qraj-t xejn.
NEG read.PFV-1SG nothing.
'I read nothing.'

Examples such as (74) indicated that the NCI satisfying this requirement needn't be local, and can be deeply embedded.

(74) Maltese (Camilleri & Sadler 2017: 153)
Ma smaj-t li qal-u li
NEG hear.PFV.1SG COMP say.PFV.3-PL COMP
qal-t-i-l-hom li gèand-hom
say.PFV.3SGF-EPENT.VWL-DAT-3PL COMP have-3PL-GEN
j-i-xtr-u xejn.
3-FRM.VWL-buy.IPFV-PL nothing
'I didn't hear that they said she told them they have to buy anything.'

This is prohibited if the embedded clause containing the negative indefinite is itself marked with *ma*. Camilleri & Sadler (2017) propose the following lexical entries to account for this:

(75) *xejn* N $(\uparrow \textsc{nvm}) = +$
(Camilleri & Sadler 2017: 159)

(76) *-x* $(\uparrow \textsc{nvm}) = +$
$\neg(\uparrow \{\textsc{xcomp}|\textsc{comp}|\textsc{adj}\}^{*}\ \textsc{gf}^{+}\ \textsc{nvm}) = +$
$\neg(\rightarrow \textsc{eneg})$
(Camilleri & Sadler 2017: 160)

The entry for *xejn* in (75) ensures that its f-structure has the NVM value +. The entry for *-x* – which should really be understood to be part of the lexical entry for the verb form of which it is part – does a similar thing. It ensures that its f-structure instantiates the NVM feature with the value +. But the second line of (76) further stipulates that this form is incompatible with any XCOMP, COMP, ADJ or grammatical function with NVM+ (e.g. *xejn*), except embedded clauses which are themselves marked for sentential negation.

The entry for *ma* in (77) contributes the ENEG feature with the value +. The underscore following the + marks the feature as 'instantiated'. This means it is required to be uniquely contributed, so expressed only once in the f-structure. It also places the requirement that an element NVM is present, but this may be non-local or local. The path definition for GF is given in (78).

(77) *ma* ENEG $(\uparrow \textsc{eneg}) = +\_$
$\{\, (\uparrow \{\textsc{xcomp}|\textsc{comp}|\textsc{adj}\}^{*}\ \textsc{gf}^{+}\ \textsc{nvm}) \mid (\uparrow \textsc{nvm})\, \} =_{c} +$
$\neg(\rightarrow \textsc{eneg})$
(Camilleri & Sadler 2017: 160)

(78) $\textsc{gf} \equiv \{\, \textsc{subj} \mid \textsc{obj} \mid \textsc{obj}_{\theta} \mid \textsc{obl} \mid \textsc{poss} \mid \underset{\neg(\rightarrow \textsc{tense})}{\textsc{adj}} \in \}$

Camilleri & Sadler's (2017) observation that some formal elements that at first sight look like negator (e.g. *-x*) may actually be better described as non-veridical markers is an important development not only in terms of descriptive linguistics, but also in the context of how co-occurrence of different elements in negative construction can be constrained.

6 Conclusion

Negation is found in every language, yet can be manifested in a vast number of ways and forms that can occur in practically every position in c-structure. While Chomskian models of syntax usually adopt an approach in which negators head their own functional projection NegP, with LFG, negators occupy the structural position that most closely accounts for their distribution. This allows for an approach in which cross-linguistic variation in the distribution and category of negative word forms is captured using existing means for determining and modelling constituency. Indeed, in many languages negators exhibit properties of non-projecting heads, indicating that adopting a single functional phrase type fails to capture the variation encountered across languages.

While a range of approaches have been proposed to model the featural properties of negation, recent research into modelling negation with LFG suggests that two different f-structure features are required to account for the distribution of negative forms and the syntactic and semantic domains that they license. These are known as ENEG, or eventuality negation, and CNEG or constituent negation. The presence of ENEG is typically associated with a broader range of syntactic and semantic effects than CNEG. The pragmatic distribution is also different, with CNEG notably employed in cases where there is a negated proposition.

While they typically occur independently of one another, a formal analysis of negation requires the availability of both features for negation, such that both may simultaneously be present in f-structure. The distribution of Negative Concord Items (NCIs), Negative Polarity Items (NPIs) and case-forms licensed by negation also suggests that multiple features must also play an important role in accounting for restrictions on the occurrence of certain forms in antiveridical contexts.

As a lexicalist model of grammar, many facets of the distribution of negative formatives is accounted for by their lexical entry. This is most clearly observed when the presence of one negator places a stipulation on the occurrence of another, or some other marker of non-veridicality.

Acknowledgements

I am grateful to Mary Dalrymple and the three anonymous reviewers for very helpful feedback on earlier versions of this chapter, and to Dávid Győrfi for discussion of Hungarian and Polish.

Abbreviations

Besides the abbreviations from the Leipzig Glossing Conventions, this chapter uses the following abbreviations.

AFF	affirmative
API	Affected Polarity Item
CNEG	constituent negation
ENEG	eventuality negation
EPENT.VWL	epenthetic vowel
FRM.VWL	form vowel
HAB	habitual
JUSS	jussive
MOD	modal
MSA	Modern Standard Arabic
NCI	Negative Concord Item
NEG1	first negative formative in multipartite expression of negation
NEG2	second negative formative in multipartite expression of negation
NegP	negation phrase
NPI	Negative Polarity Item
NSI	Negative Sensitive Item
NVM	non-veridical marker
NW	n-word
POL	polarity
POSTNEG	post verbal negator
SFP	scalar focus particle

References

Aboh, Enoch O. 2010. C-type negation markers on the right edge. In Enoch O. Aboh & James Essegbey (eds.), *Topics in Kwa syntax*, 109–139. Amsterdam: Springer.

Acquaviva, Paolo. 1997. *The logical form of negation: A study of operator-variable structures in syntax*. New York/London: Garland.

Alruwaili, Shatha & Louisa Sadler. 2018. Negative coordination in (Turaif) Arabic. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '18 conference*, 25–45. Stanford: CSLI Publications.

Alsharif, Ahmad. 2014. *Negation in Arabic*. Colchester, UK: University of Essex. (Doctoral dissertation).

Alsharif, Ahmad & Louisa Sadler. 2009. Negation in Modern Standard Arabic: An LFG approach. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '09 conference*, 5–25. Stanford: CSLI Publications.

Ameka, Felix. 1991. *Ewe: Its grammatical constructions and illocutionary devices*. Canberra: Australian National University. (Doctoral dissertation).

Baker, C. Lee. 1970. Double negatives. *Linguistic Inquiry* 1(169–186).

Benmamoun, Elabbas. 2000. *The feature structure of functional categories: A comparative study of Arabic dialects*. Oxford: Oxford University Press.

Bond, Oliver. 2013. A base for canonical negation. In Dunstan Brown, Marina Chumakina & Greville G. Corbett (eds.), *Canonical morphology and syntax*. Oxford: Oxford University Press.

Bond, Oliver. 2016. Negation through reduplication and tone: Implications for the lexical functional grammar/paradigm function morphology interface. *Journal of Linguistics* 52. 277–310.

Bond, Oliver. Forthcoming. Negation. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Borsley, Robert D. & Mohamed Krer. 2012. An HPSG approach to negation in Libyan Arabic. *Essex Research Reports in Linguistics* 61(2). 1–24.

Bresnan, Joan. 2001a. Explaining morphosyntactic competition. In Mark Baltin & Chris Collins (eds.), *Handbook of contemporary syntactic theory*, 11–44. Oxford: Blackwell.

Bresnan, Joan. 2001b. *Lexical-Functional Syntax*. Oxford: Blackwell.

Bresnan, Joan & Sam A. Mchombo. 1995. The lexical integrity principle: Evidence from Bantu. *Natural Language & Linguistic Theory* 13(2). 181–254. DOI: 10.1007/bf00992782.

Brown, Dunstan & Peter Sells. 2016. Archi as a basis for comparing different frameworks. In Oliver Bond, Greville G. Corbett, Marina Chumakina & Dunstan Brown (eds.), *Archi: Complexities of agreement in cross-theoretical perspective*, 233–260. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198747291.003.0008.

Brown, Sue. 1999. *The syntax of negation in Russian: A Minimalist approach*. Stanford: CSLI Publications.

Butt, Miriam, Tracy Holloway King, María-Eugenia Niño & Frédérique Segond. 1999. *A grammar writer's cookbook*. Stanford: CSLI Publications.

Camilleri, Maris & Louisa Sadler. 2017. Negative sensitive indefinites in Maltese. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '17 conference*, 146–166. Stanford: CSLI Publications.

Chung, Inkie. 2007. Suppletive negation in Korean and Distributed Morphology. *Lingua* 117(1). 95–148. DOI: 10.1016/j.lingua.2005.10.002.

Collins, Chris, Paul M. Postal & Elvis Yevudey. 2018. Negative polarity items in Ewe. *Journal of Linguistics* 54. 331–365.

Dahl, Östen. 1979. Typology of sentence negation. *Linguistics* 17. 79–106. DOI: 10.1515/ling.1979.17.1-2.79.

Dalrymple, Mary. 2001. *Lexical Functional Grammar* (Syntax and Semantics 34). New York: Academic Press. DOI: 10.1163/9781849500104.

Dalrymple, Mary, John J. Lowe & Louise Mycock. 2019. *The Oxford reference guide to Lexical Functional Grammar*. Oxford: Oxford University Press. DOI: 10.1093/oso/9780198733300.001.0001.

Dalrymple, Mary & Irina Nikolaeva. 2011. *Objects and information structure* (Cambridge Studies in Linguistics). Cambridge, UK: Cambridge University Press.

De Clercq, Karen. 2020. Types of negation. In Viviane Déprez & Teresa Espinal (eds.), *The Oxford handbook of negation*, 58–74. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780198830528.013.2.

de Haan, Ferdinand. 1997. *The interaction of negation and modality: A typological study*. New York/London: Garland.

de Swart, Henriette. 2009. *Expression and interpretation of negation: An OT typology*. Dordrecht: Springer.

Dryer, Matthew S. 1989. Universals of negative position. In Michael Hammond, Edith A. Moravcsik & Jessica Wirth (eds.), *Studies in syntactic typology*, 93–124. Amsterdam: John Benjamins.

Dryer, Matthew S. 2013. Negative morphemes. In Matthew S. Dryer & Martin Haspelmath (eds.), *The World Atlas of Language Structures Online*. Chapter 112. Leipzig: Max Planck Institute for Evolutionary Anthropology. https://wals.info/chapter/112.

Falk, Yehuda N. 2001. *Lexical-Functional Grammar: An introduction to parallel constraint-based syntax*. Stanford: CSLI Publications.

Frampton, John. 2009. *Distributed reduplication*. Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/9780262013260.001.0001.

Fry, John. 1999. Proof nets and negative polarity licensing. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication), 91–116. Cambridge, MA: The MIT Press.

Giannakidou, Anastasia. 1998. *Polarity sensitivity as (non)veridical dependency*. Amsterdam: John Benjamins. DOI: 10.1075/la.23.

Giannakidou, Anastasia. 2002. Licensing and sensitivity in polarity items: From downward entailment to (non)veridicality. In Mary Andronis, Erin Debenport, Anne Pycha & Keiko Yoshimura (eds.), *Papers from the 38th regional meeting* *of the Chicago Linguistic Society: Parasession on polarity and negation*, 29–54. Chicago: Chicago Linguistic Society.

Giannakidou, Anastasia. 2006. N-words and negative concord. In Martin Everaert & Henk van Riemsdijk (eds.), *The Wiley Blackwell companion to syntax*, vol. 3, 327–391. Malden, MA: Blackwell.

Givón, Talmy. 1979. *On understanding grammar*. New York: Academic Press. DOI: 10.1075/z.213.

Haegeman, Liliane. 1995. *The syntax of negation*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511519727.

Haspelmath, Martin. 1997. *Indefinite pronouns*. Oxford: Oxford University Press. DOI: 10.1093/oso/9780198235606.001.0001.

Haugen, Jason D. 2011. Reduplication in Distributed Morphology. In Jessamyn Schertz, Alan Hogue, Dane Bell, Dan Brenner & Samantha Wray (eds.), *Proceedings of the Arizona Linguistics Circle 4. Coyote Papers Working Papers in Linguistics*, vol. 18. Tucson, AZ: University of Arizona Linguistics Circle. http://hdl.handle.net/10150/143067.

Horn, Laurence R. 2020. Negation and opposition: contradiction and contrariety in logic and language. In Viviane Déprez & Teresa Espinal (eds.), *The Oxford handbook of negation*, 7–25. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780198830528.013.1.

Jackendoff, Ray. 1969. An interpretive theory of negation. *Foundations of Language* 5(2). 218–241.

Jespersen, Otto. 1917. *Negation in English and other languages*. København: A. F. Høst & Søn.

Keenan, Edward L. 1976. Towards a universal definition of subject. In Charles N. Li (ed.), *Subject and topic*, 303–333. New York: Academic Press.

Kim, Jong-Bok. 2000. *The grammar of negation: A constraint-based approach* (Dissertations in Linguistics). Stanford: CSLI Publications.

Kim, Jong-Bok. 2021. Negation. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: the handbook*. Berlin: Language Science Press.

King, Tracy Holloway. 1995. *Configuring topic and focus in Russian*. Stanford: CSLI Publications.

Klima, Edward S. 1964. Negation in English. In Jerry A. Fodor & Jerrold J. Katz (eds.), *The structure of language: Readings in the philosophy of language*, 246–323. Englewood Cliffs, NJ: Prentice Hall.

Krifka, Manfred. 1995. The semantics and pragmatics of polarity items. *Linguistic Analysis* 25. 209–258.

Kupść, Anna & Adam Przepiórkowski. 2002. Morphological aspects of verbal negation in Polish. In Peter Kosta & Jens Frasek (eds.), *Current approaches to formal Slavic linguistics: Proceedings of the Second European Conference on Formal Description of Slavic Languages, Potsdam, 1997*, 337–346. Frankfut am Main: Peter Lang.

Labov, William. 1972. Negative attraction and negative concord in English grammar. *Language* 48. 773–818. DOI: 10.2307/411989.

Laczkó, Tibor. 2014. Essentials of an LFG analysis of Hungarian finite sentences. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 325–345. Stanford: CSLI Publications.

Laczkó, Tibor. 2015. On negative particles and negative polarity in Hungarian. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '15 conference*, 166–186. Stanford: CSLI Publications.

Miestamo, Matti. 2005. *Standard negation: The negation of declarative verbal main clauses in a typological perspective*. Berlin: Mouton de Gruyter.

Neidle, Carol. 1988. *The role of case in Russian syntax*. Dordrecht: Kluwer Academic Publishers. DOI: 10.1007/978-94-009-2703-2.

Niño, María-Eugenia. 1997. The multiple expression of inflectional information and grammatical architecture. In Francis Corblin, Daniéle Godard & Jean-Marie Marandin (eds.), *Empirical issues in syntax and semantics 1*, 127–147. Bern: Peter Lang.

Pak, Majorie. 2019. Logoori grammatical tone: A Distributed Morphology analysis. In *Florida linguistics papers* (Proceedings of the Florida Linguistics Yearly Meeting (FLYM) 5 6), 51–62.

Patejuk, Agnieszka & Adam Przepiórkowski. 2014. Structural case assignment to objects in Polish. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 429–447. Stanford: CSLI Publications.

Payne, John R. 1985. Negation. In Timothy Shopen (ed.), *Language typology and syntactic description*, vol. I: Clause, 197–242. Cambridge, UK: Cambridge University Press.

Penka, Doris. 2010. *Negative indefinites*. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780199567263.001.0001.

Penka, Doris. 2016. Negation and polarity. In Nick Riemer (ed.), *The Routledge handbook of semantics*, 303–319. London: Routledge.

Pollock, J.-Y. 1989. Verb movement, Universal Grammar, and the structure of IP. *Linguistic Inquiry* 20. 365–424.

Przepiórkowski, Adam & Agnieszka Patejuk. 2015. Two representations of negation in LFG: Evidence from Polish. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '15 conference*, 322–336. Stanford: CSLI Publications.

Reinhart, Tanya. 1979. Syntactic domains for semantic rules. In Franz Günthner & Siegfried J. Schmidt (eds.), *Formal semantics and pragmatics for natural languages*. Dordrecht: Reidel.

Repp, Sophie. 2009. *Negation in gapping*. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780199543601.001.0001.

Rolle, Nicholas. 2018. *Grammatical tone: Typology and theory*. Berkeley: University of California, Berkeley. (Doctoral thesis). DOI: 10.5070/bf211040767.

Rowlett, Paul. 1998. *Sentential negation in French*. Oxford: Oxford University Press.

Sells, Peter. 2000. Negation in Swedish: Where it's not at. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '00 conference*, 1–19. Stanford: CSLI Publications.

Stassen, Leon. 1997. *Intransitive predication*. Oxford: Clarendon Press.

Szabolcsi, Anna. 2012. Scope and binding. In Claudia Maienborn, Klaus von Heusinger & Paul Portner (eds.), *Semantics: An international handbook of natural language meaning* (Handbücher zur Sprach- und Kommunikationswissenschaft / Handbooks of Linguistics and Communication Science (HSK), 33/2 2), 1605–1641. Berlin: De Gruyter Mouton.

Toivonen, Ida. 2003. *The phrase structure of non-projecting words: A case study of Swedish particles* (Studies in Natural Language and Linguistic Theory 58). Dordrecht: Kluwer Academic Publishers.

Zanuttini, Raffaella. 1997. *Negation and clausal structure*. Oxford: Oxford University Press.

Zanuttini, Raffaella. 2001. Sentential negation. In Mark R. Baltin & Chris Collins (eds.), *The handbook of contemporary syntactic theory*, 511–535. Malden, MA: Blackwell.

Zwarts, Frans. 1995. Nonveridical contexts. *Linguistic Analysis* 25(3-4). 286–312.

Chapter 2

Pronoun incorporation

Ida Toivonen
Carleton University

In LFG, so-called 'pro-drop' is analyzed as pronoun incorporation, where the person and number marking on the head is the pronoun. The morphology on the head thus serves a dual function: it is an agreement marker when an independent noun or pronoun is present in the clause, and it is an incorporated pronoun when no independent nominal element is present. This chapter spells out the basic analysis of the interplay between pronoun incorporation and agreement marking in LFG. The analysis is illustrated with examples from subject, object, and possessive marking in multiple languages. The chapter also discusses cases where the agreement marker displays markedly different characteristics than the homophonous incorporated pronoun.

1 Introduction

In LFG, *pro-drop* is analyzed as pronoun incorporation. The term pro-drop (from the longer *pronoun/pronominal dropping*) refers to certain instances where a morphologically independent pronoun is not pronounced even though the sentence involves a pronominal interpretation. The pro-drop example in (1) is from Italian, a language that allows subject pro-drop (the example is from Burzio (1986: 92)):

(1) Italian
Ho mangiato bene.
have.1sg eaten well
'I have eaten well.'

English is not a pro-drop language, and pronouns cannot be left unpronounced like subject pronouns can in Italian. However, possible pronoun omission is not

Ida Toivonen. Forthcoming. Pronoun incorporation. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

an all-or-nothing phenomenon. Haegeman (1990) and Weir (2008) discuss the restricted omission of subjects which can occur in certain registers in English (especially so-called 'diary drop'), and Cardinaletti (2014) shows that there is variation within Italian dialects regarding when pronouns can be dropped. The generalization remains that pronouns are omitted quite freely in most languages (e.g., Italian, Arabic, Chicheŵa), although some languages resist it (e.g., English, French).

The Italian example in (1) illustrates what is traditionally called pro-drop, where pronoun omission goes hand-in-hand with rich agreement marking on the verb (see, e.g., Chomsky 1981). The person, number and sometimes gender of the subject is indicated by the morphology on the verb, rendering the independent pronoun in a sense superfluous. This type of pro-drop is analyzed as pronoun incorporation in LFG: the agreement morpheme doubles as an incorporated pronoun.

Section 2 spells out the basics of this incorporation analysis of pro-drop, where the so-called agreement marker is in fact ambiguous between an agreement morpheme and a pronoun. When the independent pronoun is absent ('dropped'), the morpheme is analyzed as a pronoun whose form is morphologically incorporated into the head. When the independent pronoun is present, the morpheme merely agrees with it.

Section 3 provides examples of pro-drop that illustrate the richness of the phenomenon. The term pro-drop is often used to refer exclusively to the omission of a subject pronoun, as in Italian, but the phenomenon is in fact not limited to subjects of finite verbs: any instance of INDEX agreement (Haug forthcoming [this volume], Wechsler & Zlatić 2003) can involve pronoun incorporation.

Section 4 discusses the LFG analysis of pro-drop in light of the standard view of how agreement marking emerges through language change. The section reviews previous work which argues that the standard LFG analysis, positing ambiguity between agreement markers and pronouns, is natural given the grammaticalization path from independent pronoun to bound agreement morpheme.

Section 5 explores ambiguous forms that have grown apart beyond their mere status as pronoun or agreement marker. Many puzzling agreement phenomena from a variety of languages can be explained by the insight that the pronoun/agreement ambiguity assumed in LFG pro-drop analyses can lead to more radical differences between lexical entries that share a form.

Finally, Section 6 turns to a brief discussion of *discourse pro-drop* and *topic drop*. These two types of pro-drop have received less attention in the LFG literature, and, it seems, in the linguistics literature more generally. These types of pro-drop are not tied to rich agreement and therefore tend to be analyzed with different syntactic mechanisms than the Italian-style pro-drop that is the main concern of this chapter.

Reflecting the majority of LFG research concerned with pro-drop, this chapter focuses on the morphosyntactic aspects of pronoun incorporation. However, discourse-pragmatic factors are also highly relevant for a full understanding of the phenomenon. In cases where pro-drop is syntactically optional, the distribution of pronouns is determined by discourse factors. This is illustrated by the Spanish examples in (2) provided by Pešková (2013). In (2a), the independent subject *yo* of the second verb is obligatorily expressed, but in (2b), the inclusion of a subject before the second verb would be infelicitous on the intended interpretation where Pedro is the subject of both verbs:

(2) Spanish

a. Juan habla checo, pero yo hablo eslovaco.
John speak.3SG.PRES.IND Czech but I.NOM speak.1SG.PRES.IND Slovak
'John speaks Czech, but I speak Slovak.'

b. Pedro canta y toca la guitarra.
Peter sing.3SG.PRES.IND and play.3SG.PRES.IND the guitar
'Peter sings and plays the guitar.'

Example (2a) differs from (2b) in that the subject of the second verb in (2a) is a contrastive topic, and contrastive topics are cross-linguistically often marked by emphatic forms or stress. In Spanish and many other languages, pro-drop only occurs when an appropriate antecedent is readily accessible in the discourse context. However, establishing what counts as an appropriate antecedent is nontrivial and seems to vary across languages and dialects (see Alonso-Ovalle et al. 2002; Holmberg 2010; and references provided in those works).

The pragmatic aspect of pro-drop has been addressed within the LFG literature. For example, Dahlstrom (1991: chapters 4–5) shows that Plains Cree independent pronouns are only included when they are used contrastively. A few other LFG proposals that address pro-drop at the discourse-pragmatic level are referred to in Section 6. However, unlike the morphosyntax of incorporation-style pro-drop (the Italian, Spanish and Finnish type), there is no unique analysis of the discourse factors that is uniformly adopted across the LFG community, and the important question of exactly when "optional" pronouns are expressed will therefore not be discussed in detail.

2 Pronoun incorporation and agreement in LFG

The standard analysis[1] of pro-drop in LFG posits that the person and number morphology on the head (which is typically a verb) *is* the pronoun. The "agreement" morphology can thus be thought of as an incorporated pronoun when no corresponding independent pronoun or NP is present in the string. This has been the basic analysis of regular pro-drop in LFG since Fassi Fehri (1984, 1988, 1993) and Bresnan & Mchombo (1987). However, the insight predates Fassi Fehri, Bresnan and Mchombo and indeed the LFG framework. The same underlying idea has long been adopted by some traditional grammarians describing languages with prolific pro-drop. It is, for example, implicity assumed by Ashton (1944), who notes in her Swahili grammar "…in a Bantu language function is more important than form, and one affix often has more than one function" (1944:8).

The formal LFG analysis of pro-drop does not actually involve dropping or deleting a pronoun. There is no phonologically null pronoun present in the phrase structure. There is also no movement involved: the pronominal information is not assumed to have moved into the verbal position in order to be incorporated into the verb.

The separation of constituent structure (c-structure) and functional structure (f-structure) is key to understanding how LFG models pro-drop. C-structure and f-structure concern different aspects of syntactic structure. C-structure is typically modeled using phrase structure trees and displays information about syntactic category (e.g., noun, verb), word order and constituency. F-structure is modeled as feature structures (attribute-value matrices, AVMs) that contain information about formal features such as tense and case. Importantly, LFG also models syntactic functions (e.g., SUBJECT, ADJUNCT) using f-structures.

The basic LFG analysis of pro-drop is described in Haug forthcoming [this volume] and will also be illustrated here with the help of example (3) from Finnish (Finno-Ugric):[2]

(3) Finnish
Join kahvia.
drink.PAST.1SG coffee.PART
'I drank coffee.'

[1] Alternative analyses of pro-drop have been proposed within LFG; see Alsina (2020) for a recent example.

[2] The examples given here are from standard Finnish, which is the variety used in formal settings and in writing. Pro-drop is in fact less common in informal Finnish. Moreover, the discussion here only covers first and second person pronouns; third person pro-drop in Finnish is more constrained (Holmberg Forthcoming).

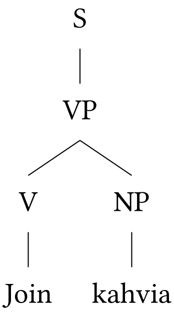
Finnish verbs inflect for three persons and two numbers. The full past tense paradigm for *juoda* 'to drink' is given in (4):

(4) *juoda* 'to drink' (Finnish)

SG	1	*join*
	2	*joit*
	3	*joi*
PL	1	*joimme*
	2	*joitte*
	3	*joivat*

The verb forms provide information about the subject's person and number. In an example like (3), there is no syntactically independent subject. A standard LFG analysis would postulate that the morphological information concerning the subject on the verb *is* the subject. The c-structure and f-structure of (3) are given in (5):

(5) C-structure and f-structure for (3)



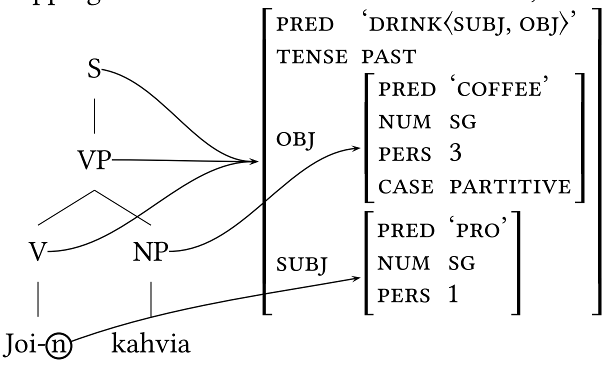
$$\left[\begin{array}{ll} \text{PRED} & \text{`DRINK}\langle\text{SUBJ, OBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{OBJ} & \left[\begin{array}{ll} \text{PRED} & \text{`COFFEE'} \\ \text{NUM} & \text{SG} \\ \text{PERS} & 3 \\ \text{CASE} & \text{PARTITIVE} \end{array}\right] \\ \text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & \text{`PRO'} \\ \text{NUM} & \text{SG} \\ \text{PERS} & 1 \end{array}\right] \end{array}\right]$$

The verb *join* in (3) is not formed in c-structure or f-structure; it is fully formed in the lexicon.[3] The c-structure does not have access to the internal structure of *join*: the terminal nodes in the phrase structure are morphologically complete words.

The mapping between c-structure and f-structure is not necessarily one-to-one, it allows for mismatches. Several f-structures can therefore receive featural information from the same word. In a sentence such as (3), the main f-structure of the sentence (the outer f-structure) and the subject f-structure both receive information from the verb *join*:

[3] The modular architecture of LFG is compatible with different theories of morphology (Dalrymple 2015; Dalrymple et al. 2019: Chapter 12; Bond 2016).

(6) Mapping between c-structure and f-structure, example (3)

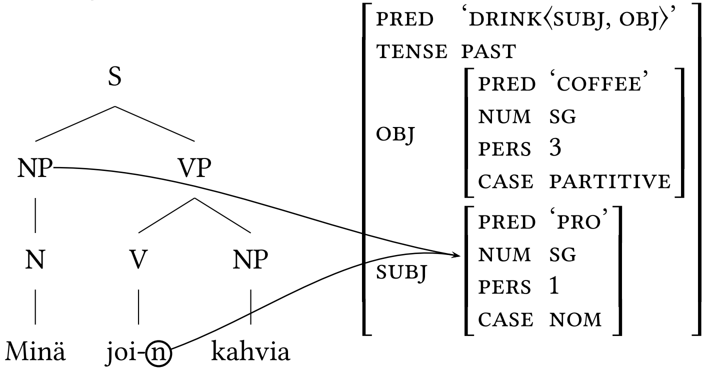


Information from several different words can also map onto the same f-structure. For example, in the Finnish sentence (7), information about the subject comes from both the pronoun *minä* and the agreement morphology on the verb.

(7) Finnish
Minä join kahvia.
I.nom drink.past.1sg coffee.part
‘I drank coffee.’

The c-structure and f-structure for (7) are provided in (8):

(8) Mapping between c-structure and f-structure, example (7)



In sum, in Finnish and other subject pro-drop languages, the pronominal subject information can be provided by the morphology on the verb alone (as in (3)) or from the subject and the verb jointly (as in (7)).

According to the LFG analysis outlined above, the first person singular ending *-n* has a different function in (3) than in (7). In (3), the ending is the pronoun, but in (7) it is a mere agreement marker. In pro-drop languages, the agreement morphology thus doubles as pronominal incorporation. Central to capturing this dual function formally is the PRED feature: pronouns have a PRED feature and agreement markers do not. The PRED feature value is a *semantic form* and is therefore of a different nature than other feature values:[4] it is an indicator of the semantics of the form and it also contains information about its possible argument structure (Bresnan et al. 2016: Chapter 4), although fuller treatment of these aspects are given at the independent grammatical levels of argument structure and semantic structure.[5] The PRED feature also differs from other features in that its value is unique and can therefore not unify with another PRED feature, even if it is identical. This characteristic is crucial for understanding pro-drop in LFG, as will be illustrated below.

The lexical entry for the Finnish first person singular ending -*n* is provided here:

(9) -*n* (↑ SUBJ NUM) = SG
(↑ SUBJ PERS) = 1
((↑ SUBJ PRED) = ‘PRO’)

The first two lines of the lexical entry indicate that the subject of the verb hosting ending -*n* is singular and first person. The third line states that a PRED feature with the value ‘pro’ (a pronominal referential feature) is optionally contributed to the subject. The parentheses indicate the optionality. The optional feature in effect yields two very similar yet not identical lexical entries, one with a PRED feature and one without:

(10) (a) -n_1 (↑ SUBJ NUM) = SG
(↑ SUBJ PERS) = 1
(↑ SUBJ PRED) = ‘pro’

(b) -n_2 (↑ SUBJ NUM)= SG
(↑ SUBJ PERS) = 1

The ending -*n* maps onto the SUBJ f-structure and cannot combine with an independent subject that is not first person singular, as that would violate the LFG principle of *uniqueness*. Uniqueness states that every attribute has a unique value. Since LFG allows feature unification, the -n_2 ending in (10b) can combine its PERS and NUM values with those of the independent pronoun *minä* where there is no

[4] Formal syntactic features such as TENSE and NUM take symbols such as PAST and PLURAL as values. Features can also take feature structures as values. For example, the values of grammatical function attributes (e.g., SUBJ, OBJ) are feature structures. Different types of features are illustrated in the f-structures above.

[5] For more references and discussion of the PRED feature, see Dalrymple et al. (2019: Section 8.2).

feature conflict, but not with those of the pronoun *te* 'you (plural)', for example. The lexical entries for *minä* and *te* are given in (11a) and (11b), respectively:

(11) a. *minä* (↑ PRED) = 'PRO'
(↑ NUM) = SG
(↑ PERS) = 1
(↑ CASE) = NOM

b. *te* (↑ PRED) = 'PRO'
(↑ NUM) = PL
(↑ PERS) = 2
(↑ CASE) = NOM

The second person plural pronoun *te* will not co-occur with the first person singular *-n* because of the mismatch in features. The first person singular *minä* does co-occur with *-n* (see (7), for example). There is no mismatch in PERS or NUM, and *minä* can occur with the agreement marking ending in (10)b. However, as mentioned above, each PRED value is assumed to be unique, and the pronoun *minä* can therefore not map onto the same f-structure as the pronominal *-n* ending in (10)a, which itself contributes a PRED feature. The PRED feature of *minä* would also be 'pro', but since every PRED feature value is unique, the two cannot combine. The single quotes around the semantic form indicate that it is unique. The uniqueness is sometimes also indicated with a subscript notation.

Again, the agreement marker *-n*$_2$ in (10)b can co-occur with *minä*: *-n*$_2$ has no PRED feature that could clash with the PRED feature of *minä*, the PERS and NUM features match and can unify, and the CASE feature is contributed by *minä* alone. In fact, *-n*$_2$ would have to co-occur with some lexical entry in the string contributing a PRED feature, otherwise the f-structure of the sentence would end up containing a SUBJ feature without a PRED. This is only acceptable for syntactic arguments that are not semantic arguments (e.g., expletives). Each semantic argument needs a PRED feature, by the LFG principle of completeness. The following formulation of completeness is provided by Bresnan et al. (2016: 62):

(12) COMPLETENESS:

i. Every function designated by a PRED must be present in the f-structure by that PRED.

ii. If a designator (↑ GF) is associated with a semantic role by the PRED, the f-structure element satisfying the designator must itself contain a semantic feature [PRED *v*].

The features provided by *minä* will map onto the subj function at f-structure by the regular mapping principles between c-structure and f-structure (Bresnan et al. 2016: Chapter 4), and so will the features provided by -n_1. In terms of feature content, the only difference between *minä* and -n_1 is that *minä* has a nominative case feature. The two entries are strikingly different in form: one is an independent word and the other a bound morpheme, and they also differ phonologically. However, the entries are nevertheless almost identical in terms of the feature content they contribute to f-structure. Since both *minä* and -n_1 have a pred feature 'pro', they both function as pronouns, despite the differences in morphophonological realization. The LFG parallel architecture allows for the possibility that forms look different at c-structure but nevertheless have the same function at f-structure.

LFG also allows for mismatches in the other direction: same form, different function. This is illustrated by the ambiguity of the -*n* form. The optionality of the pred feature has an important effect on the function of the -*n* morpheme: -n_1, with a pred 'pro' feature, is a pronoun, and the ending -n_2, without a pred feature, is an agreement marker.

The examples considered concern subjects. The pronominal possessors in standard Finnish also display pro-drop, as illustrated in (13). The possessive suffix (-*ni* for first person singular) is obligatory but the independent pronoun is optional:[6]

(13) Finnish

a. (Minun) auto-ni on vanha.
my car-1sg.Px is old
'My car is old.'

b. *Minun auto on vanha.
my car is old

Just like subject pro-drop, the analysis of possessor pro-drop relies on the pred feature of the possessive suffix -*ni*. The suffix contributes a pred 'pro' feature when it stands alone, and it lacks a pred feature when it is doubled by the independent pronoun *minun*.

Examples of object pro-drop are also attested cross-linguistically. Object pro-drop is common across the Bantu languages, for example (Bresnan & Mchombo 1987; Hualde 1989; Barrett-Keach 1995; Riedel 2009: a. o.). The examples below,

[6] However, there are varieties of Finnish where the example in (13)b, without the suffix, is grammatical.

adapted from Hualde (1989) are from the Bantu language KiRimi (also known as Nyaturu):[7]

(14) KiRimi

a. N-a-kU-on-aa (veve).
1-TNS-OM-saw-FV you
'I saw you.'

b. *N-a-on-aa veve.

Parallel to the Finnish subject and possessor examples above, the object markers that agree with independent pronouns in KiRimi are obligatory, while the independent pronouns themselves are optional. The analysis presented above can be applied in this case as well: the prefix *-kU-* has an optional PRED 'pro' feature and contributes its PRED feature only when *veve* is absent.

3 Pronominal marking across languages

This section explores some of the different ways languages make use of morphology on the head to provide information about dependents. The previous section presented the standard LFG analysis of pro-drop, which rests on the insight that the morpheme on the head has a dual function as an agreement marker and an incorporated pronoun. Of course, this does not mean that agreement morphemes *must* be able to double as pronouns. The English third person singular marker on present tense verbs (*-s* in *Mia walks*) functions solely as an agreement marker, for example. Like English, French does not allow pro-drop, even though French verbs display more detailed subject agreement marking than English, especially in the written forms. The paradigm for the verb *finir* 'to end, to finish' in (15) serves as an illustration:

(15) *finir* 'to end, to finish' (French)

SG	1	*finis*
	2	*finis*
	3	*finit*
PL	1	*finissons*
	2	*finissez*
	3	*finissent*

[7] FV in the gloss stands for ”final vowel”: this “final vowel” in Bantu has received some attention in the literature for reasons not relevant here.

In LFG terms, the subject endings on French and English verbs are mere agreement markers that do not have PRED features, not even optional ones. Rich agreement with no pro-drop is cross-linguistically very rare (Siewierska 1999).

Conversely, an incorporated pronoun does not necessarily double as an agreement marker. For example, Bresnan & Mchombo (1987) argue that object markers in the Bantu language Chicheŵa are unambiguously incorporated pronouns.[8] Chicheŵa object markers are exemplified by the morpheme *chí* in (16) (Bresnan & Mchombo's example (12)):

(16) Chicheŵa
Fîsi anadyá chímanga. Á-tá-chí-dya, anapítá ku San Francîsco.
hyena ate corn(7) he-SERIAL-it(7)-eat he-went to San Francisco
'The hyena ate the corn. Having eaten it, he went to San Francisco.'

The object marker *chí* is specified as noun class seven,[9] and is naturally interpreted as referring back to *corn* in (16). It is possible to also include a free-standing pronoun as in (17) (Bresnan & Mchombo's example (13)) below, but the pronoun is then not interpreted as referring back to the *corn* object from the previous sentence:

(17) Chicheŵa
Fîsi anadyá chímanga. Á-tá-chí-dya icho, anapítá ku San Francîsco.
hyena ate corn(7) he-SERIAL-it(7)-eat it he-went to San Francisco
'The hyena ate the corn. Having eaten it (something other than corn), he went to San Francisco.'

The grammatical object in the second sentence of (17) is the object marker, and the independent pronoun *icho* is a topic anaphorically linked to the object. In Chicheŵa, independent pronouns are used only for introducing new topics or for contrast (Bresnan & Mchombo 1987: 748).

Object markers can also co-occur with NPs headed by non-pronominal nouns:

[8] Bresnan & Mchombo (1987) use the term *anaphoric agreement* for markers that have a pronominal function and *grammatical agreement* for markers that have a mere agreement marking function and no referential properties.

[9] Bantu languages are well-known for their rich noun class (gender) system; see Katamba (2003) for an extensive overview. Chicheŵa has 18 noun classes that are listed in Bresnan & Mchombo (1987: Table 1). Agreement markers and pronouns reflect the class of the noun they agree with or refer to.

(18) Chichewa̐
Njâchi zi-ná-wá-lúm-a alenje.
bees SM-PAST-OM-bite-INDIC hunters
'The bees bit them, the hunters.'

In (18) (Bresnan & Mchombo's example (2)), *alenje* is a floating topic linked to the object marker *wá*, which is an incorporated pronoun. However, if the full NP is a regular object with no special discourse status, the object marker does not appear:

(19) Chichewa̐
Njâchi zi-ná-lúm-a alenje.
bees SM-PAST-bite-INDIC hunters
'The bees bit the hunters.'

The object marker cannot co-occur with a regular object as that would result in a 'PRED clash': they would both contribute a PRED feature value and thus violate the uniqueness principle.

Bresnan & Mchombo (1987) provide ample evidence based on word order, intonation, tonal marking, and other phenomena showing that the Chichewa̐ pronominal object markers differ from subject markers. Chichewa̐ subjects display regular pro-drop. The subject markers are obligatory, unlike object markers. A subject marker can be an agreement marker as in (19) above or an incorporated pronoun as in (20):

(20) Chichewa̐
Zi-ná-lúm-a alenje.
SM(10)-PAST-bite-INDIC hunters
'They bit the hunters.'

The Chichewa̐ data show that different classes of morphemes (subject markers and object markers) can display different pro-drop characteristics within a single language. While the object marker functions as an incorporated pronoun only, the subject marker has a dual function as an agreement marker and a pronoun.

Agreement marking often shows sensitivity to animacy. Specifically, nouns that refer to entities higher on the animacy scale are more likely to trigger agreement. This effect is observed in many Bantu languages (Riedel 2009), for example Swahili (Barrett-Keach 1995) and KiRimi (Hualde 1989). KiRimi object markers agree with animate but not inanimate objects (Hualde 1989).

KiRimi object pro-drop was illustrated in (14) in Section 2, and is further illustrated in (21). The KiRimi examples in (21–22) and (24) are from Hualde (1989).

(21) KiRimi

a. N-a-**mU**-on-aa Maria.
1-TNS-OM-saw-FV Maria
'I saw Maria.'

b. N-a-**mU**-on-aa.
1-TNS-OM-saw-FV
'I saw her.'

Like Chicheŵa subjects, the KiRimi animate object marker has a dual function as an agreement marker (21a) and a pronoun (21b). This is captured here with an optional PRED 'pro' in the lexical entries for animate object markers. Inanimate object markers, on the other hand, cannot co-occur with independent objects:

(22) KiRimi

a. N-a-**ki**-on-aa.
1-TNS-OM-saw-FV

b. *N-a-**ki**-on-aa kItabu.
1-TNS-OM-saw-FV book
'I saw it.'

Inanimate object markers can function as pronouns (22a), but they cannot agree with an object (22b). KiRimi inanimate object markers thus have an obligatory PRED feature, like the object markers in Chicheŵa. The lexical entry for the noun class 7 object marker *-ki-* is given in (23):

(23) *-ki-* (↑ OBJ PRED) = 'PRO'
(↑ OBJ ANIMATE) = –
(↑ OBJ PERS) = 3
(↑ OBJ DEF) = +

The presence of an agreeing object marker further indicates a definite interpretation of the object. This is shown in (24a) and (24b), where the difference in interpretation is indicated by the translation:

(24) KiRimi

a. N-a-**mU**-on-aa mwalimu.
1-TNS-OM-saw-FV teacher
'I saw the teacher.'

b. N-a-on-aa mwalimu.
1-TNS-saw-FV teacher
‘I saw a teacher.’

The object in example (24a) with the object marker receives a definite interpretation, whereas the object in (24b) without an object marker receives an indefinite interpretation. The lexical entry for the noun class 1 object marker *-mU* is provided in (25):

(25) *-mU-* ((↑ OBJ PRED) = ‘PRO’)
(↑ OBJ DEF) = +
(↑ OBJ ANIMATE) = +
(↑ OBJ PERS) = 3

The PRED feature for animate *-mU-* is optional: the feature is present when the object marker is pronominal and absent when the object marker functions as an agreement marker. Both the pronoun and the agreement marker are definite: personal pronouns are in general definite, and the agreement marker ensures a definite interpretation of non-pronominal objects.

The generalizations that KiRimi object markers only double objects that are both definite and animate are captured here with simple lexical specifications and the LFG principle of uniquess. The analysis is straightforward, but it does not explain the fact that the KiRimi facts follow certain cross-linguistic generalizations: Dependents that are definite and high in animacy are cross-linguistically more likely to trigger agreement on the head. We will return to this point in Section 4.[10]

Like KiRimi, Irish shows that there can be differences with respect to pronouns and agreement marking within a single paradigm. However, in Irish, the variation is not governed by definiteness or animacy, the pattern instead seems

[10] An anonymous reviewer points out that there might be noun classes with both animates and inanimates. Hualde (1989) does not address this possibility, but the description of KiRimi noun classes in Olson (1964) indicates that noun classes 9-10 and possibly 12-13 (diminutives) include both animates and inanimates. This is corroborated by Beletskiy & Diyammi’s (2019) notes on noun classes in the closely related dialect/language Isanzu. I have not found a discussion of what the agreement data are in these noun classes. Hualde makes the categorical claim that only definite animates trigger agreement. If this is correct, then each relevant prefix is best represented with two quite different lexical entries and are thus examples of *lexical splits* (discussed below in Section 5). However, Olson (1964: 171) provides a few examples where inanimate objects from class 9 (*gardens, beehive, meat*) cooccur with an object marker. This would indicate that nouns referring to biological inanimates from class 9 carry a grammatical [+ANIMATE] feature. For other examples of misalignment between biological animacy and grammatical animacy, see Bayanati & Toivonen (2019) and references cited therein.

idiosyncratically determined by form. In Irish, some verb forms (synthetic forms) provide person-number information about the subject that other forms (analytic forms) do not. The following conditional paradigm from Ulster Irish is from McCloskey & Hale (1984):

(26) *cuir* 'to put' (Irish)

SG	1	*chuirfinn*
	2	*chuirfeá*
	3	*chuirfeadh sé* (MASC), *chuirfeadh sí* (FEM)
PL	1	*chuirfimis*
	2	*chuirfeadh sibh*
	3	*chuirfeadh siad*

The synthetic forms *chuirfinn, chuirfeá* and *chuirfimis* contain information about the pronominal subjects, but *chuirfeadh* does not. The analytic *chuirfeadh* allows the subject to be expressed independently as a pronoun (*sé, sí, sibh,* or *siadh* in (26)) or a full NP. The synthetic forms cannot co-occur with independent pronouns, as evidenced by the ungrammaticality of (27) from McCloskey & Hale (1984):

(27) Irish:

*Chuirfinn	mé	isteach	ar	an	phost	sin.
put.COND.1SG	I	in			on that job	

'I would apply for that job.' (intended)

The fact that independent subject pronouns are ruled out indicates that the pronominal PRED features in the lexical entries of the synthetic forms *chuirfinn, chuirfeá* and *chuirfimis* are obligatory, unlike the optional subject PRED 'pro' features in Finnish and Chichewâ. The PRED features contributed by the synthetic verb forms cannot unify with the PRED features of independent pronouns. In second person plural and third person singular and plural, however, the verb form does not contain any information about the subject. This information is instead contributed by independent pronouns. For more examples and discussion of variation within Modern Irish, see McCloskey & Hale (1984). For detailed LFG analyses, see Andrews (1990) and Sulger (2010).

This brief overview provides a sample of the variety of patterns that pro-drop languages put on display cross-linguistically. The cross-linguistic differences are captured lexically in LFG: an incorporated pronoun has a PRED 'pro' feature, an agreement marker has no PRED feature, and morphemes that lead a double life as pronouns and agreement markers have an optional PRED feature. The data we have examined here illustrate that languages vary with respect to how they employ these possibilities. The data also illustrate that there can be differences within the same language between paradigms and, perhaps surprisingly, also within paradigms.

For more LFG analyses of pro-drop, drawn from a wide variety of languages and also a variety of types of pro-drop, see Dahlstrom (1991: ch. 5) for Plains Cree subjects and objects, Sadler (1997) for Welsh subject and object clitics, Toivonen (2000, 2001) for Finnish infinitives, Strunk (2004, 2005) for nominal possessive constructions in Low Saxon, Rákosi & Laczkó (2011) for Hungarian spatial particles, Bayram (2013) for Turkish subjects and possessors, Laczkó (2017) for Hungarian possessors, and Dione (2019) for subjects in Wolof.

4 Grammaticalization

A stage where an affix is ambiguous between an agreement marker and a pronoun is unsurprising in light of the typical grammaticalization path of pronoun to agreement marker (Givón 1976; Mithun 1988; Hopper & Traugott 2003; van Gelderen 2011):

(28) independent pronoun > weak pronoun > clitic pronoun > agreement affix > fused agreement marker

The naturalness of pronoun/agreement ambiguities given the grammaticalization cline in (28) has been noted in many previous analyses of pro-drop, including Fassi Fehri (1984); Bresnan & Mchombo (1987); Austin & Bresnan (1996); Toivonen (2001); Morimoto (2002); Butt (2007); Coppock & Wechsler (2010), Barbu & Toivonen (2018) and Haug forthcoming [this volume]. These authors and others have pointed out that when pronouns transition into agreement affixes, there can be a stage where the forms are not immediately reanalyzed as wholesale agreement, but instead are agreement markers when they double an NP and pronouns when they do not.

The grammaticalization cline in (28) conflates multiple linguistic dimensions. One such dimension regards the function: Does the marker have pronominal referential capacity or is it a mere agreement marker? This is modelled at f-structure in LFG. Other dimensions concern the morphophonological realization as an independent word, a clitic, a bound agglutinative morpheme, or a fused morpheme. This is modelled at c-structure, m-structure and p(rosodic)-structure in LFG. A lexical entry can in principle be ambiguous between a pronoun and an agreement marker regardless of its morphophonological realization.

The grammaticalization path in (28) thus conflates common sequences of changes that are often but not always parallel. One sequence concerns c-structural realization:

(29) projecting word > non-projecting word > true clitic > affix > fused affix

A projecting word is a word that projects a phrase and a non-projecting word is a morphologically and phonologically independent word that does not project a phrase. A "true clitic" is here intended to refer to a form that does not project a phrase and is phonologically dependent on a host, but is not a bound morpheme. Projecting words can also be phonologically dependent on a host, which illustrates that prosody has in fact its own relevant dimension which could be separated from (29). Toivonen (2003: 45) provides examples of different types of projecting and non-projecting words and clitics. See also Lowe (2016) for a detailed treatment of clitics in LFG.

Another relevant scale concerns referential capacity:

(30) noun > pronoun > ambiguous pronoun/agreement marker > agreement marker > transitivity marker

The prosodic or phrase-structural realization in (29) is orthogonal to the scale in (30), which is a nominal scale of referential strength. This is modeled here to a large extent with the PRED feature. As seen in the sections above, nouns, pronouns and agreement markers differ in their PRED feature: nouns have a contentful nominal PRED feature, pronouns have the PRED feature 'pro', and agreement markers have no PRED feature at all. A transitivity marker is referentially very weak, as it simply indicates that there *is* an object and does not say anything about what the object refers to.

Changes along the cline in (29) tend to be closely tied to changes along (30). In Siewierska's (1999) survey of 272 languages, most pronouns (forms with obligatory PRED 'pro') are independent words; ambiguous forms (optional PRED) are small words, clitics, or affixes; and pure agreement markers are affixes. However, the scales in (29) and (30) are not inherently connected. This disconnect is carefully argued for in van Rijn (2016), who draws on a sample of personal possessors from 39 different languages. She concludes that "loss of referentiality correlates with a loss in form, but in a relative rather than an absolute sense […] function and form evolve in the same direction, but need not evolve at the same pace" (233).

The insight that function and form can change independently of each other is not difficult to capture within LFG, since the framework models different types of linguistic information at distinct levels such as c-structure, p-structure and f-structure. The changes are also not difficult to formalize, and in fact the directionality of change seems natural within the framework. As explained in Bresnan & Mchombo (1987), the step from pronoun to optional agreement marker is modelled by the PRED feature changing from obligatory to optional. The step from ambiguous pronoun/agreement marker to pure agreement marker is modelled by the loss of the PRED feature. It is important to note, however, that even though this grammaticalization path is naturally modelled formally within LFG, the LFG framework does not dictate the directionality of the change. An explanation for this directionality needs to come from a substantive theory of language change. I will not provide such a theory here, but I will refer to a few insights from the previous literature.

As indicated by the hierarchies above, independent pronouns can be incorporated into the verb. Such a change does not necessarily occur, and it is not predictable exactly when it will occur. However, it is not surprising that such incorporation is common, given the fact that pronouns are typically unstressed and often positioned close to the verb. Pronouns are also often doubled by a full NP or a stressed pronoun, sometimes marked by some special morphology or intonation: *(As for) Carina, I really love her.* It is easy to see how such topic/focus NP + pronoun could come to be reanalyzed as argument NP + agreement marker. For example, recall that Chicheŵa object markers are incorporated pronouns that can double an object that is a discourse topic (Bresnan & Mchombo 1987). The string SUBJECT verb-*pronoun* TOPIC (where the TOPIC and the pronominal OBJECT are co-referential, e.g., (18)) could then in principle easily be reanalzyed as SUBJECT verb-*agreement* OBJECT. Bresnan & Mchombo (1987) indicate that this is precisely what has happened in some other Bantu languages, for example Makua. In light of this, it also makes sense that many agreement markers cross-linguistically agree exclusively with arguments that are high in topicality (Comrie 1981; Woolford 1999; Coppock & Wechsler 2010; Dalrymple & Nikolaeva 2011): it follows from the observation that the pronouns that were reanalyzed as agreement markers originally doubled topics. Since topics tend to be animate (Comrie 1981: 225; Arnold 2013, a.o.), it is also unsurprising that animates are more likely to agree than inanimates.

Other cross-linguistic observations follow from the very fact that agreement markers used to be pronouns. Agreement marking is often restricted to definite or specific arguments (see, e.g., the discussion of Romanian below). Personal pronouns are in general inherently definite and specific, so it is easy to see how such restrictions could remain when the markers lose their pronominal status.

Several cross-linguistic tendencies thus follow from an understanding of the history of agreement marking: agreement can be restricted to topics and to nominals with animate, definite or specific reference. It is important to note that although these generalizations can be readily captured with the LFG formalism, the formalism itself neither predicts nor dictates these tendencies. In LFG, it would be just as easy to formally specify that only indefinites agree in a given language, for example. However, given what research in historical linguistics and psycholinguistics has shown us, it would be unlikely for such a system to emerge.

One further important cross-linguistic generalization concerns the asymmetry between subjects and objects: object agreement marking is less common than subject agreement. In fact, Siewierska (1999) argues that there is no pure object agreement marking. According to Siewierska, apparent examples of object agreement are actually cases of ambiguous marking: the agreement morphemes double as pronouns. Siewierska (1999) offers some possible explanations for this asymmetry, but stresses that those explanations are tentative. In LFG, it is formally no harder to model object agreement than it is to model subject agreement. The forms would simply lack a PRED feature, like the English and French subject agreement markers mentioned in Section 3. The explanation for Siewierska's generalization thus does not come from the LFG formalism.

In general, I assume that insights about language use and change are largely independent of the formal tools that are used to model grammar. However, it is in principle possible to formulate a substantive theory of language change that is compatible with the LFG framework and that might shed light on attested cross-linguistic generalizations.

Up until now, we have mainly focused on the role of the PRED feature. However, other features are also involved and those features can change and erode as well. Coppock & Wechsler (2010) carefully detail the loss of PRED features alongside changes affecting other features such as PERS, NUM, TOPICALITY and DEFINITENESS in different ways in the Finno-Ugric languages Northern and Eastern Ostyak (Khanty) and Hungarian. Toivonen (2001) similarly traces the change of various features that lead to differences in the possessive systems of different dialects of Finnish and Saami. These works trace historical changes that target features other than PRED features, and such changes can lead to differences that reach beyond the PRED feature when a morpheme is at the ambiguous stage. The next section is devoted to examples where the pronominal morpheme is quite different from the agreement marker, even though they are identical in form.

5 Lexical splits

The LFG approach to pro-drop presented above relies on the insight that a form can have a dual function as an agreement marker and an incorporated pronoun. This duality opens the door to the possibility that the morphemes might grow further apart due to language change: since the morphological form corresponds to two similar but distinct lexical entries (one with and one without a PRED feature), the two entries might develop separately. This is in fact cross-linguistically common, and several examples will be given in this section.

One of the first languages for which the LFG theory of pro-drop was developed was Arabic. Abdelkader Fassi-Fehri explored the subject agreement system in Modern Standard Arabic as well as local varieties of Arabic in several talks and papers. Fassi Fehri (1988) shows that some of the affixes are exclusively pronominal (this is the case for the first and second person affixes) and others are ambiguous between pronouns and agreement markers. He further argues that in some cases the pronominal affix is remarkably different from the agreement marking affix, which indicates that their lexical entries differ beyond the PRED feature.

Fassi Fehri's (1988) analysis of feminine subjects in MSA will be reviewed here. Fassi-Fehri shows that the affix *-at* (also sometimes *-ati* in Fassi Fehri's examples) is ambiguous. In its pronominal use, it is a third person feminine singular. However, as an agreement marker, the same affix is less restricted. For example, *-at* (here *-ati*) agrees with a plural subject in (31):

(31) Modern Standard Arabic
ja:-ati l-bana:tu
came-FEM.SG the-girls
'The girls came.'

Fassi Fehri (1988) proposes the lexical entries in (32) for the *-at* affix, indicating that the agreeing affix is only constrained by gender.

(32)

Pronoun:			Agreement:		
(↑ SUBJ PRED)	=	'pro'			
(↑ SUBJ GEND)	=	FEM	(↑ SUBJ GEND)	=	FEM
(↑ SUBJ NUM)	=	SG			
(↑ SUBJ PERS)	=	3			

Fassi Fehri (1988) further proposes that strong forms of pronouns are never directly assigned subcategorized functions in Arabic. Instead, they are always assigned the FOCUS function, which is a grammaticalized discourse function. As such, emphatic pronouns in MSA do not co-occur with the agreement marking version of *-at* even when they are feminine. It would result in a coherence violation: neither the emphatic pronoun nor the agreement marker contributes a PRED feature to the SUBJ.

The *-at* ending can be contrasted with the third person feminine plural affix *-na*, which, unlike *-at*, is a pronoun only and cannot agree:

(33) Modern Standard Arabic

a. ji:-na
came-FEM.PL.HUM
'They came.'

b. *ji:-na l-bana:tu
came-FEM.PL.HUM the-girls

The feminine plural pronoun *-na* can only co-occur with independently expressed nouns when they are topics:

(34) Modern Standard Arabic
al-bana:tu ji:-na
the-girls came-FEM.PL.HUM
'As for the girls, they came.'

In (34), the pronominal affix *-na* is the true subject. The noun *al-bana:tu* is a topic, as evidenced in part by the word order: the unmarked word order in Standard Arabic is VSO. When *al-bana:tu* precedes the verb, *-at* is not felicitous:

(35) Modern Standard Arabic
*al-bana:tu ja:-at
the-girls came-FEM.SG
(intended) 'As for the girls, they came.'

The pronominal *-at* is singular and cannot refer to the plural *al-bana:tu*. The agreement marking *-at* does not contribute a PRED feature. As the TOPIC, the NP *al-bana:tu* also does not contribute a PRED feature to the SUBJ. The agreement marker cannot alone correspond to the SUBJ function, since the subject needs a PRED feature due to the LFG completeness condition, provided in (5a) above. In our specific examples, the verb 'to come' requires a subject with a semantic role, and that subject needs a PRED. In (31), *l-bana:tu* is the subject, and provides the PRED feature. In (33) and (34), the pronominal affix *-na* contributes a pronominal

PRED feature to the SUBJ f-strucure. In (35), *al-bana:tu* provides a PRED feature to the TOPIC function, not the SUBJ function. The agreement marking affix on the verb does not provide a PRED feature at all.

Fassi Fehri (1988) introduces further lexical entries and also specific rules to cover the complex pronominal and agreement system in Standard Arabic. Additional examples accompanied by discussions of computational implementations of Arabic agreement are provided by Hoyt (2004) and Attia (2008). Crucial to the point here is that already one of the first treatments of pro-drop in LFG pointed out that an agreement affix can diverge from a homophonous pronominal affix in features other than just the PRED feature. The agreement marking *-at* differs from the pronominal version of the same form, and Fassi Fehri captures the differences straightforwardly with the lexical entries.

Next we consider so-called 'clitic doubling' in Romanian. In Romanian, objects can be 'doubled' by a morpheme that agrees in person, number and gender. This morpheme is typically referred to as a clitic, but its morphophonological status is controversial (Dobrovie-Sorin 1994; Monachesi 1998; Popescu 2000; Luís 2004). Romanian clitic doubling is exemplified in (36), where the object *pe băiat* is doubled by the clitic *l-*:

(36) Romanian:
L-am văzut pe băiat.
3SG.M.ACC-have.1SG seen ACC boy
'I saw the boy.'

In some dialects of Romanian, all definite objects are doubled (Tomić 2006: Chapter 4; Tomić 2008: 84; Hill 2013, Barbu & Toivonen 2018). This is the case in the Aromanian dialect (spoken in Albania, Macedonia, Romania, Bulgaria, Serbia and Croatia) and the Megleno-Romanian dialect (spoken in Greece and Macedonia). Since the relevant pronouns are inherently definite, these dialects can be analyzed in LFG with an optional PRED feature in the lexical entry for the clitic, just like most of the pro-drop examples discussed above.

However, in other dialects of Romanian, including the standard variety, doubling is restricted to *pe*-marked, human, definite objects. For example, the non-human direct object 'snail' in (37) cannot be doubled by a clitic:

(37) Romanian:

a. Am văzut melcul.
have.1SG seen snail.DEF
'I saw the snail.'

b. *L-am văzut (pe) melc.
3sg.m-have.1sg seen acc snail

The clitic can refer to non-humans when it stands alone. For example, the *l-* in (38) can refer back to *melcul*, the snail:

(38) Romanian:
L-am văzut.
3sg.m.acc-have.1sg seen
'I saw it/him.'

The clitic in (38) could also refer to a human participant.

The restrictions on doubling in this variety of Romanian indicate that the agreement marking clitic and the pronominal clitic differ beyond the presence or absence of the pred feature. Barbu & Toivonen (2018) spell out the details of such an analysis, and their account is summarized here. They follow the Romanian tradition of treating *pe* as an accusative case marker (e.g, Cornilescu 2000) that is specified for human animacy, and they posit the lexical entries in (39) for the pronominal and agreement-marking clitics.

(39)

Pronoun:			Agreement:		
(↑ pred)	=	'pro'			
(↑ pers)	=	α	(↑ pers)	=	α
(↑ num)	=	β	(↑ num)	=	β
(↑ gend)	=	γ	(↑ gend)	=	γ
(↑ case)	=	acc	(↑ case)	$=_c$	acc
(↑ def)	=	+	(↑ def)	=	+

The variables α, β and γ simply stand for different pers, num and gend features that vary according to which form is used: *mă/m-* for first person singular, *te* for second person singular, *îl/l-* for third person singular masculine, etc.

The two entries in (39) only differ very slightly. The pronouns have a pred 'pro' feature and the agreement markers do not, just like we have seen in several examples above. However, there is one small but important further difference: the *case* is specified as a defining equation for the pronoun and a constraining equation for the agreement marker. The regular defining equation of the pronoun directly contributes a [case acc] feature to the object f-structure. The constraining equation requires a [case acc] feature, but does not itself provide it. If the feature is not provided in some other way, the agreement marker is illicit. The marker *pe* provides the acc feature that is needed. This explains why the clitic cannot occur without *pe*. When *pe* functions as a case marker (*pe* has an additional function as the preposition 'on'), it is also specified for human animacy, and this indirectly explains why only objects with human reference can be doubled.

Tigău (2010, 2014) reports that some speakers of Romanian allow clitic doubling with indefinites:

(40) Romanian:

Petru	(l-)a	vizitat	pe	un	prieten.
Peter	3SG.M-have.3SG	visited	ACC	a	friend

'Peter visited a friend.'

Even the speakers who allow doubling with indefinite objects allow it only sometimes. Tigău (2010, 2014) argues that doubled indefinite objects get a *specific* interpretation (see also Aoun 1981: Chapter 3).

The difference between the standard variety of Romanian (captured by (39)) and the indefinite-doubling dialect described by Tigău is captured by the lexical entries in (41):

(41)

Pronoun:			Agreement:		
(↑ PRED)	=	'pro'			
(↑ PERS)	=	α	(↑ PERS)	=	α
(↑ NUM)	=	β	(↑ NUM)	=	β
(↑ GEND)	=	γ	(↑ GEND)	=	γ
(↑ CASE)	=	ACC	(↑ CASE)	$=_c$	ACC
(↑ DEF)	=	+	(↑ SPECIFIC)	=	+

In this dialect, the pronoun is the same as in the standard dialect, but the agreement marker is marked for specificity instead of definiteness.

In two of the dialects of Romanian that have been considered here, the difference between the agreement marking clitic and the pronominal clitic goes beyond the PRED feature. Again, this kind of 'split' is not unexpected under the LFG account of pro-drop, since the optional PRED feature in effect means there are two lexical entries: one agreement marker and one pronoun.

Romanian is not the only Romance language in which the agreement marking clitic and pronominal clitic are markedly distinct. Varieties of Spanish display clitic systems very similar to that of Romanian (see, e.g., Mayer 2017). Andrews (1990) and Estigarribia (2013) analyze Rioplatense Spanish within an LFG framework, and they both propose entries for pronominal clitics that differ from the agreement clitics beyond the PRED feature. Estigarribia specifically proposes that the agreement marker has a specificity feature that the pronominal clitic lacks, which would indicate that Rioplatense Spanish clitics are very similar to the Romanian clitics represented in (41).

Finnish possessive suffixes provide yet another example of 'lexical splits'. Pronominal possessors in standard Finnish are marked by an independent pronoun and a suffix on the possessed noun or by a suffix alone (42):

(42) Finnish
Jukka näkee (minun) ystävä-ni.
J. sees my friend-1sg
'Jukka sees my friend.'

In first and second person, the independent pronoun is optional, and our basic LFG pro-drop analysis can be employed: first and second person possessive suffixes have an optional pred 'pro'.

The optionality of the PRED pro in Finnish possessive suffixes was already mentioned in Section 2. However, the third person suffix displays a more significant split. When a third person independent pronoun is omitted and possession is marked by just a third person suffix, the possessor is necessarily bound by a subject within the minimal finite clause:

(43) Finnish
Jukka$_i$ näkee ystävä-nsä$_{i/*j}$.
J. sees friend-3
'Jukka sees his (own) friend.'

Conversely, when an independent pronoun is present, the possessor *cannot* be bound by a subject:

(44) Finnish
Jukka$_i$ näkee hänen$_{*i/j}$ ystävä-nsä.
J. sees his/her friend-3
'Jukka sees his/her friend.'

In Toivonen's (2000) analysis, the suffix in (43) is an anaphoric pronoun with a pred feature, and the suffix in (44) is an agreement marker without a pred feature. The entries further differ in that the agreement suffix is restricted to agreement with human personal pronouns (45a–45d), even though the pronominal suffix can be bound by both nouns and pronouns with human or non-human referents (45e):

(45) Finnish

a. Jukka näkee Pekan ystävän.
J. sees Pekka's friend.ACC
'Jukka sees Pekka's friend.'

b. *Jukka näkee Pekan ystävä-nsä.
J. sees Pekka's friend-3Px

c. ˙Jukka näkee sen hännän.
J. sees its tail.ACC
'Jukka sees its tail.'

d. *Jukka näkee sen häntää-nsä.
J. sees its tail-3Px

e. Se/koira$_i$ heiluttaa häntää-nsä$_i$.
It/dog wags tail.PART-3Px
'It/the dog is wagging its tail.'

The Finnish pronominal possession system thus provides a further example where pro-drop involves two homophonous but syntactically quite distinct lexical entries: one agreement marker and one pronoun. In the case of Finnish third person possessive suffixes, the pronoun is anaphorically bound and has no animacy retrictions. The agreement marker agrees only with personal, human pronouns that are not anaphorically bound. For a lexical formalization similar to the analyses of Arabic subject markers and Romanian object clitics outlined above, see Toivonen (1996, 2000). For a different analysis, and also more data and references as well as a critique of the LFG analysis, see Humarniemi & Brattico (2015).

The final language we will consider in this section is Pakin Lukunosh Mortlockese. The Mortlockese data and generalizations come from Odango (2014). Odango argues that the third person singular object marker in this Micronesian language shows a split between incorporated pronoun and transitivity marker. He further shows that other object suffixes (the first and second person suffixes and the third person plural suffix) do not involve a split; they function exclusively as incorporated object pronouns (Odango uses the term 'anaphoric agreement', following Bresnan & Mchombo 1987). Example (46) illustrates the second person singular object suffix, which cannot co-occur with an independent pronoun:

(46) Mortlockese
I=aa wor-o-k (*een).
1SG.SBJ-REALIS see-TH-2SG.OBJ 2SG
'I see you.'

The third person singular marker is also an incorporated pronoun when there is no independent object:

(47) Mortlockese

anga-i-tou	mwo
take-3SG.OBJ-downward	please

‘Please take it down.’

The object marker is translated here as *it*, but it can also be translated as *him* or *her*. The pronominal third person singular marker has a PRED feature ‘pro’.

Unlike the other object suffixes, the third person singular suffix can co-occur with an object. When it does, there are no number restrictions on the object. Odango argues that the suffix is a *general transitivity marker* when it co-occurs with an object. In (48), the suffix agrees with a third person plural object:

(48) Mortlockese

Ngaan	i=sán	mwo	shuu-{nge-i/*nge-er}
1SG.EMPH	1SG.SBJ=NEG.POT	yet	meet-TH-3SG.OBJ/TH-3PL.OBJ

mwáán=kewe.
man=DIST.PL

‘As for me, I have not yet met those men.’

Note that the third person plural marker is not admissible in (48), because it functions solely as a pronoun with a PRED ‘pro’ and can therefore not co-occur with the object *mwáán=kewe*.

According to Odango, the transitivity marking suffix is generally limited to third person for many speakers, but some speakers also accept examples where the transitivity marker co-occurs with a first or second person independent pronoun.[11] He provides the following example, which is accepted by some younger speakers:

(49) Mortlockese

R-aa	wér-e-i	kiish.
3PL.SUBJ-REALIS	see-TH-3SG.OBJ	1PL.INCL

‘They see us (incl.).’

For most speakers, however, it seems that the transitivity marker is restricted to third person. Odango (2014) reports on one further restriction on the use of the

[11] The independent pronouns only appear with borrowed verbs and a few verbs that cannot be inflected (Odango 2014).

transitivity marker: it seems to be restricted either for definiteness or specificity. Odango also points to interesting age and geographical variation regarding the exact use of the marker. The variation details are interesting, but will nevertheless be set aside here.

The basic generalization that the third person singular object marker has split into a pronominal suffix and a transitivity marker is clear. Odango (2014) ties his discussion to Bresnan & Mchombo (1987), but he does not provide a formal analysis of Mortlockese. However, the generalizations he provides evidence for can be captured by the following lexical entries for the marker *-i*:

(50)	Pronoun:			Transitivity marker:		
	(↑ OBJ PRED)	=	'pro'	(↑ OBJ DEFINITE)	=	+
	(↑ OBJ PERS)	=	3	(↑ OBJ PERS)	=	3
	(↑ OBJ NUM)	=	SG			

The lexical entries in (50) are tentative but serve to illustrate the relevant lexical split. The pronominal version of the third person singular suffix is straightforward. Since it provides a PRED feature, it cannot co-occur with an independent object. However, the transitivity marker version of the suffix *requires* an independent object. The presence of (↑ OBJ) features ensures the presence of an OBJ function in the f-structure corresponding to the verb that the ending is attached to. This object function needs a PRED feature because of the completeness condition, and this feature is provided by an appropriate object in the c-structure. The lexical entry for the transitivity marker includes a third person object feature. However, for speakers that allow it to co-occur with first and second person pronouns (see (49)), the lexical entry will not include a PERS feature. I assume here that the transitivity marker is specified for definiteness, but Odango hints that it is unclear whether the relevant feature is definiteness or specificity. It is possible that this point is also a matter of speaker variation. In any event, the transitivity marking entry can be modified to include a specificity feature instead of a definiteness feature.

Although the Pakin Lukunosh Mortlockese data involve variations and points to be further investigated, it is clear that the third person singular object marker involves a split. Odango argues that the split is between a pronoun and a transtivity marker. From a historical perspective, the emergence of this split is unsurprising: object markers often grammaticalize into transitivity markers, sometimes via object agreement marking (Lehmann 2002; Mayer 2017; Widmer 2018).

In sum, a pro-drop analysis where an incorporated morpheme is assumed to have a dual function and correspond to both an agreement marker and a pronoun leads to the prediction that the two versions of the morpheme can change independently and grow further apart. This section has considered multiple examples that indicate that such cases do, in fact, occur. The examples we have considered come from Standard Arabic subject marking, Romanian object clitic doubling, Finnish possessive marking, and Mortlockese object marking. In the first three cases, the pronominal version of a morpheme displays different characteristics than the corresponding agreement-marking morpheme. In the Mortlockese case, we adopted Odango's proposal that the non-pronominal version of the third person singular incorporated pronoun is a transitivity marker.

6 Pro-drop without agreement marking

The focus of this chapter has been on cases where information about the dropped arguments is encoded on the head as an incorporated pronoun. However, sometimes pronouns are omitted even though there is no corresponding morphology on the head. This is the case in *discourse pro-drop*. Some LFG work on this type of pro-drop will be briefly reviewed in this section, even though it does not involve morphological pronoun incorporation.

Chinese and Japanese lack morphological agreement marking but nevertheless allow argument omission. A Cantonese example, originally from Luke et al. (2001), is given in (51):

(51) Cantonese (Talking about dogs)
wui5-m4-wui5 beng6 gaa3
will-not-will ill PART
'Would (they = the dogs) get ill?'

This kind of pronoun omission is referred to as *discourse pro-drop* or *radical pro-drop*. Discourse pro-drop is substantially different from pro-drop linked to agreement (Neeleman & Szendrői 2007; Sigurðsson 2011; Irgens 2017),[12] although they occur under similar pragmatic conditions, which are also conditions under which omission of weak pronouns occur in the Germanic languages (Sigurðsson 2011; Rosén 1998: and references therein). Focussing on omitted subjects, Luke et al. (2001) analyze Cantonese discourse pro-drop in LFG. They propose specific discourse-pragmatic criteria to explain how empty subjects receive an interpretation. They also posit an empty subject node in the c-structure, which renders their analysis unusual from a mainstream LFG perspective, where empty

[12] Discourse pro-drop has been argued to in fact resemble general nominal ellipsis more than pro-drop (Irgens 2017).

c-structure material is avoided since it is deemed unnecessary and computationally costly.

Rosén (1998) develops a different LFG analysis for Vietnamese. Vietnamese allows the subject, object and second object (obj$_{\theta}$) to be dropped, even though there is no morphology on the head to indicate the characteristics of the omitted element. Two examples from Rosén (1998: 146) are given in (52):

(52) Vietnamese

a. Ăn ít cỏ lắm.
eat few grass very
'(It) eats very little grass.'

b. Ông Ba tặng một bó hoa hồng hôm nọ.
Mr. Ba give one bunch flower pink day other
'Mr. Ba gave (her) a bunch of roses the other day.'

In Rosén's analysis, the dropped pronouns (*it* and *her* in the examples above) are not represented in the c-structure. In the f-structure, they are represented as the relevant grammatical functions. The pred 'pro' features are contributed by optional equations in the phrase structure rule for S for the subj and the VP rule for obj and obj$_{\theta}$. The f-structures of Vietnamese examples with pro-drop will thus look quite similar to examples where the c-structure does contain expressed pronouns, and also similar to the f-structures of Italian-style pro-drop languages, where other morphology provides the pronominal information. A difference is that the f-structures for the pro-dropped grammatical functions in Vietnamese do not contain person and number information. The key to understanding how empty pronouns assign reference in Vietnamese lies in semantic structure (s-structure) and discourse structure (d-structure), according to Rosén. Like the f-structure information, the semantic schemata needed for the s-structure of the unpronounced pronoun are contributed by the c-structure rules. These schemata include basic semantic information, such as specifications regarding the argument-function mapping.

Rosén (1998) stresses that the interpretation of the dropped pronouns does not depend on guessing. According to Rosén (1998: chapter 7), one condition for pronoun omission is *referential givenness*, meaning the existence of a presupposition of unique reference. Another important condition is *relational givenness:* the intended referent is clear with relation to the verb in context.[13] This is for example

[13] According to Rosén's formal analysis of the discourse conditions, empty pronouns must always be part of the tail value at d-structure, where the tail is understood as the s-structure of the sentence minus the value of the link and the focus.

the case when the verb is the same as in an immediately preceding context. In this case, the participants of the event referred to by the verb remain the same and can be omitted. For example, if someone asks *Did Sarah cook the meat?* and the response repeats the verb *cooked*, no pronouns are included in Vietnamese as it is clear that the participants remain the same. Another example of relational givenness would be *Sarah bought some meat and (she) cooked (it)*, where in the Vietnamese equivalent both the subject and the object pronoun can be omitted. The use of empty pronouns signals that the speaker is sure that the propositional content makes clear which referents to supply for the arguments (Rosén 1998: 137).

Butt & King (1997) show that pro-drop in Hindi/Urdu is not necessarily tied to agreement, and like Rosén, they argue for a discourse-based account. They argue that pronouns can only be omitted if they are continuing topics or backgrounded information, and they model their analysis on the independent linguistic level of i(nformation)-structure.[14] Butt (2007) extends Butt & King's analysis to Punjabi. The analyses developed by Butt & King (2000), Luke et al. (2001) and Rosén (1998) differ significantly from each other, and this indicates that there is room for more (perhaps cross-linguistic) research on discourse pro-drop within LFG. In general, discourse structure has received less attention in LFG than other levels of linguistic representation, but see King & Zaenen (2004); Dalrymple et al. (2018); and references cited in those works for important proposals.

Yet another type of pronoun omission is *topic drop*, which is found in several Germanic languages and illustrated in the Swedish example in (53):

(53) Swedish
Kommer kanske att sakna det.
come perhaps to miss it
'[I/We/They…] will perhaps miss it.'

Swedish verbs bear no agreement and the interpretation of the dropped elements is provided by the context. In these two respects, topic drop is similar to discourse pro drop. However, topic drop is more restricted and only elements in the left periphery of the sentence can be dropped (Neeleman & Szendrői 2007; Sigurðsson & Maling 2008; Sigurðsson 2011). Topic drop has not been treated extensively in LFG, but Berman (1996) provides an analysis of the phenomenon in German.

[14] Rosén (1998) uses the label d(iscourse)-structure and Butt & King (1997) uses i(nformation)-structure to formalize the same type of phenomena. Zaenen forthcoming [this volume] provides a comprehensive overview of LFG research on i-s and d-s in LFG. She reserves the term i-s for sentence-internal information, and d-s for larger units of discourse.

7 Summary

The focus of this chapter has been the LFG theory of pronominal incorporation and the interactions between nouns, independent pronouns, incorporated pronouns, and agreement markers. The analysis of regular pro-drop centers on the person, number and gender marking on the head, which is often ambiguous between an agreement marker and a pronoun. In other words, the marker is an incorporated pronoun, or else it simply agrees with an independent pronoun or noun.

Languages vary with respect to exactly how pronominal information is expressed morphosyntactically, and many different systems have been captured with LFG analyses that take the basic agreement marker-pronoun ambiguity as its starting point. The overview of the literature provided in this chapter illustrates how the typological diversity can be formally understood by appealing to features, feature unification and the mappings between independent linguistic levels.

The LFG theory of pronoun incorporation and pro-drop aligns well with the research on the grammaticalization of pronominal forms and agreement marking. In Section 4 it was argued that although LFG does not technically offer substantive historical explanations, the framework provides formal tools which are suitable for modelling the attested diachronic changes and trends.

The ambiguity between agreement marker and pronoun can give rise to changes that further differentiate between pronominal and agreement morphemes. Such drifts are not uncommon, as illustrated by the examples in Section 5. In a variety of languages, we find agreement morphemes that differ in clear and significant ways from incorporated pronouns. For example, the Finnish third person possessive suffix is restricted to non-anaphoric human personal pronouns in its agreement use, but it has no animacy restrictions and must be anaphorically bound in its pronominal use.

Finally, Section 6 of this chapter briefly reviewed some LFG accounts of pro-drop that do not involve pronominal incorporation or any morphology indicating the person and number of the omitted discourse participant. These cases are interesting for many reasons. First, they illustrate the importance of discourse-pragmatic principles for pronominal interpretation. Second, these cases pose an interesting challenge for the theory of LFG f-structure. The principle of completeness dictates that a semantic arguments needs a PRED feature, and it is not obvious where that feature comes from in cases of discourse pro-drop, where the participant does not have a phonological realization in the linguistic string.

In conclusion, the basic LFG theory of pronominal incorporation and agreement that was first formulated by Fassi Fehri (1984, 1988) and Bresnan & Mchombo (1987) is still adopted today. Over the past four decades, that theory has been used as a tool to gain insight about the details of pro-drop in a large number of languages.

Acknowledgements

I would like to thank three anonymous reviewers for very thorough and insightful comments. I also want to thank Raj Singh, Victoria Rosén, Mary Dalrymple, Roxana-Maria Barbu and Katie Van Luven for their helpful comments on earlier versions of this chapter. The paper has been much improved thanks to the feedback that I have received, but I want to stress that all remaining mistakes and misunderstandings are my own.

References

Alonso-Ovalle, Luis, Charles Clifton, Susana Fernández-Solera & Lyn Frazier. 2002. Null vs. overt pronouns and the topic-focus articulation in Spanish. *Journal of Italian Linguistics* 14(2). 151–169.

Alsina, Alex. 2020. Obligatory clitic expression, clitic omission, and the morphology-syntax interface. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 5–25. Stanford: CSLI Publications.

Andrews, Avery D. 1990. Unification and morphological blocking. *Natural Language & Linguistic Theory* 8(4). 507–557. DOI: 10.1007/bf00133692.

Aoun, Joseph. 1981. *The formal nature of anaphoric relations*. Cambridge, MA: Massachusetts Institute of Technology. (Doctoral dissertation).

Arnold, Jennifer. 2013. Information status relates to production, distribution, and comprehension. *Frontiers in Psychology* 4. 235. DOI: 10.3389/fpsyg.2013.00235.

Ashton, Ethel O. 1944. *Swahili grammar (including intonation)*. London: Longmans, Green & Co.

Attia, Mohammed. 2008. Alternate agreement in Arabic. Presented at the Parallel Grammar (ParGram) Meeting, Istanbul, Turkey.

Austin, Peter K. & Joan Bresnan. 1996. Non-configurationality in Australian aboriginal languages. *Natural Language & Linguistic Theory* 14. 215–268. DOI: 10.1007/bf00133684.

Barbu, Roxana-Maria & Ida Toivonen. 2018. Romanian object clitics: Grammaticalization, agreement and lexical splits. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '18 conference*, 67–87. Stanford: CSLI Publications.

Barrett-Keach, Camillia. 1995. Subject and object markers as agreement and pronoun incorporation in Swahili. In Akinbiyi Akinlabi (ed.), *Theoretical approaches to African linguistics* (Trends in African Linguistics 1), 109–116. Trenton, NJ: Africa World Press.

Bayanati, Shiva & Ida Toivonen. 2019. Humans, animals, things and animacy. *Open Linguistics* 5. 156–170. DOI: 10.1515/opli-2019-0010.

Bayram, Faith. 2013. *The acquisition of Turkish by heritage speakers: A processability approach*. Newcastle: Newcastle University. (Doctoral dissertation).

Beletskiy, Stanislav & Mark Paul Diyammi. 2010. A phonological and morphological sketch of Isanzu lect (Bantu, Tanzania). *Journal of Siberian Federal University, Humanities & Social Sciences* 12(1). 4–19. DOI: 10.17516/1997-1370-0376.

Berman, Judith. 1996. Topicalization vs. left dislocation of sentential arguments in German. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '96 conference*. Stanford: CSLI Publications.

Bond, Oliver. 2016. Negation through reduplication and tone: Implications for the lexical functional grammar/paradigm function morphology interface. *Journal of Linguistics* 52. 277–310.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2016. *Lexical-Functional Syntax*. 2nd edn. (Blackwell Textbooks in Linguistics 16). Malden, MA: Wiley-Blackwell.

Bresnan, Joan & Sam A. Mchombo. 1987. Topic, pronoun, and agreement in Chicheŵa. *Language* 63(4). 741–782. DOI: 10.2307/415717.

Burzio, Luigi. 1986. *Italian syntax: A Government-Binding approach*. Dordrecht: Reidel.

Butt, Miriam. 2007. The role of pronominal suffixes in Punjabi. In Annie Zaenen, Jane Simpson, Tracy Holloway King, Jane Grimshaw, Joan Maling & Christopher Manning (eds.), *Architectures, rules, and preferences: Variations on themes by Joan W. Bresnan*. Stanford: CSLI Publications.

Butt, Miriam & Tracy Holloway King. 1997. Null elements in discourse structure. Unpublished manuscript intended for publication in the Proceedings of the NULLS Seminar. http://ling.uni-konstanz.de/pages/home/butt/main/papers/nulls97.pdf.

Cardinaletti, Anna. 2014. Cross-linguistic variation in the syntax of subjects. In M. Carme Picallo (ed.), *Linguistic variation in the Minimalist framework*, 82–107. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198702894.003.0005.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris Publications. DOI: 10.1515/9783110884166.

Comrie, Bernard. 1981. *Language universals and linguistic typology*. Chicago: University of Chicago Press.

Coppock, Elizabeth & Stephen Wechsler. 2010. Less-travelled paths from pronoun to agreement: The case of the Uralic objective conjugations. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '10 conference*, 165–185. Stanford: CSLI Publications.

Cornilescu, Alexandra. 2000. Notes on the interpretation of the prepositional accusative. *Bucharest Working Papers in Linguistics* 2(1). 91–106.

Dahlstrom, Amy. 1991. *Plains Cree morphosyntax*. Garland.

Dalrymple, Mary. 2015. Morphology in the LFG architecture. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '15 conference*, 64–83. Stanford: CSLI Publications.

Dalrymple, Mary, Dag T. Haug & John J. Lowe. 2018. Integrating LFG's binding theory with PCDRT. *Journal of Language Modelling* 6(1). 87–129. DOI: 10.15398/jlm.v6i1.204.

Dalrymple, Mary, John J. Lowe & Louise Mycock. 2019. *The Oxford reference guide to Lexical Functional Grammar*. Oxford: Oxford University Press. DOI: 10.1093/oso/9780198733300.001.0001.

Dalrymple, Mary & Irina Nikolaeva. 2011. *Objects and information structure* (Cambridge Studies in Linguistics). Cambridge, UK: Cambridge University Press.

Dione, Cheikh M. Bamba. 2019. Clause structure, pro-drop and control in Wolof: An LFG/XLE perspective. *Nordic Journal of African Studies* 28(3). 1–26.

Dobrovie-Sorin, Carmen. 1994. *The syntax of Romanian: comparative studies in Romance*. Berlin/New York: Walter de Gruyter. DOI: 10.1515/9783110886597.

Estigarribia, Bruno. 2013. Rioplatense Spanish clitic doubling and "tripling" in Lexical-Functional Grammar. In Chad Howe, Sarah E. Blackwell & Margaret Lubbers Quesada (eds.), *Selected proceedings of the 15th Hispanic Linguistics Symposium*, 297–309. Somerville, MA: Cascadilla Proceedings Project.

Fassi Fehri, Abdelkader. 1984. Agreement in Arabic, binding and coherence. Presented at the Conference on Agreement in Natural Language, Stanford University.

Fassi Fehri, Abdelkader. 1988. Agreement in Arabic, binding and coherence. In Michael Barlow & Charles A. Ferguson (eds.), *Agreement in natural language*, 107–158. Stanford: CSLI Publications.

Fassi Fehri, Abdelkader. 1993. *Issues in the structure of Arabic clauses and words*. Dordrecht: Kluwer Academic Publishers. DOI: 10.1007/978-94-017-1986-5.

Givón, Talmy. 1976. Topic, pronoun and grammatical agreement. In Charles N. Li (ed.), *Subject and topic*, 151–188. New York: Academic Press.

Haegeman, Liliane. 1990. Understood subjects in English diaries. *Multilingua* 9(2). 157–199.

Haug, Dag. Forthcoming. Agreement. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Hill, Virginia. 2013. The direct object marker in Romanian: A historical perspective. *Australian Journal of Linguistics* 33(2). 140–151. DOI: 10.1080/07268602.2013.814527.

Holmberg, Anders. 2010. Null subject parameters. In Theresa Biberauer, Anders Holmberg, Ian Roberts & Michelle Sheehan (eds.), *Parametric variation: Null subjects in minimalist theory*, 88–124. Cambridge, UK: Cambridge University Press.

Holmberg, Anders. Forthcoming. Null subjects in Finnish and the typology of pro-drop. In Anne Tamm & Anne Vainikka (eds.), *Uralic syntax*, 1–30. Cambridge, UK: Cambridge University Press.

Hopper, Paul J. & Elizabeth Closs Traugott. 2003. *Grammaticalization*. 2nd edn. (Cambridge Textbooks in Linguistics). Cambridge, UK: Cambridge University Press. DOI: 10.1017/CBO9781139165525.

Hoyt, Frederick. 2004. Subject-verb agreement in Modern Standard Arabic: An LFG implementation in the Xerox Language Engineering Environment. Unpublished manuscript, University of Texas, Austin.

Hualde, José Ignacio. 1989. Double object constructions in KiRimi. In Paul Newman & Robert Botne (eds.), *Current approaches to African linguistics 5*, 179–190. Dordrecht: Foris Publications. DOI: 10.1515/9783112420089-014.

Humarniemi, Saara & Pauli Brattico. 2015. The development of bound pronominal paradigms. *Finno-Ugric Languages and Linguistics* 9(1–2). 2–41.

Irgens, Benedicte Mosby. 2017. *Person deixis in Japanese and English: A contrastive functional analysis*. Bergen: University of Bergen. (Doctoral dissertation).

Katamba, Francis. 2003. Bantu nominal morphology. In Derek Nurse & Gérard Philippson (eds.), *The Bantu languages*, 103–120. London: Routledge.

King, Tracy Holloway & Annie Zaenen. 2004. F-structures, information structure, and discourse structure. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '04 conference*. Stanford: CSLI Publications.

Laczkó, Tibor. 2017. Modelling (in)definiteness, external possessors and (typological) variation in Hungarian possessive DPs. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '17 conference*, 243–263. Stanford: CSLI Publications.

Lehmann, Christian. 2002. *Thoughts on grammaticalization*. 2nd edn. Erfurt: Arbeitspapiere des Seminars für Sprachwissenschaft der Universität Erfurt.

Lowe, John J. 2016. Clitics: Separating syntax and prosody. *Journal of Linguistics* 52(2). 375–419.

Luís, Ana. 2004. *Clitics as morphology*. Colchester, UK: University of Essex. (Doctoral dissertation).

Luke, Kang Kwong, Adams Bodomo & Owen Nancarrow. 2001. The subject condition in Cantonese. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '01 conference*, 359–379. Stanford: CSLI Publications.

Mayer, Elisabeth. 2017. *Spanish clitics on the move: Variation in time and space*. Berlin: De Gruyter Mouton. DOI: 10.1515/9781614514213.

McCloskey, James & Kenneth Hale. 1984. On the syntax of person-number inflection in Modern Irish. *Natural Language & Linguistic Theory* 1. 487–533.

Mithun, Marianne. 1988. The development of bound pronominal paradigms. In Winfred P. Lehmann & Helen-Jo Jakusz Hewitt (eds.), *Language Typology 1988: Typological models in the service of reconstruction*, 85–104. Amsterdam: John Benjamins. DOI: 10.1075/cilt.81.07mit.

Monachesi, Paola. 1998. The morphosyntax of Romanian cliticization. In P. Coppen, H. van Halteren & L. Teunissen (eds.), *Proceedings of Computational Linguistics in the Netherlands 1997*, 99–118.

Morimoto, Yukiko. 2002. Prominence mismatches and differential object marking in Bantu. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '02 conference*, 292–314. Stanford: CSLI Publications.

Neeleman, Ad & Kriszta Szendrői. 2007. Radical pro drop and the morphology of pronouns. *Linguistic Inquiry* 4(38). 671–714. DOI: 10.1162/ling.2007.38.4.671.

Odango, Emerson Lopez. 2014. The status of subject and object markets in Pakin Lukunosh Mortlockese. *Oceanic Linguistics* 53(1). 110–135. DOI: 10.1353/ol.2014.0004.

Olson, Howard S. 1964. *The phonology and morphology of Rịmi*. Hartford, CT: Hartford Seminary. (Doctoral dissertation).

Pešková, Andrea. 2013. Experimenting with *pro-drop* in Spanish. *SKY Journal of Linguistics* 26. 117–149.

Popescu, Alexandra. 2000. The morphophonology of the Romanian clitic sequence. *Lingua* 110. 773–799. DOI: 10.1016/s0024-3841(00)00016-4.

Rákosi, György & Tibor Laczkó. 2011. Inflecting spatial particles and shadows of the past in Hungarian. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '11 conference*, 440–460. Stanford: CSLI Publications.

Riedel, Kristina. 2009. *The syntax of object marking in Sambaa: A comparative Bantu perspective*. Utrecht: LOT – Netherlands Graduate School of Linguistics. (Doctoral dissertation).

Rosén, Victoria. 1998. *Topics and empty pronouns in Vietnamese*. Bergen: University of Bergen. (Doctoral dissertation).

Sadler, Louisa. 1997. Clitics and the structure-function mapping. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '97 conference*. Stanford: CSLI Publications.

Siewierska, Anna. 1999. From anaphoric pronoun to grammatical agreement marker: Why objects don't make it. *Folia Linguistica* 33(2). 225–252. DOI: 10.1515/flin.1999.33.1-2.225.

Sigurðsson, Halldór Ármann. 2011. Conditions on argument drop. *Linguistic Inquiry* 2(42). 267–304. DOI: 10.1162/ling_a_00042.

Sigurðsson, Halldór Ármann & Joan Maling. 2008. Argument drop and the Empty Left Edge Condition. *Working Papers in Scandinavian Syntax* 81. 1–27.

Strunk, Jan. 2004. *Possessive constructions in Modern Low Saxon*. Stanford: Stanford University. (MA thesis).

Strunk, Jan. 2005. Pro-drop in nominal possessive constructions. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '05 conference*, 447–467. Stanford: CSLI Publications.

Sulger, Sebastian. 2010. Analytic and synthetic verb forms in Irish – An agreement-based implementation in LFG. In Manfred Pinkal, Ines Rehbein, Sabine Schulte im Walde & Angelika Storrer (eds.), *Semantic approaches in natural language processing: Proceedings of the Conference on Natural Language Processing 2010*, 169–174. Saarbrücken: Saarland University Press.

Tigău, Alina Mihaela. 2010. Towards an account of differential object marking in Romanian. *Bucharest Working Papers in Linguistics* 12(1). 137–158.

Tigău, Alina Mihaela. 2014. Specificity effects with clitic doubling and *pe* marking. *Bucharest Working Papers in Linguistics* 16(1). 43–62.

Toivonen, Ida. 1996. Finnish possessive suffixes in Lexical-Functional Grammar. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '96 conference*. Stanford: CSLI Publications.

Toivonen, Ida. 2000. The morphosyntax of Finnish possessives. *Natural Language & Linguistic Theory* 18(3). 579–609.

Toivonen, Ida. 2001. Language change, lexical features and Finnish possessors. In Miriam Butt & Tracy Holloway King (eds.), *Time over matter: Diachronic perspectives on morphosyntax*, 209–225. Stanford: CSLI Publications.

Toivonen, Ida. 2003. *The phrase structure of non-projecting words: A case study of Swedish particles* (Studies in Natural Language and Linguistic Theory 58). Dordrecht: Kluwer Academic Publishers.

Toivonen, Ida. Forthcoming. Pronoun incorporation. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Tomić, Olga Mišeska. 2006. *Balkan sprachbund morphosyntactic features* (Studies in Natural Language and Linguistic Theory 67). Dordrecht: Springer. DOI: 10.1007/1-4020-4488-7.

Tomić, Olga Mišeska. 2008. Towards grammaticalization of clitic doubling: Clitic doubling in Macedonian and neighbouring languages. In Dalina Kallulli & Liliane Tasmowski (eds.), *Clitic doubling in the Balkan languages*, 65–87. Amsterdam/Philadelphia: John Benjamins. DOI: 10.1075/la.130.06mis.

van Gelderen, Elly. 2011. *The linguistic cycle: Language change and the language faculty*. Oxford: Oxford University Press.

van Rijn, Marlou. 2016. The grammaticalization of possessive person marking: A typological approach. *Transactions of the Philological Society* 114(2). 233–276. DOI: 10.1111/1467-968x.12077.

Wechsler, Stephen & Larisa Zlatić. 2003. *The many faces of agreement*. Stanford: CSLI Publications.

Weir, Andrew. 2008. *Subject pronoun drop in informal English*. Edinburgh: University of Edinburgh. (MA thesis).

Widmer, Manuel. 2018. Transitivity markers in West Himalayish: Synchronic and diachronic considerations. *Linguistics of the Tibeto-Burman Area* 41(1). 75–105. DOI: 10.1075/ltba.17011.wid.

Woolford, Ellen. 1999. Animacy hierarchy effects on object agreement. In Paul Ketey (ed.), *New dimensions in African linguistics* (Trends in African Linguistics 3), 109–116. Trenton, NJ: Africa World Press.

Zaenen, Annie. Forthcoming. Information structure. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Chapter 3

Formal and computational properties of LFG

Ronald M. Kaplan
Stanford University

Jürgen Wedekind
University of Copenhagen

This chapter first reviews the basic architectural concepts that underlie the formal theory of Lexical-Functional Grammar. The LFG formalism provides a simple set of devices for describing the common properties of all human languages and the particular properties of individual languages. It postulates two levels of syntactic representation for a sentence, a constituent structure and a functional structure. These are related by a piecewise correspondence that permits the abstract functional structure to be described in terms of configurations of constituent structure phrases. We then survey the mathematical and computational properties of this simple framework. We demonstrate that the recognition/parsing, realization/generation, emptiness, and other more specific decision problems are unsolvable for grammars in the unrestricted LFG formalism. A first set of restrictions guarantees decidability of recognition, realization, and other problems for grammars that are still suitable for linguistic description, but the solutions to these problems in the worst case are computationally impractical. The class of LFG grammars that meet an additional set of restrictions is equivalent to the class of mildly context-sensitive grammars, and the recognition and realization problems for grammars in this class are thus not only decidable but tractable as well.

1 Introduction

The basic features of the LFG formalism are quite simple and have remained remarkably stable since they were first introduced by Kaplan & Bresnan (1982). An

Ronald M. Kaplan & Jürgen Wedekind. Forthcoming. Formal and computational properties of LFG. in Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

LFG grammar assigns to each sentence in its language at least one constituent structure (c-structure) and at least one functional structure (f-structure). The c-structure is a phrase-structure tree that represents the order of words and their grouping into phrases. The f-structure is a hierarchical attribute-value matrix that represents the underlying grammatical relations that are expressed by configurations of c-structure nodes. The c-structure is determined in the traditional way by the rules of a context-free grammar. The f-structure is a minimal model for the functional description (f-description) that is constructed from annotations associated with the categories of rules that license the nodes of the c-structure. The f-description is obtained by instantiating those annotations on the assumption that there is a piece-wise correspondence ϕ between the nodes of the c-structure and the units of a satisfying f-structure.

This simple correspondence architecture still lies at the core of LFG theory even as it has been extended and refined to provide more insightful accounts of long distance dependencies (Kaplan & Zaenen 1989), coordination (Kaplan & Maxwell 1988), and other syntactic phenomena. In this chapter we focus on the mathematical and computational properties of the basic formalism. As is well known, its expressive power goes far beyond the capabilities of the context-free c-structure grammar. This is because the annotations may associate information that originates from different (and possibly arbitrarily distant) nodes with the same f-structure unit. The result is that such a unit must satisfy requirements that come from words in the string or nodes in the tree that do not stand in a local mother-daughter relationship. A string with an otherwise well-formed c-structure is excluded from the language if such context-sensitive f-structure requirements are inconsistent. We know that some degree of context sensitivity is needed for recognizing and parsing natural languages (Bresnan et al. 1982; Culy 1985; Shieber 1985), but the basic LFG formalism may allow for more expressive power than is actually required.

Indeed, Kaplan & Bresnan (1982) used a reduction from the Turing machine halting problem to show that the recognition/parsing problem is undecidable for unrestricted LFG grammars (see also Johnson 1988). This is the computationally important problem of determining whether or not a given string belongs to the language of the grammar and is assigned at least one c-structure and corresponding f-structure. Wedekind (2014) proved the undecidability of the realization problem, also of practical significance. This is the problem of determining whether the language contains at least one string to which an arbitrary given f-structure is assigned. Wedekind's undecidability proof used a reduction from the emptiness problem for the intersection of context-free languages. He also used that reduction to show the undecidability of the emptiness problem for unrestricted LFGs

(Wedekind 1999). This is the problem of determining whether or not there are any strings at all in the language of a given LFG grammar. The emptiness problem for LFGs was previously shown to be undecidable by reductions from Hilbert's Tenth Problem (Roach 1983) and Post's Correspondence Problem (Nishino 1991).

We revisit these undecidability results in Section 4. We provide alternative proofs within a single, conceptually simple, framework. In Appendix A we use this framework to show that other more specific decision problems are also unsolvable.

We consider in Section 5 some formal conditions that are sufficient to guarantee decidability of the recognition and realization problems. Kaplan & Bresnan (1982) showed that recognition is decidable if c-structures with non-branching dominance (NBD) chains and/or unlimited empty nodes are excluded, and they argued that this is a reasonable restriction for LFG grammars that describe natural languages. This parsing-oriented limitation does not reduce the complexity of generation (Wedekind 2014), but an unrelated restriction has been shown to ensure the decidability of that problem (Wedekind & Kaplan 2012). This raises the question whether there is a single, linguistically plausible, condition that applies indifferently to both parsing and generation. We introduce in Section 5 such a uniform condition, proper anchoring, but we also demonstrate that this particular condition is not strong enough to guarantee that these problems can be solved with practical efficiency. In the worst case recognition and generation may take an amount of time that is exponential in the length of an input sentence or f-structure.

This leads us to examine in Section 7 a stronger set of restrictions that not only guarantee decidability of recognition and realization as well as emptiness but also ensure that those problems can be solved in polynomial time. This follows from the fact that LFG grammars that meet these additional restrictions are mildly context-sensitive in their expressive power and thus also have the known mathematical and computational properties of that class of formal grammars.

2 Basic LFG formalism

We show in Figure 1 the c-structure and f-structure that the annotated c-structure rules in (1) and lexical entries in (2) would assign to the sentence *He sees the girl*.

(1)
$$\mathrm{S} \rightarrow \begin{array}{cc} \mathrm{NP} & \mathrm{VP} \\ (\uparrow \textsc{subj}) = \downarrow & \uparrow = \downarrow \\ & (\uparrow \textsc{tense}) \end{array}$$

$$\mathrm{NP} \rightarrow \begin{array}{cc} (\mathrm{Det}) & \mathrm{N} \\ \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

$$\mathrm{VP} \rightarrow \begin{array}{cc} \mathrm{V} & \mathrm{NP} \\ \uparrow = \downarrow & (\uparrow \textsc{obj}) = \downarrow \end{array}$$

(2)

he	N	$(\uparrow \textsc{pred}) = \text{‘}\textsc{pro}\text{’}$ $(\uparrow \textsc{agr pers}) = 3$ $(\uparrow \textsc{agr num}) = \textsc{sg}$	*the*	Det	$(\uparrow \textsc{spec}) = \textsc{def}$
sees	V	$(\uparrow \textsc{pred}) = \text{‘}\textsc{see}\langle \textsc{subj obj} \rangle\text{’}$ $(\uparrow \textsc{tense}) = \textsc{pres}$ $(\uparrow \textsc{subj agr pers}) = 3$ $(\uparrow \textsc{subj agr num}) = \textsc{sg}$	*girl*	N	$(\uparrow \textsc{pred}) = \text{‘}\textsc{girl}\text{’}$ $(\uparrow \textsc{agr pers}) = 3$ $(\uparrow \textsc{agr num}) = \textsc{sg}$ $(\uparrow \textsc{spec})$

The correspondence function ϕ is indicated by the arrows between the c-structure nodes and the f-structure units and also, redundantly, by the columns of node identifiers *root*, $n_1, n_2, \ldots$ attached to the f-structure units. We see even in this simple example that the function ϕ is typically many-to-one (heads and coheads of grammatical constituents are mapped into the same f-structure) but is not onto (the AGR/agreement f-structure units are not the image of any node). The function ϕ may also be partial, if nodes necessary for c-structure well-formedness have no f-structure significance.

The phrasal categories of this c-structure obviously meet the node admissibility conditions of the annotated rewriting rules (1). Lexical entries are interpreted also as annotated rewriting rules that relate the lexical categories of the c-structure to the words of the sentence. The entry for *the*, for example, is interpreted as the rule

(3)
$$\mathrm{Det} \rightarrow \begin{array}{c} \mathrm{the} \\ (\uparrow \textsc{spec}) = \textsc{def} \end{array}$$

and the normal node admissibility conditions also license the proper lexical expansions for the tree.

The description that the f-structure must satisfy is constructed from the annotations associated with the daughter categories of the rules that license particular nodes in the c-structure. Each side of an equation designates an element of a corresponding f-structure, and the equation is satisfied if both sides designate the same element. The metavariable ↓ in an annotation designator instantiates to the

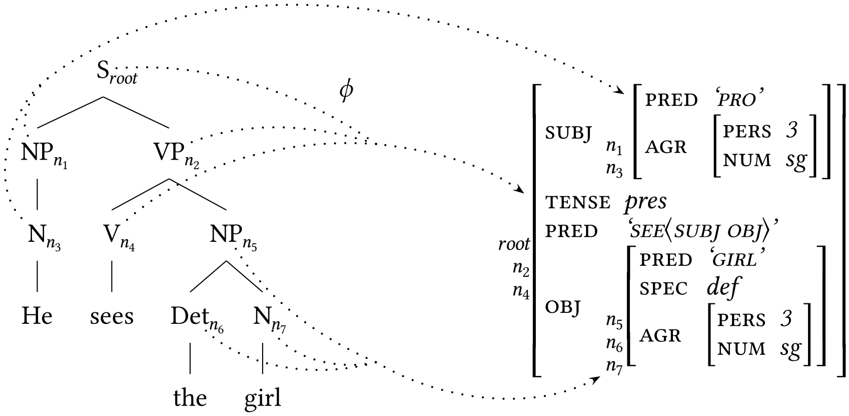


Figure 1: Illustration of the basic LFG architecture: A c-structure c and f-structure f related by the correspondence function ϕ from the nodes of c to the units of f. The f-structure units are indexed by the nodes to which they correspond.

f-structure corresponding to the node that matches the associated rule category (n_1 for ↓ in the annotation $(\uparrow \textsc{subj}) = \downarrow$ attached to the NP in the S rule), and the metavariable ↑ denotes the f-structure corresponding to the mother of that node (the node *root* for that rule). To be precise, if $*$ instantiates to the matching node and $M(*)$ instantiates to its mother, then ↓ and ↑ are abbreviations for $\phi(*)$ and $\phi(M(*))$ respectively. The metavariable instantiations are easy to read from the *annotated c-structure* in Figure 2. This is a phrase-structure tree whose nodes are labeled with the category-annotation pairs that appear in grammar rules and lexical entries.

The first NP is identified as n_1 and its mother is *root*, so the annotation $(\uparrow \textsc{subj}) = \downarrow$ instantiates directly to $(\phi(root)\ \textsc{subj}) = \phi(n_1)$. Since a parenthesized designator denotes the element reached by traversing a path of attributes from a starting f-structure, the f-structure in Figure 1 satisfies this equation because $\phi(n_1)$ is the SUBJ of $\phi(root)$ under the illustrated ϕ correspondence. The full f-description for this annotated c-structure is the conjunction of instantiated equations collected from all of its nodes, shown in (4).

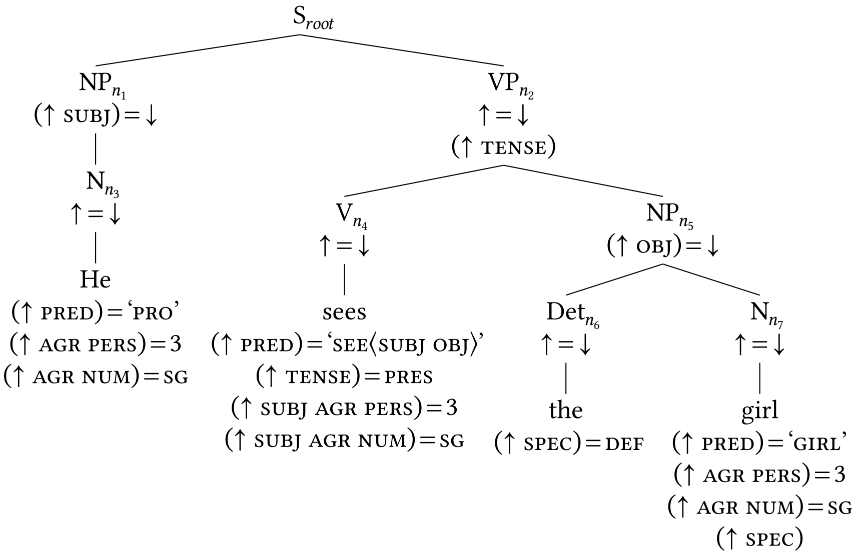


Figure 2: Annotated c-structure for *He sees the girl* with the rules in (1) and lexicon in (2).

(4) $(\phi(\mathit{root})\ \textsc{subj}) = \phi(n_1)$
$\phi(\mathit{root}) = \phi(n_2)$
$(\phi(\mathit{root})\ \textsc{tense})$
$\phi(n_1) = \phi(n_3)$
$(\phi(n_3)\ \textsc{pred}) = \text{‘}\textsc{pro}\text{’}$
$(\phi(n_3)\ \textsc{agr pers}) = 3$
$(\phi(n_3)\ \textsc{agr num}) = \textsc{sg}$
$\phi(n_2) = \phi(n_4)$
$(\phi(n_4)\ \textsc{pred}) = \text{‘}\textsc{see}\langle\textsc{subj obj}\rangle\text{’}$
$(\phi(n_4)\ \textsc{pred}) = \textsc{pres}$
$(\phi(n_4)\ \textsc{subj agr pers}) = 3$
$(\phi(n_4)\ \textsc{subj agr num}) = \textsc{sg}$
$(\phi(n_2)\ \textsc{obj}) = \phi(n_5)$
$\phi(n_5) = \phi(n_6)$
$\phi(n_5) = \phi(n_7)$
$(\phi(n_6)\ \textsc{spec}) = \textsc{def}$
$(\phi(n_7)\ \textsc{pred}) = \text{‘}\textsc{girl}\text{’}$
$(\phi(n_7)\ \textsc{agr pers}) = 3$
$(\phi(n_7)\ \textsc{agr num}) = \textsc{sg}$
$(\phi(n_7)\ \textsc{spec})$

We can test each equation separately to verify that the f-structure in Figure 1 meets all the specifications in (4). The equation $(\phi(n_4)\ \textsc{subj agr num}) = \textsc{sg}$ is satisfied, for example, because ϕ maps n_4 to the outermost f-structure, and that f-structure has a path from SUBJ through AGR to NUM, ending in the atomic value SG. That value is consistent with the requirement that the equation $(\phi(n_3)\ \textsc{agr num}) = \textsc{sg}$ imposes on the f-structure of n_3. In contrast, this grammar would assign no f-structure to the string *They sees the girl* because the f-description for its c-structure would require its subject f-structure to have inconsistent values for AGR NUM, a violation of the *Uniqueness Condition* of Kaplan &

Bresnan (1982). The correspondence ϕ and the instantiated metavariables ensure that the properties of the subject NP are consistent with the verb's agreement specification even though they do not appear together in a local mother-daughter configuration.

The f-structure in this configuration also meets the additional well-formedness conditions of LFG theory. We see that it is a *minimal* model of the f-description in the sense that at least one equation or combination of equations will no longer be satisfied if any attribute or value is removed (for example $(\phi(n_6)$ SPEC$)=$DEF fails without the SPEC feature of the OBJ).[1] Conversely, a structure with any features beyond those already present, say if the SUBJ is extended with TENSE PAST, is not minimal, because the f-description is still satisfied when that feature is removed. The minimal model is unique[2] for a given annotated c-structure and contains all and only the linguistically relevant features that are expressed by the words of a sentence.

The minimal model is important in LFG theory for another reason. It is the basis for the distinction between *defining annotations* and *constraining annotations*. The defining annotations are the simple equalities between two designators whose instantiations determine the attributes and values of the minimal model. That f-structure must then also satisfy the instantiations in the f-description of any constraining annotations. The grammar in (1) contains two constraining annotations, the positive existential constraints ($\uparrow$ TENSE) and ($\uparrow$ SPEC). The instantiation $(\phi(\mathit{root})$ TENSE) is satisfied because the conjunction of defining equations in the f-description specify a particular value (PRES) for the attribute TENSE in the f-structure corresponding to the S node. This constraint excludes strings whose main verb is a participle instead of a tensed form (e.g. **He seeing the girl*) without depending on participles setting up a uniqueness clash by also adding a TENSE feature with an otherwise unnecessary and uninformative value (e.g. NONE). Similarly the instantiation $(\phi(n_7)$ SPEC) excludes singular common nouns that have no specifier (e.g. **He sees girl*). The formalism also allows for con-

[1] Strictly speaking, a minimal model of the f-description includes not only the attributes and values of the f-structure but also the association of those elements with the nodes of the c-structure as instantiated via the ϕ correspondence, as depicted in Figure 1. Technically, what we usually regard as the f-structure is the restriction of such a model to just those attributes and values.

[2] As a notational convenience, the LFG formalism allows for primitive annotations to be embedded in disjunctive formulas that then might have several solutions. There is an obvious transformation of the grammar that converts disjunctions of annotations within a rule to an equivalent set of alternative rules with annotations that are no longer disjunctive. The minimal models are unique for the annotated c-structures assigned by the rules of such a transformed grammar.

straints that test the minimal model for the absence of a feature, e.g. $\neg(\uparrow \textsc{obj})$; for the presence or absence of a specific attribute value, e.g. $(\uparrow \textsc{voice}) =_c \textsc{passive}$ or $(\uparrow \textsc{voice}) \neq \textsc{active}$; for the identity of two f-structures, e.g. $(\uparrow \textsc{subj}) =_c (\uparrow \textsc{obj})$; and for any value other than a specific one, e.g. $(\uparrow \textsc{subj num}) \neq \textsc{sg}$. Constraining annotations help to avoid clutter in the f-structure by assigning syntactic significance to the presence or absence of unmarked or default features and also by capturing the difference between constituents that provide values for features and constituents that check those values. See Kaplan (2019) for a fuller discussion of underspecified values in LFG.

The quoted values of the $\textsc{pred}$ attributes in Figure 1 carry the subcategorization restrictions of the predicates they represent, and they characterize the essential interaction between syntax and semantics while staying agnostic about the details of any particular underlying semantic theory. The semantic form '$\textsc{see}\langle\textsc{subj obj}\rangle$' contains a list of grammatical-function designators that the predicate subcategorizes for. The *Completeness Condition* requires that all listed functions appear locally in the minimal f-structure, and the *Coherence Condition* precludes the local appearance of any governable functions ($\textsc{comp}$, $\textsc{obl}$, $\textsc{xcomp}$...) not included in the list. The semantic form also indicates that $\textsc{see}$ is the semantic relation and, by virtue of their order in the list, that $\textsc{subj}$ and $\textsc{obj}$ respectively map to the first and second arguments of that relation. Semantic forms do not require special treatment in our formal analysis because they can be interpreted as succinct abbreviations for collections of other annotations. Thus the positive and negative constraints (5a-b) express the subcategorization requirements of '$\textsc{see}\langle\textsc{subj obj}\rangle$'. The semantic relation and the mapping of functions to arguments can be coded with distinguished attributes $\textsc{rel}$, $\textsc{arg1}$, $\textsc{arg2}$ as in (5c-d).[3]

(5) a. Completeness: $(\uparrow \textsc{subj}) \quad (\uparrow \textsc{obj})$
b. Coherence: $\neg(\uparrow \textsc{comp}) \quad \neg(\uparrow \textsc{obl}) \quad \neg(\uparrow \textsc{xcomp})$ ...
c. Semantic relation: $(\uparrow \textsc{rel}) = \textsc{see}$
d. Argument mapping: $(\uparrow \textsc{arg1}) = (\uparrow \textsc{subj})$
$(\uparrow \textsc{arg2}) = (\uparrow \textsc{obj})$
e. Instantiation: $(\uparrow \textsc{pred instance}) = *$

[3] Halvorsen & Kaplan (1988) introduced a separate semantic projection, σ, as an alternative to distinguished attributes in formulating these essential properties of the syntax-semantics interface. In that more explicit arrangement σ would be a qualifier on the (5c-d) designators. In Glue Semantics they are elaborated in collections of linear logic premises (Dalrymple et al. 1993; Dalrymple 1999).

Semantic forms are instantiated in LFG theory to mark the difference between syntactically-implied semantic coreference (as in constructions of functional control) and unrelated repetitions of similar expressions. The effect of equation (5e) is to make each occurrence unique by assigning the matching node as an index.

3 Technical preliminaries

In preparation for the mathematical analysis in the following sections we now introduce more precise specifications of the LFG derivation machinery.

The annotated c-structure is often described as the result of a special derivation process for an LFG grammar G that treats categories and annotations separately. But it is helpful for formal reasoning to regard it as a normal derivation of the *annotated c-structure grammar* for G, an ordinary context-free grammar with a systematically modified set of rules. Suppose $X{:}A$ is an annotated category in the right side of a rule in the traditional LFG grammar format. Then for every rule expanding X the annotated grammar contains a version in which the left side is also decorated with those particular annotations. For example, because NP in (1) is annotated in S with the SUBJ assignment and in VP with the OBJ assignment, the NP rule is replaced by the rules in (6).

(6)
$$\begin{array}{cccc} \text{NP} & \rightarrow & \text{(Det)} & \text{N} \\ (\uparrow \textsc{subj}) = \downarrow & & \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

$$\begin{array}{cccc} \text{NP} & \rightarrow & \text{(Det)} & \text{N} \\ (\uparrow \textsc{obj}) = \downarrow & & \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

With this reformulation the normal category matching of context-free derivations allows us to make direct use of all established properties (decidability, closure, pumping) of context-free grammars and their derivations. The traditional LFG c-structure in Figure 1 is obviously just the annotation-free projection of the annotated c-structure in Figure 2.

For every annotated c-structure there is an instantiated f-description that defines a function ϕ mapping its nodes to their corresponding minimal-model f-structure units, if the f-description is satisfiable. There is also a function *Yield* that maps its nodes to the substrings of the sentence that they dominate. The set of G's derivations is then characterized by the relation Δ_G defined in (7).

(7) $\Delta_G(s, c, f)$ iff c is an annotated c-structure of G, s is the terminal string of c, and f is the minimal model for the satisfiable f-description instantiated from c.

Note that an annotated c-structure c uniquely determines both the string s and f-structure f in a derivation triple. Moreover, without further stipulation we know that the length of the string $|s|$ and the number of units $|f|$ in the f-structure are both bounded by (functions of) $|c|$, the number of nodes in the c-structure. That is, there are grammar-dependent functions $\overleftarrow{b}_G$ and $\overrightarrow{b}_G$ such that

(8) For all $(s, c, f) \in \Delta_G$, $|s| \leq \overleftarrow{b}_G(|c|)$ and $|f| \leq \overrightarrow{b}_G(|c|)$.

The function $\overleftarrow{b}_G$ depends on the number of daughters in the longest c-structure rule and $\overrightarrow{b}_G$ depends on the most complicated annotated category.

The language, f-structure, parsing, and generating projections of Δ_G are defined in (9).

(9) $L(G) = \{s \mid \Delta_G(s, c, f)$ for some c and $f\}$ = the language of G
$F(G) = \{f \mid \Delta_G(s, c, f)$ for some s and $c\}$ = the f-structures of G
$Par_G(s) = \{f \mid \Delta_G(s, c, f)$ for some $c\} \subseteq F(G)$
$Gen_G(f) = \{s \mid \Delta_G(s, c, f)$ for some $c\} \subseteq L(G)$

A parser for an LFG grammar G provides for any given string s the set of f-structures (if any) that are related to it by the grammar, and a generator provides all the strings that the grammar relates to a given f-structure (if any).

These projections allow for succinct statements of the emptiness, recognition, and realization decision problems (10).

(10) Emptiness: is $L(G)$ empty? (equivalently, are $F(G)$ or Δ_G empty?)
Recognition: for any string s is $Par_G(s)$ empty?
Realization: for any f-structure f is $Gen_G(f)$ empty?

We show in the next section that the emptiness, recognition, and realization problems are all undecidable for unrestricted LFG grammars. This implies immediately that the parsing and generation are also unsolvable. Our demonstrations involve simple phrase-structure rules with elementary defining annotations as exemplified in (11).

(11)	(↑/↓ SUBJ NUM) = SG	assign an atomic value
	(↑ SUBJ) = ↓	assign a function to a daughter f-structure
	(↓ OBJ) = (↑ SUBJ)	daughter-mother control
	(↑ XCOMP SUBJ) = (↑ SUBJ)	traditional functional control

Annotations of these types are not exceptional, they are commonly found in linguistic grammars.

4 Undecidable problems

A standard method for showing that a formal problem of interest is undecidable is the reduction technique. A problem P is said to be *reducible* to problem P' if for any instance of P an instance of P' can be constructed such that solving the instance of P' will solve the instance of P as well. Thus, if P reduces to P' and P is undecidable, then P' must also be undecidable. As noted, this general strategy has been applied with reductions from different source problems (Turing machine halting, Hilbert's Tenth, Post Correspondence, emptiness of context-free intersection) to address the LFG emptiness, recognition, and realization problems. Here we present a single reduction-source framework, based on the emptiness problem of context-free intersection, that recapitulates these previous results.

4.1 The emptiness problem

The emptiness problem for context-free intersection is the problem of determining whether or not the languages generated by two given context-free grammars G_1 and G_2 have an empty intersection ($L(G_1) \cap L(G_2) = \emptyset$). This problem is known to be undecidable. The reduction of this emptiness problem to questions of the LFG formalism depends on the ability to construct for every context-free grammar G an LFG grammar whose f-structures contain encodings of all and only the strings of $L(G)$. We show in (12) one way in which a string pqr can be encoded in the attributes and values of an f-structure, as a H(ead)-T(ail) list representation.

(12) $\begin{bmatrix} \textsc{h} & \text{p} \\ \textsc{t} & \begin{bmatrix} \textsc{h} & \text{q} \\ \textsc{t} & \begin{bmatrix} \textsc{h} & \text{r} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

Without loss of generality, let G be an arbitrary context-free grammar in Chomsky Normal Form, that is, a context-free grammar with only binary branching rules of the form $A \rightarrow B\ C$ for nonterminal expansions and unary rules $A \rightarrow a$ for terminals. The schematic rules in (13) provide a template for an LFG grammar *String*(G) that creates head-tail encodings (12) for the strings of $L(G)$.

(13) a. $A \rightarrow \begin{array}{cc} B & C \\ (\uparrow \textsc{l})=\downarrow & (\uparrow \textsc{r})=\downarrow \\ (\downarrow \textsc{b})=(\uparrow \textsc{b}) & (\downarrow \textsc{e})=(\uparrow \textsc{e}) \\ & (\uparrow \textsc{l e})=(\uparrow \textsc{r b}) \end{array}$

b. $A \rightarrow \begin{array}{c} a \\ (\uparrow \textsc{b h})=a \\ (\uparrow \textsc{b t})=(\uparrow \textsc{e}) \end{array}$

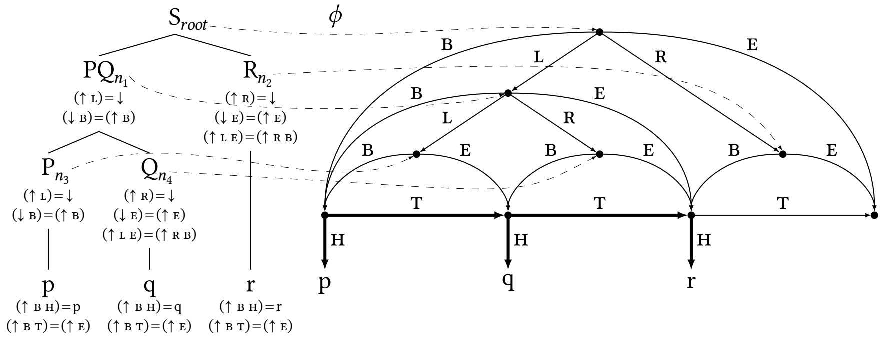


Figure 3: An annotated c-structure and f-structure derived with head-tail string encoding rules of the form in (13). Thick lines show the string encoding, thin lines show the construction scaffolding. The ϕ correspondence is depicted with dashed lines.

The annotations on the binary rules (13a) transmit the string encodings from their daughter f-structures to their mother f-structure. The attributes l(eft) and r(ight) are the scaffolding needed to concatenate the encodings from the daughters by linking the end of the left-daughter encoding to the beginning of the right. Rules of the form (13b) create for each terminal the one-element head-tail encoding of their right side, with b and e attributes marking its beginning and end. Control equations such as (↓ b)=(↑ b) and (↑ b t)=(↑ e) are the essential ingredient in this and other string-encoding formulations: Crucially, they allow terminal-string information to propagate transparently through all intermediate nodes to the f-structure of the root. Figure 3 shows the annotated c-structure and a graphical f-structure representation for a derivation containing a head-tail encoding of a single string.

Now suppose that G_1 and G_2 are arbitrary context-free grammars in Chomsky Normal Form and assume without loss of generality that their nonterminals are disjoint and that the strings of each language end with a marker # distinct from all other terminals. We construct a new LFG grammar G by combining the rules of $String(G_1)$ and $String(G_2)$ with root categories S_1 and S_2 respectively and introducing a new root category S with start rule (14).

(14)
$$\begin{array}{lccc} \mathrm{S} & \rightarrow & \mathrm{S}_1 & \mathrm{S}_2 \\ & & (\downarrow \textsc{b})=(\uparrow \textsc{b}) & (\downarrow \textsc{b})=(\uparrow \textsc{b}) \end{array}$$

By construction of the string grammars and the b annotations of the start rule, only the string encodings of the two derived f-structures can interact. Because the string encodings are compatible only if the derived strings are identical, the

LFG language $L(G)$ contains all and only strings ss for $s \in L(G_1) \cap L(G_2)$. The emptiness of context-free intersection is undecidable so the question whether $L(G)$ is empty must also be undecidable.

4.2 The recognition problem

We prove that the LFG recognition problem is undecidable by exhibiting one particular string that belongs to $L(G)$ only if $L(G_1) \cap L(G_2) \neq \emptyset$. We modify the string grammars for G_1 and G_2 by treating each terminal a other than # as a nonterminal category and adding for each of them a trivial rule

(15) $a \rightarrow \epsilon$

The effect is that only the string # belongs to the language of each of the modified string grammars, but that single string is assigned all and only the f-structures that respectively encode the original context-free languages. Again with the starting rule (14) the concatenation ## belongs to the language of the modified grammar if and only if $L(G_1) \cap L(G_2) \neq \emptyset$, that is, if and only if $Par_G(\#\#)$ is not empty.

Empty nodes are disfavored in some modern versions of LFG, particularly when long-distance dependencies are characterized by functional uncertainty rather than traces (Kaplan & Zaenen 1989; Dalrymple et al. 2015). But the undecidability of recognition can also be demonstrated with grammars $String(G_1)$ and $String(G_2)$ redefined so as to produce the same head-tail string encodings from nonbranching dominance chains without the benefit of empty nodes.

For each binary rule $A \rightarrow B\ C$ the string encoding grammars will now contain a nonbranching rule of the form (16a). This immediately derives only the left daughter B but pushes the right-daughter category C on a simulated stack for expansion lower in the derivation. Since B is the left daughter of A, the encodings of their terminal strings have a shared B(eginning).

(16) a. $A \rightarrow \begin{array}[t]{c} B \\ (\downarrow \textsc{stk cat})=C \\ (\downarrow \textsc{stk nxt})=(\uparrow \textsc{stk}) \\ (\downarrow \textsc{b})=(\uparrow \textsc{b}) \end{array}$ b. $A \rightarrow \begin{array}[t]{c} C \\ (\uparrow \textsc{stk cat})=C \\ (\uparrow \textsc{stk nxt})=(\downarrow \textsc{stk}) \\ (\uparrow \textsc{b t})=(\downarrow \textsc{b}) \\ (\uparrow \textsc{b h})=a \end{array}$ c. $A \rightarrow \begin{array}[t]{c} \# \\ (\uparrow \textsc{b h})=\# \end{array}$

Corresponding to each terminal rule $A \rightarrow a$, for $a \neq \#$, there is a collection of rules of the form (16b), one for each right-daughter category C whose expansion may have been deferred until it reemerges at the top of the stack. The annotations pop that category from the stack while adding the terminal a to the front of the head-tail encoding of the terminal string under C. Finally, for each unary rule

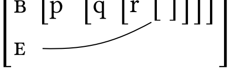
$A \rightarrow \#$ the string grammar contains a rule of the form (16c) to terminate the NBD derivations and install # as the final item of every string encoding. Because # is the distinguished end-of-string marker, these preterminals never appear as left daughters of binary rules and are thus always the last categories to be removed from the stack. As before, if NBD string grammars $String(G_1)$ and $String(G_2)$ are combined into an LFG grammar G with rule (14), then $Par_G(\#\#) \neq \emptyset$ if and only if $L(G_1) \cap L(G_2) \neq \emptyset$.

The complexity of recognition arises from the fact that, in order to assign f-structures to the strings of infinite languages, annotated c-structure grammars must include rules for recursive subderivations (rule sequences that derive a node labeled with an annotated category A from an A-labeled dominating node), and such recursive subderivations must be allowed to stack one above another. The string grammars in our undecidability proofs show that recursive subderivations can assign to a single string (#) a set of f-structures each encoding one of the strings of an infinite context-free language. Unlike the f-structures that correspond to the sentences of natural languages, those f-structures are determined only by the annotations on nonterminal categories without regard to any lexical information carried by the input string or even its length (there is no function of $|s|$ that bounds the sizes of c and f in all derivation triples).

4.3 The realization problem

We turn now to the realization problem. Also using a reduction from the emptiness of context-free intersection, Wedekind (2014) proved that realization is undecidable for unrestricted LFG grammars if there are cyclic paths in the input f-structure.[4] Whereas the emptiness and recognition demonstrations are based on head-tail string encodings, Wedekind's proof is formulated in terms of an alternative way of encoding the strings of a language, a descending chain of attributes as illustrated in (17).

(17)



The beginning of the encoding for string pqr is still accessible as the value of the B attribute, but now the end is identified by the reentrant value of the top-level E

[4] Wedekind & Kaplan (2012) established that the realization problem is decidable if the input f-structure f contains no cycles. For an acyclic f-structure the string-set $Gen_G(f)$ can be described by a context-free grammar, and the emptiness problem for context-free grammars is decidable.

attribute. Grammars that encode context-free languages in this way are created by replacing the annotations on the terminal rules (13b) with functional control annotations as in (18).

(18)
$$A \rightarrow \begin{array}{c} a \\ (\uparrow \textsc{b}\ a)=(\uparrow \textsc{e}) \end{array}$$

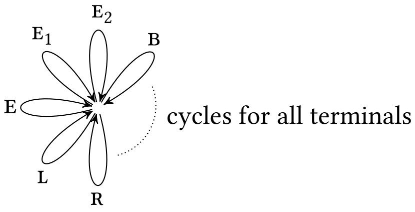
The scaffolding illustrated in Figure 3 is unchanged but the H attributes at the bottom are removed and the sequence of T attributes is replaced by the sequence of terminal-attributes.

The essence of Wedekind's (2014) proof is then captured by combining attribute-chain string-encoding grammars for arbitrary context-free grammars G_1 and G_2 into an LFG grammar G with start rule (19).

(19)
$$\text{S} \rightarrow \begin{array}{c} \text{S}_1 \\ (\uparrow \textsc{l})=\downarrow \\ (\downarrow \textsc{b})=(\uparrow \textsc{b}) \\ (\downarrow \textsc{e})=(\uparrow \textsc{e}_1) \end{array} \quad \begin{array}{c} \text{S}_2 \\ (\uparrow \textsc{r})=\downarrow \\ (\downarrow \textsc{b})=(\uparrow \textsc{b}) \\ (\downarrow \textsc{e})=(\uparrow \textsc{e}_2) \end{array} \quad \begin{array}{c} \# \\ (\uparrow \textsc{e}_1\ \textsc{e}_1)=\uparrow \\ \bigwedge\limits_{x \text{ an attribute}} (\uparrow \textsc{e}_2\ \textsc{e}_1)=(\uparrow \textsc{e}_2\ \textsc{e}_1\ x) \end{array}$$

In the absence of atomic values there can be no atom-value clashes to exclude mismatching combinations, and the language $L(G)$ therefore contains all strings $s_1 s_2\#$ for $s_1 \in L(G_1)$ and $s_2 \in L(G_2)$. However, strings belonging to the intersection of $L(G_1)$ and $L(G_2)$ are distinguished by the fact that the end points $\textsc{e}_1$ and $\textsc{e}_2$ of their descending attribute-chain encodings are the same. In that case $(\phi(\mathit{root})\ \textsc{e}_1)=(\phi(\mathit{root})\ \textsc{e}_2)$ and the annotations on the terminal # entail by simple substitutions that $(\phi(\mathit{root})\ x)=\phi(\mathit{root})$ for all attributes x. Thus all and only strings $ss\#$ for $s \in L(G_1) \cap L(G_2)$ receive the one-element cyclic f-structure f in (20).

(20)



The realization problem is undecidable because $Gen_G(f) \neq \emptyset$ if and only if $L(G_1) \cap L(G_2) \neq \emptyset$. This shares with the recognition proof the property that infinitely many annotated c-structures of arbitrary size may have to be inspected to determine whether there is at least one that is related to a single input of a fixed size (a cyclic f-structure in this case).[5]

The undecidability results we have demonstrated here, together with other simple reductions from the emptiness problem of context-free intersection, can be used to show that other properties of the unrestricted LFG formalism are also undecidable. The following is a partial list of these undecidable questions.

(21) a. Generation from underspecified f-structures: Is there a sentence that realizes an f-structure with more features than a given one? (Wedekind 1999)

b. Ambiguity-preserving generation: Is there a single string that realizes two different f-structures? (Wedekind & Kaplan 1996)

c. Finite versus infinite ambiguity: Is any string in the language infinitely ambiguous? (Jaeger et al. 2005)

d. Ranking in Optimality-theoretic LFG: Can an optimal derivation always be identified? (Kuhn 2003)

e. Economy of Expression: Can the smallest c-structure for a given f-structure be identified?[6]

Appendix A includes simple proofs showing that a number of more specific questions are also undecidable. Additional restrictions are clearly necessary to provide a linguistic formalism that is mathematically manageable.

5 Conservation and decidability

The recognition and realization problems are undecidable for unrestricted LFG grammars because there is no finite number of (size-bounded) annotated c-structures whose inspection is sufficient to determine whether there is a valid derivation for a given input string/f-structure. As a consequence, there is no systematic relationship between the length of a string and the sizes of its f-structure parses or the size of an f-structure and the lengths of its generated strings. Moreover, a grammar can assign infinitely many f-structures to a single string and

[5] Cyclic f-structures have been proposed in the analysis of complex adjunction and coordination constructions (Zweigenbaum 1988; Fang & Sells 2007; Haug & Nikitina 2012; Przepiórkowski & Patejuk 2012) and thus cannot be excluded from the LFG formalism. More to the point, example (52b) in the Appendix A shows that it is undecidable whether an arbitrary LFG grammar produces cyclic f-structures.

[6] This follows from the fact that realization is undecidable in the general case (as just sketched): if it cannot be decided whether there are any c-structures at all for an f-structure input, then the smallest such structure cannot be determined.

infinitely many strings to a single f-structure. These properties seem implausible for language as a medium of communication.

From a broader perspective, these excesses can be cast in terms of the "grammatical mapping problem", the problem of characterizing in an explanatory and computable way the relation Γ between the sentences of a language and representations of their meanings (presumably logical formulas that can be interpreted in a representation of the world) (Kaplan & Bresnan 1982; Kaplan 1987). If s is a sentence of a language and m represents one of its meanings (that is, $(s, m) \in \Gamma$), then pretheoretically we expect the derivational machinery that translates between s and m to be information-conserving in the following sense.

(22) *Principle of Conservation*
For all $(s, m) \in \Gamma$, $|m|$ is bounded by $|s|$ and $|s|$ is bounded by $|m|$.

The size of the meaning representation can be defined in any reasonable way. The crucial claim is that the derivational machinery does not by itself add or subtract, in either direction, arbitrary amounts of information. The additional linguistically appealing property of bidirectional finite ambiguity follows as an immediate corollary.

(23) *Finite Ambiguity*
If Γ is conservative, then each sentence expresses only a finite number of meanings and each meaning is expressed by only a finite number of sentences.

In LFG-based approaches the grammatical mapping Γ is typically conceptualized as the composition of the grammar-defined syntactic derivations Δ_G and the semantic derivations Σ that map primarily between syntactic f-structures and corresponding representations of meaning.[7]

(24) $(s, m) \in \Gamma_G$ iff $(f, m) \in \Sigma$ and $(s, c, f) \in \Delta_G$ for some c-structure c.

An end-to-end mapping $(s, m) \in \Gamma_G$ is conservative if the semantic derivation $(f, m) \in \Sigma$ has grammar-dependent bounds in both directions and is thus information-conserving, and s and f of the triple $(s, c, f) \in \Delta_G$ are also co-bounded (the syntactic derivation is also conservative). Recalling that $|s|$ and $|f|$ are both bounded by $|c|$ in any derivation triple (8), it follows that

[7] This is not to discount the influence of linguistic features that may be formalized in other projections within the LFG correspondence architecture. The bounding requirements of the Conservation Principle would also govern mappings that include those other projections.

(25) An LFG syntactic derivation $(s, c, f) \in \Delta_G$ is *conservative* if also $|c|$ is bounded by both $|s|$ and $|f|$.

The syntactic recognition/parsing and realization/generation problems are solvable if only conservative derivations are defined to be linguistically relevant, in accordance with principle (22). In each direction only a finite number of size-limited annotated c-structures must be enumerated and inspected to determine whether a derivation belongs to Δ_G.[8]

With respect to Σ, Glue Semantics (Dalrymple et al. 1993; Dalrymple 1999) determines a meaning representation m for a string by a linear-logic deduction applied to a collection of premises associated with an f-structure f assigned to that string. The resource-sensitive nature of linear logic suggests that m will naturally be bounded by $|f|$, but that has not yet been clearly established. It is also unknown whether or under what additional conditions the f-structures that correspond to a given meaning representation m are bounded by $|m|$.[9] With the expectation that those issues will be resolved in future research, we return here to our focus on Δ_G, the syntactic component of Γ_G.

Kaplan & Bresnan (1982) were the first to show the undecidability of the recognition problem for unrestricted LFG grammars and the first to address it by imposing an information-conserving constraint on the derivations in Δ_G. Their constraint restricts the derivations of the annotated c-structure grammar so as to limit the distribution of empty nodes and nonbranching dominance chains.[10] The effect is to include as NBD-valid c-structures only those where every recursive subderivation contains at least one pair of terminal-dominating sisters. This specifically excludes the derivations that our demonstrations of recognition undecidability rely on. All NBD-valid derivations are conservative in the parsing direction, since the annotated c-structure is bounded by the length of the string, and the recognition and parsing problems are therefore solvable.

[8] However, the emptiness problem remains undecidable even if attention is confined only to conservative derivations. All derivations for the grammars constructed with rules (13) and (14) are conservative in the sense of (25). Emptiness requires consideration of all possible string or f-structure inputs, not just particular ones that are presented for parsing or generation. By the same token, it is undecidable whether all derivations for a given grammar are conservative.

[9] Generation from an f-structure not bounded by $|m|$ can be reduced to the undecidable problem of generating from an arbitrarily underspecified f-structure (Wedekind 1999).

[10] The NBD constraint in LFG was a specific and early example of a family of what have become known generically as *Off-line Parsability* conditions. A number of variants of Off-line Parsability have been proposed for other grammatical frameworks. See Jaeger et al. 2005 for a survey.

By a symmetrical argument, syntactic derivations will be conservative in the generation direction if they are restricted so that the size of the annotated c-structure is bounded as a function of the size of the f-structure. Unfortunately, the NBD condition is not sufficient to pick out just those information-conserving derivations and thus to ensure also that the realization and generation problems are decidable (cf. Wedekind (2014); Wedekind & Kaplan (2020)). The attribute-chain string-encoding grammars and the combining start rule (19) used in the undecidability proof for realization are ϵ-free, and it is only (nonrecursive) terminal rules that do not branch. A condition stronger than the NBD restriction is needed to guarantee that generation is conservative and thus decidable.

It has also been noted, on the other hand, that the original NBD condition may be too strong. It disallows recursive nonbranching dominance chains in every context, even when an errant subderivation is a component of a discontinuous constituent supported intuitively by an element elsewhere in the string. For example, Johnson (1986) observed that it proscribes the straightforward analysis of the Dutch double infinitive construction as provided by the grammar of Bresnan et al. (1982) and illustrated in (26).

(26) (dat) hij het boek heeft kunnen lezen
(that) he the book has able read
'(that) he has been able to read the book'

Recursive applications of the nonbranching VP rule (27) would be required to match the level of the OBJ 'het boek' with the level of its governing predicate in the discontinuous, extended-head configuration.[11]

(27)
$$\text{VP} \rightarrow \underset{(\uparrow \textsc{xcomp})=\downarrow}{\text{VP}}$$

We address these shortcomings of Kaplan and Bresnan's NBD restriction by introducing an alternative way of identifying a subclass of conservative derivations that is better attuned to the natural flow of linguistic information. It takes into

[11] Johnson's particular example does not violate the very early refinement of the constraint wherein functional annotations are also taken into account in determining whether a category is recursive. This was introduced soon after the original formulation and later described by Kaplan & Maxwell (1996) and Dalrymple 2001. But this slightly weaker version would still disallow the intended analyses of sentences with more intransitive verbs and deeper XCOMP embeddings as in

(dat) hij het boek moet haben kunnen lezen
(that) he the book must have able read
'(that) he must have been able to read the book'

account the architectural correspondence between c-structures and f-structures to impose a new bound on the size of generation c-structures while relaxing the bound in the parsing direction. Our new condition makes use of the following definitions.

Let c be an annotated c-structure and let n and n' be two distinct nodes in c with n dominating n'. The *subderivation from n to n'*, denoted by $c^n_{n'}$, is the derivation that we obtain from c by removing from the subderivation rooted by n the subtree under n'. Two subderivations $c^n_{n'}$ and $c^{\check{n}}_{\check{n}'}$ are said to be *stacked* if the bottom node of one dominates the top node of the other. A subderivation $c^n_{n'}$ is *recursive* if n and n' are both labeled with the same annotated category.

The admissibility of recursive subderivations is then defined in terms of f-structure and string anchors.

(28) Let c be an annotated c-structure with terminal string s and f-structure f. We say that a recursive subderivation $c^n_{n'}$ is

a. *f-anchored in f_k* if there is a node $\bar{n}$ of $c^n_{n'}$ such that $\phi(\bar{n}) = f_k$ and

b. *s-anchored in s_j* if there is a node $\bar{n}$ of $c^n_{n'}$ such that $\bar{n}$ or a node in $\phi^{-1} \circ \phi(\bar{n})$ dominates s_j.

We refer to f_k and s_j as the f- and s-anchors of $c^n_{n'}$. The subclass of properly anchored derivations is then defined as follows.

(29) A derivation $(s, c, f) \in \Delta_G$ is *properly anchored* iff

a. every recursive subderivation $c^n_{n'}$ of c is f- and s-anchored and

b. the f-anchors of any two recursive subderivations in a stack are distinct, and so are their s-anchors.

If (s, c, f) is a properly anchored derivation, then every recursive subderivation is anchored in both a functional unit of the f-structure and an element of the string (29a). Moreover, requirement (29b) ensures that the anchoring for stacked recursive subderivations is one-to-one. The anchoring of stacked recursive subderivations of such a c-structure is illustrated in Figure 4.

If N is the set of annotated nonterminal categories for a grammar G, any subderivation $c^n_{n'}$ with a path length equal to $|N|$ must be recursive. The annotated c-structures of properly anchored derivations are thus bounded by the respective sizes of their corresponding strings and f-structures, as stated in the following lemma.

(30) The depth of the c-structure c of all properly anchored derivations $(s, c, f) \in \Delta_G$ is bounded by $|N|(|s| + 1)$ and $|N|(|f| + 1)$, respectively, for a string of length $|s|$ and an f-structure of $|f|$ units.

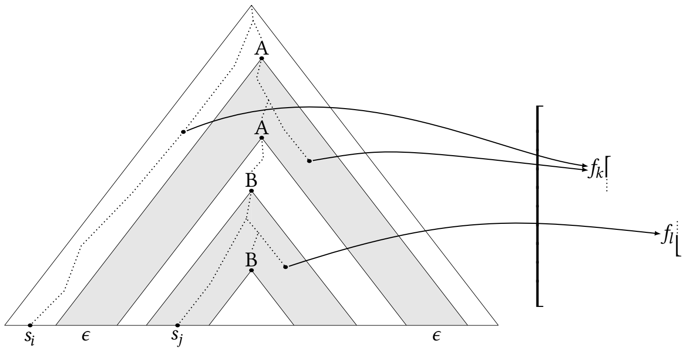


Figure 4: A c-structure with two stacked subderivations, highlighted in gray. The subderivations are f- and s-anchored at f_k, s_i and f_l, s_j, respectively, with $k \neq l$ and $i \neq j$. The upper subderivation is discontinuously string-anchored because none of its internal nodes dominates a terminal (it derives the empty string ϵ) while the lower subderivation is continuously string-anchored.

Lemma (30) implies that the properly anchored derivations for an unrestricted LFG grammar are conservative, and that the recognition and realization problems are therefore decidable if unanchored derivations are excluded from linguistic consideration. This is because only a finite number of size-bounded annotated c-structures need to be inspected in order to solve these problems.

The conditions for proper anchoring include derivations that the NBD condition does not admit and exclude derivations that NBD classifies as valid. NBD and proper anchoring, however, do agree on the status of derivations for the schematic grammars in (31).

(31)	a.	S →	S (↑ GF) = ↓	a (↑ PRED) = ‘P⟨GF⟩’	S →	a (↑ PRED) = ‘A’	NBD-valid anchored
	b.	S →	S (↑ GF) = ↓ (↑ PRED) = ‘P⟨GF⟩’		S →	a (↑ PRED) = ‘A’	NBD-invalid shared s-anchor

The recursive subderivations for (31a) are branching and they are thus both valid and properly string-anchored. Each subderivation is also f-anchored to a distinct unit in its f-structure’s GF hierarchy. If GF is a governable grammatical function, then the f-structures of all derivations are complete and coherent and correspond to the *lexical* meanings carried by the repetitively longer strings. The grammar (31b) provides the same set of complete and coherent f-structures but associates all of them to the single one-element string. That string is infinitely ambiguous and there is no bound on the size of the constructional meaning representations determined by the recursive subderivations. These linguistically implausible subderivations are nonbranching and their string anchors cannot be distinct (29b). They are appropriately excluded as neither NBD-valid nor properly anchored.

In (32) we show grammars that provide branching derivations for every string in the set $\{a^n \mid n > 1\}$, and every derivation is thus NBD-valid but properly anchored only with respect to strings. The recursive subderivations of these grammars are excluded because they do not meet the f-anchoring conditions of (29).

(32) a. $\mathrm{S} \rightarrow \mathrm{S} \quad \mathrm{a} \qquad\qquad \mathrm{S} \rightarrow \mathrm{a}$ NBD-valid, no f-anchor

b. $\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{S}} \quad \mathrm{a} \qquad\qquad \mathrm{S} \rightarrow \underset{(\uparrow \textsc{pred})=\text{‘}\textsc{a}\text{’}}{\mathrm{a}}$ NBD-valid, shared f-anchor

The subderivations of (32a) have no f-anchors (28a) while the f-anchors for the subderivations of (32b) are not pairwise distinct (29b). The effect of the proper anchoring conditions for these configurations is consistent with other exclusionary proposals, in particular, the Different-Words version of Economy of Expression (Dalrymple et al. 2015).

As a final point of comparison, we note that the branching requirement of the NBD condition is essentially a special case of the string-anchor conditions (29) when recursive subderivations are stacked. The string anchors for valid derivations must be dominated by nodes contained within each particular subderivation. In contrast, (28b) admits stacked recursive subderivations whose distinct anchors may be dominated by nodes elsewhere in the c-structure. The linguistically significant relationship is captured in the composition $\phi^{-1} \circ \phi$. It requires only that the dominating node is an extended (co-)head of a node in a recursive subderivation, a component of the same discontinuous constituent (Zaenen & Kaplan 1995; Bresnan et al. 2016). The situation is schematized by the grammar (33).

(33) $\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{A}} \quad \underset{\uparrow=\downarrow}{\mathrm{P}}$ NBD-invalid, anchored

$$\mathrm{A} \rightarrow \underset{(\uparrow \textsc{gf})=\downarrow}{\mathrm{A}} \qquad\qquad \mathrm{P} \rightarrow \underset{(\uparrow \textsc{gf})=\downarrow}{\mathrm{P}} \quad \underset{(\uparrow \textsc{pred})=\text{‘}\textsc{p}\langle\textsc{gf}\rangle\text{’}}{\mathrm{p}}$$

$$\mathrm{A} \rightarrow \underset{(\uparrow \textsc{pred})=\text{‘}\textsc{a}\text{’}}{\mathrm{a}} \qquad\qquad \mathrm{P} \rightarrow \underset{(\uparrow \textsc{pred})=\text{‘}\textsc{p}\langle\textsc{gf}\rangle\text{’}}{\mathrm{p}}$$

The highest A and P nodes each dominate a separate stack of recursive subderivations. The subderivations of the P stack contain their distinct p string anchors, but the A stack is a nonbranching (invalid) chain over the single terminal. Because of the parallel GF function assignments, the P nodes serve as extended heads for the A nodes of the A–P discontinuous constituents, and the p terminals can thus act as distinct s-anchors for the A subderivations.[12] The number of A subderivations in each properly anchored derivation is thus bounded by the length of the p substring, and only those finitely-many derivations are made available for further filtering by the Completeness and Coherence subcategorization conditions. Derivations of this type are the basis for a natural account of the discontinuous constituents in Johnson's (1986) Dutch double infinitive examples.[13]

The proper anchoring condition (29) establishes a manageable relationship between strings and f-structures by virtue of the mediating role that recursive c-structures play in the LFG syntactic architecture. This relationship is information-conserving in the sense of (22) and (25). It crucially depends on the ϕ correspondence and the linguistically motivated notion of extended heads to correlate the depth of c-structure recursion with the sizes of strings and f-structures, as indicated by Lemma (30). The set of properly anchored derivations for a given string or f-structure is finitely enumerable. It follows that recognition and realization are decidable for that restricted subset of derivations and so are other input-specific problems as listed in (21) and in Appendix A. It is possible, for example, to identify the most economical (properly anchored) derivation for a given f-structure because there are only a finite number of candidates whose c-structures must be compared. Proper anchoring, however, is not sufficient to ensure decidability of the emptiness problem (the demonstration in Section 4.1 involves only properly anchored derivations), and other questions that require consideration of all possible string and f-structure inputs also remain undecidable.

6 Intractability of parsing and generation

The recognition/parsing, realization/generation, and other problems are decidable for the conservative, properly-anchored derivations of arbitrary LFG grammars. But the fact that the number of derivations for a given input is finite does

[12] This arrangement of parallel function assignments gives rise to the so-called "zipper" configuration discussed below and by Maxwell & Kaplan (1996) and Kaplan & Wedekind (2020).

[13] Note also that the same verbs could be reused as anchors for a different stack of recursive subderivations, for example, in the hypothetical case that the language allows an elaboration of this construction with a ditransitive lower verb and a dislocated OBL NP. This is because pairwise distinctness (29b) applies on a per-stack basis.

not mean that it is small, and indeed the computational cost of solving these problems may be very high. We show in this section that recognition and realization are intractable in the worst case, that is, for arbitrary grammars they cannot be solved in a number of processing steps polynomial in the size of a given input. Their intractability is demonstrated by the usual technique of reducing these problems to another problem that is already known to be intractable. The technique requires that the reduction itself is computable in polynomial time so that we know that the reduction procedure does not hide the complexity of the problems of interest.

The 3-SAT problem is the problem in the NP-complete complexity class often used for polynomial-time reductions that establish the intractability of other problems. This is the problem of determining the satisfiability of a Boolean formula in conjunctive normal form where each of the conjoined clauses is a disjunction of three literals. That is, each formula is a conjunction of the form $C_1 \wedge .. \wedge C_n$, each clause C_j is a disjunction of the form $l_{j_1} \vee l_{j_2} \vee l_{j_3}$, and each literal l_{j_i} is a propositional variable p_k or a negated variable $\neg p_k$. The question is whether there is at least one way of assigning truth values to the variables that makes all the clauses be true. The three-clause formula in (34a) is a simple problem that is satisfiable under several assignments among which is the one in (34b).

(34) a. $(p_1 \vee p_2 \vee p_3) \wedge (\neg p_1 \vee \neg p_2 \vee p_3) \wedge (\neg p_1 \vee p_2 \vee \neg p_3)$

b. p_1=TRUE, p_2=FALSE, p_3=FALSE

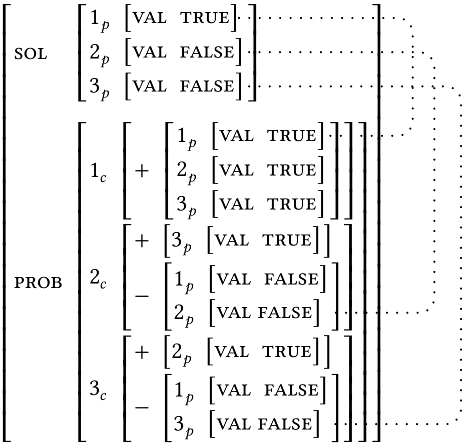
We show that the recognition problem is intractable by providing a small LFG grammar G such that the set of f-structures $Par_G(s) \neq \emptyset$ if and only if the string s is an encoding of a satisfiable Boolean problem in conjunctive normal form. A formula is presented as a sequence of substrings one corresponding to each clause. The substring for the i^{th} clause begins with the letter c followed by the string of digits $d_1 .. d_j$ that represents the integer i. This is followed by substrings that identify the literals that make up that clause. Every occurrence of a positive literal p_k is encoded as the character + followed by the digits representing the integer k, and every occurrence of a negative literal $\neg p_k$ is represented as the character − followed by the digits for k. According to this scheme the formula (34a) is presented as the string of characters (35).[14]

(35) c1 +1+2+3 c2 −1−2+3 c3 −1+2−3

[14] We would of course see longer digit strings, not just singletons, for problems with ten or more clauses or variables.

There is a simple information-conserving LFG grammar *G* that maps a string representing any satisfiable Boolean problem into f-structures that recapitulate the problem and make explicit the truth-value assignments that solve it. The linear order of clause and literal substrings is recast into descending chains of digit attributes in the f-structure. The sequences for the signed propositional variables of all literals are attached at the bottom of the attribute chain of their containing clause, and the grouping of literals within clauses is thus maintained. The lower PROB(lem) substructure shown in (36)[15] corresponds to the problem string (35).

(36)



The upper SOL(ution) substructure corresponds to the truth-value assignment (34b) that makes all clauses be true.

Let $\mathrm{S_{num}}$ be the root category of a descending attribute-chain grammar for the regular language Digit^{+} of arbitrarily long digit sequences, with the scaffolding attributes B and E giving access to the top and bottom of the descending chains (as in (17) above). Then the rules in (37) provide a c-structure and an f-structure for the string encoding of every well-formed and satisfiable Boolean formula. In particular, the f-structure for one of the derivations for string (35) appears as (36) when the innocuous scaffolding attributes are not displayed.

(37) a. S → Clause⁺
(↑ PROB)=(↓ B)
(↑ SOL)=(↓ SOL)

[15] The clause and propositional variable subscripts *c* and *p* are provided just for readability; they are not actually part of the formal structure.

$$\text{b.}\quad \text{Clause} \rightarrow \begin{array}{ccccc} \text{c} & \text{Snum} & \text{Lit}^* & \text{Lit} & \text{Lit}^* \\ & (\uparrow \textsc{b})=(\downarrow \textsc{b}) & (\uparrow \textsc{ce})=\downarrow & (\uparrow \textsc{ce})=\downarrow & (\uparrow \textsc{ce})=\downarrow \\ & (\downarrow \textsc{e})=(\uparrow \textsc{ce}) & & (\uparrow \textsc{sol})=(\downarrow +) & \\ & & & (\uparrow \textsc{sol})=(\downarrow -) & \end{array}$$

$$\text{c.}\quad \text{Lit} \rightarrow \left\{ \begin{array}{cc} + & \text{Snum} \\ & (\uparrow +)=(\downarrow \textsc{b}) \\ & (\downarrow \textsc{e val})=\textsc{true} \end{array} \middle| \begin{array}{cc} - & \text{Snum} \\ & (\uparrow -)=(\downarrow \textsc{b}) \\ & (\downarrow \textsc{e val})=\textsc{false} \end{array} \right\}$$

The start rule (37a) recognizes the conjunction of arbitrarily many clause constituents.[16] Every clause consists of one or more literals, and every literal consists of a positive or negative marking followed by the identifier of its propositional variable. The $(\uparrow \textsc{prob})=(\downarrow \textsc{b})$ annotation promotes all the clause attribute chains to the problem substructure. The additional clause-ending scaffolding attribute CE makes it possible to connect the positive and negative literals to the bottom of their containing-clause chains. The truth-value assignments in (37c) attach the value TRUE at the bottom of the variable chains of positive literals and FALSE at the bottom of negative literals, thereby encoding the truth-value assignments that make each literal be true. Finally, just one of the true literals is selected to make the clause true, and the SOL annotations incorporate the variable and truth-assignment of that literal (whether it happens to be positive or negative) into the global solution.

A derivation in G will succeed only if the literals chosen locally and independently for each clause result in SOL truth-value assignments that are globally consistent. If a problem is unsatisfiable, then the f-description for every c-structure derivation will be inconsistent. Thus for a string s encoding an arbitrary Boolean problem, the set $Par_G(s) \neq \emptyset$ if and only if that problem is satisfiable for at least one consistent set of truth-value assignments.[17]

With this abstract formal grammar it is easy to see the potential source of computational complexity for LFG recognition. For each literal of every clause, rule (37b) produces an alternative annotated c-structure that makes a different contribution to SOL. The number of properly anchored derivations that must be inspected for global consistency thus grows in the worst case as an exponential in

[16] For succinctness and clarity we use LFG's traditional Kleene $^+$ and * notations to specify repeating category sequences rather than their right-recursive equivalents. For example, the single rule (37a) is equivalent to the two rules $\text{S} \rightarrow \begin{array}{c} \text{Clause} \\ (\uparrow \textsc{prob})=(\downarrow \textsc{b}) \\ (\uparrow \textsc{sol})=(\downarrow \textsc{sol}) \end{array}$ and $\text{S} \rightarrow \begin{array}{cc} \text{Clause} & \text{S} \\ (\uparrow \textsc{prob})=(\downarrow \textsc{b}) & \uparrow=\downarrow \\ (\uparrow \textsc{sol})=(\downarrow \textsc{sol}) & \end{array}$.

[17] Berwick (1982) provided the first NP-completeness proof for the LFG recognition problem and Stanley Peters (p.c. 1982) offered a different argument. The demonstration here uses far less of the LFG machinery than those earlier proofs and generalizes to problems with arbitrary numbers of clauses and variables.

the number of clauses. For example, there will be 3^n derivations to consider in the case of a 3-SAT problem with n clauses each of which has three literals. Linguistic grammars will also have this exponential complexity profile if their fixed number of rules describe morphosyntactic agreement dependencies that range over the full length of the input string. We can also see, schematically, a configuration that is sufficient to guarantee tractability while still allowing for input strings of arbitrary length. Suppose there is a constant k that limits the number of clauses that a single S can expand to, as in (38), but with a new starting category S′ that allows for the concatenation of an arbitrary number of k-limited S's.

(38) $\mathrm{S}' \rightarrow \mathrm{S}^{+}$ $\qquad \mathrm{S} \rightarrow \begin{array}{c}\mathrm{Clause}^{\leq k}\\ (\uparrow \textsc{prob}) = (\downarrow \textsc{b})\\ (\uparrow \textsc{sol}) = (\downarrow \textsc{sol})\end{array}$

Crucially, there are no annotations on S′ to link the f-structures of the S nodes, and thus there can be no interaction among the truth assignments of the embedded clauses. The worst case complexity for a string of n 3-literal clauses is proportional to $\frac{n}{k} \cdot 3^k$. This is exponential in the grammar-dependent constant k but polynomial in the length of the input. This foreshadows the tractability of k-bounded LFG grammars that we discuss in the next section.

For recognition the Boolean problem string and f-structure are organized so that the signed propositional variables are grouped within clauses, and the grammar checks for consistency of variable truth values in the global SOL structure. For the reduction of the LFG realization problem to Boolean satisfiability, the string and corresponding f-structure are transposed so that a Boolean problem is presented with its clauses grouped within its propositional variables. We again provide a small LFG grammar G' now with the property that the string set $Gen_{G'}(f) \neq \emptyset$ if and only f is the encoding of a satisfiable Boolean problem.[18] The transposed string presentation and equivalent f-structure for problem (34a) are shown in (39).

(39) a. p1 +1−2−3 p2 +1−2+3 p3 +1+2−3

[18] See Wedekind & Kaplan (2021) for a fuller discussion of the technical issues particularly concerning the realization problem.

b.
$$\begin{bmatrix} 1_p & \begin{bmatrix} + & \begin{bmatrix} 1_c \end{bmatrix} \\ - & \begin{bmatrix} 2_c \\ 3_c \end{bmatrix} \end{bmatrix} \\ 2_p & \begin{bmatrix} + & \begin{bmatrix} 1_c \\ 3_c \end{bmatrix} \\ - & \begin{bmatrix} 2_c \end{bmatrix} \end{bmatrix} \\ 3_p & \begin{bmatrix} + & \begin{bmatrix} 1_c \\ 2_c \end{bmatrix} \\ - & \begin{bmatrix} 3_c \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The string indicates that variable 1 occurs in a positive literal in the first clause but in negative literals in the second and third clauses. The linear order of variables and clauses in the string is reflected in the f-structure's descending attribute chains. The clause identifiers are grouped according to the signed propositional variables of the literals that they contain.

The LFG grammar G' in (40) establishes the relationship between the equivalent string and f-structure expressions of any well-formed Boolean formula, whether satisfiable or not.

(40) a.
$$\begin{array}{lc} \mathrm{S} \rightarrow & \mathrm{Var}^{+} \\ & (\uparrow \textsc{prob})=(\downarrow \textsc{b}) \\ & (\uparrow \textsc{sol})=(\downarrow \textsc{sol}) \end{array}$$

b.
$$\begin{array}{lcccccccc} \mathrm{Var} \rightarrow & \mathrm{p} & \mathrm{Snum} & \{+ & \mathrm{Snum} & \mid & - & \mathrm{Snum} & \}^{+} \\ & & (\uparrow \textsc{b})=(\downarrow \textsc{b}) & & (\uparrow \textsc{ve}\ +)=(\downarrow \textsc{b}) & & & (\uparrow \textsc{ve}\ -)=(\downarrow \textsc{b}) & \\ & & (\downarrow \textsc{e})=(\uparrow \textsc{ve}) & & & & & & \end{array}$$

A sentence consists of a sequence of proposition-variable substrings each of which begins with a variable identifier followed by any number of digit substrings representing the clauses in which that variable appears. Each clause is prefixed with + and − to indicate whether the variable appears in a positive or negative literal. The annotations promote the variable's descending digit-chain to the top and attach the clause identifiers under the + or − attributes at the bottom of each variable chain, according to whether the clause is positively or negatively marked. This produces the f-structure displayed in (39b) (again with omission of the scaffolding attributes B/E and now VE). If f is an input f-structure for realization that expresses an arbitrary Boolean problem in this way, then the set $Gen_{G'}(f)$ includes a string of the form (39a).

Both the input f-structure and the grammar must be elaborated so that LFG realization distinguishes between satisfiable and unsatisfiable Boolean problems.

Along with the encoding of a particular problem the f-structure must specify the necessary and sufficient conditions for a solution, namely, that every clause is true under at least one consistent assignment of truth values to the variables. The input f-structure represents this requirement by attaching a value TRUE at the bottom of every clause identifier in the top-level SOL substructure and wherever the clause appears in the problem encoding under PROB. F-structure (41) is the elaboration of (39b) with this additional information.

(41)
$$\left[\begin{array}{ll}
\text{SOL} & \left[\begin{array}{ll} 1_c & [\text{VAL TRUE}] \\ 2_c & [\text{VAL TRUE}] \\ 3_c & [\text{VAL TRUE}] \end{array}\right] \\
\text{PROB} & \left[\begin{array}{ll}
1_p & \left[\begin{array}{ll} + & \left[1_c\ [\text{VAL TRUE}]\right] \\ - & \left[\begin{array}{ll} 2_c & [\text{VAL TRUE}] \\ 3_c & [\text{VAL TRUE}] \end{array}\right] \end{array}\right] \\
2_p & \left[\begin{array}{ll} + & \left[\begin{array}{ll} 1_c & [\text{VAL TRUE}] \\ 3_c & [\text{VAL TRUE}] \end{array}\right] \\ - & \left[2_c\ [\text{VAL TRUE}]\right] \end{array}\right] \\
3_p & \left[\begin{array}{ll} + & \left[\begin{array}{ll} 1_c & [\text{VAL TRUE}] \\ 2_c & [\text{VAL TRUE}] \end{array}\right] \\ - & \left[3_c\ [\text{VAL TRUE}]\right] \end{array}\right]
\end{array}\right]
\end{array}\right]$$

In this depiction the dotted line shows that the clause identifiers and their truth values in the solution are equated to all of their occurrences in the problem-statement substructure.

F-structure (41) is correctly assigned to the satisfiable problem string (39a) if the single Var expansion rule above is replaced by the alternatives in (42).

(42) a.
$$\text{Var} \rightarrow \begin{array}{c} \text{p} \\ \\ \\ \end{array} \begin{array}{c} \text{Snum} \\ (\uparrow \text{B}) = (\downarrow \text{B}) \\ (\downarrow \text{E}) = (\uparrow \text{VE}) \\ \\ \end{array} \left\{ \begin{array}{c} +\quad \text{Snum} \\ (\uparrow \text{VE} +) = (\downarrow \text{B}) \\ (\uparrow \text{SOL}) = (\downarrow \text{B}) \\ (\downarrow \text{E VAL}) = \text{TRUE} \end{array} \middle| \begin{array}{c} -\quad \text{Snum} \\ (\uparrow \text{VE} -) = (\downarrow \text{B}) \\ (\uparrow \text{SOL}) = (\downarrow \text{B}) \\ \\ \end{array} \right\}^{+}$$

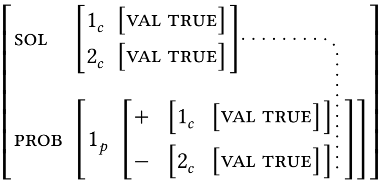
b.
$$\text{Var} \rightarrow \begin{array}{c} \text{p} \\ \\ \\ \end{array} \begin{array}{c} \text{Snum} \\ (\uparrow \text{B}) = (\downarrow \text{B}) \\ (\downarrow \text{E}) = (\uparrow \text{VE}) \\ \\ \end{array} \left\{ \begin{array}{c} +\quad \text{Snum} \\ (\uparrow \text{VE} +) = (\downarrow \text{B}) \\ (\uparrow \text{SOL}) = (\downarrow \text{B}) \\ \\ \end{array} \middle| \begin{array}{c} -\quad \text{Snum} \\ (\uparrow \text{VE} -) = (\downarrow \text{B}) \\ (\uparrow \text{SOL}) = (\downarrow \text{B}) \\ (\downarrow \text{E VAL}) = \text{TRUE} \end{array} \right\}^{+}$$

The SOL annotations in both versions lift all the clause identifiers, whether positive or negative, to the top-level. The rules differ in that (42a) also attaches the value TRUE only at the bottom of every positive-clause chain while (42b) attaches TRUE only to the bottom of every negative clause. Thus for every variable there is a choice in every derivation between the two expansions, corresponding to a guess of consistent truth-value assignments for every variable.

If a problem is satisfiable, then each clause will be assigned TRUE under at least one variable, that value will be carried with the clause identifier into the SOL structure, and it will propagate by equality to all of the other (positive or negative) occurrences of that clause. The result will be an f-structure configured as in (41), and the string corresponding to the problem substructure will be a realization of that f-structure.

Grammar G' will also derive annotated c-structures and f-structures for a string that represents an unsatisfiable problem, but each of those f-structures will be missing a required truth value for at least one of the clauses. For the trivially unsatisfiable problem $p_1 \wedge \neg p_1$ the input f for realization is the f-structure (43a) and (43b) is its corresponding string expression.

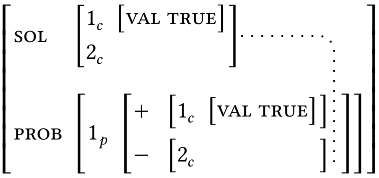
(43) a.



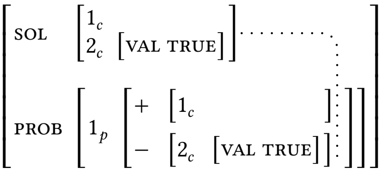
b. p1 +1−2

With just one variable there is only one choice between the alternative Var expansion rules, giving rise to two derivations. Assigning TRUE to the positive literal produces f-structure (44a) and (44b) results if the negative literal is selected. Neither of these is complete for all the attributes and values of (43a) and thus string (43b) (and any other string that corresponds to the problem substructure) does not belong to $Gen_{G'}(f)$.

(44) a.



b.



Any Boolean satisfiability problem can thus be reduced to the realization problem for the simple LFG grammar G' if the problem is translated to an input f-structure that encodes the problem and the requirement of truth for all clauses. A derivation for G' will map a string to that f-structure if and only if the Boolean problem is satisfiable. As for recognition, realization is intractable because the number of derivations whose f-structures must be compared to the input is exponential in the size of the problem, in this case the number of variables it contains.

7 *k*-Bounded LFG grammars and tractability

These intractability results for the conservative, properly anchored derivations of arbitrary grammars raise the question whether there are other formal restrictions that will guarantee that the computationally important problems of recognition and realization can be solved in polynomial time. Seki et al. 1993 first established the connection between a much more restricted subclass of LFG grammars and Linear Context-Free Rewriting Systems (LCFRS), formal systems that can describe only mildly context-sensitive dependencies and for which the recognition problem is tractable (Kallmeyer 2010). The Seki et al. *finite-copying grammars* permit rules with the very limited functional annotations in (45a) provided that all their derivations also satisfy the bounding condition (45b).

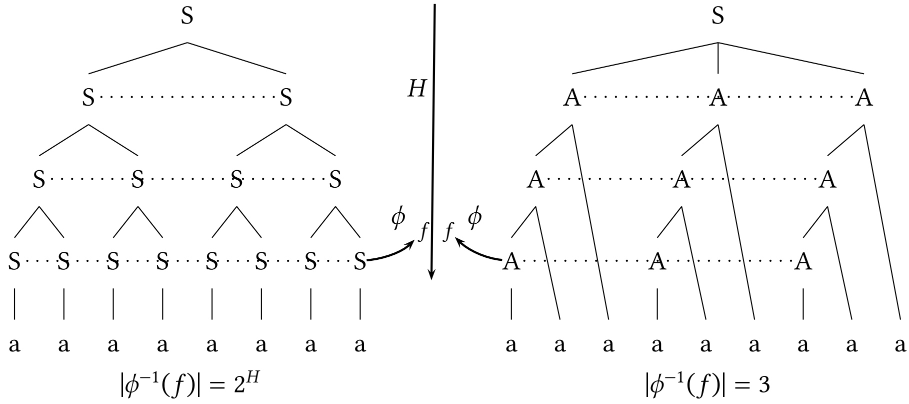
(45) a. Each category on the right-side of a rule can be annotated with at most one function assignment of the form $(\uparrow \textsc{f}) = \downarrow$ and any number of atom-value assignments only of the form $(\uparrow \textsc{a}) = \textsc{v}$.

b. There is a constant k such that no more than k nodes map to the same f-structure element f in any derivation. That is, $|\phi^{-1}(f)| \leq k$ for every f.[19]

Structure sharing in finite-copying grammars can only be achieved through instantiated function-assigning annotations. This specific type of structure sharing is occasionally referred to as “zipper” unification. That is, if two distinct nodes n_1 and n_2 map to the same f-structure in a derivation, then there must always be a node $\hat{n}$ dominating these nodes such that the sequences of function-assigning annotations on the paths from $\hat{n}$ to n_1 and n_2, respectively, must be identical, that is, form a “zipper”.

The bounding condition (45b) limits the number of non-local dependencies that can arise through structure sharing and thus proscribes c-structure recursions that give rise to zippers of size greater than the constant k. Indeed, Seki et al. have shown that the recognition problem is NP-complete for grammars

[19] This condition can also be expressed in terms of an extended-head formulation: $|\phi^{-1} \circ \phi(n)| \leq k$ for every c-structure node n. The parameter k may also be regarded as a formal characterization of the linguistic notion *degree of discontinuity* (Chomsky 1953).



a. C-structure for grammar (46a)

b. C-structure for grammar (46b)

Figure 5: Zipper nodes in depth-balanced c-structures

that meet the notational restrictions (45a) but do not satisfy the bounding condition (45b). Thus the bounding condition is crucial for tractable performance even with the severe notational restrictions of the finite-copying formalism.

Grammars with these limited annotations are expressive enough to specify the kinds of derivations depicted in Figure 5. The derivation on the left is produced by the simple recursive rules in (46a) while the one on the right is derived with the grammar (46b).

(46) a. $\mathrm{S} \rightarrow \underset{(\uparrow \textsc{l})=\downarrow}{\mathrm{S}} \quad \underset{(\uparrow \textsc{l})=\downarrow}{\mathrm{S}} \qquad \mathrm{S} \rightarrow \underset{(\uparrow \textsc{l})=\#}{\mathrm{a}}$

b. $\mathrm{S} \rightarrow \underset{(\uparrow \textsc{l})=\downarrow}{\mathrm{A}} \quad \underset{(\uparrow \textsc{l})=\downarrow}{\mathrm{A}} \quad \underset{(\uparrow \textsc{l})=\downarrow}{\mathrm{A}} \qquad \mathrm{A} \rightarrow \underset{(\uparrow \textsc{l})=\downarrow}{\mathrm{A}} \; \mathrm{a} \qquad \mathrm{A} \rightarrow \underset{(\uparrow \textsc{l})=\#}{\mathrm{a}}$

These grammars both meet the finite-copying notational restrictions (45a), and the derivations of both grammars have nodes that share structure in the zipper configurations indicated by the dotted lines. But the difference in these structure-sharing configurations corresponds to a difference in computational complexity. For all derivations of grammar (46a) the number of nodes in the set $\phi^{-1}(f)$ is an exponential in the height H of those nodes, as indicated in Figure 5a. In contrast, for all derivations of grammar (46b) the number of nodes in a structure-sharing set is bounded by a constant (3 in this case) that is independent of their height (Figure 5b). Grammar (46b) but not (46a) meets the finite-boundedness property

(45b), and this is a decidable property for all derivations of such notationally restricted grammars. Note that the string and f-structure sizes are correlated in the derivations of both grammars. They are thus not distinguished by the conditions of proper anchoring.

The restrictions (45a) are obviously too severe for linguistic description. The notation disallows, for example, the trivial $\uparrow = \downarrow$ annotations that mark the heads and coheads in the functional domain of a predicate, the $(\uparrow \textsc{xcomp subj}) = (\uparrow \textsc{obj})$ equations of functional control, and all other ways of relating the f-structures of different nodes. They also exclude multi-attribute value specifications, such as $(\uparrow \textsc{subj num}) = \textsc{sg}$, that encode agreement requirements, and any direct specification of feature values on daughter nodes, as in $(\downarrow \textsc{case}) = \textsc{nom}$.

Wedekind & Kaplan 2020 take the Seki et al. 1993 finite-copying grammars as the starting point for developing a subclass of LFG grammars that are more suitable for linguistic description but are similarly limited in their expressive power. The *k-bounded* LFG grammars of Wedekind and Kaplan allow the richer set of annotations in (47).

(47) Basic annotations

$\uparrow = \downarrow$	(co)head identifier
$(\uparrow \textsc{f}) = \downarrow$	function assignment
$(\uparrow/\downarrow \textsc{a b c} \cdots) = \textsc{v}$	general atom-value assignments

Reentrancies

$(\uparrow \textsc{f}) = (\uparrow \textsc{h})$	local-topic link
$(\downarrow \textsc{g}) = (\uparrow \textsc{h})$	daughter-mother control
$(\downarrow \textsc{g}) = (\downarrow \textsc{h})$	daughter sharing
$(\downarrow \textsc{g}) = \uparrow$	promotion
$(\uparrow \textsc{f}) = \uparrow$	mother cycle
$(\downarrow \textsc{g}) = \downarrow$	daughter cycle
$(\uparrow \textsc{f g}) = (\uparrow \textsc{h})$	functional control

The annotations in this enlarged set include those that are commonly used in natural language grammars and that remain compatible with theoretical conventions such as the Principle of Functional Locality (Kaplan & Bresnan 1982). In *k*-bounded grammars these more flexible annotations are accompanied with additional conditions that also limit the number of non-local dependences that can arise through structure sharing. The *k*-bounded LFG grammars thus enjoy the same mathematical and computational properties that Seki et. al identified: They characterize only mildly context sensitive languages for which recognition is tractable. The additional conditions that a *k*-bounded grammar *G* must meet are listed in (48).

(48) a. Each right-side category is annotated with at most one function assignment ($\uparrow$ F) = $\downarrow$, and (co)head identifiers $\uparrow = \downarrow$ and function assignments always appear in complementary distribution (to keep separate the properties of heads and their complements).

b. The *functional domains* of G (the collections of $\uparrow = \downarrow$-annotated nodes that map to the same f-structure) are bounded by a grammar-dependent constant h (so G can be converted to an equivalent grammar $G^{\backslash\uparrow=\downarrow}$ that is free of $\uparrow = \downarrow$ annotations).

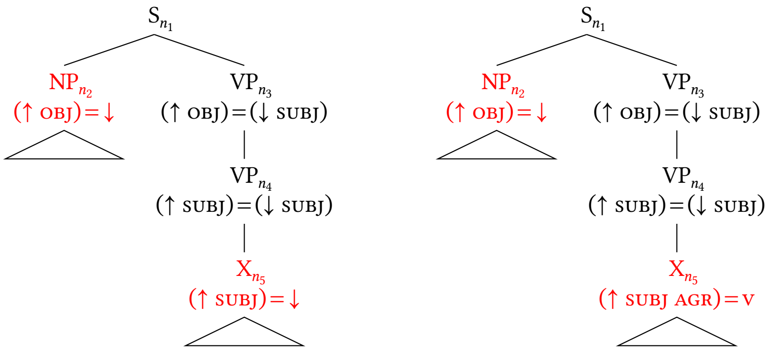
c. The derivations of the grammar formed by removing all reentrancies from $G^{\backslash\uparrow=\downarrow}$ are bounded by a grammar-dependent constant k, as in (45b). (Wedekind & Kaplan 2020 call this the *reentrancy-free kernel* of G.)

d. Reentrancies are nonconstructive.

Nonconstructivity is an implicit property of derivations in broad coverage LFG grammars that has been mentioned (but not well formalize) in the LFG literature as a requirement for functional uncertainty and off-path constraints (Crouch et al. 2008; Zaenen & Kaplan 1995) (Dalrymple et al. 1995: page 133). The reentrancies of a grammar are nonconstructive if they cannot extend the ϕ mapping from c-structure nodes to f-structure units beyond the correspondences established by simple function assignments (the zipper-forming annotations of finite-copying grammars).

The difference between constructive and nonconstructive reentrancies is illustrated in Figure 6. On the left side the reentrancies are constructive because they cause the nodes n_2 and n_5 to map to the same f-structure element. If reentrancies are nonconstructive, as in the derivation on the right side, they do not introduce node-to-f-structure mappings that are not entailed by function assignments alone, and thus they do not affect the bounds that function assignments establish on the ϕ^{-1} node classes. Nonconstructive reentrancies only propagate the limited atom-value information that the grammar attaches to individual nodes and not the unregulated amount of information that might be associated recursively with entire subtrees.

Wedekind & Kaplan 2020 have shown that the nonconstructivity condition (48d) is decidable if the ϕ^{-1} node classes of a grammar are k-bounded and if any two-attribute functional control annotations can be reduced to shorter ones (e.g. shrinking ($\uparrow$ XCOMP SUBJ) = ($\uparrow$ OBJ) to ($\downarrow$ SUBJ) = ($\uparrow$ OBJ) when conjoined with ($\uparrow$ XCOMP) = $\downarrow$). While it is undecidable in general whether every functional control annotation can be shortened (see example (52c) in Appendix A), they can



Constructive

Nonconstructive

$\phi(n_2) = \phi(n_5)$

$(\phi(n_2)$ OBJ AGR$)=$V

Figure 6: Constructive and nonconstructive reentrancies.

always be reduced to daughter-mother controls in derivations that meet the requirements of the Coherence Condition. Wedekind & Kaplan 2020 provide a formal specification of nonconstructivity, this expected consequence of Coherence, and other technical requirements that are sufficient to decide whether an arbitrary LFG grammar belongs to the k-bounded subclass and therefore describes only mildly context-sensitive languages.

Wedekind & Kaplan 2020 also prove that for any LFG grammar G with the properties defined in (47) and (48) there is a linear context free rewriting system that accepts all and only the strings in $L(G)$ and allows recovery of the f-structures that G assigns to each such string. The tractability of LCFRS recognition thus establishes for k-bounded LFG grammars that recognition of individual input strings can be accomplished in time polynomial in their length. Here we sketch a simpler demonstration that is framed entirely within the LFG formalism. This is based on a line of argument that Lang 1994 and others have developed for the recognition problem of context-free grammars.

On this approach to context-free recognition the solution is partitioned into two phases. Given an input string s and an arbitrary context-free grammar G with $|G|$ rules, the first step is to specialize G to a context-free grammar G_s with the property that $s \in L(G)$ if and only if $L(G_s) \neq \emptyset$. The second step then is to determine whether or not the language $L(G_s)$ is empty. In the context-free case the procedure for specializing G to s and the size of the resulting grammar are both polynomial in the length of the input, and for context-free grammars the emptiness problem is bounded by a polynomial in grammar size. It follows on this particular argument (among many others) that context-free recognition is bounded by a polynomial in $|s|$.

This two-part strategy immediately carries over to LFG recognition. The specialization of an arbitrary LFG grammar G to a given input s can be extracted from the chart data structures provided by any number of context-free parsing algorithms modified to keep track of the annotations of matching c-structure categories (equivalently, to operate unmodified on left-side annotated rules as in (6) above). This is a polynomial process that results in an annotated LFG grammar G_s of size also polynomial in $|s|$ that assigns to s all and only the f-structures that G assigns to s. In particular, $Par_G(s) = \emptyset$ if and only if $Par_{G_s}(s) = \emptyset$, and this is equivalent to the question whether $L(G_s) = \emptyset$.

We noted above that the emptiness problem for arbitrary LFG grammars remains undecidable even if only properly anchored derivations are taken into account. However, if G belongs to the subclass of k-bounded grammars then so does G_s, and the emptiness problem for arbitrary k-bounded grammars is not only decidable but solvable with worst-case complexity that is polynomial in grammar size. A proof of this property is outlined in Appendix B. Thus, following the context-free argument, for any input string s and k-bounded LFG grammar G, in time polynomial in $|s|$ it can be determined whether $s \in L(G)$.

Wedekind & Kaplan 2012 applied a similar two-phase strategy to prove that the realization problem is decidable for an arbitrary LFG grammar G and an arbitrary acyclic input f-structure f (see also Kaplan & Wedekind (2000)). They specialized G to a grammar G_f with the property that the string-set $Gen_G(f) = \emptyset$ if and only if $L(G_f) = \emptyset$. The grammar G_f is context-free and its emptiness is therefore decidable. In the general case the specialization phase is not tractable and the resulting G_f may be exponentially larger than G. If G is k-bounded, however, the consistency and completeness of all LFG derivations for any f, even cyclic ones, can be simulated with an annotation-free polynomial expansion of the categories and rules of G.

Thus the recognition and realization problems for k-bounded grammars can be solved in polynomial time: for arbitrary inputs it can be determined whether the sets $Par_G(s)$ and $Gen_G(f)$ are empty. But the k-bounded restrictions are not sufficient to guarantee that those sets contain only a finite number of elements. The context-free grammar G_f, for example, can describe a language with arbitrarily long strings, if G allows for unlimited morphological markers in subtrees with nodes that are not in the domain of the ϕ projection. And the f-structures for a given string can also be arbitrarily large, if the grammar permits stacked recursive subderivations. If useless rules are removed from G_f and if annotations are carried along in the grammar G_s^* as described in Appendix B, then the generation algorithm for context free grammars can be used to enumerate the elements of $Gen_G(f)$ and $Par_G(s)$, one after the other and each in linear time. But obviously the generation and parsing enumerations will never terminate in the face of infinite ambiguity. The derivations for a k-bounded grammar are not necessarily conservative in the sense of (25), even though the emptiness tests for recognition and realization have tractable solutions.

The proper-anchoring/conservation and k-bounded restrictions target different sources of mathematical and computational complexity. Proper anchoring limits the height of recursive subderivations in a stack but imposes no constraint on the number of stacks in a single derivation. The k-bounded restrictions limit the degree of discontinuity but say nothing to relate the sizes of strings and f-structures. The combination of constraints provides for conservative, finitely-ambiguous, derivations with tractable recognition and realization. We have suggested above that conservation is a plausible pretheoretic property of natural communication, and we have also argued that the k-bounded patterns of information flow are compatible with other linguistic principles (Kaplan & Wedekind 2019; Wedekind & Kaplan 2020). The k-bounded restrictions (47-48) and the proper anchoring condition (29) are different ways of moderating the excessive mathematical and computational power of the basic LFG formalism while preserving in different ways its suitability for linguistic description.

8 Summary

Lexical-Functional Grammar is equipped with a simple architecture that formalizes a piecewise correspondence between structures of different types, the phrase-structure trees of the constituent structure and the attribute-value matrices of the functional structure. We have shown that f-structure encodings of the strings of arbitrary context-free grammars can be produced by straightforward application of the formalism's most primitive annotations. From that it follows that recognition/parsing, realization/generation, and other mathematical and computational questions are easily proved to be undecidable.

One source of this excessive power, at least for the recognition and realization problems, is the fact that an unrestricted grammar may establish no systematic relationship between the sizes of input strings and the sizes of corresponding f-structures. This is inconsistent with the Principle of Conservation (22) that we suggest is a pretheoretic property of language as a medium of communication: the derivational machinery that maps in both directions between strings and their f-structures does not add or subtract arbitrary amounts of information.

Problems that relate to specific inputs, including recognition/parsing and realization/generation, become decidable if unconservative derivations are excluded from consideration.

The annotated c-structure is the generative component of the LFG formalism and serves as the intermediary between strings and f-structures. Thus we have proposed a condition on recursive c-structure subderivations that ensures that strings and f-structures stand in a conservative relationship. A derivation is properly anchored if each recursive subderivation is anchored in elements of both the string and f-structure and recursive subderivations in a stack do not share the same anchors. For parsing this condition improves on the original prohibition of derivations with nonbranching dominance chains but applies to the generation problem as well.

The proper anchoring condition is strong enough to ensure decidability but we show that it is not strong enough to guarantee tractability. Tractability for recognition and realization is the computationally important property of the k-bounded LFG grammars and derivations. These grammars are in the class of mildly context-sensitive grammars, even though their derivations are not necessarily conservative. The subclass of LFG grammars and derivations that meet the conditions of both proper anchoring and k-boundedness has attractive mathematical and computational properties and may serve as a better foundation for a formal theory of natural language syntax.

Appendix A: Other undecidable questions

In Section 4 we used the descending attribute-chain string encoding (17) for arbitrary Chomsky Normal Form context-free grammars to prove the undecidability of the realization problem. We apply that same encoding here to show that several more specific properties are undecidable for unrestricted LFG grammars. The start rule (49) follows the pattern laid out earlier in (19). It denotes the ends of the S_1 and S_2 substrings as E_1 and E_2 respectively and includes a place-holder P for grammatical fragments that we will use to encode other decision problems. As noted before, there are no atomic values and therefore no atom-value clashes in the attribute-chain string encodings, and the set of derivations can only be filtered by properties spelled out in P.

(49)
$$\begin{array}{lccc}
\mathrm{S} \;\rightarrow & \mathrm{S}_1 & \mathrm{S}_2 & P \\
 & (\uparrow \textsc{l}) = \downarrow & (\uparrow \textsc{r}) = \downarrow & \\
 & (\downarrow \textsc{b}) = (\uparrow \textsc{b}) & (\downarrow \textsc{b}) = (\uparrow \textsc{b}) & \\
 & (\downarrow \textsc{e}) = (\uparrow \textsc{e}_1) & (\downarrow \textsc{e}) = (\uparrow \textsc{e}_2) &
\end{array}$$

If any realization of P expresses a particular property that is satisfied only if $F(G)$ contains f-structures with equal $\textsc{e}_1$ and $\textsc{e}_2$ values, then that property must be undecidable.

As a first example, the alternative annotations on the terminal # in (50) shows that it is undecidable whether a minimal model satisfies either defining or constraining equalities between two f-structure units.

(50)
$$\begin{array}{lcccc}
\text{S} & \rightarrow & \text{S}_1 & \text{S}_2 & \# \\
 & & (\uparrow \textsc{l}) = \downarrow & (\uparrow \textsc{r}) = \downarrow & \left\{ \begin{array}{c} (\uparrow \textsc{e}_1) = (\uparrow \textsc{e}_2) \\ (\uparrow \textsc{e}_1) =_c (\uparrow \textsc{e}_2) \\ (\uparrow \textsc{e}_1) \neq (\uparrow \textsc{e}_2) \end{array} \right\} \\
 & & (\downarrow \textsc{b}) = (\uparrow \textsc{b}) & (\downarrow \textsc{b}) = (\uparrow \textsc{b}) & \\
 & & (\downarrow \textsc{e}) = (\uparrow \textsc{e}_1) & (\downarrow \textsc{e}) = (\uparrow \textsc{e}_2) &
\end{array}$$

The function assignments on X and Y in (51) show that it is in general undecidable whether there are derivations with nodes that ϕ maps to the same f-structure.

(51)
$$\begin{array}{lccccc}
\text{S} & \rightarrow & \text{S}_1 & \text{S}_2 & \text{X} & \text{Y} \\
 & & (\uparrow \textsc{l}) = \downarrow & (\uparrow \textsc{r}) = \downarrow & (\uparrow \textsc{e}_1) = \downarrow & (\uparrow \textsc{e}_2) = \downarrow \\
 & & (\downarrow \textsc{b}) = (\uparrow \textsc{b}) & (\downarrow \textsc{b}) = (\uparrow \textsc{b}) & (\uparrow \textsc{e}_1) = (\uparrow \textsc{e}_2) & \\
 & & (\downarrow \textsc{e}) = (\uparrow \textsc{e}_1) & (\downarrow \textsc{e}) = (\uparrow \textsc{e}_2) & &
\end{array}$$

It follows from this that any other property that depends on nodes mapping to the same f-structure is also undecidable.

Thus, expanding the nonterminals X and Y with the rules (52a) shows that the satisfiability of existential constraints or constraints between atomic values is undecidable and, as a consequence, that Completeness and Coherence are also undecidable. The annotations (52b) establish that it is undecidable whether an arbitrary LFG grammar gives rise to cyclic f-structures, and (52c) shows that functional control annotations cannot decidably be reduced to combinations of function assignments and daughter-mother controls.

(52) a.
$$\begin{array}{lcc}
\text{X} & \rightarrow & x \\
 & & (\uparrow \textsc{f}) = \textsc{v}
\end{array}
\qquad
\begin{array}{lcc}
\text{Y} & \rightarrow & y \\
 & & \left\{ \begin{array}{c} (\uparrow \textsc{f}) \\ \neg(\uparrow \textsc{f}) \\ (\uparrow \textsc{f}) =_c \textsc{v} \\ (\uparrow \textsc{f}) \neq \textsc{v} \end{array} \right\}
\end{array}$$

b.
$$\begin{array}{lcc}
\text{X} & \rightarrow & x \\
 & & (\uparrow \textsc{f g}) = (\uparrow \textsc{h})
\end{array}
\qquad
\begin{array}{lcc}
\text{Y} & \rightarrow & y \\
 & & (\uparrow \textsc{f}) = (\uparrow \textsc{h})
\end{array}$$

c.
$$\begin{array}{lcc}
\text{X} & \rightarrow & x \\
 & & (\uparrow \textsc{f g}) = (\uparrow \textsc{h})
\end{array}
\qquad
\begin{array}{lcc}
\text{Y} & \rightarrow & y \\
 & & (\uparrow \textsc{f}) = \downarrow
\end{array}$$

Appendix B: Emptiness of k-bounded LFG grammars

We sketch here the proof that the complexity of the emptiness problem for an arbitrary k-bounded LFG grammar G is polynomial in $|G|$, the size of its rule set. The argument makes use of the three grammar transformations listed in (53). Each of these can be carried out in polynomial time, as indicated below, and each guarantees that G and the transformed grammar G' are *co-empty*, that is, that the set of derivations $\Delta_G = \emptyset$ if and only $\Delta_{G'} = \emptyset$.

(53) a. $\uparrow=\downarrow$ removal: For any k-bounded LFG grammar G there is a co-empty $\uparrow=\downarrow$-free k-bounded grammar $G^{\setminus\uparrow=\downarrow}$.

b. Zipper removal: For any $\uparrow=\downarrow$-free k-bound LFG grammar G there is a co-empty 1-bounded (zipper-free) LFG grammar G^z.[20]

c. Annotation removal: For any 1-bounded LFG grammar G there is a co-empty annotation-free grammar G^a, and G^a is context-free.

Applying these transformations in sequence to an arbitrary k-bounded LFG grammar G results in a co-empty context free grammar $G^* = G^{\setminus\uparrow=\downarrow,z,a}$ whose size $|G^*|$ is a polynomial function of $|G|$. The string-set $L(G) = \emptyset$ if and only if the context free language $L(G^*) = \emptyset$, and this can be determined by the well-known emptiness algorithm for context free grammars, which is polynomial in the size of the grammar.

For (53a), the $\uparrow=\downarrow$ annotations in an arbitrary k-bounded grammar G are eliminated by replacing each $\uparrow=\downarrow$-annotated category in one rule with the right-side of each of the rules that expand that category. Let R be the smallest set that includes the rules of G and is closed under the convention (54). In this template δ, θ, and ψ are strings of annotated categories, and α may be a set of annotations with $\uparrow$ substituted for $\downarrow$.

(54) If R contains rules of the form

$$\mathrm{A} \rightarrow \delta \underset{\alpha}{\underset{\uparrow=\downarrow}{\mathrm{B}}} \theta \quad \text{and} \quad \mathrm{B} \rightarrow \psi$$

then R also contains the rule $\mathrm{A} \rightarrow \delta \underset{\alpha}{\psi} \theta$

The $\uparrow=\downarrow$-free grammar $G^{\setminus\uparrow=\downarrow}$ is constructed by removing from R any rules with $\uparrow=\downarrow$ annotations. Note that a replacement sequence can never be longer than the

[20] Unlike the transformations that are often used in proofs of other formal-language properties, zipper removal does not preserve the language $L(G)$: the grammars G and G' generally are not weakly equivalent.

limit on the number of nodes in a functional domain, the parameter h of condition (48b). As a consequence, the growth of the grammar is bounded by a polynomial in $|G|$. Moreover, the resulting grammar $G^{\setminus\uparrow=\downarrow}$ accepts exactly the same strings as G and assigns them the same f-structures, although with c-structures that are not as deep.

For (53b), the rules of a zipper-free 1-bounded grammar are created from sets of up to k rules of a $\uparrow=\downarrow$-free k-bounded grammar G. The zipper daughters, occurrences of right-side categories with the same function assignments, are replaced with a single new daughter labeled by the concatenation (notated with ·) of the labels of the zipper daughters and annotated with the union of the zipper-daughter annotations. Let R now be the smallest set that includes the rules of a $\uparrow=\downarrow$-free grammar G and is closed under the following:

(55) a. If $\mathrm{A}_1 \rightarrow \delta_1, \ldots, \mathrm{A}_j \rightarrow \delta_j \quad (1 \leq j \leq k)$ are rules in R,

then R also contains the rule $\mathrm{A}_1 \cdot \ldots \cdot \mathrm{A}_j \rightarrow \delta_1 \ldots \delta_j$

b. If R contains a rule of the form

$$\begin{array}{cccccc} \mathrm{A} \rightarrow & \delta & \mathrm{B}_1 & \theta & \mathrm{B}_2 & \psi \\ & & (\uparrow \textsc{f})=\downarrow & & (\uparrow \textsc{f})=\downarrow & \\ & & \alpha_1 & & \alpha_2 & \end{array}$$

then R also contains the rule $\begin{array}{cccc} \mathrm{A} \rightarrow & \mathrm{B}_1 \cdot \mathrm{B}_2 & \delta\ \theta\ \psi \\ & (\uparrow \textsc{f})=\downarrow & \\ & \alpha_1\ \ \alpha_2 & \end{array}$

The zipper-free grammar G^z is then created by removing from R any rule with multiple assignments for the same function or with annotations that are locally unsatisfiable. Local (within-rule) satisfiability of a rule with n daughters is tested by instantiating all metavariables with distinct constants $b_0, b_1, \ldots, b_n$ that stand for a putative mother node and its daughters. b_0 is substitute for ↑ in all annotations and b_i is substituted for ↓ in the annotations of the i^{th} daughter. The local f-description thus created is then solved using standard deductive-closure techniques.

The size of a zipper-free grammar G^z is exponential in k but polynomial in $|G|$, because there are at most $|G|^k$ rule combinations that must be considered. For every derivation in G of a string s with discontinuous subtrees for a particular grammatical function there is a corresponding derivation in G^z that assigns the same f-structure to a string s^z. The two strings contain the same words but not necessarily in the same order: the words are permuted so that the words of discontinuous subtrees for s are contiguous in s^z.

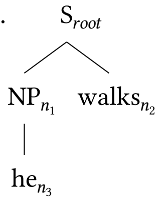
The annotation-removal transformation (53c) is based on the fact that atomic values in a 1-bounded grammar can only propagate between mothers and daughters within a single subtree. This is because, by definition, there are no nodes n and n' in separate subtrees with $\phi(n) = \phi(n')$. Atomic values in sister subtrees may have different values for a particular feature, but that can only result in an overall unsatisfiable f-description if annotation chains relative to a common mother put them in contact. Chains of atom-value annotations carried by the categories of a 1-bounded LFG derivation can be simulated by an elaborated set of refined c-structure categories in a corresponding annotation-free derivation. An annotation-free derivation is context-free and will fail if and only if the f-description for the 1-bounded LFG derivation is unsatisfiable.

The ↑=↓-free and zipper-free rules in (56) provide the derivation (57a) for the sentence *He walks.*

(56)
$$\begin{array}{llcc}
\text{S} & \rightarrow & \text{NP} & \text{walks} \\
 & & (\uparrow \textsc{subj}) = \downarrow & (\uparrow \textsc{pred}) = \text{`}\textsc{walk}\langle\textsc{subj}\rangle\text{'} \\
 & & (\downarrow \textsc{case}) = \textsc{nom} & (\uparrow \textsc{tense}) = \textsc{pres} \\
 & & & (\uparrow \textsc{subj num}) = \textsc{sg}
\end{array}
\qquad
\begin{array}{llc}
\text{NP} & \rightarrow & \text{he} \\
 & & (\uparrow \textsc{pred}) = \text{`}\textsc{pro}\text{'} \\
 & & (\uparrow \textsc{gend}) = \textsc{m} \\
 & & (\uparrow \textsc{num}) = \textsc{sg} \\
 & & (\uparrow \textsc{case}) = \textsc{nom}
\end{array}$$

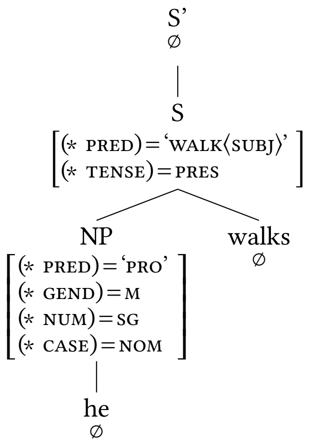
The f-description is satisfiable because the case assigned to the subject NP matches the case of *he*, and the subject's number, entailed by the combination (↑ SUBJ)=↓ and (↑ SUBJ NUM)=SG, also matches the number of *he*. The connection between the S and NP feature annotations is simulated by the refined NP category in (57b).

(57) a.



$$_{root}\left[\begin{array}{ll}
\textsc{subj} & _{n_1}\left[\begin{array}{ll}
\textsc{pred} & \text{`}\textit{pro}\text{'} \\
\textsc{gend} & m \\
\textsc{num} & sg \\
\textsc{case} & nom
\end{array}\right] \\
\textsc{pred} & \text{`}\textit{walk}\langle\textit{subj}\rangle\text{'} \\
\textsc{tense} & pres
\end{array}\right]$$

b.



Starting from a new category S’, tree (57b) is the context free derivation provided by the category-refined, annotation-free rules in (58).

(58) $\underset{\varnothing}{\mathrm{S'}} \rightarrow \underset{\left[\begin{array}{l}(\ast\ \textsc{pred}) = \text{‘}\textsc{walk}\langle\textsc{subj}\rangle\text{’}\\ (\ast\ \textsc{tense}) = \textsc{pres}\end{array}\right]}{\mathrm{S}}$

$$\underset{\left[\begin{array}{l}(\ast\ \textsc{pred}) = \text{‘}\textsc{walk}\langle\textsc{subj}\rangle\text{’}\\ (\ast\ \textsc{tense}) = \textsc{pres}\end{array}\right]}{\mathrm{S}} \rightarrow \underset{\left[\begin{array}{l}(\ast\ \textsc{pred}) = \text{‘}\textsc{pro}\text{’}\\ (\ast\ \textsc{gend}) = \textsc{m}\\ (\ast\ \textsc{num}) = \textsc{sg}\\ (\ast\ \textsc{case}) = \textsc{nom}\end{array}\right]}{\mathrm{NP}} \quad \underset{\varnothing}{\text{walks}} \qquad \underset{\left[\begin{array}{l}(\ast\ \textsc{pred}) = \text{‘}\textsc{pro}\text{’}\\ (\ast\ \textsc{gend}) = \textsc{m}\\ (\ast\ \textsc{num}) = \textsc{sg}\\ (\ast\ \textsc{case}) = \textsc{nom}\end{array}\right]}{\mathrm{NP}} \rightarrow \underset{\varnothing}{\text{he}}$$

Note that the c-structure derivation for the string *Him walks* would have an unsatisfiable f-description. The corresponding category mismatch excludes a derivation with refined categories.

For an arbitrary 1-bounded grammar G the co-empty annotation-free grammar G^a produces derivation trees whose nodes are labeled with refined categories of this form. A refined category is a pair $c{:}m$ consisting of a c-structure category label c of G together with a refinement matrix m of atom-value feature specifiers $(\ast\ \textsc{p q r} \ldots) = \textsc{v}$. The feature specifiers simulate in a G^a derivation the possible interactions of atomic values in the f-description of a corresponding G derivation, as illustrated. Importantly, Wedekind & Kaplan 2020 show that a finite set of specifiers is sufficient to simulate all possible atom-value interactions. These are the specifiers containing no more than ℓ of G’s attributes, where ℓ is the number of attributes in the longest atom-value assignment in G.

Let N be the smallest set of refined categories and let R be the smallest set of refined rules, rules with refined-category labels, that are closed under the following conditions (see Wedekind & Kaplan 2020 for additional technical details).

(59) a. If S is the start symbol of G and S' is a category distinct from other G categories, N contains S:∅ and S’:∅ and R contains S’:∅ → S:∅.

b. If *term* is a terminal symbol of G, N contains *term*:∅.

c. If r is a refinement of a rule $A_0 \rightarrow \underset{\alpha_1}{A_1} \ldots \underset{\alpha_n}{A_n}$ of G with a sequence of refined categories $A_0{:}m_0, \ldots, A_n{:}m_n$ in N, then R contains r and N contains the refined categories of r.

The refinement of a rule $A_0 \rightarrow \underset{\alpha_1}{A_1} \ldots \underset{\alpha_n}{A_n}$ of G with a sequence of refined categories $A_0{:}m_0, \ldots, A_n{:}m_n$ is produced by instantiating the ↑ and ↓ metavariables with distinct mother-daughter constants $b_0, b_1, \ldots, b_n$, as above, but also including in the local f-description atom-value equations instantiated from the feature-specifier matrices. The additional equations are created by substituting b_i for all of the asterisks in each m_i. A refined rule r is constructed if this augmented f-description is satisfiable. Each category A_i in the original G rule (including the mother category) is replaced by a refined category $A_i{:}m_i'$ where the feature specifiers of m_i' are formed by substituting * for b_i in each length-limited atom-value equation (b_i P Q R…)=V that the f-description entails. The newly refined categories are added to N.

The annotation-free grammar G^a is then constructed in the following way. S’:∅ is its starting category, its terminals categories are of the form *term*:∅ for each terminal *term* of G, and its context-free rules are constructed from the refined rules in R by using standard context-free algorithms to eliminate useless rules, those that cannot participate in successful derivations, and then removing their functional annotations. The context-free derivations in G^a correspond to all and only the c-structures of G with satisfiable f-descriptions. Because the feature specifiers in a refined category are limited in length by the grammar parameter ℓ, $|G^a|$ is only polynomially larger than $|G|$ and its emptiness can be determined in polynomial time. We also note that if the annotations are not removed from the useful rules of R, the set of f-structures for a grammar with those still-annotated rules will be exactly the f-structures of G.

References

Berwick, Robert C. 1982. Computational complexity and Lexical-Functional Grammar. *American Journal of Computational Linguistics* 8(3–4). 97–109. DOI: 10.3115/981923.981926.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2016. *Lexical-Functional Syntax*. 2nd edn. (Blackwell Textbooks in Linguistics 16). Malden, MA: Wiley-Blackwell.

Bresnan, Joan, Ronald M. Kaplan, Stanley Peters & Annie Zaenen. 1982. Cross-serial dependencies in Dutch. *Linguistic Inquiry* 13(4). 613–635. https://www.jstor.org/stable/4178298. Reprinted in Savitch, Bach, Marsh & Safran-Naveh (1987: 286-319).

Chomsky, Noam. 1953. Systems of syntactic analysis. *Journal of Symbolic Logic* 18(3). 242–256. DOI: 10.2307/2267409.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III & Paula Newman. 2008. *XLE Documentation*. Xerox Palo Alto Research Center. Palo Alto, CA. https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle_toc.html.

Culy, Christopher D. 1985. The complexity of the vocabulary of Bambara. *Linguistics and Philosophy* 8. 345–351.

Dalrymple, Mary (ed.). 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication). Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/6169.001.0001.

Dalrymple, Mary. 2001. *Lexical Functional Grammar* (Syntax and Semantics 34). New York: Academic Press. DOI: 10.1163/9781849500104.

Dalrymple, Mary, Ronald M. Kaplan & Tracy Holloway King. 2015. Economy of Expression as a principle of syntax. *Journal of Language Modelling* 3(2). 377–412. DOI: 10.15398/jlm.v3i2.82.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.

Dalrymple, Mary, John Lamping & Vijay Saraswat. 1993. LFG semantics via constraints. In *Proceedings of the 6th conference of the European chapter of the ACL (EACL 1993)*, 97–105. Association for Computational Linguistics. DOI: 10.3115/976744.976757.

Fang, Ji & Peter Sells. 2007. A formal analysis of the verb copy construction in Chinese. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '07 conference*, 198–213. Stanford: CSLI Publications.

Halvorsen, Per-Kristian & Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, 1116–1122. Tokyo. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 279–292).

Haug, Dag & Tatiana Nikitina. 2012. The many cases of non-finite subjects: The challenge of "dominant" participles. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 292–311. Stanford: CSLI Publications.

Jaeger, Efrat, Nissim Francez & Shuly Wintner. 2005. Unification grammars and off-line parsability. *Journal of Logic, Language and Information* 14(2). 199–234. DOI: 10.1007/s10849-005-4511-1.

Johnson, Mark. 1986. The LFG treatment of discontinuity and the double infinitive construction in Dutch. In *Proceedings of the 5th West Coast Conference on Formal Linguistics*, 102–118. Stanford: CSLI Publications.

Johnson, Mark. 1988. *Attribute-value logic and the theory of grammar.* Stanford: CSLI Publications.

Kallmeyer, Laura. 2010. On mildly context-sensitive non-linear rewriting. *Research on Language and Computation* 8. 341–363. DOI: 10.1007/s11168-011-9081-6.

Kaplan, Ronald M. 1987. Three seductions of computational psycholinguistics. In Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod Johnson & Paul Bennett (eds.), *Linguistic theory and computer applications*, 149–188. London: Academic Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 339–367).

Kaplan, Ronald M. 2019. Formal aspects of underspecified features. In Cleo Condoravdi & Tracy Holloway King (eds.), *Tokens of meaning: Papers in honor of Lauri Karttunen*, 349–369. Stanford: CSLI Publications.

Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 173–281. Cambridge, MA: The MIT Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 29–130).

Kaplan, Ronald M. & John T. Maxwell III. 1988. Constituent coordination in Lexical-Functional Grammar. In *COLING '88: Proceedings of the 12th Conference on Computational Linguistics*, 303–305. Budapest. DOI: 10.3115/991635.991696. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 199–210).

Kaplan, Ronald M. & John T. Maxwell III. 1996. *LFG Grammar Writer's Workbench.* Xerox Palo Alto Research Center. Palo Alto, CA. https://www.researchgate.net/profile/John_Maxwell5/publication/2760068_Grammar_Writer's_Workbench/links/0c96052405e97928e9000000.pdf.

Kaplan, Ronald M. & Jürgen Wedekind. 2000. LFG generation produces context-free languages. In *COLING 2000 volume 1: the 18th International Conference on Computational Linguistics*, 425–431. DOI: 10.3115/990820.990882.

Kaplan, Ronald M. & Jürgen Wedekind. 2019. Tractability and discontinuity. In Miriam Butt, Tracy Holloway King & Ida Toivonen (eds.), *Proceedings of the LFG '19 Conference*, 130–148. Stanford: CSLI Publications.

Kaplan, Ronald M. & Jürgen Wedekind. 2020. Zipper-driven parsing for LFG grammars. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 169–189. Stanford: CSLI Publications.

Kaplan, Ronald M. & Jürgen Wedekind. Forthcoming. Formal and computational properties of LFG. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Kaplan, Ronald M. & Annie Zaenen. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In Mark Baltin & Anthony Kroch (eds.), *Alternative conceptions of phrase structure*, 17–42. Chicago: University of Chicago Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 137–165).

Kuhn, Jonas. 2003. *Optimality-Theoretic Syntax – A declarative approach*. Stanford: CSLI Publications.

Lang, Bernard. 1994. Recognition can be harder than parsing. *Computational Intelligence* 10(4). 486–494.

Maxwell, John T., III & Ronald M. Kaplan. 1996. Unification-based parsers that automatically take advantage of context freeness. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '96 conference*, 1–31. Stanford: CSLI Publications.

Nishino, Tetsuro. 1991. Mathematical analysis of Lexical-Functional Grammars —complexity, parsability, and learnability. *Language Research* 27(1). 867–915.

Przepiórkowski, Adam & Agnieszka Patejuk. 2012. The puzzle of case agreement between numeral phrases and predicative adjectives in Polish. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 490–502. Stanford: CSLI Publications.

Roach, Kelly. 1983. LFG languages over a one-letter alphabet. Unpublished manuscript, Xerox Palo Alto Research Center.

Savitch, Walter J., Emmon Bach, William Marsh & Gila Safran-Naveh (eds.). 1987. *The formal complexity of natural language* (Studies in Linguistics and Philosophy). Dordrecht: Springer.

Seki, Hiroyuki, Ryuichi Nakanishi, Yuichi Kaji, Sachiko Ando & Tadao Kasami. 1993. Parallel multiple context-free grammars, finite-state translation systems, and polynomial-time recognizable subclasses of Lexical-Functional Grammars. In *Proceedings of the 31st annual meeting of the Association for Computational Linguistics*, 130–139. Columbus, OH: Association for Computational Linguistics. DOI: 10.3115/981574.981592.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8(3). 333–343. DOI: 10.1007/978-94-009-3401-6_12.

Wedekind, Jürgen. 1999. Semantic-driven generation with LFG- and PATR-style grammar. *Computational Linguistics* 25(2). 277–281. https://www.aclweb.org/anthology/J99-2006.

Wedekind, Jürgen. 2014. On the universal generation problem for unification grammars. *Computational Linguistics* 40(3). 533–538. DOI: 10.1162/coli_a_00191.

Wedekind, Jürgen & Ronald M. Kaplan. 1996. Ambiguity-preserving generation with LFG- and PATR-style grammars. *Computational Linguistics* 22(4). 555–558. https://www.aclweb.org/anthology/J96-4005.

Wedekind, Jürgen & Ronald M. Kaplan. 2012. LFG generation by grammar specialization. *Computational Linguistics* 38(4). 867–915. DOI: 10.1162/coli_a_00113.

Wedekind, Jürgen & Ronald M. Kaplan. 2020. Tractable Lexical-Functional Grammar. *Computational Linguistics* 46(2). 515–569. DOI: 10.1162/coli_a_00384.

Wedekind, Jürgen & Ronald M. Kaplan. 2021. LFG generation from acyclic f-structures is NP-hard. *Computational Linguistics*. DOI: 10.1162/coli_a_00419.

Zaenen, Annie & Ronald M. Kaplan. 1995. Formal devices for linguistic generalizations: West Germanic word order in LFG. In Jennifer S. Cole, Georgia M. Green & Jerry L. Morgan (eds.), *Linguistics and computation*, 3–27. Stanford: CSLI Publications. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 215–240).

Zweigenbaum, Pierre. 1988. *Attributive adjectives, adjuncts and cyclic f-structures in Lexical-Functional Grammar*. DIAM Rapport Interne RI-58a. Paris: Departement Intelligence Artificielle et Medecine.

Chapter 4

LFG and Dependency Grammar

Dag Haug
University of Oslo

This chapter discusses Dependency Grammar from the perspective of LFG. We first introduce the key ideas behind Dependency Grammar and how they relate to LFG concepts. We then show how both LFGs and Dependency Grammars can be translated into Multiple Context-Free Grammars to study formal differences between the frameworks. Next we discuss two recent efforts to translate from LFG analyses to the version of Dependency Grammar adopted in Universal Dependencies. Finally we show how Glue semantics can be applied to dependency structures.

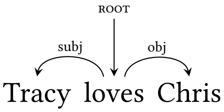
1 Introduction

Dependency grammar (DG) is a tradition for syntactic analysis based on binary, asymmetric relations (called dependency relations or just *dependencies*) between words. These relations are typically labelled, giving rise to a set of labels that can be thought of as grammatical functions, which are of course also important in LFG. In fact, the correspondence between dependencies in DG and grammatical functions in LFG and their central role in both theories is the main similarity, formally and conceptually, between the two frameworks.

The primacy of dependencies is what holds together work in the DG tradition. As we will see, it is characteristic of almost all DG theories that they acknowledge a level of syntax that we will call the *core dependencies*. This is a set of dependencies restricted so as to form a tree over the words of a sentence, i.e. a structure where each word has exactly one head, except the root word, which has none (or equivalently, is attached to a synthetic root node). (1) shows a very simple example of this.

Dag Haug. Forthcoming. LFG and Dependency Grammar. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

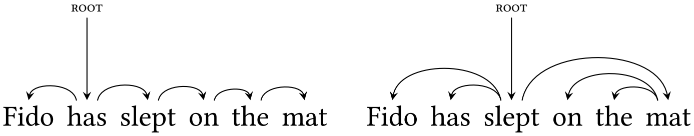
(1)



Most theoretical work and concrete analyses have seen the need to introduce additional mechanisms or levels of structure beyond core dependencies to give the theory more analytical bite; this goes all the way back to Tesnière (1959), the founding work of modern DG. But there is typically little agreement about the additional mechanisms or levels of structure between individual scholars working in the DG tradition. So, while the core dependency representation is often acknowledged as theoretically inadequate, it has enjoyed considerable popularity as a simplified representation with practical applications in computational linguistics and natural language processing.

But even restricting attention to core dependencies, there are a number of choice points where different dependency frameworks make different decisions. For example, if we model structures with a lexical word and one or more function words (for example, articles and nouns, auxiliaries and full verbs, or prepositions and their complements) in the core dependencies, we must take a stance on whether the lexical or the functional word is the head: the co-head option often used in LFG is not available. (2) shows what the (unlabelled) dependency structure of a simple sentence would look like if we take function words as heads (left) or lexical words as heads (right).

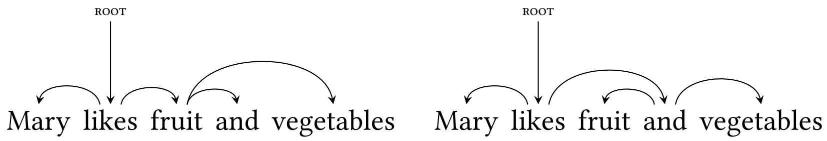
(2)



It is obviously not necessary to treat all function words the same, and so there are intermediate variants between these two extremes, taking for example prepositions and articles as heads, but not auxiliary verbs.

Another point at which dependency grammarians diverge is the treatment of coordination. Because coordination is normally thought of as symmetric, it is not easy to represent with directed dependencies. Here the most common competing analyses, shown in (3), involve taking the first conjunct as the head (left), which entails giving up on symmetry; or to make the conjunction the head (right) and maintain symmetry, but at the cost of dissociating the conjuncts from their normal head (e.g. the verb), which is the basis for most morphsyntactic and semantic constraints.

(3)


ROOT
Mary likes fruit and vegetables
ROOT
Mary likes fruit and vegetables

Faced with the choices illustrated in (2) and (3) many linguists in the DG tradition have felt that neither analysis is satisfactory, and they have therefore reacted by enriching the dependency formalism in various ways that result in data structures that have more in common with LFG. We discuss some key examples of this in Section 2. But even if much theoretical work in DG assumes such enriched data structures, most practical applications of DG rely on core dependencies, thereby forcing choices that, at least from an LFG perspective, are somewhat arbitrary.

One key difference between DG and LFG is that dependency grammarians typically do not formalize their work and in many cases do not provide (even informal) rules that generate the constructions they are interested in but content themselves with providing analyses of the whole structure. This goes back to the earliest dependency grammarians such as Tesnière, but has become even more prominent with the increasing use of dependency structures in data-driven parsing, where the goal is not to define a grammar that recognizes (or generates strings from) a formal language, but to parse strings into a single plausible structural representation. Nevertheless, it *is* possible to conceive of DGs as formal grammars. In Section 3 we discuss how this can be done using the framework of Multiple Context-Free Grammars. While this is not an approach that most dependency grammarians follow, it yields a useful a framework for comparing DG and LFG. Another useful perspective on DG and LFG is offered by recent efforts to translate LFG resources into DG resources, which we discuss in Section 4. Section 5 explores the potential for combining dependency grammars with Glue semantics, the standard semantic framework in LFG.

2 The dependency grammar tradition and LFG

The idea of using binary, labelled, asymmetric relations to analyse syntax is found in the work of Pāṇini, Ancient Greek and Roman grammarians and the speculative grammarians of the Middle Ages (Covington 1984). On its own, this idea is too vague to define a theoretical framework and both Pāṇini and the speculative grammarians have also been seen as forerunners of generative grammar (Kiparsky 1993; Chomsky 1966). What defines the modern dependency grammar tradition, which started with Tesnière (1959), is the attempt to base syntax primarily, or even exclusively, on the concept of core dependencies, as opposed to the concept of constituency developed in American structuralism and the generative tradition. Although there have been a number of attempts to develop dependency grammar into a full-fledged grammatical theory (the most well-known ones being Functional Generative Description (Sgall et al. 1986); Meaning–Text Theory (MTT) (Mel'čuk 1988); and Word Grammar (Hudson 1984, 2010)), none of these are very widespread beyond the environments where they originated and hence there is no single, coherent version of DG as a formal framework. The focus of this section is therefore not to identify assumptions made in specific frameworks, but rather to compare ideas that are common in the dependency grammar tradition with LFG.

2.1 Dependency graphs and f-structures

There is an obvious similarity between dependencies, as found in DG, and the binary, labelled, asymmetric relations between the nodes of an LFG f-structure.[1] In both cases, the relations form a directed labelled graph over nodes corresponding to linguistic material. The similarity even extends to the set of labels used, which in both cases contain traditional grammatical functions such as subject and object. Formally, however, there are two important differences: First, the nodes of the f-structure are not words, but correspond to zero, one or several words/c-structure terminals. This is how LFG escapes the indeterminacy of direction of headedness in constructions which combine lexical and functional words that we saw in (2.) Second, labelled dependencies are not necessarily functional, i.e. there may be two or more daughters bearing the same relation to the same head, in violation of LFG's uniqueness condition.[2]

In addition to these two formal differences, there are in practice many more differences, because DG analyses rarely use the full power of a directed graph and instead typically emphasize the core dependencies, which form a tree spanning the words of the sentence. To the extent that e.g. multiple heads are used, one of the heads is typically considered "primary". Even so, the formal similarities between dependencies and f-structures mean that similar theoretical questions can arise in both DG and LFG and even that one can think of LFG's f-structures as

[1] To emphasize the parallelism between f-structures and dependency graphs, we rely here on the graph-theoretic interpretation of attribute-value matrices, where feature-structures and atomic values are nodes, and attributes are labelled edges between these nodes, and not the "official" interpretation of f-structures as functions (Kaplan 1995; Kaplan & Bresnan 1982). The graph-theoretic interpretation is standard in HPSG; see Przepiórkowski forthcoming: section 4 [this volume] for discussion of the differences between the two views.

[2] LFG can deal with several dependents bearing the same relation by using set-valued attributes e.g. for ADJUNCT; this introduces the concept of sets, which also has no counterpart in DG.

dependency graphs that take a particular view on certain foundational questions in DG.[3]

An overarching question in the DG tradition (see e.g. de Marneffe & Nivre 2019: 199f.) is whether dependency relations are sufficient for analyzing syntax. In one sense, the answer is obviously no. Like f-structures, dependency structures say nothing about word order. This is dealt with in the c-structure in LFG, and scholars within dependency grammar have also seen the need to enrich the theory with a mechanism for constraining word order. We return to this in Section 3. But more fundamentally, one might ask whether core dependencies, tree structures over words, are sufficient to capture functional aspects of syntax like f-structures do in LFG.

In fact, it is not too hard to see that core dependencies cannot fully represent the functional relations of a sentence. Consider for example, the subject in a raising construction.

(4) It seems to rain.

The expletive *it* bears a functional relation to the raising verb *seems* as witnessed by agreement; but the form of the expletive is licensed by the lexical verb *rain* (and would be different in e.g. *There seems to be a problem*), giving evidence for a second functional relation. If one insists on core dependencies, one of the two relations must be privileged.

The alternative is to increase the expressivity of the theory, and this is in fact what Tesnière did when he introduced two other kinds of relations beside dependencies that can hold between words, namely junction (*jonction*) and transfer (*translation*). Junction is the relation that holds between coordinated items that are either dependents of the same head or heads of the same dependent. Translation is the relation that holds between lexical words and functional words that license their appearing in various dependencies. For example, complementizers "translate" verbs so as to license their appearing in object position according to the analysis in Tesnière (1959: 24); similar analyses are given for determiners and adpositions.

Crucially, words that are linked by junction or transfer form a complex node (*nucleus dissocié*) in the dependency graph and jointly contract dependency relations. In this way, their dependents end up having more than one head; and they can collectively bear a single dependency relation to their head. In this respect,

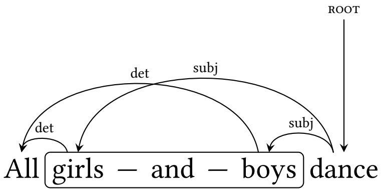
[3] Furthermore, on the implementation side, Bröker (1998) shows how DGs can be encoded as LFGs and implemented in the XLE platform.

Tesnière's analyses are in fact quite close to standard LFG f-structures, where coordination is analysed in terms of a set-valued attribute (5) and function words such as e.g. auxiliaries form a single f-struture node with their lexical verb (6).

(5) a. All boys and girls dance.

b.
$$\left[\begin{array}{ll} \text{PRED} & \text{`DANCE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \left\{\begin{array}{l} \left[\begin{array}{ll} \text{PRED} & \text{`BOY'} \\ \text{SPEC} & \boxed{1}\left[\begin{array}{ll}\text{PRED} & \text{`ALL'}\end{array}\right]\end{array}\right] \\ \\ \left[\begin{array}{ll}\text{PRED} & \text{`GIRL'} \\ \text{SPEC} & \boxed{1}\end{array}\right]\end{array}\right\}\end{array}\right]$$

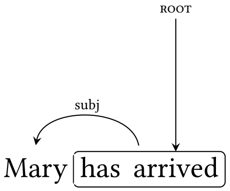
c.



(6) a. Mary has arrived.

b.
$$\left[\begin{array}{ll} \text{PRED} & \text{`ARRIVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PERFECT} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`MARY'}\end{array}\right]\end{array}\right]$$

c.



In this respect, both Tesnière's theory and LFG's f-structure reject the idea that syntactic dependencies can be adequately captured in a tree structure over the words of a sentence. Nevertheless, LFG's approach is much more general than Tesnière's. Tesnière allows many-to-one relations between words and dependency nodes based on relations that are not dependencies, but he maintains the tree structure over dependency nodes. Therefore, the only way a word can have two heads is if those heads form a single node by junction or transfer, as in (5c) and (6c); but LFG also allows for a word to have two heads that do not form a group, as in the analysis of functional control verbs (7).

(7) a. Chris persuaded Mary to come

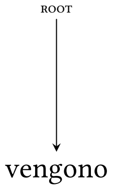
b. $$\left[\begin{array}{ll} \textsc{pred} & \text{‘}\textsc{persuade}\langle\textsc{subj, obj, xcomp}\rangle\text{’} \\ \textsc{subj} & \left[\textsc{pred}\ \text{‘}\textsc{Chris}\text{’}\right] \\ \textsc{obj} & \boxed{1}\left[\textsc{pred}\ \text{‘}\textsc{Mary}\text{’}\right] \\ \textsc{xcomp} & \left[\begin{array}{ll}\textsc{pred} & \text{‘}\textsc{come}\langle\textsc{subj}\rangle\text{’}\\ \textsc{subj} & \boxed{1}\end{array}\right]\end{array}\right]$$

Such dependencies cannot be expressed in Tesnière’s formalism, because *persuade* and *come* share the dependent *Mary*, despite not forming a group. Moreover, *Mary* bears a different syntactic relation to each of them, which again is not possible in Tesnière’s formalism. More recent versions of dependency grammar have typically accounted for control and raising verbs by positing more levels of representation, see Section 2.2.

Finally, an important difference between Tesnière’s dependency graphs and f-structures is that f-structures may contain nodes that correpond to no overt word. A typical case is pro-drop, as in (8) from Italian.

(8) a. vengono
come-PRS.3PL

b. $$\left[\begin{array}{ll} \textsc{pred} & \text{‘}\textsc{come}\langle\textsc{subj}\rangle\text{’} \\ \textsc{subj} & \left[\textsc{pred}\ \text{‘}\textsc{pro}\text{’}\right]\end{array}\right]$$

c. 

Again, Tesnière’s formalism cannot capture this: dependency nodes may correspond to one word, or more words if they form a group by junction or transfer, but not to zero. The strategy in later versions of DG has been the same as that used to address phenomena where LFG uses structure sharing, namely to introduce more levels of representation.

In sum, one can say from an LFG perspective that Tesnière’s dependency graphs, while certainly more expressive than core dependencies, are insufficiently general to deal with the complex functional relations that exist in natural language sentences.

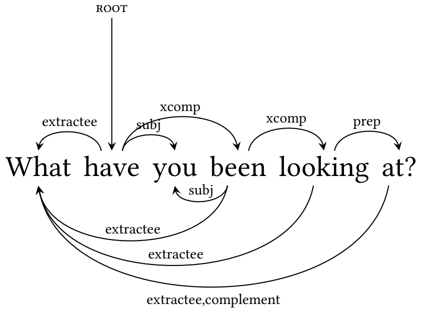
2.2 Other levels of syntactic representation

Tesnière’s strategy was to enrich dependency graphs so as to be able to represent more functional relations than core dependencies can do. More recent versions of DG have instead opted to keep the core dependencies simple and instead go beyond a single level of grammatical description to accommodate more information. One prominent example is the so-called tectogrammatical layer found in Functional Generative Description (Sgall et al. 1986) and the associated Prague Treebanks (Hajič et al. 2020). This layer is annotated with an enriched dependency tree that will contain nodes that do not correspond to words (e.g. pro-dropped subjects) and secondary edges capturing multiple head-phenomena such as control.[4]

Melčuk's Meaning–Text Theory explicitly distinguishes a deep syntactic level between the semantic level and surface syntax. However, as pointed out by Kahane (2003), the deep syntactic level is the least defined level of MTT and it is not clear how much information it is supposed to contain. What is clear, however, is that grammatically imposed coreference relations are resolved in deep syntax, opening up a way to deal with e.g. control.

In Word Grammar (Hudson 1984, 2010), too, control is treated by loosening the tree constraint on dependency structures. (9), from Hudson (2003: 521), illustrates how structure sharing is used to analyze raising (*you* shared by *have* and *been*)[5] and extraction (*what* shared by *have*, *been*, *looking* and *at*).

(9)



The dependency graph in (9) is essentially identical to the standard LFG analysis (except that in extraction, LFG usually has structure sharing only between the gap and the filler position, without involving the intermediate f-structures).

[4] The status of the tectogrammatical layer is not entirely clear: the Prague Dependency Treebank annotation guidelines (https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/ch02.html) say that it "represents the semantic structure of the sentence", but Hajič et al. (2020) describe it as "deep syntax". The difference may be merely terminological.

[5] Instead of Hudson's *sharer*, I have used the LFG relation XCOMP which Hudson explicitly mentions as an alternative name for the same concept. The diagram in Hudson (2003: 521) does not have a subject relation between *you* and *looking*, although *looking* is an XCOMP of *been*. It is unclear whether this is just an error.

However, in Word Grammar, the edges above and below the words have different status: “This diagram also illustrates the notion ‘surface structure’ […]. Each dependency is licensed by the grammar network, but when the result is structure-sharing just one of these dependencies is drawn above the words; the totality of dependencies drawn in this way constitutes the sentence’s surface structure. In principle any of the competing dependencies could be chosen, but in general only one choice is compatible with the ‘geometry’ of a well-formed surface structure, which must be free of ‘tangling’ (crossing dependencies – i.e. discontinuous phrases) and ‘dangling’ (unintegrated words). There are no such constraints on the non-surface dependencies.”[6]

This illustrates the point that we made in the introduction: different varieties of dependency grammar may have different notions of “deep syntax”, but they all share the idea that there is an interesting representation of syntactic dependencies that is a rooted tree over nodes that stand in a one-to-one correspondence with the words of the sentence. This is very different from LFG: all edges of an f-structure graph are equal. The subject edge that connects the subject of a control construction to the control verb has exactly the same status as the subject edge that connects the subject to the non-finite verb. Thus, there is no “privileged subgraph” of the f-structure that forms a rooted tree over the words. By contrast, Hudson’s distinction between the surface structure and the non-surface dependencies gives rise to such a privileged subgraph, although it must be said that the distinction between surface and non-surface dependencies is not further developed in Word Grammar.

Dependency grammars also differ in their treatment of “null words”, i.e. cases where LFG would have an f-structure node that does not correspond to any surface word, as in e.g. pro-drop. Most dependency analyses would simply leave out such subjects, as we saw in (8). But here too, many dependency grammars introduce the missing subjects in “deeper” projections, for example in the tectogrammatical layer of Functional Generative Description. In fact, Word Grammar is one of the few dependency grammar frameworks that acknowledge empty elements in the core syntactic graph. Creider & Hudson (2006) present an argument for this that runs along standard lines of LFG thinking. In Ancient Greek, predicate nouns and adjectives agree in case (and adjectives also in number and gender) with their subjects; and subjects of infinitives are in the accusative.

(10)	nomízo:	gàr	humâ:s	emoì	eînai	kaì	patrída	kaì
	think-1.PRS	for	you-ACC	me-DAT	be-INF	and	fatherland-ACC	and

[6] Hudson (2003: 521)

phílous
friends-ACC
‘For I think you are to me both fatherland and friends’ (Xenophon, Anabasis 1.3.6)

But crucially, the predicative is accusative also when the accusative subject is absent (11), even in cases where there is a coreferential element in the higher clause (12).

(11) philánthro:pon eînai deî
humane-ACC be-INF must
‘one must be humane’ (Isocrates 2.15)

(12) exarkései soi túrannon genésthai
suffice-FUT you-DAT king-ACC become-INF
‘it will be enough for you to become king’ (Plato, Alcibiades 2, 141a7)

In (12), we observe that the predicate noun *turannon* does not agree directly with its logical subject *soi*, but rather with the unexpressed subject of the infinitive. Since case agreement is generally agreed to be syntactic (whereas agreement in number and gender could potentially be semantic), Creider & Hudson (2006) conclude that the unexpressed subject of the infinitive must nevertheless be present in the syntax. This is unsurprising from an LFG point of view, but does not seem to be generally accepted in DG. It is unclear, for example, how Functional Generative Description would deal with this data, since null words are inserted only at the tectogrammatical layer, where there is no case feature.

3 Word order and generative power in DG and LFG

In most versions of dependency grammar, it is assumed that the nodes of a dependency structure are not linearly ordered in themselves: a dependency relation implies no particular linear order between a head and its dependents, but can be related to different surface linearizations. This view goes back to Tesnière (1959: chapter 7), who distinguishes sharply between structural order (dependencies) and linear order. The main exception to this is Functional Generative Description, which assumes a linear order on the nodes even in the tectogrammatical layer, to capture information structure.

But even if the nodes of the dependency structures are not linearly ordered, it is possible (and in fact necessary for most languages) to constrain the relation between dependency structure and linearization. One much-discussed constraint is *projectivity*.[7]

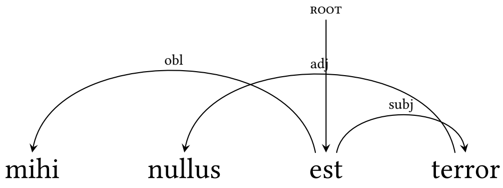
(13) A dependency graph is projective iff for every edge $n_h \rightarrow n_d$ it contains, n_h dominates all nodes that occur between n_h and n_d (where domination is the transitive closure of the edge/dependency relation)

An early result due to Gaifman (1965) is that projective dependency grammars are weakly equivalent to context-free grammars.[8] This result may in fact have led to a lack of interest in dependency grammar because it was widely believed in the sixties and seventies (and eventually proved in the eighties) that natural languages are *not* context-free. On the other hand, the recognition problem for a dependency grammar with no linearization constraints at all (thus allowing arbitrary discontinuities) is NP complete (Neuhaus & Bröker 1997).[9]

With the increasing popularity of dependency grammars in the 2000s, this led to the search for *intermediate* linearization constraints between strict projectivity and arbitrary non-projectivity. One important class of constraints is based on the notion of *block degree* (Holan et al. 1998). Intuitively, projectivity as defined in (13) ensures that the subgraph of n_h (i.e. n_h and the set of nodes it dominates) forms a single block of adjacent nodes. We can instead allow the subgraph to form *two* blocks of adjacent nodes, interrupted by a continuous set of words. We say that n_h has block degree 2; and the block degree of a dependency tree is the highest block degree of any of its nodes. Equivalently, we can speak of gap degree, which is block degree minus 1 (i.e., the number of allowed gaps). (14) illustrates this with an example from Latin.

(14) a. Mihi nullus est terror
me.DAT none.NOM is fear.NOM
'I have no fear.'

b.



[7] It seems that this term originated with a technical report by P. Ihm and Y. Lecerf "Eléments pour une grammaire générale des langues projectives", Bruxelles 1960, but I have been unable to find this paper.

[8] See also Hays (1964).

[9] As we will see in Section 4, this is not an issue in data-driven parsing, which sidesteps the recognition problem and aims directly at providing a contextually plausible parse.

The gap degree of *est* is 0, since its subgraph is continuous; but the gap degree of *terror* is 1, since there is one gap in its subgraph – *est* intervenes between *terror* and *nullus*, but is not dominated by *terror*. As a result, the gap degree of the whole tree is 1.

To study the computational complexity of the dependency grammars that could generate structures like (14), and their relationship to LFG grammars, it is convenient to use phrase structure-based systems that allow discontinuities, so-called Linear Context-Free Rewriting Systems (LCFRS, Vijay-Shanker et al. 1987) or the notational variant Multiple Context-Free Grammars (MCFG, Seki et al. 1991). The MCFG formalism is a generalization of CFG which retains ordinary CFG productions for the expression of categorial structure, but uses explicit *yield functions* to compute the yield of the mother node from the yields of the daughters. In an ordinary CFG, yield computation is conflated with category formation: a rule such as DP → D NP says both that the category DP is formed of a D and an NP, and that the yield of the resulting DP is formed by concatenating the yields of D and NP. In effect, then, a CFG can be seen as an MCFG with concatenation as the only yield function.[10]

To allow for greater expressivity, MCFG allows yields to be *tuples* of strings. For example, we may want to say that the yield of DP is a pair (2-tuple) consisting of the yields of D and NP. This pair will then be the input to further yield functions that apply to productions with DP on the right-hand side. More generally, we may allow yields to be *n*-tuples of strings. The interesting point is that there is a close correspondence between yield components in an MCFG and blocks in a corresponding dependency structure. We can extract MCFG rules from dependency trees, as shown in Kuhlmann (2013), where a formal exposition is given. Here I just provide an intuitive understanding of how the tree in (14b) gives rise to the rules in Table 1.

Table 1: Rules extracted from the tree in (14b)

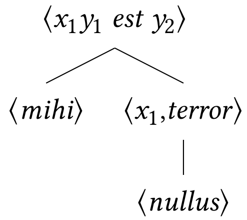
rule	yield function	compact notation
adj → $g()$	$g = \langle \textit{nullus} \rangle$	adj → $\langle \textit{nullus} \rangle$
obl → $h()$	$h = \langle \textit{mihi} \rangle$	obl → $\langle \textit{mihi} \rangle$
subj → i(adj)	$i = \langle x_1, \textit{terror} \rangle$	subj → $\langle x_1, \textit{terror} \rangle$ (adj)
root → j(obl subj)	$j = \langle x_1 y_1\ \textit{est}\ y_2 \rangle$	root → $\langle x_1 y_1\ \textit{est}\ y_2 \rangle$(obl subj)

[10] See Clark (2014) for an accessible introduction for linguists and Kallmeyer (2010: chapter 6) for a more formal introduction.

Looking at *nullus* in (14b), we see that it has no dependents, hence the right-hand side of the first rule is a constant function which fixes the yield to the string *nullus*, and similarly for *mihi*. For *terror*, things are more interesting. It takes one dependent, an adj, and hence its yield function *i* depends on the value of that argument. Concretely, the yield of the node *terror* is a tuple, consisting of the yield of the adj dependent which is represented as x_1,[11] and the string *terror*. Finally, the verb takes two arguments, subj and obl. The yield is constructed by concatenating the yield of the obl (i.e. x_1), the first component of the subj(i.e. y_1), the string *est*, and the second component of subj (y_2).

With the rules in figure 1, we can construct the MCFG derivation tree in (15).

(15)


$\langle x_1 y_1$ *est* $y_2\rangle$
⟨*mihi*⟩
$\langle x_1$,*terror*$\rangle$
⟨*nullus*⟩

But notice that because the MCFG grammar is lexical, i.e. each rule introduces exactly one lexical item, the tree in (15) is isomorphic to the dependency tree in (14b). In other words, a lexicalized MCFG can simply be interpreted as a dependency grammar which simultaneously restricts word order.

This allows us to compare the generative capacity and the parsing complexity of dependency grammars with other formalisms. Under a reasonable constraint on discontinuities,[12] the expressivity of an MCFG depends only on the maximal block degree of the grammar, giving rise to a hierarchy of *k*-MCFGs, where *k* is the block degree of the most complex yield function in the grammar. It turns out that 2-MCFGs (and hence dependency grammars that allow maximally one gap) are weakly equivalent to Tree Adjoining Grammars and 'classical' Combinatory Categorial Grammar, as was proven by Bodirsky et al. (2005).[13]

Even more interesting from an LFG perspective, there is also a result that a subclass of LFG grammars, so-called *finite copying LFGs*, can be translated into weakly equivalent MCFGs/LCFRSs (Seki et al. 1993). Finite copying LFGs are quite restricted in what functional annotations they allow, in particular they do

[11] The convention is that we use *x* for the yield of the first dependent and *y* for the yield of the second dependent, and subscript those variables with an index referring to blocks of the yield.

[12] Namely wellnestedness; a tree is wellnested if there are no disjoint subtrees that overlap linearly.

[13] See also Kuhlmann (2007, 2010).

not allow head annotations (↑=↓) or reentrancies, and also impose the crucial constraint that the grammar puts an upper bound on the number of c-structure nodes corresponding to a single f-structure. Wedekind & Kaplan (2020) show that we can impose this upper bound while still allowing head annotations and reentrancies, as long as they are nonconstructive. This allows most functional equations that are used in linguistic work, including functional control equations of the type (↑ F G)=(↑ H). Kaplan and Wedekind call these grammars *k*-bounded LFGs and prove that for any *k*-bounded LFG, a weakly equivalent *k*-MCFG can be constructed. Moreover, the MCFG rules can be annotated with functional descriptions that allow us to construct the f-structure that the corresponding *k*-bounded LFG assigns to the sentence, yielding a strongly equivalent MCFG.

These results allow us to compare dependency grammars and LFGs in a precise way. First of all, dependency grammars and *k*-bounded LFGs are weakly equivalent. Nevertheless, although strongly equivalent MCFGs can be constructed from both dependency grammars and *k*-LFGs, it is not the case that we can construct a strongly equivalent dependency grammar from an LFG. The interpretation of an MCFG as a dependency grammar relies on unique lexicalization: each rule contains a single lexical item interpreted as the head. The MCFGs that Kaplan and Wedekind construct from LFGs are not lexicalized in this way. They do contain functional descriptions that allow us to identify the head but, since LFG allows coheads, the head is not guaranteed to be unique. Moreover, the functional descriptions in the MCFG constructed from an LFG may contain reentrancies, i.e. words having more than one head, which have no interpretation on the dependency grammar side, thus losing information. A final, minor point is that Kuhlmann's interpretation of MCFGs as dependency grammars say nothing about edge labels; it would be natural and straightforward to interpret LFG's ordinary function assignments as such labels.

In sum, then, the formal analysis tells us that the difference between *k*-bounded LFGs and dependency grammars resides exactly in the availability of co-heads and reentrancies, which provide important information from a linguistic point of view. Finally, it should be noted that the restriction to *k*-bounded LFGs, while preserving coverage of many, perhaps most, linguistic phenomena, is nevertheless not trivial. Rambow (2014) argued that unbounded scrambling as found in German and other free word order languages falls outside the generative capacity of MCFGs (and mildly context sensitive grammar formalisms in general) and hence *k*-bounded LFGs.

The comparison of dependency grammars and LFGs through MCFGs is also interesting from other points of view. As Kaplan and Wedekind point out, the effect of converting an LFG to an MCFG is to precompute the interaction between f- and c-structure and construct a grammar that recognizes all and only the c-structures whose f-descriptions are satisfiable. From a practical point of view, this may be an advantage in parsing. But from the perspective of theoretical LFG, it can be argued that MCFGs and the dependency grammars they give rise to conflate c- and f-structure, making it harder to state linguistic generalizations. The advantage of LFG's projection architecture is precisely to "to account for significant linguistic generalizations in a factored and modular way by means of related but appropriately dissimilar representations" (Kaplan 1989: 309). Seen from the dependency grammar side, the formal results offer a choice: Kuhlmann's translation to MCFGs makes it possible to enrich dependency grammars with an account of word order in a single component; but Kaplan and Wedekind's results show that MCFGs can be "modularized" into a word order component and a functional component (which is not surprising given that MCFGs generalize CFGs precisely by dissociating dominance and linearization) to give something very close to LFG. Either way, the formal analysis exposes similarities and differences between the frameworks. In principle, this paves the way for cross-fertilization on the theoretical side, but in practice such gains are limited by the fact that, as we pointed out in Section 1, dependency grammarians typically do not think in terms of (formal or informal) rules that generate the constructions they are interested in but content themselves with providing analyses of the whole structure.

4 DG and LFG in computational linguistics

4.1 Data-driven dependency parsing

On the computational side, there is a similar difference between DG on the one hand, and LFG and most other formal linguistic traditions, in that there has generally been little interest in developing formal grammars that can generate or parse languages. There are some exceptions to this: in the framework of Constraint Dependency Grammar (Maruyama 1990), there is for example a broad-coverage parser of German (Foth et al. 2005); and Constraint Grammar (Karlsson et al. 1995) is a widely used system in which implemented grammars have been created for a wide variety of languages. Many of these grammars content themselves with assigning syntactic function labels to words, without building a full syntax tree, but even so, many have proven useful in practical tasks.

Nevertheless, the dominant use of DG in computational linguistics is closely associated with machine learning approaches where computers find patterns in human annotated data. For such approaches, it is sufficient that annotators provide case-by-base analyses of the corpus without actually abstracting the rules that would create these analyses. Consistency remains a goal, since it makes the patterns easier to learn, but it is not enforced in the way it would be in grammar-based annotation such as typical scenarios for creating LFG parsebanks, where annotators choose between alternative analyses provided by the underlying grammar.

As we have seen several times so far, the constraints on core dependency syntax, namely the unique mother and the one-to-one correspondence between nodes and tokens, mean that many theoretically relevant distinctions cannot be encoded. On the flip side, this makes the annotation task easier as the annotator does not have to be trained in drawing the distinctions. The result is also often more accessible to end users: while grammar-based treebanks contain much more information than dependency trees, this information is typically encoded in a specific theoretical framework and not always easily accessible to users without training in that framework. In short, core dependency trees offer a tradeoff between practical considerations and theoretical depth, which may be attractive for many applications where the deeper linguistic distinctions do not matter much.

On top of that, the simple target structure makes it possible to train very efficient statistical dependency parsers. This approach is fundamentally different from the formal grammar approach to DG developed by Kuhlmann (2013), which we saw in Section 3. Data-driven parsers learn from human annotation and try to provide the most plausible parse in context, without judging acceptability or enumerating possible parses. In this context, non-projective dependencies are not an issue and can be captured efficiently (McDonald et al. 2005). Nivre (2008) introduced algorithms that could produce projective dependency parses in time linear of the input and algorithms that allow non-projective parses and run in quadratic time. Such results led to a huge increase of interest in dependency parsing, which quickly became dominant in statistical approaches to computational linguistics.

Data-driven parsing requires annotated data and the last decade has seen a large increase in the number of dependency treebanks that are available, especially driven by the Universal Dependencies (UD) initiative.[14] UD developed out of the Stanford dependencies for English (de Marneffe & Manning 2008) (which means that there is a certain amount of LFG heritage) as an effort to create an annotation scheme that can be used across languages. Though it has been driven mainly by practical considerations in NLP research, it has in recent years also been used for linguistic research (e.g. Hahn et al. 2020; Berdicevskis & Piperski 2020).

[14] See https://universaldependencies.org/ and de Marneffe et al. (2021).

As of release 2.9 (November 2021), UD contains 217 treebanks from 122 languages. A comparison with LFG's ParGram approach reveals the strengths and weaknesses of the approach.[15] Drawing on the long tradition for using DG to provide case by case analyses rather than abstracting grammars has made it possible to achieve an unprecedented breadth of coverage. On the other hand, the analyses are more shallow than those provided by LFG grammars and the lack of underlying grammars makes the UD project much more prone to inconsistencies both within and across treebanks.

4.2 Converting LFG parsebanks to dependency treebanks

The existence of annotated resources in both LFG and DG formats makes it possible to study differences between the two from a different perspective than the formal language approach we adopted in Section 3. In this section, we look at work on converting LFG-based resources to dependency structures to see how the two formats compare and to what extent information can be preserved when converting to the less expressive DG format.

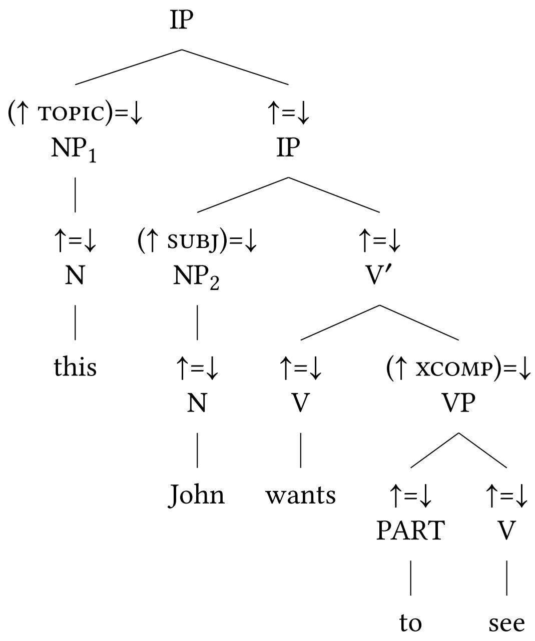
For completeness, we mention that there has also been some work on enriching them to yield LFG-structures, e.g. by Forst (2003) and Haug (2012). However, both Forst and Haug started from relatively rich dependency annotations (with secondary edges), so that the conversion to f-structures was not difficult and other issues were more important (e.g. the creation of c-structures from the dependency representations by Haug).

Several conversion algorithms have been developed to convert LFG structures to dependency structures. Here we discuss two recent approaches, by Meurer (2017) and Przepiórkowski & Patejuk (2020),[16] which contrast in interesting ways, since Meurer starts from the c-structure and Przepiórkowski and Patejuk from the f-structure. Both are natural starting points: the f-structure represents grammatical functions, just like the target dependency structure; but the c-structure has the advantage that its terminal nodes are in one-to-one correspondence with the words of the sentence, just like in the dependency structure. Both algorithms target the particular style of dependency annotation adopted in UD, but proceed in two steps, namely first the creation of a dependency structure, and second, the modification of that structure to comply with the exact representation chosen in UD. Here we focus on the first step. To illustrate how the two algorithms work, we consider the LFG structure in (16)–(17).

[15] For more on ParGram, see Forst & King forthcoming [this volume].

[16] Dione (2020) presents an approach that combines Meurer (2017) and Przepiórkowski & Patejuk (2020). For older work, see Øvrelid et al. (2009) and Çetinoğlu et al. (2010).

(16)

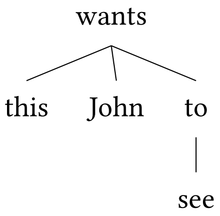


(17)

$$\begin{bmatrix} \textsc{pred} & \text{`}\textsc{want}\langle\textsc{subj, xcomp}\rangle\text{'} \\ \textsc{topic} & \boxed{1}\begin{bmatrix}\textsc{pred} & \text{`}\textsc{pro}\text{'}\end{bmatrix} \\ \textsc{subj} & \boxed{2}\begin{bmatrix}\textsc{pred} & \text{`}\textsc{John}\text{'}\end{bmatrix} \\ \textsc{xcomp} & \begin{bmatrix} \textsc{pred} & \text{`}\textsc{see}\langle\textsc{subj, obj}\rangle\text{'} \\ \textsc{subj} & \boxed{1} \\ \textsc{obj} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

In Meurer's approach, the first step is to "lexicalize" the c-structure tree by recursively replacing each non-terminal node with its functional head node, as determined by the annotation ↑=↓. This straightforward for IP, NP$_1$ and NP$_2$ in (16): *wants*, *this* and *John* are uniquely linked to these nodes via an unbroken chain of ↑=↓. But more generally, the challenge here is the same as in lexicalizing an MCFG that results from the Kaplan-Wedekind construction: co-heads and absence of heads mean there might be no unique daughter to lift. To find a unique head in such cases the algorithm proceeds as follows: 1) if no daughter of x is a functional head, attach all daughters to the mother of x and proceed as before. 2) if more than one daughter of x is a functional head, choose the one with the shortest embedding path. 3) if there is a tie, choose the leftmost node. For the VP in our example, case 3 applies and we choose *to* as the head; it is therefore lifted to the VP node, while *see* is only lifted to the V node. These lifting operations yield the tree in (18).

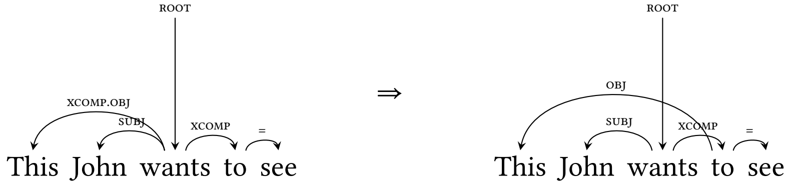
(18)



We then need to label the edges. Meurer's algorithm does that by labelling the edge between nodes x and their daughter y in the resulting tree with the f-structure path from $\phi(x)$ to $\phi(y)$. So, the edge from *John* to *wants* is labelled subj since that is the path from the f-structure of *wants* to the f-structure of *John*. But because of reentrancies in the f-strucure, the path between two f-structures is not always unique: for example, there is a path from the f-structure of *wants* to the f-structure of *this* that is labelled topic, but there is another path that is labelled xcomp obj. In such cases, Meurer chooses the shortest path that contains only grammatical functions (i.e. no discourse functions); in our case that yields the complex label xcomp.obj where the two elements of the f-structure path have been concatenated with a dot. Co-heads present another problem for the labelling approach: *to* and *see* share the same f-structure, so there is no path. In such cases a dummy relation = is used.

This algorithm produces a projective dependency graph with complex labels, as shown for our example in the lefthand side of (19). In the next step, the complex labels are resolved and nodes attached accordingly, potentially introducing non-projectivity. For our example, when the complex relation xcomp.obj is resolved, we obtain the non-projective tree on the right-hand side of (19).

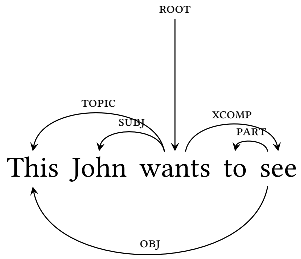
(19)



This then is the input to the final step, where the dependency tree is normalized according to the UD annotation standard.

Przepiórkowski & Patejuk (2020), by contrast, start from the f-structure, which already represents the syntactic dependencies. This means the challenge is different, namely to match the nodes of the f-structure to the words of the sentence, which are the nodes of the target dependency graph. F-structure nodes may correspond to zero, one or several words; they are given by the ϕ^{-1}, which is part of the source annotation. F-structures that correspond to no words (e.g. in pro-drop) may simply be ignored in the dependency structure; but for f-structures that correspond to more than one word, the "true" head that will take the f-structure's place in the corresponding dependency structure must be identified, and the other words in ϕ^{-1} must be attached with appropriate relations. The basic algorithm is simple: if there is a verbal token in ϕ^{-1}, choose that as the true head; otherwise, choose a nominal or adjectival token; otherwise an explicit lexical conjunction. The other nodes are then attached to the true head with a relation labelled by their own preterminal category. This produces the structure in (20) from the f-structure in (17).

(20)



As we can see, the output from the algorithm of Przepiórkowski & Patejuk (2020) is not a tree, but a graph, where all f-structure relations are preserved, including two incoming edges to *this*. This is exploited to produce enhanced UD, which allow for this kind of graph structure; but the output is also trimmed to produce a basic UD structure.

(19)–(20) illustrate the output of the first steps in the conversions, where the target is to produce the desired data structure, namely a dependency tree or graph over words. As mentioned, the next step is to normalize this structure to the concrete requirements of the UD annotation standard. This is less interesting from our point of view, but it is worth looking at a few topics that display divergences between standard LFG solutions and choices that are made in the dependency grammar community as exemplified by UD.

First, UD subscribes to the primacy of content words. This means that content words are typically heads of function words, for example in structures consisting of auxiliary and verb, adposition and noun, and determiner and noun, as illustrated in the lower graph of (2) in Section 1. Also, there are no nested structures of function words, so e.g. in structures with multiple auxiliaries (*may have been understood*), all the auxiliaries attach directly to the lexical verb. While UD may be extreme among dependency grammar approaches in adopting this principle across the board, similar analyses are found for some of these structures in other frameworks. By contrast, such analyses are non-existing in the LFG literature, except for noun-determiner structures (where the determiner is often analysed as a SPEC dependent of the noun): function words are typically either co-heads, or lone heads, taking a lexical word as their dependent. For example, there are analyses of auxiliaries as co-heads specifying features of the f-structure where the lexical verb contributes the PRED, and alternative analyses where auxiliary verbs take XCOMP dependents, potentially in a cascading sequence ending in the lexical verb.

In fact, the difference between the co-head analysis and the UD dependent analysis of function words is rather slight, as revealed by the conversion procedure of Przepiórkowski & Patejuk (2020). In f-structures that have functional co-heads, the lexical head will be chosen as the head during conversion, and hence the function words will end up as dependents. And in fact, given that UD uses a flat structure for multiple function words means that the two representations are more or less equivalent, a point made in the UD documentation too,[17] where it is said that function word relations are different from dependency relations between content words and in fact form Tesnière-style nuclei.

This in turn opens the door to theoretical cross-fertilization. What are good criteria for choosing between the two analyses? The UD argument is that primacy of lexical words maximizes parallelism across languages, and the exact same argument has been raised in the LFG literature (Butt et al. 1996). On the other hand, Dyvik (1999) has countered that this leads to a stipulative, rather than empirical, notion of language universality and also that it can lead to analyses that are language-internally unmotivated. Recently, Osborne & Gerdes (2019) criticized the UD approach and argued that functional words should always be heads. They were apparently unaware of the LFG literature on the topic, perhaps because it is cast in terms of co-heads vs. XCOMPs. But as we have seen, the difference between a co-head analysis and a UD-style annotation is very slight, and so the arguments made in the LFG context are certainly relevant also for the DG community.

The other main divergence between initial dependencies, as resulting from the conversion algorithms, and the target UD structures concern coordination. Here LFG makes use of an additional data structure, sets, which have no equivalents in standard dependency grammar or in UD. (Although as we saw above, Tesnière's junction comes close.) There are many competing analyses of coordination in the dependency literature,[18] maybe suggesting that the basic data structure of dependency trees is ill-suited to model coordination, as Tesnière argued. The UD choice

[17] https://universaldependencies.org/u/overview/syntax.html

[18] And also in (pre-UD) dependency treebanks, see Popel et al. (2013).

is to take the first conjunct as the head and attach the other conjuncts to it with a special dependency relation CONJ, whereas conjunctions and punctuation marks are attached to their following conjunct with CC and PUNCT. It is known that this annotation style cannot capture all important structural differences, such as the difference between a dependent of the first conjunct and a shared dependent of multiple conjuncts, or different style of nested coordinations. The conversion procedure exposes this lack of expressivity, but also makes it possible to quantify its effect. As observed by Przepiórkowski & Patejuk (2020), only twelve out of 21,732 utterances in the Polish LFG structure bank are effected.[19]

More generally, Przepiórkowski & Patejuk (2020) conclude that the information loss in converting from LFG to (enhanced) UD is in fact negligible, except in the case of pro-drop structures. As the UD effort continues to expand, there is therefore considerable potential for theoretical cross-fertilization.

5 Semantics

Tesnière in general pays much less attention to meaning than to structure, but at various points he does talk about semantic dependencies. Several versions of dependency grammars (Functional Generative Description, Meaning–Text Theory) have taken this up and operate with a separate level of semantic structure. There are also various graph-based semantic representation languages such as Abstract Meaning Representation (AMR, Banarescu et al. 2013), which arguably are semantic dependency representations without an accompanying syntactic representation. All such semantic dependency graphs, whether they are coupled to syntax or not, differ considerably from standard logic-based formalizations of meaning as used in LFG and most other formal frameworks. They will not be further discussed here.

Robaldo's Dependency Tree Semantics (Robaldo 2006) is much closer to standard conceptions of formal semantics, as it aims to transform dependency trees into structures that can be interpreted model-theoretically. But for the purposes of comparison with LFG, it is more interesting to observe that Bröker (2003: p. 308), in his discussion of the formal foundations of dependency grammar, briefly suggested that the similarity between dependency trees and LFG's functional structure could make the application of Glue semantics (Dalrymple et al. 1993; Dalrymple 1999; see also Asudeh forthcoming [this volume]) to dependency grammar a promising research area. Gotham & Haug (2018) flesh out this idea

[19] See Przepiórkowski & Patejuk (2019) for a proposal as to how nested coordination could be captured in UD.

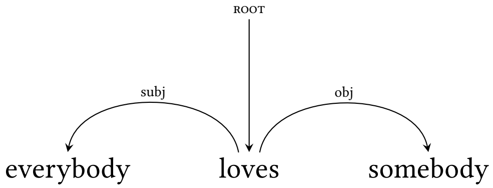
and show how to combine Universal Dependencies with Partial Compositional Discourse Representation Theory Haug (2014).

On the formal side, there are few if any obstacles to such an application. The fundamental idea behind glue semantics is to have linear logic terms guide the composition of corresponding lambda terms. In the first order glue setting, the terms of the linear logic are the f-structures, atomic formulae are formed by applying predicates to the f-structures (in type-theoretic terms, these predicates act like unary type constructors) and complex formulae are formed with ⊠, which acts as a binary function type constructor. Consider (21), which gives the meanings, dependency structure and f-structure for *Everybody loves somebody*. We write e_1 for $e(1)$, i.e. the application of the type constructor/predicate e to the syntactic object/term with index 1.

(21) a.

everybody	$\lambda P.\forall x.person(x) \rightarrow P(x)$	$(e_1 \boxtimes t_2) \boxtimes t_2$
somebody	$\lambda P.\forall x.person(x) \wedge P(x)$	$(e_3 \boxtimes t_2) \boxtimes t_2$
loves	$\lambda x.\lambda y.love(x, y)$	$e_1 \boxtimes e_3 \boxtimes t_2$

b.



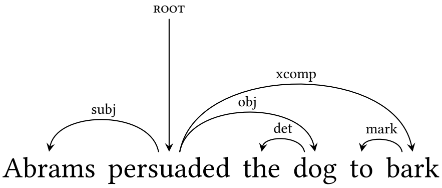
c.

$$\left[\begin{array}{ll} \text{PRED} & \text{‘LOVE}\langle\ \text{SUBJ}\ \rangle\text{’} \\ \text{SUBJ} & \left[\text{PRED}\ \ \text{‘EVERYBODY}\right] \\ \text{OBJ} & \left[\text{PRED}\ \ \text{‘SOMEBODY’}\right] \end{array}\right]$$

Clearly, it makes no difference whether we interpret the glue types in (21a) over the dependency tree in (21b) or the f-structure in (21c): in both cases we just need the same mapping between the indices 1, 2, 3 and the corresponding f-structures or dependency nodes.

However, while the formal properties of the two theories are similar enough that Glue semantics can be used for both LFG and DG, a practical consideration is that dependency trees typically do not contain all the semantically relevant information that we find in the corresponding f-structure. Control structures are a case in point (22).

(22)



The dependency tree in (22) lacks the information that *the dog* is the subject of *to bark*. However, the label XCOMP does tell us that the missing subject of *to bark* is one of the dependents of *persuaded*. As a result, the best we can do is to introduce a discourse referent x_2 that is the subject of the infinitive clause and must be linked to one of the participants in the matrix event, though we do not know which one, unless we have access to the lexical information that *persuade* is an object control verb. We see that it is possible to compensate for some of the information loss in dependency trees, although the result only becomes useful if we have other, lexical information sources available: dependency trees on their own don't typically come with the rich semantic lexical entries that glue (and other formal semantic theories) require. We refer to Gotham & Haug (2018) for more details.

6 Summary

We have seen that the basic relations for analyzing functional syntax, dependencies in DG and grammatical functions in LFG, are very similar, both formally and conceptually. Nevertheless, the focus on core dependencies that is often seen in DG work leaves other levels of analysis less well developed than in LFG. Word order, in particular, has not received much attention, but we have seen that it can be interestingly restricted through the use of lexicalized MCFGs, offering a point of comparison to LFGs, which can also be translated to MCFGs. Another point of comparison is offered by work on converting LFG parsebanks to dependency treebanks. Finally, we saw that the similarity between DG and LFG also means that they can use the same syntax-semantics interface in the form of Glue semantics.

All in all, the considerable similarities between the two theories suggest there is ample room for mutually benefiting discussion, especially if the increasing use of DG in computational linguistics triggers a corresponding interest in theoretical DG.

Acknowledgements

I thank the two anonymous reviewers, one non-anonymous reviewer (Adam Przepiórkowski) and the editor Mary Dalrymple for very helpful comments on a previous version of this chapter.

References

Asudeh, Ash. Forthcoming. Glue Semantics. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer & Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186. Sofia: Association for Computational Linguistics. https://www.aclweb.org/anthology/W13-2322.

Berdicevskis, Aleksandrs & Alexander Piperski. 2020. Corpus evidence for word order freezing in Russian and German. In *Proceedings of the fourth workshop on Universal Dependencies (UDW 2020)*, 26–33. Barcelona: Association for Computational Linguistics. https://aclanthology.org/2020.udw-1.4.

Bodirsky, Manuel, Marco Kuhlmann & Mathias Möhl. 2005. Well-nested drawings as models of syntactic structure. In James Rogers (ed.), *Proceedings of FG-MoL 2005: the 10th conference on Formal Grammar and the 9th meeting on Mathematics of Language*, 195–203. Stanford: CSLI Publications. http://cslipublications.stanford.edu/FG/2005/bodirsky.pdf.

Bröker, Norbert. 1998. How to define a context-free backbone for DGs: implementing a DG in the LFG formalism. In *Processing of dependency-based grammars*. https://aclanthology.org/W98-0504.

Bröker, Norbert. 2003. Formal foundations of dependency grammar. In Vilmos Ágel (ed.), *Dependency and valency* (Handbook of Linguistics and Communication Sciences 25), 294–310. Berlin: De Gruyter.

Butt, Miriam, María-Eugenia Niño & Frederique Segond. 1996. Multilingual processing of auxiliaries in LFG. In D. Gibbon (ed.), *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS conference*, 111–122. Berlin: Mouton de Gruyter.

Çetinoğlu, Özlem, Jennifer Foster, Joakim Nivre, Deirdre Hogan, Aoife Cahill & Josef van Genabith. 2010. LFG without c-structures. In *NEALT proceedings*, vol. 9, 43–54.

Chomsky, Noam. 1966. *Cartesian linguistics: A chapter in the history of rationalist thought*. New York: Harper & Row. DOI: 10.1017/cbo9780511803116.

Clark, Alexander. 2014. *An introduction to multiple context free grammars for linguists*. https://alexc17.github.io/papers/mcfgsforlinguists.pdf.

Covington, Michael A. 1984. *Syntactic theory in the High Middle Ages*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511735592.

Creider, Chet & Richard Hudson. 2006. Case agreement in Ancient Greek: Implications for a theory of covert elements. In Kensei Sugayama & Richard Hudson (eds.), *Word Grammar: New perspectives on a theory of language structure*, 35–53. London: Continuum.

Dalrymple, Mary (ed.). 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication). Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/6169.001.0001.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.

Dalrymple, Mary, John Lamping & Vijay Saraswat. 1993. LFG semantics via constraints. In *Proceedings of the 6th conference of the European chapter of the ACL (EACL 1993)*, 97–105. Association for Computational Linguistics. DOI: 10.3115/976744.976757.

de Marneffe, Marie-Catherine & Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *COLING 2008: Proceedings of the Workshop on Cross-framework and Cross-domain Parser Evaluation*, 1–8. Manchester. DOI: 10.3115/1608858.1608859.

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre & Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics* 47(2). 255–308. DOI: 10.1162/coli_a_00402.

de Marneffe, Marie-Catherine & Joakim Nivre. 2019. Dependency grammar. *Annual Review of Linguistics* 5(1). 197–218. DOI: 10.1146/annurev-linguistics-011718-011842.

Dione, Cheikh M. Bamba. 2020. From LFG to UD: a combined approach. In *Proceedings of the fourth workshop on Universal Dependencies (UDW 2020)*, 57–66. Barcelona: Association for Computational Linguistics. https://aclanthology.org/2020.udw-1.7.

Dyvik, Helge. 1999. The universality of f-structure: Discovery or stipulation? The case of modals. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '99 conference*, 1–11. Stanford: CSLI Publications.

Forst, Martin. 2003. Treebank conversion – Establishing a testsuite for a broad-coverage LFG from the TIGER treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*, 205–216. Association for Computational Linguistics. https://www.aclweb.org/anthology/W03-2404.

Forst, Martin & Tracy Holloway King. Forthcoming. Computational implementations and applications. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Foth, Kilian, Wolfgang Menzel & Ingo Schröder. 2005. Robust parsing with weighted constraints. *Natural Language Engineering* 11(1). 1–25. DOI: 10.1017/S1351324903003267.

Gaifman, Haim. 1965. Dependency systems and phrase-structure systems. *Information and Control* 8(3). 304–337. DOI: 10.1016/s0019-9958(65)90232-9.

Gotham, Matthew & Dag Haug. 2018. Glue semantics for Universal Dependencies. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '18 conference*, 208–226. Stanford: CSLI Publications.

Hahn, Michael, Dan Jurafsky & Richard Futrell. 2020. Universals of word order reflect optimization of grammars for efficient communication. *Proceedings of the National Academy of Sciences* 117(5). 2347–2353. DOI: 10.1073/pnas.1910923117.

Hajič, Jan, Eduard Bejček, Jaroslava Hlavacova, Marie Mikulová, Milan Straka, Jan Štěpánek & Barbora Štěpánková. 2020. Prague Dependency Treebank - Consolidated 1.0. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC'20)*, 5208–5218. Marseille: European Language Resources Association (ELRA). https://www.aclweb.org/anthology/2020.lrec-1.641.

Haug, Dag. 2012. From dependency structures to LFG representations. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 271–291. Stanford: CSLI Publications.

Haug, Dag. Forthcoming. LFG and Dependency Grammar. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Haug, Dag Trygve Truslew. 2014. Partial dynamic semantics for anaphora: Compositionality without syntactic coindexation. *Journal of Semantics* 31(4). 457–511. DOI: 10.1093/jos/fft008.

Hays, David G. 1964. Dependency theory: a formalism and some observations. *Language* 40(4). 511–525. DOI: 10.2307/411934.

Holan, Tomas, Vladislav Kuboň, Karel Oliva & Martin Platek. 1998. Two useful measures of word order complexity. In *Processing of dependency-based grammars*. https://aclanthology.org/W98-0503.

Hudson, Richard. 1984. *Word Grammar*. Oxford: Blackwell.

Hudson, Richard. 2003. Word Grammar. In Vilmos Ágel (ed.), *Dependency and valency*, vol. 25 (Handbook of Linguistics and Communication Sciences), 508–526. Berlin: De Gruyter. DOI: 10.1093/oxfordhb/9780199738632.013.0019.

Hudson, Richard. 2010. *An introduction to Word Grammar*. Cambridge, UK: Cambridge University Press.

Kahane, Sylvain. 2003. The Meaning-Text Theory. In Vilmos Ágel (ed.), *Dependency and valency* (Handbook of Linguistics and Communication Sciences 25). Berlin: De Gruyter.

Kallmeyer, Laura. 2010. *Parsing beyond context-free grammars*. Berlin: Springer. DOI: 10.1007/978-3-642-14846-0.

Kaplan, Ronald M. 1989. The formal architecture of Lexical-Functional Grammar. *Journal of Information Science and Engineering* 5. 305–322. Revised version published in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 7–27).

Kaplan, Ronald M. 1995. The formal architecture of Lexical-Functional Grammar. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.), *Formal issues in Lexical-Functional Grammar*, 7–27. Stanford: CSLI Publications. Earlier version published as Kaplan (1989).

Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 173–281. Cambridge, MA: The MIT Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 29–130).

Karlsson, Fred, Atro Voutilainen, Juha Heikkilae & Arto Anttila. 1995. *Constraint grammar: A language-independent system for parsing unrestricted text*. Berlin: Walter de Gruyter.

Kiparsky, Paul. 1993. Paninian linguistics. In Ronald Asher & Seumas Simpson (eds.), *Encyclopedia of languages and linguistics*. Oxford: Pergamon.

Kuhlmann, Marco. 2007. *Dependency structures and lexicalized grammars*. Saarbrücken: Universität des Saarlandes. (Doctoral dissertation). http://www.ida.liu.se/~marku61/pdf/kuhlmann2007dependency.pdf.

Kuhlmann, Marco. 2010. *Dependency structures and lexicalized grammars: an algebraic approach*. Berlin: Springer. DOI: 10.1007/978-3-642-14568-1.

Kuhlmann, Marco. 2013. Mildly non-projective dependency grammar. *Computational Linguistics* 39(2). 355–387. DOI: 10.1162/COLI_a_00125.

Maruyama, Hiroshi. 1990. Structural disambiguation with constraint propagation. In *Proceedings of the 28th annual meeting of the Association for Computational Linguistics*, 31–38. Pittsburgh: Association for Computational Linguistics. DOI: 10.3115/981823.981828.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov & Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology conference and Conference on Empirical Methods in Natural Language Processing*, 523–530. Vancouver: Association for Computational Linguistics. DOI: 10.3115/1220575.1220641.

Mel'čuk, Igor A. 1988. *Dependency syntax: Theory and practice.* Albany: State University of New York Press.

Meurer, Paul. 2017. From LFG structures to dependency relations. *Bergen Language and Linguistics Studies* 8. 183–201. DOI: 10.15845/bells.v8i1.1341.

Neuhaus, Peter & Norbert Bröker. 1997. The complexity of recognition of linguistically adequate dependency grammars. In *Proceedings of the 35th annual meeting of the Association for Computational Linguistics and 8th conference of the European chapter of the ACL* (ACL '98/EACL '98), 337–343. Association for Computational Linguistics. DOI: 10.3115/976909.979660.

Nivre, Joakim. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics* 34(4). 513–553. DOI: 10.1162/coli.07-056-r1-07-027.

Osborne, Timothy & Kim Gerdes. 2019. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa: A Journal of General Linguistics* 4(1). 17. DOI: 10.5334/gjgl.537.

Øvrelid, Lilja, Jonas Kuhn & Kathrin Spreyer. 2009. Cross-framework parser stacking for data-driven dependency parsing. *Traitement automatique des langues* 50(3). 109–138.

Popel, Martin, David Mareček, Jan Štěpánek, Daniel Zeman & Zděněk Žabokrtský. 2013. Coordination structures in dependency treebanks. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: long papers)*, 517–527. Sofia. http://www.aclweb.org/anthology/P13-1051.

Przepiórkowski, Adam. Forthcoming. LFG and HPSG. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar.* Berlin: Language Science Press.

Przepiórkowski, Adam & Agnieszka Patejuk. 2019. Nested coordination in Universal Dependencies. In *Proceedings of the third workshop on Universal Dependencies (UDW SyntaxFest 2019)*, 58–69. Paris: Association for Computational Linguistics. DOI: 10.18653/v1/W19-8007.

Przepiórkowski, Adam & Agnieszka Patejuk. 2020. From Lexical Functional Grammar to Enhanced Universal Dependencies: The UD-LFG treebank of Polish. *Language Resources and Evaluation* 54. 185–221. DOI: 10.1007/s10579-018-9433-z.

Rambow, Owen. 2014. *Formal and computational aspects of natural language syntax.* Philadelphia: University of Pennsylvania. (Doctoral dissertation).

Robaldo, Livio. 2006. *Dependency tree semantics*. Turin: University of Turin. (Doctoral dissertation).

Seki, Hiroyuki, Takahashi Matsumura, Mamoru Fujii & Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science* 88(2). 191–229. DOI: 10.1016/0304-3975(91)90374-b.

Seki, Hiroyuki, Ryuichi Nakanishi, Yuichi Kaji, Sachiko Ando & Tadao Kasami. 1993. Parallel multiple context-free grammars, finite-state translation systems, and polynomial-time recognizable subclasses of Lexical-Functional Grammars. In *Proceedings of the 31st annual meeting of the Association for Computational Linguistics*, 130–139. Columbus, OH: Association for Computational Linguistics. DOI: 10.3115/981574.981592.

Sgall, Petr, Eva Hajičová & Jarmila Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Prague: Academia.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Paris: Klincksieck.

Vijay-Shanker, K., David J. Weir & Aravind K. Joshi. 1987. Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th annual meeting of the Association for Computational Linguistics*, 104–111. Stanford: Association for Computational Linguistics. DOI: 10.3115/981175.981190.

Wedekind, Jürgen & Ronald M. Kaplan. 2020. Tractable Lexical-Functional Grammar. *Computational Linguistics* 46(2). 515–569. DOI: 10.1162/coli_a_00384.

Chapter 5

LFG and Minimalism

Peter Sells
University of York

I compare and contrast LFG and the Minimalist Program (Chomsky 1995) with regard to different overall aspects of the frameworks: fundamental design properties, the representation of phrase structure, the representation of clausal grammatical information, the nature and role of syntactic features, and the analysis of agreement.

1 A framework for comparison

LFG and the Minimalist Program (MP; Chomsky 1995, 2000) are not straightforwardly comparable, as they are articulated in quite different ways. Going back to the 1980s, it could be said that LFG and Government-Binding Theory (Chomsky 1981) had a certain amount of commonality of approach, but as MP has developed from the earlier Government-Binding theory (GB), more and more emphasis has been placed in MP on derivation (see e.g. Hornstein et al. 2006; Hornstein 2018), rather than information and representation, which are of course cornerstones of LFG.

As both LFG and GB were responses to theoretical concerns about "classical" transformational grammar, which was developed during the 1970s, it is useful to start with the legacy of the early transformational period, which I summarize in (1):

(1) a. the overt part of syntax is represented in a phrase structure tree
b. all information in syntax is structured
c. different parts of a syntactic representation may share information

Peter Sells. Forthcoming. LFG and Minimalism. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

As GB has developed into MP, it has been assumed that (1b-c) refer to the same structure as (1a): that only the structures of phrase structure represent syntactic information, and that relationships are expressed in that structure, being established by movement operations. For instance, topicalization of an object creates a relationship between a topic position and an object position, as a result of a derivational operation in the MP.

LFG is a framework which is also based on the principle that all syntactic information is structured, but importantly that not all syntactic information is structure in the sense of phrase structure, and so it embodies (1) by having (at least) three aspects to the overall representation of a sentence:

(2) a. overt phrase structure (c-structure)
 b. a clause-level representation of the information it conveys (f-structure)
 c. an argument-structure representation for predicate-argument structure (a-structure)

All syntactic frameworks have a means to represent argument structure, and for any given predicate, its argument structure is structured according to the Thematic Hierarchy (e.g. Jackendoff 1972) or something equivalent. This is a-structure in LFG (see Findlay & Kibort forthcoming [this volume]). There is a mapping between this structure and the surface grammatical properties, f-structure, which is the representation of (2b). These properties include the GFs such as SUBJ and OBJ. The representation of clausal grammatical information (2b) is not part of the phrase structure representation (2a), but rather is the information that the overt structure conveys. This clausal representation is nevertheless structured in the sense that the information it contains is organized and grouped, according to principles of organization pertinent to this level.

This is different to the approach to clausal information in the MP, where information may start out quite distributed throughout the overall derivation, but can be aggregrated through successive movements, but also modified (e.g. a feature specification being used to drive one operation, then being deleted subsequent to the application of that operation; see Section 4.2). The core arguments of a predicate are merged first into a vP-VP structure (see Chomsky 1995: 315ff.) and this is the representation of argument structure; the internal argument(s) merge into VP and the external argument is the specifier of vP. Then further functional structure such as TP or CP is projected above vP. Functional or relational properties such as ‘subject’ and ‘object’ are characterized by the particular ‘Agree’ relations between v and Obj and between T and Subj.

Broadly speaking, the "subjecthood" properties identified by Keenan (1976) divide into those which properly refer to argument structure, and those which refer to clausal grammatical information (Manning 1996). Different syntactic phenomena may relate to either representation. For instance, anaphor binding is determined by the argument structure hierarchy in some languages (e.g. Schachter 1976 on Tagalog; Wechsler & Arka 1998 on Balinese). In other languages, the hierarchy of grammatical functions holding over f-structure is most relevant (see e.g. Bresnan 2001: 212–213; Bresnan et al. 2016: 217 on 'syntactic rank').

The representations of argument structure and of clausal information are largely language-invariant, though the mapping between them shows more variation, as do the ways in which different syntactic phenomena which refer to them. The information that they represent is carried by the overt phrase structures, (2a), which are of course subject to the most variation, and therefore the least revealing about "deep" properties of language.

In this chapter I evaluate different aspects of the LFG and MP approaches to grammatical theory. In Section 2 I consider overall "design features" of the frameworks, and what motivates them. In particular I outline how LFG took a different direction from transformational grammar. In Section 3 I contrast the approaches of the frameworks to phrase structure, and how the balance of analysis between c-structure and f-structure falls in LFG. Finally in Section 4 I compare the role(s) that features play in LFG and in MP, and how featural specifications participate in agreement.

2 Design features of a grammatical framework

Kaplan (2019a) gives a personal statement of how the passage below from Chomsky (1965) inspired his research which became part of the foundation of LFG (see e.g. Kaplan & Bresnan 1982: 173–174):

> "No doubt, a reasonable model of language use will incorporate, as a basic component, the generative grammar that expresses the speaker-hearer's knowledge of the language; but this generative grammar does not, in itself, prescribe the character or functioning of a perceptual model or a model of speech production." (Chomsky 1965: 9)

Kaplan also pursued the idea that linguistic complexity will be best modelled through (possibly complex) interactions of different (relatively simple) components, different representational dimensions, inspired by Simon (1962).

In this section I will consider how LFG addresses the core aims of a generative grammar, and how it has done so according to certain key foundational properties which set it apart from the procedural approach which has characterized the GB/MP approach led by Chomsky.

2.1 Levels of adequacy

One way to approach how a given framework takes up the agenda for Generative Grammar is to consider how the framework concerns itself with Chomsky's successive levels of adequacy:

> "To summarize briefly, there are two respects in which one can speak of "justifying a generative grammar." On one level (that of descriptive adequacy), the grammar is justified to the extent that it correctly describes its object, namely the linguistic intuition – the tacit competence – of the native speaker. In this sense, the grammar is justified on external grounds, on grounds of correspondence to linguistic fact. On a much deeper and hence much more rarely attainable level (that of explanatory adequacy), a grammar is justified to the extent that it is a principled descriptively adequate system, in that the linguistic theory with which it is associated selects this grammar over others, given primary linguistic data with which all are compatible. In this sense, the grammar is justified on internal grounds, on grounds of its relation to a linguistic theory that constitutes an explanatory hypothesis about the form of language as such. The problem of internal justification – of explanatory adequacy – is essentially the problem of constructing a theory of language acquisition, an account of the specific innate abilities that make this achievement possible." (Chomsky 1965: 26–27)

Since the GB era, Chomsky has taken explanatory adequacy to be the focus of syntactic theorizing (Rizzi 2016; D'Alessandro 2019). Yet to do this presupposes that there is a core of facts and generalizations so that there is a stable set of grammars which satisfy descriptive adequacy. Hornstein (2018: 55) presents a list of structural properties that syntacticians might agree are the "mid-level generalizations" of grammar; see also D'Alessandro (2019: 8) for a summary. For instance, data as in (3)–(4), from Chomsky (1973: 261), lead to well-established generalizations to classify verbs as being raising or control predicates, and binding conditions on anaphors and pronouns:

(3) a. They appeared to John to like each other.

b. * They appealed to John to like each other.

(4) a. * We appeared to John to like us.
b. We appealed to John to like us.

At the level of what facts and what kinds of facts are in the domain of syntax – such as those just given – frameworks such as GB/MP, LFG, and HPSG (Pollard & Sag 1987, 1994) are roughly commensurate, and so can be compared as to how they embody descriptive adequacy. Of course the formal details of a syntactic system which is intended to have a good "correspondence to linguistic fact" vary between each framework, but these are the easiest points of comparison. I take up this kind of comparison in sections 3 and 4 below.

LFG is a framework which has been developed to address descriptive adequacy, and which can be part of broader cognitive or computational approaches to human language, following the first Chomsky quote above. In this sense, it perhaps could be argued that LFG committed 40 years ago to what has become known in the MP literature as Chomsky's "third factor" (Chomsky 2005: 6) for explaining the format of grammatical knowledge:

(5) "… we should, therefore, be seeking three factors that enter into the growth of language in the individual:

1. Genetic endowment, apparently nearly uniform for the species, which interprets part of the environment as linguistic experience, …
2. Experience, which leads to variation, within a fairly narrow range, …
3. Principles not specific to the faculty of language …
– principles of data analysis that might be used in language acquisition
– principles of structural architecture
– principles of efficient computation"

The GB perspective on the language faculty put a great burden on an innate Universal Grammar which is essentially a parameterized blueprint for any individual grammar. This corresponds to Chomsky's first factor. Over the last 25 years, the trend in the development of the MP has been to reduce reliance on this purely innate component of grammar, in favor of the third factor. The reference in that factor to principles of data analysis and of structural architecture is quite salient as these are the principles at the basis of the considerations I raise in the next subsection, though of course this is not to imply that frameworks such as LFG deny that there are any 'first factor' properties or principles of our language capacity. However, as O'Grady (2012: 498) comments: "… the shift of focus to third-factor effects in generative grammar marks a milestone of sorts. Not because the idea is new, for it is not. Broadly speaking, the rest of the field has been committed to the primacy of third-factor explanations for decades."

2.2 Foundational properties of syntactic systems

What kinds of property are fundamental to syntax, to be emergent from a theory which "constitutes an explanatory hypothesis about the form of language", as in Chomsky's notion of explanatory adequacy? From the Minimalist perspective, the key notion here is the binary merge of abstract syntactic elements – 'External Merge' for initial structure-building, and 'Internal Merge' for movement from an existing position to another one. The structure is built up incrementally, with steps in the derivation driven by categorial requirements of combination or by features (see Section 3.2 and Section 4.2 below); the terminal nodes of the structure are spelled out morphologically after the syntactic operations have taken place.

LFG has taken a different starting point as to what the key properties of syntax are; in the rest of this subsection I highlight the consequences of a few examples which determine the 'lexicalist' and 'functional' (that is, information-based) aspects of LFG.

2.2.1 Lexicalist

LFG is a lexicalist framework, built on the assumption that the terminals in the phrase structure are word-level entities, the X^0s of X′-theory. The roots of this approach are in the Lexicalist Hypothesis of Chomsky (1970). Chomsky argued that the shared properties of different words based on the same lexeme could be accounted for without recourse to transformation (a nominalization transformation for the specific examples considered in that paper), and he introduced X′-theory to account for structural similarities across categories. LFG, like other declarative frameworks, expands on this perspective, using other syntactic information not directly represented in the phrase structure (cf. 2) to capture the appropriate similarities. An X^0 may be internally complex, carrying the same kinds of information as may be expressed by other elements or configurations in the syntax, but formed according to its constraints on morphology, not on syntax.

The following Swedish example from Müller & Wechsler (2014: 29) illustrates several properties which motivate the lexicalist analysis. It involves coordination of an active and a passive verb:

(6) Swedish
Golfklubben begärde och beviljade-s marklov för
golf.club.DEF requested and granted-PASS ground.permit for
banbygget efter en hel del förhandlingar och kompromisser
track.build.DEF after a whole part negotiations and compromises
med Länsstyrelsen och Naturvårdsverket.
with county.board.DEF and nature.protection.agency.DEF
'The golf club requested and was granted a ground permit for track construction after a lot of negotiations and compromises with the County Board and the Environmental Protection Agency.'

Müller and Wechsler argue that this example does not involve Right-Node Raising, but rather coordination of two finite verbs at the X^0-level (*begärde och beviljades*). Each verb is a syntactic word, marked for past tense (the *de* part of each), and the second one is marked for passive (the *s*). Hence the voice alternation active/passive is represented on single words, and does not involve spans of structure involving separate heads such as V, v, and Voice. Additionally, the second verb is a straightforward counterexample to the 'Mirror Principle' (Baker 1985), which is supposed to diagnose a close relationship between syntactic structure and word-internal morpheme structure. Swedish passive -*s* always appears external to other tense or aspectual suffixes on the word, even though in an expanded MP-style clausal structure the Voice head would be taken to be lower than and therefore closer to the lexeme stem with respect to Aspect or Tense heads.

The French example in (7) also motivates both the lexicalist approach, as well as the design feature that agreement is not directional.

(7) French
Je suis heureuse.
I am happy.F.SG
'I am happy.' (spoken by a female)

Neither the subject pronoun *je* nor the inflected verb *suis* are categorized or marked for gender – as in English and many other languages – yet the predicate adjective is marked as feminine (and singular). The non-formal linguistic intuition that the adjective agrees with its subject, or "agrees with a noun", has been the basis of many formalized linguistic analyses: the predicate adjective is a target and the subject should be its controller. Yet there is no plausible source in the lexical content of (7) for a feminine gender specification except for the adjective. It is certainly true of the *sentence* (7) that it expresses a meaning involving a feminine subject, but the morphosyntactic basis of that meaning could not be *je* or *suis*, under any plausible analysis of those words.

This example is very powerful. From it, it follows then that *heureuse* is a lexical item marked for feminine gender independently of the syntactic structure in which it appears, as there is no source for feminine in the rest of the structure. This entails the Lexicalist Hypothesis, as each X^0 in the syntactic structure has properties that do not refer to any other X^0 in the structure – usually referred to as ‘Lexical Integrity’ (e.g. Bresnan 2001: 92; Bresnan et al. 2016: 92).

2.2.2 Information-based clausal representation

Next, from the same example, it follows that “agreement” is the name we give to a situation in which two or more syntactic elements put constraints on a single informational unit, but that there is no priority of one element over the other(s). In (7), all three such elements (the words in this case) put constraints on what the subject is; and as the combination of those constraints is not contradictory, the example is well-formed. The f-structure contribution of each word is shown in (8). (9) shows the f-structure for the full example.

(8) a. $\begin{bmatrix} \textsc{pred} & \text{‘}\textsc{pro}\text{’} \\ \textsc{pers} & 1 \\ \textsc{num} & \textsc{sg} \end{bmatrix}$ = je

b. $\begin{bmatrix} \textsc{subj} & \begin{bmatrix} \textsc{pers} & 1 \\ \textsc{num} & \textsc{sg} \end{bmatrix} \\ \textsc{tense} & \textsc{pres} \\ \textsc{pred} & \text{‘}\textsc{be}\langle(\uparrow \textsc{subj})(\uparrow \textsc{xcomp})\rangle\text{’} \\ \textsc{xcomp} & \begin{bmatrix} \textsc{subj} & [\ ] \end{bmatrix} \end{bmatrix}$ = suis

c. $\begin{bmatrix} \textsc{subj} & \begin{bmatrix} \textsc{gend} & \textsc{fem} \end{bmatrix} \\ \textsc{pred} & \text{‘}\textsc{happy}\langle(\uparrow \textsc{subj})\rangle\text{’} \end{bmatrix}$ = heureuse

(9) $\begin{bmatrix} \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘}\textsc{pro}\text{’} \\ \textsc{pers} & 1 \\ \textsc{num} & \textsc{sg} \\ \textsc{gend} & \textsc{fem} \end{bmatrix} \\ \textsc{tense} & \textsc{pres} \\ \textsc{pred} & \text{‘}\textsc{be}\langle(\uparrow \textsc{subj})(\uparrow \textsc{xcomp})\rangle\text{’} \\ \textsc{xcomp} & \begin{bmatrix} \textsc{subj} & [\ ] \\ \textsc{pred} & \text{‘}\textsc{happy}\langle(\uparrow \textsc{subj})\rangle\text{’} \end{bmatrix} \end{bmatrix}$

Local syntactic relationships indicated by the structure-sharing seen above are typically located in a predicate – these will involve what kind of arguments the predicate takes, possibly specifications of case, whether it is a raising or control predicate, agreement information, and so on. The apparent directionality of agreement seen in canonical examples has nothing to do with "agreement" itself – as a mechanism of agreement does not exist – but rather comes from the second property of *suis*, that it is effectively a raising predicate, and so whatever is true of its complement's subject is true of its subject. In (9) the information shown as the value of SUBJ is the minimal amount of which the constraints coming from each of the entries in (8) is true.

With regard to the implications for explanatory adequacy, this simple example shows that the format of grammars is only consistent with those that lack derivation and directionality – in other words, if the hypothesis space is restricted to declaratively stated grammars, we expect that languages will quite generally show examples like (7). In Section 4.3, I take up in more detail the key information-based properties of what we informally refer to as "agreement".

Other examples also show the importance of the information that an item carries over its phrase structure properties. (10) (originally from Hudson 1977; see also Gazdar et al. 1985: 64; Bresnan 2001: 19, Bresnan et al. 2016: 14) illustrates one of the "paradoxes of movement":

(10) a. * I aren't happy.
b. Aren't I happy?

An initial positioning of an internally-negated auxiliary is taken as evidence in movement analyses that the auxiliary has undergone several movements, combining with Neg and then with T[ense] before moving to C. However, from the notional analytic source *I am not happy* there is no pre-T-to-C version **I amn't happy*, as in (standard) English there is no form *amn't*, and (10a) is also ungrammatical. In this use, then, the form *aren't* is a word which can only appear in the C, or inverted-aux, position, but not in any other position. In terms familiar from the early days of transformational grammar, (10b) would have to be analyzed as a grammatical example derived from an ungrammatical source.

The contrast in (10) shows that the syntactic properties that an item has (being a tensed negated auxiliary in this case) are not inexorably associated with structural derivations which aggregate information. A movement-based account of (10) would have to assume that the syntactic features of *aren't* can be assembled on T, and from there moved to C, but that there is no lexical item which can spell out those features on T, but only on C. In other words, what actually matters is the surface position of the realization of a set of syntactic properties, not where (or where else) those properties came from. This is exactly what a declarative framework such as LFG provides, with the same implication for explanatory adequacy – if an element in a higher position must correspond to a derivationally related version of itself in a lower position, pairs like (10) should not exist. But they do exist, and they show that the format of grammars should recognize that words have bundles of features which are associated with (sets of) syntactic positions. Within the broader Miminalist approach, the realizational account of morphology in the Distributed Morphology framework (Halle & Marantz 1993; see Bobaljik 2017 for a recent overview) has the same sensitivity to syntactic position: for *aren't*, a rule of vocabulary insertion could be made sensitive to the collection of relevant syntactic features in the context of C, but not of T.

2.3 Rules and representations

As syntactic frameworks have developed since the 1980s, they have diverged as to whether the focus is on constraints stated on representations, or on steps in a procedural derivation. Government-Binding theory has a mix of properties: conditions on rule application and conditions on representations. For instance, the examples above in (3)–(4) involve *appear* as a raising predicate which requires an operation of the rule Move-α, while *appeal* is a control predicate which requires a representational check involving a PRO subject (see e.g. Haegeman 1994 for a summary of GB). More recently, the "movement theory of Control" (e.g. Hornstein & Polinsky 2010) eliminates the representational condition on the empty category subject in favor of a derivational analysis similar in the relevant ways to the one for the raising predicate.

The GB Binding Theory Principles A and B were originally each stated as a condition on a representation. For instance, Principle A looks for a specific relationship of coindexing between antecedent and anaphor within a certain domain; reinterpreted as a condition on rule application, the principle must involve an operation of movement up to (near) the antecedent, within a certain domain, following an idea first proposed in Lebeaux (1983).

In the development of the MP, Chomsky has taken the view that as some aspects of the grammar are procedural, and so require conditions on rule application, parsimony would dictate that all grammatical conditions are of that type, with no conditions on representation. Hence the levels of GB over which representational conditions were stated were eliminated. The MP is an attempt to deconstruct GB along purely procedural aspects (see e.g. Hornstein 2018: 54) –

in the limit, there is no "stopping off" at any point to evaluate a representation. In fact, though, each step in a derivation must involve a local representation – but one within which or to which some further operation should take place. The proposed operation of 'Minimal Search' for the operation *Agree* (Chomsky 2007: 9) must inspect a structure to find something within it – here, an element W probing within a structure Z – and the outcome of that will constrain what (procedurally) can happen next: "Since W contains all information relevant to further computation involving Z, W is also necessarily the probe that selects a goal in any internal modification of Z. Minimal search conditions limit the goal of the probe to its complement, the smallest searchable domain."

The output of syntax is fed to the 'interfaces'. On the semantic side, the end of the syntactic derivation corresponds to the GB level of Logical Form (LF), which feeds to the 'conceptual-intentional' interface. On the phonetic side, the overt output of the derivation is spelled-out to Phonetic Form (PF), which feeds to the interface known as 'articulatory-perceptual' or 'sensorimotor' (see e.g. Chomsky 1995: 2, Chomsky 2007: 5). One leading idea of the Minimalist Programme is that LF and PF have no properties specific to them; rather, any apparent well-formedness conditions are due entirely to properties of the interfaces.

Within the core domain of syntax, there seem to be several phenomena which bear on the issue of rules vs. representations, and which appear to favor the latter – because their analysis seems irreducibly representational. I will mention two different instances and then go on to two others in more detail. First, as just noted, the MP operation of Agree has to access a representation, in order to establish a relation between Probe and Goal (see also Section 4.2). Second, the approach to case marking known as 'Dependent Case' (e.g. Baker 2015) calculates the case values of NPs by referring to larger structure – the underlying intuition being that in a clause containing two NPs, a subject c-commanding an object, the marked case value of Accusative for the object is the value assigned to an NP c-commanded by another, and in a typologically different system, the marked case value of Ergative for the subject is the value assigned to an NP which c-commands another. Hence the computation of case values must refer to a structural representation.

I now go in more depth into two instances which illustrate a different kind of representational condition – a negative condition. It is difficult to imagine how such conditions could successfully be captured procedurally. Returning to the binding conditions of GB, for Principle A, there have been different proposals to reinterpret it derivationally, for instance Lidz & Idsardi (1998), Hornstein (2001) and Boeckx et al. (2007), though others take a more traditional view, such as Safir (2008) and Charnavel & Sportiche (2016). While Principle A requires two elements to be in a certain relationship, Principle B forbids two elements from having a certain relationship – it is a negative condition. A procedural reinterpretation does not seem directly possible for Principle B, as it requires disjointness (unless perhaps the system of recording contra-indexing of Chomsky 1980 is revived), though Reuland (2011) presents a revised Binding Theory which refers to properties of predicates and semantic constraints on the interpretation of their arguments.

Principles A and B as they apply to English are familiar. In some languages, with anaphoric systems more complex than that found in English, conditions on the various elements of the system may involve both positive and negative constraints – such as in Norwegian (Dalrymple 2001: 279–288, Bresnan et al. 2016: 259–261). Norwegian has four relevant anaphor/pronoun forms, shown in (11) with their LFG binding properties. The content of the binding properties is given in (12):

(11) Featural analysis of Norwegian pronouns:

a. seg [+sbj, −ncl]

b. ham [−ncl]

c. seg selv [+sbj, +ncl]

d. ham selv [−sbj, +ncl]

(12) a. [+sbj, −ncl] – The antecedent must be a subject in the minimal finite domain outside of the minimal nucleus containing the pronoun.

b. [−ncl] – The antecedent must be outside of the minimal nucleus containing the pronoun.

c. [+sbj, +ncl] – The antecedent must be a subject in the minimal nucleus containing the pronoun.

d. [−sbj, +ncl] – The antecedent must be a nonsubject in the minimal nucleus containing the pronoun.

The negative conditions here seem to refer crucially to representations – to check that a relationship does not hold in a certain local domain, or to check that a relationship does hold, but not with a subject.

A different consideration about the role of representations comes from the distribution of the depictive *sisxoli* ‘alone’ in Tsez, a language of the Caucasus which has an ergative-absolutive case marking system. The depicitive may be associated with a preceding NP, but may not itself precede its NP associate (Polinsky 2000). Hence the depictive has two possible associates in (13a), one in (13b), and none in (13c).

(13) Tzez

a. kid-bā ziya sisxoli bišer-si
girl-ERG cow.ABS alone feed-PST.EVID
'The girl$_i$ alone$_i$ fed the cow.'
'The girl fed the cow$_i$ alone$_i$.'

b. kid-bā sisxoli ziya bišer-si
girl-ERG alone cow.ABS feed-PST.EVID
'The girl$_i$ alone$_i$ fed the cow.'
*'The girl fed the cow$_i$ alone$_i$.'

c. * sisxoli kid-bā ziya bišer-si
alone girl-ERG cow.ABS feed-PST.EVID

The linear precedence condition is reinterpred as one of c-command in later discussions of these same examples in Polinsky & Potsdam (2006) and Fukuda (2008) – the associate must c-command the depictive.

The distribution of the depictive becomes more interesting in the context of raising and control predicates. In Tsez the predicate *yoq-* 'begin' is ambiguous between control and raising, and in fact is a backward control predicate in its control use or a forward raising predicate in its raising use (Polinsky & Potsdam 2006). In LFG, the higher and lower SUBJ values of control or raising are structure-shared in f-structure, and as discussed in Sells (2006) that f-structure property is consistent with c-structure expression of the relevant argument in the matrix clause ('forward') or in the embedded clause ('backward'). (14a) is an interesting example regarding the syntax of the depictive, as it is grammatical even though the depictive apparently precedes its associate. Polinsky & Potsdam (2006) analyze this as a backward control structure: a null (absolutive) subject of 'begin', indicated by ∅ in (14b), controls the lower (ergative) subject of 'feed'. ∅ is used here as a notation to suggest the analysis of the example, but it has no actual correspondent in the c-structure, as is standard in the LFG analysis of control and raising. In this example, it is the null matrix argument indicated by ∅ which is the associate of the depictive, and both are constituents of the main clause (see the f-structure in (17) below):

(14) Tzez

a. sisxoli kid-bā ziya bišra yoq-si (backward control)
alone girl-ERG cow.ABS feed begin-PST.EVID
'The girl$_i$ alone$_i$ began to feed the cow.'
*'The girl began to feed the cow$_i$ alone$_i$.'

b. $\varnothing_i$ sisxoli [kid-bā$_i$ ziya bišra] yoq-si
   alone [girl-ERG cow.ABS feed] begin-PST.EVID

The other use of *yoq-* is as a regular forward raising predicate. Its subject is in absolutive case as the predicate is not formally transitive, and the subject in the lower clause is the empty position, again indicated here by ∅. As seen in (15a), with the syntactic analysis in (15b), the same order of elements as in (14a) is ungrammatical in this instance, as the depictive does indeed precede its associate:

(15) Tzez
   a. * sisxoli kid [ziya bišra] yoq-si (forward raising)
      alone girl-ABS [cow.ABS feed] begin-PST.EVID
   b. * sisxoli kid$_i$ [$\varnothing_i$ ziya bišra] yoq-si
      alone girl-ABS [cow.ABS feed] begin-PST.EVID

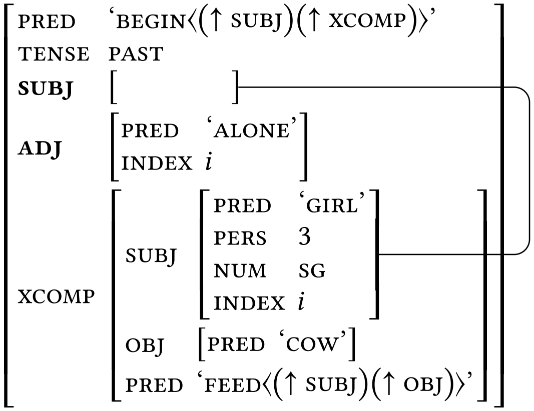
The LFG account of this data requires the concepts of f-command, which is like c-command but stated on f-structure, and of f-precedence (see Glossary for f-command and f-precedence). This latter concept makes reference to the c-structure expression(s) – if any – of f-structure elements. Crucially, an element such as a null argument which is present only in f-structure, but not in c-structure, has no (f-)precedence relations defined on it (Bresnan 2001: 195; Bresnan et al. 2016: 213). The LFG analysis of the Tsez depictive can be stated simply as in (16):

(16) a. The associate and the depictive f-command each other.
   b. The depictive must not f-precede the associate.

(16a) is essentially a clause-mate condition, and (16b) is a negative condition. It does not require that the associate f-precede the depictive, but rather that the depictive does not f-precede the associate.

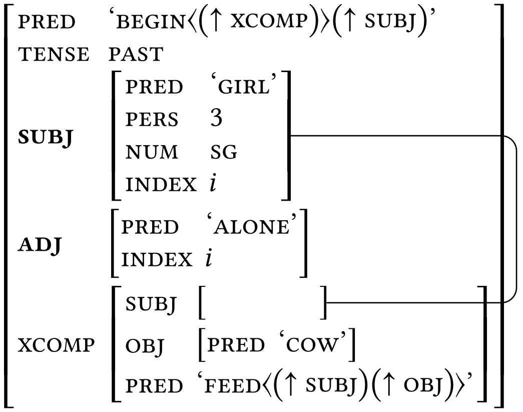
The f-structure of (14) is shown in (17), leaving out the case values of the arguments, which would formally conflict under straighforward structure-sharing (i.e. formal equality in LFG terms). The case values require a slightly nuanced analysis, whatever the framework (see Polinsky & Potsdam 2002, Sells 2006). For presentational purposes, I assume here that the formal relation between depictive and associate is that they share an INDEX value. Their GFs which f-command each other are indicated by the boldface GF names, in the matrix nucleus. The matrix SUBJ is structure-shared with the embedded SUBJ as the predicate is backward control. While there is a matrix SUBJ in f-structure, it has no c-structure expression (there is no '∅' in the c-structure); only the embedded subject is present in c-structure. Consequently, limited to the matrix f-structure in which the associate and depictive f-command each other, no precedence relation is defined on the boldface SUBJ, and so the condition in (16b) is also satisfied.

(17) F-structure of (14), ignoring the case values. **ADJ** does not f-precede **SUBJ**:



In contrast, for (15), involving a forward raising use of the predicate, the constraint in (16b) is not satisfied, because the SUBJ is overt in the matrix clause, and so f-precedes the depictive ADJ.

(18) F-structure of (15), ignoring the case conflict. **ADJ** f-precedes **SUBJ**:



The Minimalist account in Polinsky & Potsdam (2006) (see also Fukuda 2008) is stated in terms of positive conditions, of which (19b) is the important one.

(19) a. The associate and the depictive are clause-mates.
     b. The associate c-commands/binds (≡ precedes) the depictive.

What is interesting about (19b) is that it can only be successfully interpreted representationally. Suppose that at one point in the derivation, the associate (whether overt or covert) c-commands the depictive, and the relevant syntactic relationship is established, satisfying (19b). However, what is to prevent some later operation which scrambles the depictive higher, so that it c-commands its associate, in violation of (19b)? To prevent this possibility, (19b) must be interpreted as an output condition on the "final" representation, regardless of when during the derivation the relation between associate and depictive has been established. Hence even though (19b) is a positive condition, not a negative one, it is necessarily representational.

In the LFG analysis, (16b) is necessarily a negative condition, as the null subject in backward control is only represented in f-structure (14), and so could never be evaluated against a positive precedence condition like (19b). Evidence from other languages supports the position that null arguments are present in f-structure but absent from c-structure. Null pronouns in Malayalam are not sensitive to f-precedence conditions, unlike overt pronouns (Mohanan 1983: 664–665). Kameyama (1985) presents a similar argument for Japanese (summarized in Dalrymple 2001: 171ff. and 288ff.). For Malayalam, Mohanan observes that an overt pronoun may not precede its antecedent – compare (20) and (21a) with (21b) – while a null pronoun (indicated for presentational purposes by *pro* in (21b)) may 'precede' its antecedent:

(20) Malayalam

a. [kuṭṭiyute ammaye] awan n̲uḷḷi
[child.GEN mother.ACC] he.NOM pinched
'He$_i$ pinched the child$_i$'s mother.'

b. * [awante ammaye] kuṭṭi n̲uḷḷi
[he.GEN mother.ACC] child.NOM pinched
'The child$_i$ pinched his$_i$ mother.'

(21) Malayalam

a. [awan aanaye n̲uḷḷiyaṯinə ṡeeṣam] kuṭṭi$_i$ uraŋŋi
[he.NOM elephant.ACC pinched.it after] child.NOM slept
'The child$_i$ slept [after he$_{*i/j}$ pinched the elephant].'

b. [*pro* aanaye n̲uḷḷiyaṯinə ṡeeṣam] kuṭṭi$_i$ uraŋŋi
[elephant.ACC pinched.it after] child.NOM slept
'The child$_i$ slept [after he$_{i,j}$ pinched the elephant].'

The overt pronoun ‘he’ in (21a) may not take ‘child’ as its antecedent, as the former precedes the latter, but this restriction is not there with the null pronoun in (21b). Why would overt and null pronouns have different precedence conditions on them? Mohanan (1983: 664) proposes that the correct analysis is that a pronoun cannot precede its antecedent, where precedence is defined on c-structure elements, such as overt pronouns, but is not defined for null pronouns, which are present only in f-structure.

Consider the c-structure relationships of the relevant parts of the examples, shown in (22a), with the f-structure of the example shown in (22b). The subscript numbers show the c-to-f-structure correspondences:

(22) a. C-structure:
(pronoun$_1$) pinched$_2$ child$_3$
b. F-structure:

$$\left[\begin{array}{ll} \text{SUBJ} & {}_3\left[\text{PRED}\ \text{‘CHILD’}\right] \\ \text{PRED} & \text{‘SLEEP}\langle(\uparrow \text{SUBJ})\rangle\text{’} \\ \text{TENSE} & \text{PAST} \\ \text{ADJ} & {}_2\left[\begin{array}{ll} \text{PRED} & \text{‘PINCH’} \\ \text{SUBJ} & {}_1\left[\text{PRED}\ \text{‘PRO’}\right] \\ \dots & \dots \end{array}\right] \end{array}\right]$$

In both examples in (21), the adjunct clause 2 f-precedes 3, ‘child’, because the c-structure correspondent(s) of 2 precede the correspondent(s) of 3. However with regard to 1 and 3, 1 f-precedes 3 only if 1 is present in c-structure, which is only the case in (21a). Hence the apparently different binding properties of pronouns reduce to their different properties in different parts of the syntactic analysis.

The implications of this analysis are far-reaching: if certain syntactic elements can have a range of grammatical properties without being represented in phrase structure – and the above is positive evidence that they are not represented in phrase structure – then every aspect of grammatical analysis which can or must refer to those properties must also be independent of any phrase structure representation, including phenomena such as subjecthood, agreement, binding, and so on.

Declarative frameworks have different dimensions of analysis – such as c-structure and f-structure as described below – but not different levels in the sense that GB had (e.g. D-structure, S-structure). As there are no rules or operations, there are no conditions on rules, and so all conditions are stated over representations, as constraints.

3 Phrase Structure

3.1 Heads and headed structures

LFG c-structures have some similarities with the S-structures of late GB. A canonical clause (for an SVO language) is structured around what I refer to as a 'skeleton' with a 'spine' (Sells 2001: 17). (24) below shows the skeleton, and the spine corresponds to all the non-argumental parts, V, I, C and their projections. These are separate categories which participate in the familiar clausal extended projection (Grimshaw 2000: 116ff. Bresnan 2001: 100), often now referred to as the 'Hierarchy of Projections' (Adger 2003).

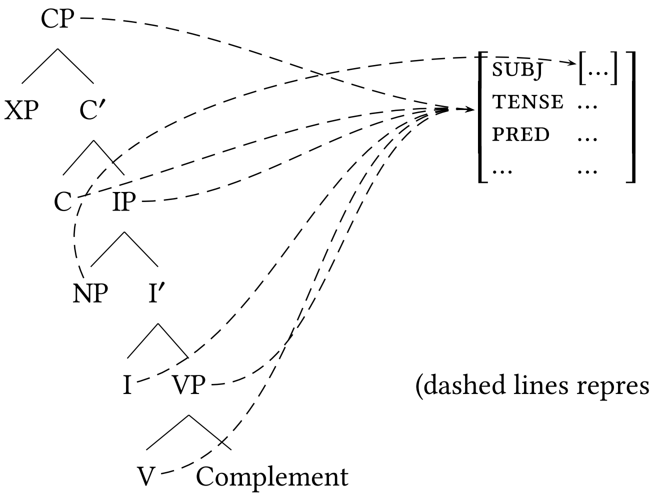
The formal relation in the c-structure between V and I and C is usually developed from the idea of 'extended projection' of Grimshaw (2000); see also Bresnan (2001: 100–101), Bresnan et al. (2016: 103). The clausal categories are all projections of the category verb, which is specified by the traditional labels [+V, −N] (Chomsky 1970).

(23) Extended Projections

a. V = [+V, −N, P0] (the zeroth-level projection of V)
b. I = [+V, −N, P1] (the first-level projection of V)
c. C = [+V, −N, P2] (the second-level projection of V)

The outline clause structure has specifiers of CP and IP, and complement position(s) within VP, schematized here with the placeholder label Complement.

(24)



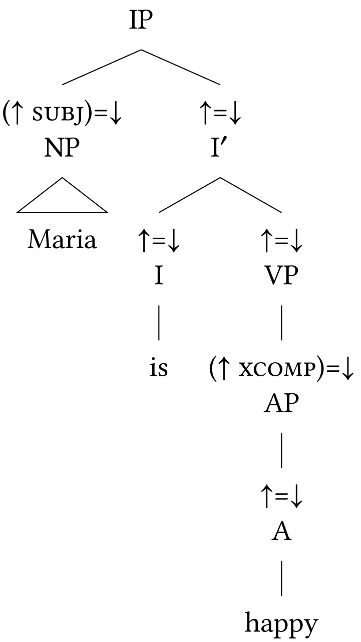
(dashed lines represent information flow)

Each node in the c-structure is annotated as to how it contributes to the f-structure. The formal annotations on nodes are not shown in (24), for simplicity, but the dashed lines represent the way that information flows from the c-structure to the f-structure.

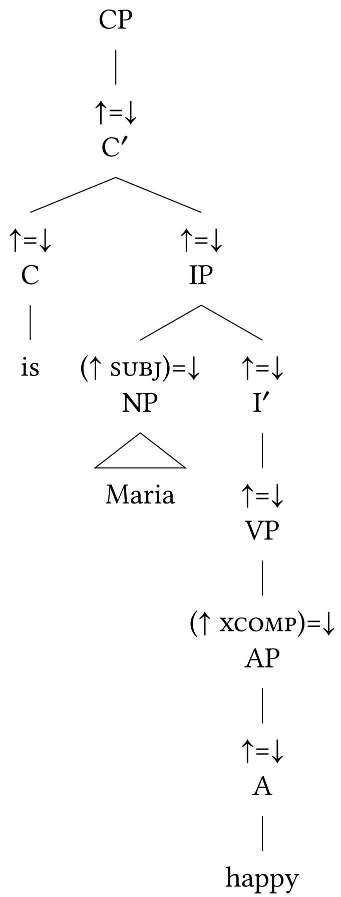
Every node in the clausal spine contributes its information to the main f-structure, as can be seen from the several lines converging on the outer f-structure, which represents the grammatical information of the clause (again for simplicity, I omit lines from the X′ nodes). The other nodes, XP, NP, and Complement, have different annotations on them, as they contribute to parts of the overall f-structure. For instance, the annotation on the node NP would indicate that its contribution is as the subject – in other words, NP as specifier of IP is the subject position. This is indicated by the dashed line going from NP in the c-structure to the value of SUBJ in the f-structure.

As far as clausal information is concerned, the verb itself contributes identically to the clause whether it is in V or in I or in C, a property usually referred to as 'head mobility' (see e.g. Bresnan 2001: 126ff. Bresnan et al. 2016: 129ff.). For instance, unless extra information is associated with the C node in (25b), both c-structures in (25) would determine the same f-structure:

(25) a.



b.



Head mobility can be illustrated with the c-structures above. On the assumption that the only VP can be the c-structure complement of I, then for the example *Maria is happy* in (25a) the VP lacks a c-structure head V, for the verb *is* is in I; and in (25b), for the string *Is Maria happy*, both IP and VP lack their X^0 heads. In these structures the finite form of *be* acts as an auxiliary verb, and so does not head a surface VP, but appears in a higher functional head position (in contrast *be* as a non-finite form would head VP, as in *Maria could [be happy]*).

Formally, the theory requires that every XP either has a c-structure head in the standard X′ sense, or that it maps to an f-structure shared with at least one YP which is headed in c-structure. Such a Y^0 is known as the 'extended head' of XP (the notion is originally due to Zaenen & Kaplan 1995: 221, revised to the formulation given here by Bresnan 2000: 353). So in (25a), I is the extended head of VP, and in (25b), C is the extended head of IP and of VP, leading to the illusion that the head is "moving". Different verbal categories may be restricted, though, to particular c-structures positions: finite auxiliaries in English may only appear in I or C, not in V; finite non-auxiliaries must appear in V. Hence finite auxiliaries have the category [+V, −N, P>0] and finite verbs have the category [+V, −N, P0].

The discourse in the MP literature over the past 25 years as to whether head movement exists or whether it is part of 'narrow syntax' (see e.g. Roberts 2011; Harizanov & Gribanova 2019 for overviews) is quite puzzling from the perspective of declarative frameworks such as LFG or HPSG, as heads are central to the syntactic analysis. The issue arose in the development of the MP as position-occupying head movement does not obey the Extension Condition of Chomsky (1995), requiring that every operation of Merge extends the root node of the current tree. Head movement violates this condition, as it formally involves adjunction to a node lower than the root node (in contrast to XP adjunction, which does adjoin at the root). Consequently Chomsky raised the issue of the status of head movement (e.g. Chomsky 2000: 136–137; Chomsky 2001: 38) within the MP approach.

As the mapping from c-structure to f-structure in LFG suggests, the crucial fact about a clausal spine is that head positions share information, each being a functional co-head (see e.g. (25)). This is directly evidenced in various core cases of multiple expresssion of the same grammatical information in a single domain, as first described in LFG by Niño (1997). The same properties of clausal information are expressed on more than one head (see also Sells 2004, Lødrup 2014). The Finnish examples in (26) and (27) (Niño 1997: 135, 137) show the phenomenon:

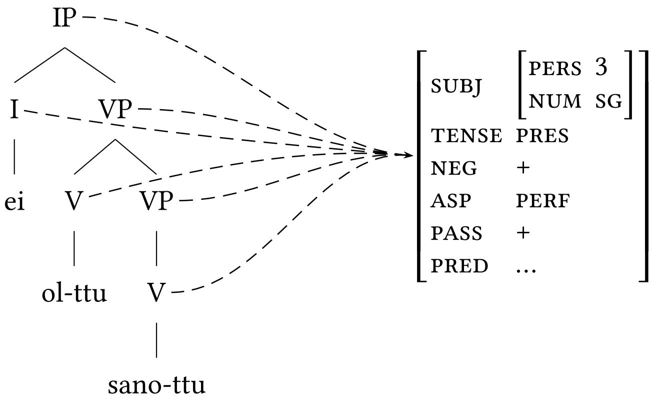
(26) Finnish
Äl-kää puhu-ko.
NEG.IMP-IMP.2.PL speak-IMP
'Don't (you pl.) speak!'

(27) Finnish
a. Ei ol-lut sano-ttu
NEG.3.SG PRF-PST.PTCP.SG say-PASS.PST.PTCP.SG
'It has not been said.'
b. Ei ol-ttu sano-ttu
NEG.3.SG PRF-PASS.PST.PTCP.SG say-PASS.PST.PTCP.SG
'It has not been said.'

(26) involves a special form of negation restricted to imperatives, as well as imperative marking on both the auxiliary and the main verb. In (27), singular

marking appears on all three words, the negative, which is a kind of auxiliary, another auxiliary, and the main verb. These examples also indicate that 'passive' is a feature in f-structure which can be accessed – see also Lødrup (2014) for evidence in Norwegian for the same conclusion. (27b) is a colloquial variant of (27a), in which the passive marking on the main verb also appears on the medial auxiliary. The c- and f-structure of (27b) are shown in (28). It can easily be seen that the constraints coming from each of the words in (27b) – using the glosses as a guide – are satisfied by this f-structure:

(28)



The co-head approach of LFG allows for different sources of the same constraint (e.g. that the value of num is sg) which will be true of just a single object (e.g. the subject). Hence feature exponence may be distributed or apparently multiplied. In the MP approach, each feature necessarily originates in only one position in the structure, and then must be copied or spread onto other positions, for data such as that above. In MP analyses, 'imperative' corresponds to a high position in the clause, so the imp feature in (27) must spread downwards. However in (27b), the pass feature would originate on the lowest verb, the only one marked in (27a), and so would have to spread upwards.

The distribution of morphological exponence is probably not related to direction of spreading, but rather concerns morphological constraints on each type of word as to what features it must express, might express, or cannot express. This can be seen clearly in the examples in (30) from Livonian (Niño 1997: 131), which obey the generalizations in (29):

(29) a. verbs are marked for number
b. participles are not marked for person

(30) Livonian

a. ä-b u-m and-ên-∅
NEG-1 be-1.SG give-PST.PTCP-SG
'I have not given.'

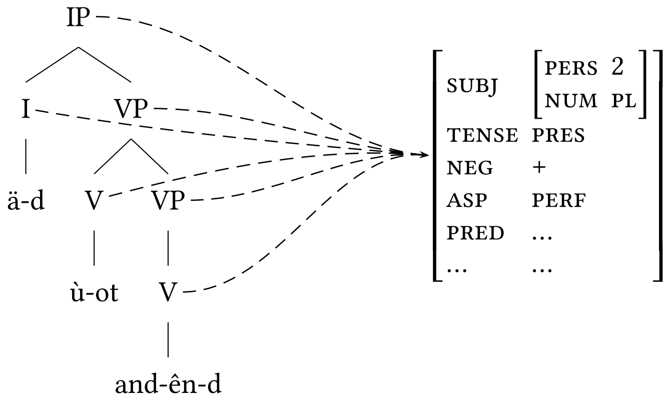
b. ä-b ù-om and-ên-d
NEG-1 be-1.PL give-PST.PTCP-PL
'We have not given.'

c. ä-d ù-od and-ên-∅
NEG-2 be-2.SG give-PST.PTCP-SG
'You have not given.'

d. ä-d ù-ot and-ên-d
NEG-2 be-2.PL give-PST.PTCP-PL
'You have not given.'

(31) shows the c- and f-structure of (30d). Again following the gloss, it can be seen that the PERS value of the subject is identically constrained by the first two words, while the NUM value is constrained by the last two words:

(31)



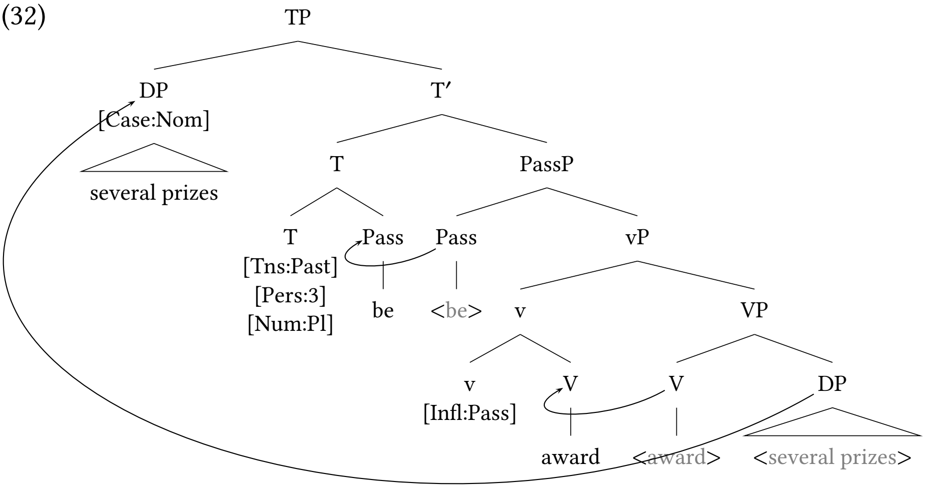
These patterns of multiple expresssion extend beyond simple clauses, into various kinds of complex predicate (see e.g. Sells 2004, Lødrup 2014), which might require a more nuanced syntactic analysis than simple embedding of f-structure nuclei – as argued for on the basis of entirely different data by Andrews & Manning (1999). The multiple expression data could profitably be analyzed in a realizational framework (as suggested in Sells 2004) – every informational element within a certain domain must have at least one rule of realization applying to it (this idea is formalized explicitly in Crysmann & Bonami 2016), but in certain circumstances one piece of information can be referred to more than once, as the generalizations in (29) suggest. Crucially, again, it is not that one piece of morphological exponence on a c-structure head is copied to another head, but rather that different (co-)heads are acting as exponents of the same grammatical information.

3.2 The MP approach to phrase structure: Merge

The legacy of the Government-Binding model of syntax into the MP is a procedural approach to structure and structure-building. Binary structures are built up by Merge of two elements, often known as External Merge or 'first merge'. The GB idea of movement is reinterpreted in the MP as Internal Merge – one element from within a given structure is (re-)merged near the top of the structure. As noted in Section 1, the argument structure of the predicate is represented in a vP-VP structure, above which there are further projections such as TP and CP. By the time the structure has built up at least to TP, this structure effectively codes clausal information.

Strictly speaking, the syntactic derivation is abstract, with syntactic relationships referring to the structural notion of c-command but not linear order, which comes in the mapping from syntax to Phonetic Form (PF). The relevant terminal nodes of the structure are spelled out as words via the principles of Distributed Morphology (for an overview of this framework, see Harley & Noyer 2003 or Embick & Noyer 2007). Consider the derivation in (32) of the example *several prizes were awarded*, which will also feature later on:

(32)



The internal argument of a transitive verb is merged with V (a kind of root) within VP, at the lower right of the structure. The structure builds up bottom-up via successive applications of Merge. The VP is immediately the complement of a "little v" vP, which introduces the external argument in a canonical transitive. The particular example here is a passive, with the external argument suppressed. The two components of the verb, V and v, are combined by head movement of V to v, as the structure shows. The notation <award> indicates the original position of V before movement.

The passive is indicated here by PassP, following the analysis of English auxiliaries in Adger (2003). The Pass head *be* merges with vP as its complement. Next, above that, a T′ is created with a formative for past tense in its T head. The auxiliary verb *be* also undergoes head movement, to combine with T. Finally, following X′-theory, T′ has a specifier which hosts the surface subject. In the case of a passive example, a DP is raised from a VP-internal position to fill the subject position.

The arrows in (32) indicate movement. Standardly in the Minimalist approach, movement leaves behind a 'copy' of the moved constituent (Chomsky 1995), which the notation such as <award> etc. is intended to convey. Principles of realization at PF determine which copy is overt (pronounced) – usually the highest copy, as in the earlier versions of transformational grammar where moved constituents leave behind a trace, which is by definition unpronounced. The formalization of MP syntactic derivations due to Collins & Stabler (2016) captures the 'copy' idea by treating each operation of movement as creating a multidominance structure from an single terminal element; that formalization is extended to head movement by Bleaman (2021). However, for presentational purposes, I show the more familiar movement-with-copies structures here.

There are complex heads in T and v in (32), both formed via head movement. They also have a representation of the various features which are present or which are valued during the course of the derivation (see Section 4.2). These complex heads will realize their lexical and featural information as the words *were* and *awarded*.

Hence, the phrase structure derivation in the Minimalist approach represents all the clausal information, somewhat like LFG's f-structure, which is then spelled out as the overt form, which corresponds to some extent to LFG's c-structure.

3.3 Phrase structures are not isomorphic to clausal information

One difference between LFG and MP concerns how far the phrase structure is a direct representation of the clausal information. As just noted, the representation of clausal information in a Minimalist approach is encoded within the phrase structure (in its configuration and its derivation), while in LFG the relation between f-structure and c-structure is fundamentally more flexible. The Mandarin Verb Copy Construction (Li & Thompson 1981, Huang 1982) serves as a good example of how clausal information at f-structure can exist independently of any particular c-structure property. Postverbal arguments and adjuncts appear to be in competition within a single VP: in order to express an argument and an adjunct, the main verb must be duplicated to form a second VP, as in (33b).

(33) Mandarin

a. * Zhangsan tan gangqin de hen hao
Zhangsan play piano LNK very well
'Zhangsan plays piano very well'

b. Zhangsan tan gangqin tan de hen hao
Zhangsan play piano play LNK very well
'Zhangsan plays piano very well'

Huang (1982) proposed a phrase structure filter which essentially disallows arguments and adjuncts in the same VP. Fang & Sells (2007) note that both arguments of a ditransitive verb appear in the first VP (underlined in (34a)), and that an object may be displaced from within the first VP, otherwise preserving the phrase structure:

(34) Mandarin

a. wo song ta zhe jian liwu song de hen hao
I give him this CL gift give LNK very well
'I gave him this gift and it turned out to be a very good idea.'

b. zhe jian liwu wo song ta _ song de hen hao
this CL gift I give him _ give LNK very well
'This gift, I gave (it) him and (it turned out to be) very good.'

However, if an object is displaced from a monotransitive VP, verb "copying" is no longer an option (also see Huang 1982: 53):

(35) Mandarin

a. * gangqin Zhangsan tan _ tan de hen hao
piano Zhangsan play _ play LNK very well

b. gangqin Zhangsan tan _ de hen hao
piano Zhangsan play _ LNK very well
'The piano, Zhangsan played (it) very well.'

If we take (36) to be the basic f-structure of what should be expressed in one structure of (33), given the constraint that arguments and adjuncts cannot be in the same c-structure VP, it follows that (33b) is the only possible expression.

(36)
$$\left[\begin{array}{ll} \textsc{pred} & \text{`}\textsc{play}\langle(\uparrow \textsc{subj})(\uparrow \textsc{obj})\rangle\text{'} \\ \textsc{subj} & \left[\textsc{pred}\ \text{`}\textsc{Zhangsan}\text{'}\right] \\ \textsc{obj} & \left[\textsc{pred}\ \text{`}\textsc{piano}\text{'}\right] \\ \textsc{adjunct} & \left[\textsc{pred}\ \text{`}\textsc{very well}\text{'}\right] \end{array}\right]$$

On the other hand, if 'piano' appears as a structural TOPIC, in clause-initial position, only a single VP is required to express the in-situ material, which consists of the PRED and its ADJUNCT in (37), as in example (35b). The identification of TOPIC and OBJ takes place only at f-structure (Kaplan & Zaenen 1989).

(37)
$$\left[\begin{array}{ll} \textsc{topic} & \left[\textsc{pred}\ \text{`}\textsc{piano}\text{'}\right] \\ \textsc{pred} & \text{`}\textsc{play}\langle(\uparrow \textsc{subj})(\uparrow \textsc{obj})\rangle\text{'} \\ \textsc{subj} & \left[\textsc{pred}\ \text{`}\textsc{Zhangsan}\text{'}\right] \\ \textsc{obj} & \left[\qquad\qquad\right] \\ \textsc{adjunct} & \left[\textsc{pred}\ \text{`}\textsc{very well}\text{'}\right] \end{array}\right]$$

Examples such as (35b) show that the competition between arguments and adjuncts for the same VP is a phrase-structure phenomenon, and is not relevant for the level of clausal grammatical information: a verb in Mandarin can perfectly well have a full array of arguments and any adjuncts, but only some of those can be expressed within a single VP. Following a careful survey of the research on this topic, Bartos (2019) proposes an MP analysis which has to appeal to haplology of V to derive (35b) from (35a) (already suggested in Huang 1982: 99), but this is merely symptomatic of an underlying misanalysis, for the core relations between a predicate and its arguments and adjuncts are not isomorphically represented in phrase structure.

4 Features and agreement

4.1 Feature theory and LFG

LFG is built on the foundation that featural specifications in morpho-syntax are of the form [attribute value], and that well-formedness requires every attribute in a given representation to have an appropriate value. The attribute-value format is used in LFG to represent functional structure, which represents the relational and featural content of a clause, but not constituent structure. F-structure is deliberately designed to not look like a phrase structure, to signify that it represents a different kind of syntactic information, and also that the parts within it are unordered. (The concept of f-precedence in LFG (see Glossary) crucially makes reference to the c-structure realization of f-structure elements.)

Adger (2010) considers features and the format of features in MP. He also concludes that features should be represented as attribute-value pairs, but rejects the idea that feature names can have structured values, because that re-creates the hierarchical structure within the phrase structure (e.g. a structured value for SUBJ corresponds to a DP in the phrase structure with internal constituency). Of course, there is no claim in LFG that every attribute in f-structure is the name of a feature – 'f' stands for 'functional', not 'feature'. Hence the closest comparison will be the atomically-valued attributes in f-structure, which will correspond most closely to features in MP, and which also accord with the general notion of morpho-syntactic features. More precisely, these will be the 'syntactic' features identified by Sadler & Spencer (2001) (see also Spencer 2013), which are the target of morpho-syntactic exponence (as in the discussion of Finnish and Livonian above in Section 3.1). In this subsection I compare the LFG and MP approaches to such features. An extended discussion of features in the MP in comparison to other frameworks can be found in Asudeh & Toivonen (2006: 409–420).

Featural information associated with each word introduces constraints on the well-formedness of the whole structure, within a 'monotonic' system: information cannot be selectively ignored, nor can it be changed. Hence declarative frameworks such as LFG necessarily have a property which has come to feature in MP discourse – the 'No Tampering Condition' (Chomsky 2007), which does not allow information on an item to be changed as it is merged in as part of the derivation (see also Section 5).

For instance, (38) is ungrammatical as not all the constraints coming from the lexical items can be satisfied simultaneously, and no part of the information can be ignored:

(38) * You am happy.

In this example, *you* will specify the value of PERSON of the subject as 2, but *am* will specify that same value as 1. There is no way to satisfy the requirements of these first two words in a single structure.

LFG introduces featural information either via lexical items or by the rules which license phrase structure. Every well-formed feature specification in f-structure is of the form [attribute value], by definition (see e.g. Kaplan & Bresnan 1982: 181–182). If any lexical item specifies a feature but without a value, that is an 'unvalued feature'; some other element in the structure must introduce the value for that feature, or else the overall structure will be ill-formed. Unvalued features play a significant role in MP analyses (because their function goes beyond that of simply representing information; see Section 4.2); they also find their place in declarative analyses, as described below, although valued features tend to be the norm.

The basic way for information to be specified is as a defining equation – for instance the information carried by the appropriate lexical entries to give the f-structures in (8). There is another kind of informational contribution, the constraining equation of LFG. Kaplan & Bresnan (1982: 207–209) motivate constraining equations with familiar facts such as those in (39), with their proposal for analysis in (40):

(39) A girl is handing (*is hands, *is handed) the baby a toy.

(40) *is*: (↑ XCOMP PARTICIPLE) $=_c$ PRESENT

The VP complement of *is* has the grammatical function XCOMP in the LFG analysis, and within that, the grammatical form *handing* would provide the value 'PRESENT' for the attribute PARTICIPLE. That fulfils the requirement in (40). The important move to a constraining equation over a defining one concerns the ungrammatical variants in (39). For instance, as a finite form, *hands* is not specified at all for the attribute PARTICIPLE, and so does not provide the information that (40) needs. However, if that information in (40) were specified as defining information, it would be unified in with the information from *hands*, and – at least on that count – the sequence **is hands* would not be ungrammatical, as nothing would be inconsistent. Kaplan and Bresnan note that in a unification-based system, constraining equations have the important consequence that negative-value specifications for otherwise unnecessary features can be avoided. (For more on features see Kaplan 2019b.)

Accounts involving a constraining-type analysis are common. This is the situation that is modelled in MP analyses with an 'uninterpretable' feature – two elements between which there is some dependency have the same feature specifications, but only one such specification is the 'real one'. An MP analysis of the English auxiliary system by Aelbrecht & Harwood (2015) involves the same idea as in (40), proposing that uninterpretable but valued features match between the governed verb and the higher auxiliary which governs it.

The use of a constraining equation can be further illustrated in the case of a strict Negative Polarity Item (NPI) – an item must appear in the context of negation, but is not the expression of negation itself. Such an NPI constrains its syntactic environment to have the NEG feature with the value + (see e.g. Sells 2000); this information must be present, but supplied by some other element, namely overt negation. From the MP perspective, Zeijlstra (2015) discusses an analysis of NPIs in which they "carry some uninterpretable negative feature [uNEG] that must be checked against a higher, semantically negative element that carries an interpretable formal negative feature [iNEG]." Again, in the relevant sense, this is a valued feature which is contentful on one element, and is on another for the purpose only of establishing an abstract syntactic relation.

Returning to the case of an attribute introduced without a value, this is an existential constraint on f-structure (see Kaplan & Bresnan 1982: 210ff. and Dalrymple 2001: 112–114). For instance, the complementizer *that* in English introduces a clause which is tensed, but it places no restriction on the value of TENSE. Hence part of the functional information associated with *that* will be the existential constraint (↑ TENSE). This constrains the f-structure of the clause to have the attribute TENSE, and any well-formed f-structure must have a value for that attribute. The value is not supplied by *that*, so that information must come from elsewhere in the clause introduced by *that*.

In summary, the notions of 'unvalued' and 'uninterpretable' features which are important in MP analyses – see immediately below – have formalized equivalents in LFG, and in LFG, neither can by itself lead to a well-formed f-structure: an f-structure cannot contain an attribute without a value, and the contribution of a constraining equation must be matched by the contribution of a (valued) defining equation. In keeping with the character of the differences between the LFG and MP approaches, a clausal f-structure in LFG is never partial nor ill-formed, unlike stages in an MP derivation. Rather, each element in the c-structure in a given example contributes to a set of constraints which the overall f-structure must satisfy. If those constraints conflict, there is no f-structure which satisfies them, and the example is thereby ungrammatical.

4.2 Features in the MP

Features are put to at least three uses in MP analyses (see Adger 2010: 200–212). The first is to represent information, the second is to establish a relationship, and the third is to make something happen. The representational aspect usually involves valued features, and might involve unvalued ones. The second use involves the notion of interpretable and uninterpretable features, which is the mechanism for establishing a relationship known as 'Agree' between a Probe and a Goal (Chomsky 2000: 101).

For instance, Adger (2010: 189) gives the following illustrative example of a feature that is unvalued, and also uninterpretable. The idea that some features are uninterpretable was originally introduced by Chomsky (1995: 277–278). In (41), the first group of features are features of the subject DP, and the second group are features on the T head of TP.

(41) {D, definite, plural} … {T, past, *u*plural}

The DP is definite and plural, and T is past and also marked as plural, but the prefix notation *u* indicates that the plural feature, though present on T, is uninterpretable on it. Adger describes (41) as follows:

> "The idea is that a feature like [plural] only has an interpretation when specified on a category which can be potentially interpreted as plural (e.g. on a noun), otherwise an instance of this feature will be uninterpretable: interpretability is detectable from a feature's syntactic/semantic context. The formal property of features (the *u* prefix) which enables them to enter into dependency relations is thus linked to the interpretation of features …".

The [plural] feature is not interpretable on T – the interpretation of tense never makes reference to singular/plural – but the matching occurence of [plural] on the subject DP establishes the Agree relation between these two groups of features. After it has been checked by a matching interpretable feature, [plural] is then eliminated on T.

From the perspective of LFG, the equivalent of [*u*plural] in (41) would be as in (42).

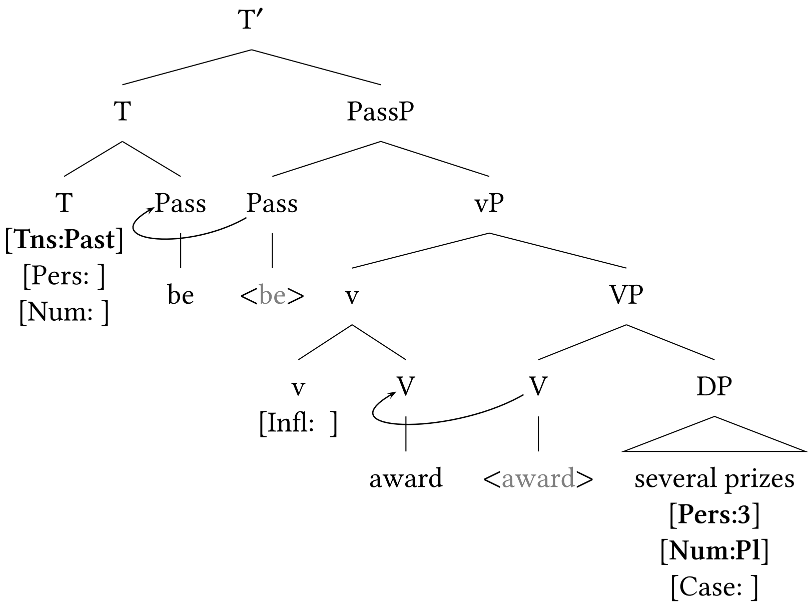
(42) (↑ SUBJ NUM) $=_c$ PL

Just as with (41), a structure described by (42) will only be well-formed if some other element (e.g. the subject) specifies the PL value for the feature, but it represents a different approach to the role of features. In (41), the feature on T is understood to convey "I am plural", which is uninterpretable; but the specification in (42) conveys "my subject's number is plural", which is actually straightforwardly interpretable.

The third use of features in the MP is to trigger an operation. Such features do not seem to overlap with the features considered above, and exist solely to make something happen. The canonical example is the 'EPP-feature' derived from GB, but used in different ways to force either XP movement or X^0 movement (head movement) in many MP analyses. This feature has been more recently cast as an 'Edge Feature' (e.g. Chomsky 2005). It is not clear formally what kind of feature this is – it must be satisfied, as an instruction for some structure to be built, and once satisfied, there are two options: either it becomes inactive, or it stays active, allowing for multiple specifiers (Chomsky 2007: 11).

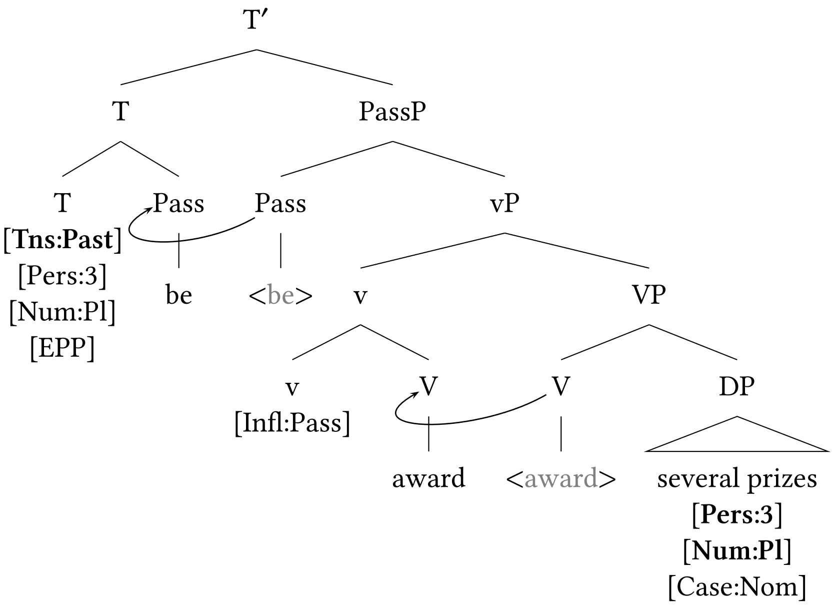
I now show in more detail how (un)valued and (un)interpretable features participate in an MP derivation. What is shown here is based on the presentation in Radford (2009: 284ff.), though using a slightly different representational format which is more internally consistent and which will also be more transparent in the context of the LFG approach to features described above. There is in fact a close relation between valued and interpretable features, as will be evident in the structures below. However, the two notions are formally distinct and can play different roles in an overall syntactic analysis (see e.g. Aelbrecht & Harwood 2015). The structure in (43) underlies the fragment *were awarded several prizes*, which is our illustrative example. The DP *several prizes* has interpretable features of person and number (it is 3rd person plural), and in the syntax it will have a value for case; but the case feature is initially unvalued, as the particular value of case will depend on the syntactic context of the DP. The v which ultimately hosts *award* has an Infl feature (sometimes referred to as VFORM in LFG), which will also be valued according to the syntactic context of the verb. Finally, the head T is specified as past tense, and it also hosts agreement features for person and number, which are unvalued at this initial stage. In the structure, the features shown in bold are those which are interpretable, and they also are the ones which are valued.

(43)



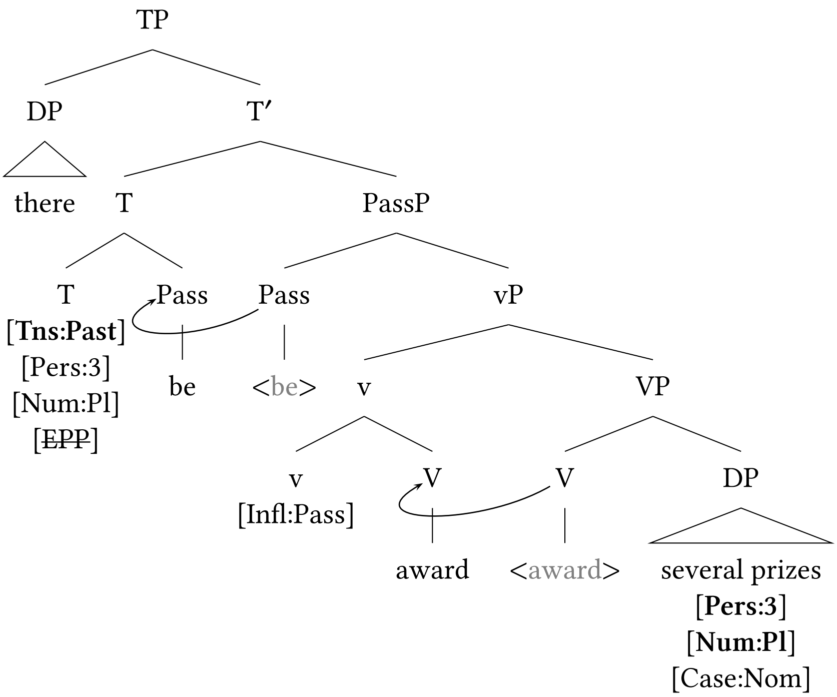
Formally, Agree establishes a relation between two nodes, a Probe and a Goal, one of which has an interpretable feature, with the other having a (matching) uninterpretable feature. In the simple example here, the uninterpretable features are the unvalued ones, and so they will become valued once Agree takes place. For instance, the Pass head *be* in (43) enters into an Agree relation with v and values the Infl feature as Pass(ive). The T head enters into an Agreement relation with the DP *several prizes*, valuing the Case on DP as Nom, and at the same time taking the values of person and number from that DP. Hence, after these Agree relations are established, all features are valued, as in (44), and these feature specifications will be relevant for morphological realization (e.g. the surface form of *be* will be *were*). Nevertheless, the non-bold (formerly unvalued) features are classed as uninterpretable, and must delete by the end of the syntactic derivation, Logical Form. Finally, (44) also shows one more feature on T, [EPP], which is discussed immediately below.

(44)



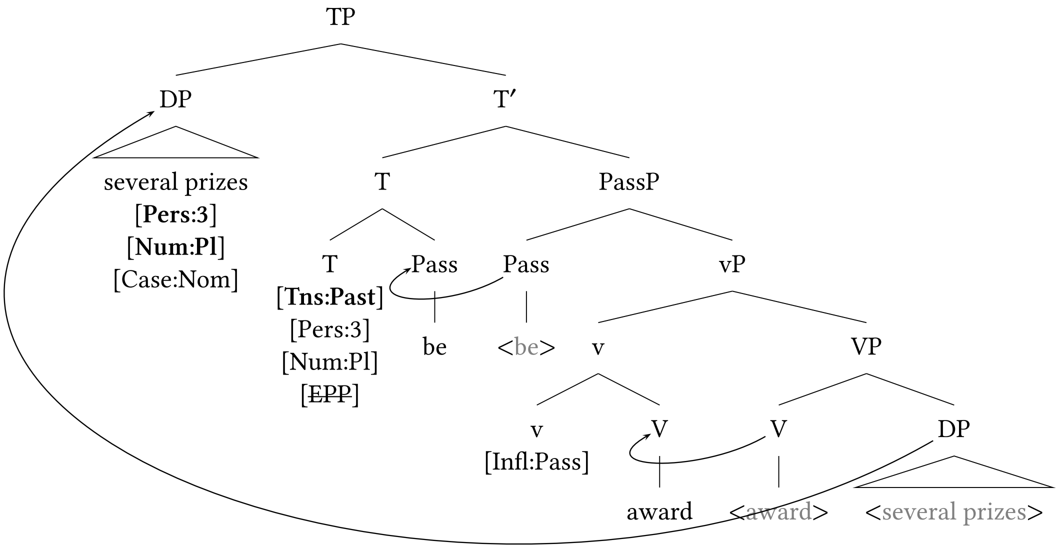
T in (44) has the [EPP] feature mentioned above, which has the effect that at the next step of the derivation a specifier must be created. This feature, then, does not represent a 'featural' property of the clause (unlike, say, 'past tense'), but represents a structural property. One option for satisfying this feature is to merge in an expletive placeholder, *there*:

(45)



Once the specifier is merged, the EPP feature is thereby satisfied, indicated in (45) by the strikethrough. Another way of satisfying this feature from the stage in (44) is to raise the object DP to the subject position, as a canonical passive:

(46)



4.3 Agreement and the direction of Agree

The second use of features in the MP noted above is that they participate in the process of Agree (Chomsky 2000), which is a prerequisite to establishing a relation in the syntax. The exemplar syntactic relation is that of agreement – say between a subject and a finite verb. As with canonical agreement (e.g. Corbett 2006), there is assumed to be a controller of agreement and a target for agreement, a directional or asymmetric relationship, formally instantiated as a Probe and a Goal in the MP. There is considerable debate in the MP literature as to the 'directionality' of Agree – is it upwards, or downwards? – as well as to whether feature valuation passes from the higher element to the lower one, or vice versa. For instance, Polinsky & Preminger (2019) make a linguistic argument about the direction of agreement (specifically, ϕ-feature agreement); they argue that agreement must be directional, looking downwards, but valuing features upwards. In contrast, Bjorkman & Zeijlstra (2019) argue for a more complex system in which a checking relation is first established, but only upwards, and then after that valuation can take place, in either direction. Some examples which bear on these issues are given below. These proposals are each 'substantive' proposals, motivated by empirical observations, as there is nothing formally about the MP system which requires a given directionality for Agree.

As noted above in Section 2.2.2, there is no sense in LFG in which agreement can be directional, as "agreement" is the informal notion we apply to a situation where more than one element provides featural information about some (other) element. The Archi examples below show that the distinction between Controller and Target, or between Probe and Goal, cannot be sustained anyway.

Polinsky & Preminger (2019) present examples such as (47) from Tsez, to support their claim that a Probe looks downwards to find a Goal – that the Probe c-commands the Goal – and the relevant feature values from the Goal are then valued upwards to the (previously incomplete) Probe. The key property of this example is that it involves long-distance agreement, in which the matrix verb does not agree with any local argument but rather agrees with the absolutive argument (object) of the embedded clause. Tsez has an ergative-absolutive case-marking system, and the verb agrees with an absolutive argument. The embedded absolutive in (47) is 'bread', class III, and both the local predicate 'eat' and the higher predicate 'know' agree with it in class, shown in boldface in the gloss:

(47) Tsez

eni-r	[už-ā	magalu	b-āc’-ru-ɬi]	
mother-DAT	[boy-ERG	bread.III(ABS)	III-eat-PST.PTCP-NMLZ]	

b-iy-xo
iii-know-pres
‘The mother knows that as for the bread, the boy ate it.’

The particular argument that Polinsky and Preminger make is based on the observation that the opposite configuration appears to be unattested – we never find a structure in which a verb in a lower clause agrees with an argument in a higher clause. To rule out this logical possibility, they argue that syntactic theory should only allow downwards Agree/upwards valuation. The detail of their argument is not crucial here – what is relevant are the relative structural relations between the two elements in the agreement relationship. In (47), as the absolutive controls agreement on the higher predicate ‘know’, I will categorize this example as one in which the target must c-command the controller (hence, valuation is upwards).

The rest of the examples in this section are taken from Archi, another language with an ergative-absolutive system. Archi has a wide range of potential targets for agreement, but the controller is always the absolutive. (48) from Bond & Chumakina (2016: 67) shows various agreement targets (boldfaced in the gloss, each corresponding to the exponent *d*-):

(48) Archi

[d-ez	χir]	d-e⟨r⟩qˤa-r-ši	d-i
[**ii.sg**-1sg.dat	behind]	**ii.sg**-⟨ipfv⟩go-ipfv-cvb	**ii.sg**-be.pres

‘She follows me.’ (male speaking)

Both the main verb and the auxiliary ‘be’ show agreement with the absolutive for gender and number. The gender system in Archi consists of four noun classes, and in this example, the gender agreement is for class ii. The controller of agreement is not overt – it is the implicit subject of the intransitive predicate, formally in absolutive case. In addition, the first singular pronoun *d-ez* which is the object within the directional PP headed by ‘behind’ agrees with the absolutive of its clause, even though the pronoun is not a direct co-argument of the absolutive in this example. The pronoun is itself first person singular, but it also has an ‘external agreement’ slot for the clausal absolutive. Now that pronoun, inside the PP, cannot c-command anything outside that PP, and yet the intuition here is that it is the target of agreement: so for this example it must be the case that the controller (a null subject absolutive) c-commands the target. Valuation, if directional, should be downwards – exactly reversed from the Tsez example (47).

The LFG analysis of Archi agreement in Sadler (2016) codes each agreeing element for the relevant features of the notional agreement controller – the argument in absolutive case. As the GF of that argument could be subj or obj depending on the transitivity of the predicate, Sadler uses the designator piv, proposed by Falk (2006). Sadler (2016: 161) also uses the template approach (Dalrymple et al. 2004) to schematize over different agreement combinations. For the form in (49), @ii.sg associates the gender and number agreement values with the word, as in the second commentary enclosed in [ ]:

(49)

d-ez, Pronoun
(↑ pers) = 1
(↑ num) = sg   [it is a first singular pronoun]
@ii.sg((gf ↑) piv)   [its external agreement features are class ii singular]
…

For (48), (gf ↑) piv instantiates as (obl obj ↑) subj. The f-structure of the example is shown in (50), where the external agreement path for the first person pronoun – the inner [pred ‘pro’] – follows this instantiation and specifies values for gender and number, shown in boldface:

(50) F-structure of (48); agreement of the pronoun with the absolutive ((obl obj ↑) subj) must be: class ii, num sg

$$\begin{bmatrix} \textbf{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘PRO’} \\ \text{PERS} & 3 \\ \textbf{NUM} & \textbf{SG} \\ \textbf{CLASS} & \textbf{II} \\ \text{CASE} & \text{ABS} \end{bmatrix} \\ \text{TENSE} & \text{PRES} \\ \text{PRED} & \text{‘GO}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBL})\rangle\text{’} \\ \textbf{OBL} & \begin{bmatrix} \text{PRED} & \text{‘BEHIND}\langle(\uparrow \text{OBJ})\rangle\text{’} \\ \textbf{OBJ} & \begin{bmatrix} \text{PRED} & \text{‘PRO’} \\ \text{PERS} & 1 \\ \text{NUM} & \text{SG} \\ \text{CASE} & \text{DAT} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Note that the first person pronoun itself does not have any “agreement slot” within its own feature structure: it has no agreement feature specification which is supposed to match or be copied somewhere else in the (f-)structure.

The informal notions of controller and target have no embodied representation, which ultimately proves to be an important fact about the LFG analysis –

because there are examples in which ‘controller’ and ‘target’ are the same single syntactic element. There are different types of example in Archi where an absolutive argument “agrees with itself” – a given syntactic element has an external agreement slot, to agree with the absolutive of its clause, but that element happens to be the absolutive itself. (See also Corbett 2006: 68–69, Borsley 2016: 137.) In these examples, the distinction between controller and target – as two distinct elements in an asymmetric relationship – is invalid, but on a co-description account of the kind illustrated by (49) the examples work out straightforwardly.

(51) is one such example. A reflexive pronoun in Archi has two slots for agreement – one for the features of its antecedent, as is familiar, and another one for the features of the absolutive of the clause. In (51) (from Bond & Chumakina 2016: 70) the subject is the pronoun ‘I’, in dative case, and the object is the reflexive, in absolutive case, and it is class II, signifying a female referent. The subject pronoun, main verb and auxiliary verb each agree in class with the absolutive, as does one of the slots in the reflexive – the whole form is 1SG, agreeing with the subject antecedent, and there is also an infixed class II agreement, again agreeing with the absolutive, which is the reflexive itself.

(51) Archi

d-ez	zona⟨r⟩u	d-ak:u-r-ši	d-i
II.SG-1SG.DAT	1SG.REFL.ABS⟨**II.SG**⟩	**II.SG**-see-IPFV-CVB	**II.SG**-be.PRS
daχon-n-a-š			
mirror(IV)-SG.OBL-IN-EL			

‘I am seeing myself in the mirror.’

So here, (GF ↑) PIV instantiates as (OBJ ↑) OBJ, one of the logical possibilities. The f-structure is shown in (52):

(52) F-structure of (51); agreement of the reflexive with the absolutive ((obj ↑) obj) must be: class ii, num sg

$$\left[\begin{array}{ll}
\mathrm{SUBJ} & \left[\begin{array}{ll} \mathrm{PRED} & \text{‘}\mathrm{PRO}\text{’} \\ \mathrm{PERS} & 1 \\ \mathrm{NUM} & \mathrm{SG} \\ \mathrm{CLASS} & \mathrm{II} \\ \mathrm{CASE} & \mathrm{DAT} \end{array}\right] \\
\mathrm{TENSE} & \mathrm{PRES} \\
\mathrm{PRED} & \text{‘}\mathrm{SEE}\langle(\uparrow \mathrm{SUBJ})(\uparrow \mathrm{OBJ})\rangle\text{’} \\
\mathbf{OBJ} & \left[\begin{array}{ll} \mathrm{PRED} & \text{‘}\mathrm{PRO}\text{’} \\ \mathrm{REFL} & + \\ \mathrm{PERS} & 1 \\ \mathbf{NUM} & \mathbf{SG} \\ \mathbf{CLASS} & \mathbf{II} \\ \mathrm{CASE} & \mathrm{ABS} \end{array}\right] \\
\mathrm{OBL} & \left[\ \text{“}\mathrm{MIRROR}\text{”}\ \right]
\end{array}\right]$$

Like other aspects of the grammar, the correct account of “agreement” does not involve moving something – in this case, featural information – from one place to another, but rather is a partial specification of featural information in a larger structure.

5 Conclusion

LFG takes up the challenges of accounting for human language precisely as Chomsky first articulated them, yet continuing with a view quite different from his as to what the core non-negotiable properties of the syntactic system should be. It was developed as systematic and coherent framework for representation of grammatical information, based on certain key design features. While these design features give these frameworks a very different character from a procedural framework such as the MP, many of the components of analysis which MP has developed are already present in declarative frameworks (see Section 4), as is the convergence of interest in exploring ‘third factor’ considerations (see (5)).

As Chomsky has noted, the choice between grammatical frameworks can be understood in terms of the “extra burdens” that an over-exuberant approach will entail (Chomsky 2007: 10–11). From a declarative perspective, any procedural approach creates such burdens, as the necessary mechanisms are either too powerful or are not well-founded. If those mechanisms can change or even delete syntactic information or syntactic substance, it is necessary to constrain those destructive operations with the 'No Tampering Condition' – indeed, a very natural property of a grammatical system, but one that should be intrinsic to it.

The 'copy theory' of movement (Chomsky 1995) is a way of expressing the intuition that some information is shared. However, copies involve duplication of substance, which amounts to more than the sharing of information. The discussion of structure in Section 3 provides a perspective on two kinds of further burden that necessarily arise in a copy-based approach. First, with regard to head mobility, the LFG view is that the issue is one of alternative positions, rather than successive positions which exist to provide hosts for position-occupying movement. The head is indeed only "in" one head position, but it makes the same contribution that it would have made from any of its alternative possible positions, and so might appear as if it were contributing from each position. The evidence from multiple expression of clausal information supports this view. A recent MP account of the syntax of heads by Arregi & Pietraszko (2021) associates only the informational part of a given head with several head positions, effectively recapitulating the LFG analysis of head mobility through various operations to create the right representations. Second, the facts of Mandarin verb copying show that certain parts of the syntactic analysis indeed call for a duplication of substance (when there are both in-situ arguments and adjuncts in Mandarin), while other parts involve more abstract syntactic information (the notion of a verb *having* arguments and having adjuncts). If that abstract information is conflated with the phrase structure substance, the system generates too much, and then extra operations have to be invoked, pruning or conflating substance.

The formalization of the MP by Collins & Stabler (2016) is designed in part to address the No Tampering Condition, and the apparent duplication of substance. Instead of creating copies, in this formalization, successive movements of a given element create new multidominance relations from that single element, which therefore does not change during the derivation. Their formalization is extended to head movement by Bleaman (2021). This particular formalization might make MP derivations slightly closer in nature to f-structures, in that each object in the derivation is a single informational unit which may have multiple grammatical relations and phrase-structural relations (e.g. a topicalized object is both an object and a topic, but with just one overt realization, in topic position).

A different kind of burden of potential complexity falls on the feature system of the MP, as features are used to do more than represent information (see Section 4). It becomes necessary to posit "bad" feature specifications, such as uninterpretable features, which by design are not interpretable on their hosts, and which must be eliminated during the derivation. LFG has constraints of a different character for checking that certain grammatical relationships exist, and which do not involve recourse to local pockets of uninterpretability.

Acknowledgments

The content of this chapter has benefitted considerably from the comments of three anonymous reviewers, whose contributions I gratefully acknowledge; and thanks to Mary Dalrymple for her continued guidance.

References

Adger, David. 2003. *Core syntax: A Minimalist approach*. Oxford: Oxford University Press.

Adger, David. 2010. A Minimalist theory of feature structure. In Anna Kibort & Greville G. Corbett (eds.), *Features: Perspectives on a key notion in linguistics*, 185–218. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780199577743.003.0008.

Aelbrecht, Lobke & Will Harwood. 2015. To be or not to be elided: VP ellipsis revisited. *Lingua* 153. 66–97. DOI: 10.1016/j.lingua.2014.10.006.

Andrews, Avery D. & Christopher D. Manning. 1999. *Complex predicates and information spreading in LFG*. Stanford: CSLI Publications.

Arregi, Karlos & Asia Pietraszko. 2021. The ups and downs of head displacement. *Linguistic Inquiry* 52(2). 241–290. DOI: 10.1162/ling_a_00377.

Asudeh, Ash & Ida Toivonen. 2006. Symptomatic imperfections. *Journal of Linguistics* 42. 395–422.

Baker, Mark C. 1985. The Mirror Principle and morphosyntactic explanation. *Linguistic Inquiry* 16. 373–415.

Baker, Mark C. 2015. *Case: Its principles and its parameters* (Cambridge Studies in Linguistics). Cambridge, UK: Cambridge University Press. DOI: 10.1017/CBO9781107295186.

Bartos, Huba. 2019. The V-copy construction in Mandarin: A case temporarily reopened. In Jianhua Hu & Haihua Pan (eds.), *Interfaces in grammar*, 167–205. Amsterdam: John Benjamins.

Bjorkman, Bronwyn M. & Hedde Zeijlstra. 2019. Checking up on (Φ-)Agree. *Linguistic Inquiry* 50(3). 527–569. DOI: 10.1162/ling_a_00319.

Bleaman, Isaac. 2021. Predicate fronting in Yiddish and conditions on multiple copy Spell-Out. *Natural Language & Linguistic Theory*. DOI: 10.1007/s11049-021-09512-3.

Bobaljik, Jonathan David. 2017. Distributed morphology. In Joost Dekkers, Frank van der Leeuw & Jeroen van de Weijer (eds.), *Oxford research encyclopedia of linguistics*. Oxford: Oxford University Press. DOI: 10.1093/acrefore/9780199384655.013.131.

Boeckx, Cedric, Norbert Hornstein & Jairo Nunes. 2007. Overt copies in reflexive and control structures: A movement analysis. *University of Maryland Working Papers in Linguistics* 15. 1–46.

Bond, Oliver & Marina Chumakina. 2016. Agreement domains and targets. In Oliver Bond, Greville G. Corbett, Marina Chumakina & Dunstan Brown (eds.), *Archi: Complexities of agreement in cross-theoretical perspective*, 43–76. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198747291.003.0003.

Borsley, Robert D. 2016. HPSG and the nature of agreement in Archi. In Oliver Bond, Greville G. Corbett, Marina Chumakina & Dunstan Brown (eds.), *Archi: Complexities of agreement in cross-theoretical perspective*, 118–149. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198747291.003.0005.

Bresnan, Joan. 2000. Optimal syntax. In Joost Dekkers, Frank van der Leeuw & Jeroen van de Weijer (eds.), *Optimality Theory: Phonology, syntax, and acquisition*, 334–385. Oxford: Oxford University Press.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2016. *Lexical-Functional Syntax*. 2nd edn. (Blackwell Textbooks in Linguistics 16). Malden, MA: Wiley-Blackwell.

Charnavel, Isabelle & Dominique Sportiche. 2016. Anaphor binding: What French inanimate anaphors show. *Syntax* 47. 35–87. DOI: 10.1162/ling_a_00204.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: The MIT Press.

Chomsky, Noam. 1970. Remarks on nominalization. In Roderick A. Jacobs & Peter S. Rosenbaum (eds.), *Readings in English transformational grammar*, 184–221. Waltham, MA: Ginn.

Chomsky, Noam. 1973. Conditions on transformations. In Stephen R. Anderson & Paul Kiparsky (eds.), *A festschrift for Morris Halle*, 232–286. New York: Holt, Reinhart & Winston.

Chomsky, Noam. 1980. On binding. *Linguistic Inquiry* 11. 1–46.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris Publications. DOI: 10.1515/9783110884166.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/9780262527347.001.0001.

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In Roger Martin, David Michaels & Juan Uriagereka (eds.), *Step by step: Essays on Minimalist syntax in honor of Howard Lasnik*, 89–155. Cambridge, MA: The MIT Press.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz (ed.), *Ken Hale: A life in language*, 1–52. Cambridge, MA: The MIT Press.

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36(1). 1–22. DOI: 10.1162/0024389052993655.

Chomsky, Noam. 2007. Approaching UG from below. In Uli Sauerland & Hans-Martin Gärtner (eds.), *Interfaces + recursion = language? Chomsky's minimalism and the view from syntax-semantics*, 1–29. Berlin: Walter de Gruyter.

Collins, Chris & Edward Stabler. 2016. A formalization of minimalist syntax. *Syntax* 19(1). 43–78. DOI: 10.1111/synt.12117.

Corbett, Greville G. 2006. *Agreement*. Cambridge, UK: Cambridge University Press.

Crysmann, Berthold & Olivier Bonami. 2016. Variable morphotactics in information-based morphology. *Journal of Linguistics* 52. 311–374.

D'Alessandro, Roberta. 2019. The achievements of Generative Syntax: A time chart and some reflections. *Catalan Journal of Linguistics Special Issue: Generative Syntax: Questions, Crossroads, and Challenges*. 7–26.

Dalrymple, Mary. 2001. *Lexical Functional Grammar* (Syntax and Semantics 34). New York: Academic Press. DOI: 10.1163/9781849500104.

Dalrymple, Mary, Ronald M. Kaplan & Tracy Holloway King. 2004. Linguistic generalizations over descriptions. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '04 conference*, 199–208. Stanford: CSLI Publications.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.

Embick, David & Rolf Noyer. 2007. Distributed morphology and the syntax-morphology interface. In Gillian Ramchand & Charles Reiss (eds.), *The Oxford handbook of linguistic interfaces*, 289–324. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780199247455.013.0010.

Falk, Yehuda N. 2006. *Subjects and universal grammar: An explanatory theory*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511486265.

Fang, Ji & Peter Sells. 2007. A formal analysis of the verb copy construction in Chinese. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '07 conference*, 198–213. Stanford: CSLI Publications.

Findlay, Jamie Y. & Anna Kibort. Forthcoming. Argument structure and mapping theory. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Fukuda, Shin. 2008. Backward control. *Language and Linguistics Compass* 2/1. 168–195. DOI: 10.1111/j.1749-818x.2007.00048.x.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum & Ivan A. Sag. 1985. *Generalized phrase structure grammar*. Cambridge, MA: Harvard University Press.

Grimshaw, Jane. 2000. Locality and extended projection. In Peter Coopmans, Martin Everaert & Jane Grimshaw (eds.), *Lexical specification and insertion*, 115–133. Amsterdam/Philadelphia: John Benjamins. DOI: 10.1075/cilt.197.07gri.

Haegeman, Liliane. 1994. *Introduction to government and binding theory*. Oxford: Blackwell.

Halle, Morris & Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Kenneth Hale & Samuel Jay Keyser (eds.), *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*, 111–176. Cambridge, MA: The MIT Press.

Harizanov, Boris & Vera Gribanova. 2019. Whither head movement? *Natural Language & Linguistic Theory* 37. 461–522.

Harley, Heidi & Rolf Noyer. 2003. Distributed morphology. In Rint Sybesma & Lisa Cheng (eds.), *The second 'Glot International' state-of-the-article book: the latest in linguistics*, 463–496. Berlin & New York: Mouton de Gruyter.

Hornstein, Norbert. 2001. *Move! A minimalist theory of construal*. Malden, MA: Blackwell.

Hornstein, Norbert. 2018. The Minimalist Program after 25 years. *Annual Review of Linguistics* 4. 49–65. DOI: 10.1146/annurev-linguistics-011817-045452.

Hornstein, Norbert, Jairo Nunes & Kleathes K. Grohmann. 2006. *Understanding minimalism*. Cambridge, UK: Cambridge University Press.

Hornstein, Norbert & Maria Polinsky (eds.). 2010. *Movement theory of control*. Amsterdam/Philadelphia: John Benjamins. DOI: 10.1075/la.154.

Huang, C.-T. James. 1982. *Logical relations in Chinese and the theory of grammar*. Cambridge, MA: Massachusetts Institute of Technology. (Doctoral dissertation).

Hudson, Richard. 1977. The power of morphological rules. *Lingua* 42. 73–89. DOI: 10.1016/0024-3841(77)90020-1.

Jackendoff, Ray. 1972. *Semantic interpretation in generative grammar*. Cambridge, MA: The MIT Press.

Kameyama, Megumi. 1985. *Zero anaphora: The case of Japanese*. Stanford: Stanford University. (Doctoral dissertation).

Kaplan, Ronald M. 2019a. Computational psycholinguistics. *Computational Linguistics* 45(4). 607–626. DOI: 10.1162/coli_a_00359.

Kaplan, Ronald M. 2019b. Formal aspects of underspecified features. In Cleo Condoravdi & Tracy Holloway King (eds.), *Tokens of meaning: Papers in honor of Lauri Karttunen*, 349–369. Stanford: CSLI Publications.

Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 173–281. Cambridge, MA: The MIT Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 29–130).

Kaplan, Ronald M. & Annie Zaenen. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In Mark Baltin & Anthony Kroch (eds.), *Alternative conceptions of phrase structure*, 17–42. Chicago: University of Chicago Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 137–165).

Keenan, Edward L. 1976. Towards a universal definition of subject. In Charles N. Li (ed.), *Subject and topic*, 303–333. New York: Academic Press.

Lebeaux, David. 1983. A distributional difference between reciprocals and reflexives. *Linguistic Inquiry* 14(4). 723–730.

Li, Charles N. & Sandra A. Thompson. 1981. *Mandarin Chinese: A functional reference grammar*. Berkeley/Los Angeles: University of California Press.

Lidz, Jeffrey & William J. Idsardi. 1998. Chains and phonological form. *University of Pennsylvania Working Papers in Linguistics* 5. 109–125.

Lødrup, Helge. 2014. How can a verb agree with a verb? Reanalysis and pseudocoordination in Norwegian. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 367–386. Stanford: CSLI Publications.

Manning, Christopher D. 1996. *Ergativity: Argument structure and grammatical relations* (Dissertations in Linguistics). Stanford: CSLI Publications.

Mohanan, K. P. 1983. Functional and anaphoric control. *Linguistic Inquiry* 14. 641–674.

Müller, Stefan & Stephen Wechsler. 2014. Lexical approaches to argument structure. *Theoretical Linguistics* 40(1-2). 1–76. DOI: 10.1515/tl-2014-0001.

Niño, María-Eugenia. 1997. The multiple expression of inflectional information and grammatical architecture. In Francis Corblin, Daniéle Godard & Jean-Marie Marandin (eds.), *Empirical issues in syntax and semantics 1*, 127–147. Bern: Peter Lang.

O'Grady, William. 2012. Three factors in the design and acquisition of language. *Wiley Interdisciplinary Reviews: Cognitive Science* 3. 493–499. DOI: 10.1002/wcs.1188.

Polinsky, Maria. 2000. Tsez beginnings. In *Proceedings of the 25th annual meeting of the Berkeley Linguistics Society: Special session on Caucasian, Dravidian, and Turkic linguistics*, 14–29. Berkeley: Berkeley Linguistics Society.

Polinsky, Maria & Eric Potsdam. 2002. Backward control. *Linguistic Inquiry* 33. 245–282. DOI: 10.1162/002438902317406713.

Polinsky, Maria & Eric Potsdam. 2006. Expanding the scope of control and raising. *Syntax* 9. 171–192. DOI: 10.1111/j.1467-9612.2006.00090.x.

Polinsky, Maria & Omer Preminger. 2019. The *Agreement Theta Generalization*. *Glossa: A Journal of General Linguistics* 4(1). 102.

Pollard, Carl & Ivan A. Sag. 1987. *Information-based syntax and semantics*. Stanford: CSLI Publications.

Pollard, Carl & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press & CSLI Publications.

Radford, Andrew. 2009. *Analysing English sentences: A minimalist approach*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511801617.

Reuland, Eric. 2011. *Anaphora and language design*. Cambridge, MA: The MIT Press.

Rizzi, Luigi. 2016. The concept of explanatory adequacy. In Ian Roberts (ed.), *The Oxford handbook of universal grammar*, 97–113. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780199573776.013.5.

Roberts, Ian. 2011. Head movement and the Minimalist Program. In Cedric Boeckx (ed.), *The Oxford handbook of linguistic minimalism*, 195–219. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780199549368.013.0009.

Sadler, Louisa. 2016. Agreement in Archi: An LFG perspective. In Oliver Bond, Greville G. Corbett, Marina Chumakina & Dunstan Brown (eds.), *Archi: Complexities of agreement in cross-theoretical perspective*, 150–183. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198747291.003.0006.

Sadler, Louisa & Andrew Spencer. 2001. Syntax as an exponent of morphological features. In Geert Booij & Jaap van Marle (eds.), *Yearbook of morphology 2000*, 71–96. Dordrecht: Kluwer Academic Publishers. DOI: 10.1007/978-94-017-3724-1_4.

Safir, Ken. 2008. Coconstrual and narrow syntax. *Syntax* 11. 330–355. DOI: 10.1111/j.1467-9612.2008.00117.x.

Schachter, Paul. 1976. The subject in Philippine languages: Actor, topic, actor-topic, or none of the above? In Charles N. Li (ed.), *Subject and topic*, 491–518. New York: Academic Press.

Sells, Peter. 2000. Negation in Swedish: Where it’s not at. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG ’00 conference*, 1–19. Stanford: CSLI Publications.

Sells, Peter. 2001. *Structure, alignment and optimality in Swedish*. Stanford: CSLI Publications.

Sells, Peter. 2004. Syntactic information and its morphological expression. In Louisa Sadler & Andrew Spencer (eds.), *Projecting morphology*, 187–225. Stanford: CSLI Publications.

Sells, Peter. 2006. Using subsumption rather than equality in functional control. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '06 conference*, 455–473. Stanford: CSLI Publications.

Sells, Peter. Forthcoming. LFG and Minimalism. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Simon, Herbert A. 1962. The architecture of complexity. *Proceedings of the American Philosophical Society* 106(6). 467–482. DOI: 10.1007/978-3-642-27922-5_23.

Spencer, Andrew. 2013. *Lexical relatedness: A paradigm-based model*. Oxford: Oxford University Press.

Wechsler, Stephen & I Wayan Arka. 1998. Syntactic ergativity in Balinese: An argument structure based theory. *Natural Language & Linguistic Theory* 16. 387–442.

Zaenen, Annie & Ronald M. Kaplan. 1995. Formal devices for linguistic generalizations: West Germanic word order in LFG. In Jennifer S. Cole, Georgia M. Green & Jerry L. Morgan (eds.), *Linguistics and computation*, 3–27. Stanford: CSLI Publications. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 215–240).

Zeijlstra, Hedde. 2015. The morpho-syntactic realization of negation. In Artemis Alexiadou & Tibor Kiss (eds.), *Syntax — Theory and analysis. An international handbook*, 274–308. Berlin: De Gruyter Mouton.

Handbook of Lexical Functional Grammar

Set blurb on back with \BackBody{my blurb}