Chapter 34

HPSG and Lexical Functional Grammar

Stephen Wechsler
The University of Texas

Ash Asudeh
University of Rochester

Here is the epigram: more people have been to Berlin than I have

Here is the abstract: concrete gets you through abstract better than abstract gets you through concrete

1 Introduction

Head-Driven Phrase Structure Grammar is similar in many respects to its cousin framework, Lexical Functional Grammar or LFG (Bresnan et al. 2015; Dalrymple 2001). Both HPSG and LFG are lexicalist frameworks in the sense that they distinguish between the morphological system which creates words, and the syntax proper which combines those fully inflected words into phrases and sentences. Both frameworks assume a lexical theory of argument structure (Müller & Wechsler 2014) in which verbs and other predicators come equipped with valence structures indicating the kinds of complements and other dependents that the word is to be combined with. Both theories treat control (equi) and raising as a lexical property of certain control or raising predicates. Both representational systems are based on unification grammar (Kay 1984a), employing directed graphs that are often represented in the form of recursively embedded feature structures. Phonologically empty nodes of the constituent structure are avoided in both theories, with the gaps appearing in long-distance dependencies as the sole

Stephen Wechsler & Ash Asudeh. 2020. HPSG and Lexical Functional Grammar. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 1239–1253. Berlin: Language Science Press. DOI:??

exception in some analyses, and complete elimination of empty categories even in those cases, in others.

At the same time, there are interesting differences. Each theory makes available certain representational resources that the other theory lacks. LFG has output filters in the form of *constraining equations*, HPSG does not. HPSG's feature structures are *typed*, those of LFG are not. The feature descriptions (directed graphs) are fully integrated with the phrase structure grammar in the case of HPSG, while in LFG they are intentionally separated in an autonomous level of representation in the form of a *functional structure* or f-structure. These differences lead some linguists to feel that certain types of generalization are more perspicuously stated in one framework than the other. Because LFG's functional structure is autonomous from the constituent structure whose terminal yield gives the order of words in a sentence, that functional structure can instead serve as a representation of the grammatical functions played by various components of a sentence. This makes LFG more amenable to a functionalist motivation, and also provides a standard representation language for capturing the more cross-linguistically invariant properties of syntax. Meanwhile, HPSG is more deeply rooted in phrase structure grammar, and thus provides a clearer representation of the locality conditions that are important for the proper functioning of grammars.

This chapter presents a comparison of the two theories with an emphasis on contrasts between the two. It is organized by grammatical topic.

2 Phrases and Endocentricity

A phrasal node shares certain grammatical features with specific daughters, such as the HEAD features that it shares with the head daughter. In HPSG this is accomplished by means of structure-sharing (reentrancies) in the immediate dominance schemata and other constraints on local sub-trees such as the Head Feature Principle. LFG employs essentially the same mechanism for feature sharing in a local sub-tree but implements it slightly differently. Each node in a phrase structure is paired with a so-called functional structure or *f-structure*, which is formally a set of attribute-value pairs. It is through the f-structure that the nodes of the phrase structure share features. The phrase structure is referred to as *c-structure*, for categorial or constituent structure, in order to distinguish it from f-structure. The grammar, in the form of a standard rewriting system, directly generates only c-structures, not f-structures. Those c-structure rules introduce equations that form a projection function from c-structure to f-structure. For example, the phrase structure grammar in 1 and lexicon in 2 generate the tree in 3.

(1) a. S → NP VP
$(\uparrow \textsc{subj}){=}\downarrow$ $\uparrow{=}\downarrow$

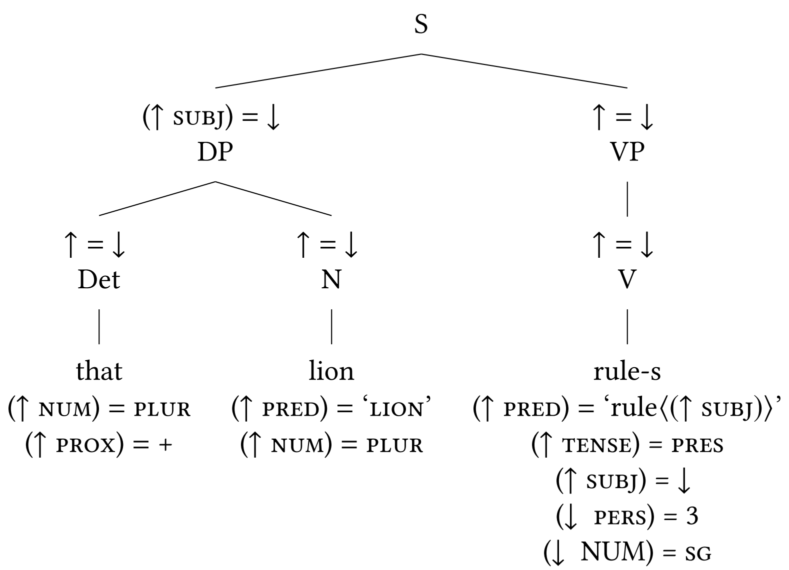
b. NP → $\begin{pmatrix}\text{Det}\\ \uparrow{=}\downarrow\end{pmatrix}$ N
$\uparrow{=}\downarrow$

c. VP → V $\begin{pmatrix}\text{NP}\\ (\uparrow \text{OBJ}){=}\downarrow\end{pmatrix}$
$\uparrow{=}\downarrow$

(2) a. *lion*: N (↑ ‘)’ = ‘lion’
(↑ NUM) = SG

b. *rule*: V (↑ ‘)’ = ‘rule⟨(↑ SUBJ)’
-*s*: *infl* (↑ TENSE) = PRES
(↑ SUBJ) = ↓
(↓ PERS) = 3
(↓ NUM) = SG

(3)



Each node in the c-structure maps to a function, that is, to a set of attribute-value pairs. Within the equations, the up and down arrows are metavariables over function names, interpreted as follows: the up arrow refers to the function to which the mother node maps, and the down arrow refers to the function that its own node maps to. To derive the f-structure from 3, we instantiate the metavariables to specific function names and solve for the function associated with the root node (here, S). In 4 the function names *f1, f2,* etc. are subscripted to the node labels. The arrows have been replaced with those function names.

(4)

- S_{f1}
  - (f1 SUBJ) = f2
 DP_{f2}
    - f2 = f4
 Det_{f4}
      - that_{f7}
 (f4 NUM) = PLUR
 (f4 PROX) = +
    - f2 = f5
 N_{f5}
      - lion_{f8}
 (f5 PRED) = ‘LION’
 (f5 NUM) = SG
  - f1 = f3
 VP_{f3}
    - f3 = f6
 V_{f6}
      - rules_{f9}
 (f6 PRED) = ‘rule⟨(*f*6SUBJ)⟩’
 (f6 TENSE) = PRES
 (f6 SUBJ) = f9
 (f9 PERS) = 3
 (f9 NUM) = SG

Collecting all the equations from this tree and solving for *f1*, we arrive at the f-structure in 5:

$$(5)\quad \left[\begin{array}{ll} \textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \textit{`lion'} \\ \textsc{num} & \textit{sg} \\ \textsc{pers} & \textit{3} \\ \textsc{prox} & + \end{array}\right] \\ \textsc{pred} & \textit{`rule}\langle(\uparrow \textsc{subj})\rangle\textit{'} \\ \textsc{tense} & \textit{pres} \end{array}\right]$$

Since the up and down arrows refer to nodes of the local subtree, LFG annotated phrase structure rules like those in 1 can often be directly translated into HPSG immediate dominance schemata and principles constraining local subtrees. By way of illustration, let FS (for *f-structure*) be an HPSG attribute corresponding to the f-structure projection function. Then the LFG rule in 6a (repeated from 1a above) is equivalent to the HPSG rule in 6b:

(6) a. $S \longrightarrow \underset{(\uparrow \text{SUBJ})=\downarrow}{\text{DP}} \quad \underset{\uparrow=\downarrow}{\text{VP}}$

b. $S[\text{FS}\ \boxed{1}] \longrightarrow \text{DP}[\text{FS}\ \boxed{2}] \quad \text{VP}[\text{FS}\ \boxed{1}[\text{SUBJ}\ \boxed{2}]]$

Let us compare the two representations with respect to heads and dependents.

Taking heads first, the VP node annotated with ↑ = ↓ is an *f-structure head*, meaning that the features of the VP are identified with those of the mother S. This effect is equivalent to the tag $\boxed{1}$ in 6b. Hence ↑ = ↓ has an effect similar to HPSG's Head Feature Principle. However, in LFG the part of speech categories and their projections such as N, V, Det, NP, VP, DP, etc. belong to the c-structure and not the f-structure. As a consequence those features are not subject to sharing, and any principled correlations between such categories, such as the fact that N is the head of NP, V the head of VP, C as head of CP, and so on, are instead captured in an explicit version of (extended) X-bar theory applying to the c-structure. The LFG based theory of endocentricity is considerably weaker (more permissive) than what is typically found in most transformation based grammars. The version of extended X-bar theory in Bresnan et al. (2015: ch.) assumes that all nodes on the right side of the arrow of the phrase structure rule are optional, with many unacceptable partial structures ruled out in the f-structure instead (see Section XXX below for well-formedness constraints on f-structures). Also not all structures need to be endocentric (i.e. not all structures have a head daughter in c-structure). The LFG category S shown in 6a is inherently exocentric, lacking a c-structure head, and is used for the analysis of copulaless clauses. (English is also assumed to have endocentric clauses of category IP, where an auxiliary verb of category I (for Inflection) serves as the c-structure head.) S is also used for flat structures in non-configurational clauses found in languages such as Warlpiri.

Functional projections like DP, IP, and CP are typically assumed to form a 'shell' over the lexical projections NP, VP, AP, and PP (plus CP can appear over S). In fact this idea of extending X-bar to functional categories has its origin in the LFG work of the late Yehuda Falk (REFS), from which it then spread to transformational theories. This is formally implemented by having the functional head (such as Det) and its lexical complement (such as NP) be f-structure co-heads. See for example the DP *that lion* in 3, where Det and N are both annotated with ↑ = ↓ . The DP, Det, and N nodes all map to the same f-structure, namely the subsidiary structure serving as the value of SUBJ (see 5). What makes this unification possible is that function words lack a PRED (for 'predicate') feature that would otherwise indicate a semantic form. Content words such as *lion* have such a feature ([PRED 'LION']), and so if the Det had one as well then they would clash in the f-structure. Note more generally that the f-structure flattens out much of the hierarchical structure of the corresponding c-structure.

Complementation works a little differently in LFG from HPSG. Note that the LFG rule 6a indicates the SUBJ grammatical function on the subject NP node, while the pseudo-HPSG rule 6b indicates the SUBJ function on the VP functor selecting the subject. A consequence of the use of functional equations in LFG is that a grammatical relation such as SUBJ can be locally associated with its formal exponents, whether a configurational position in phrase structure (as in 3), head-marking (agreement), or dependent marking (case). A subject-marking case affix can introduce a so-called 'inside out' functional designator, (SUBJ $\uparrow$), which requires that the f-structure of the DP bearing that case ending be the value of a SUBJ attribute REFS-NORDLINGER.[1] This aspect of LFG representations makes it convenient for functionalist and typological work on grammatical relations.

3 Valence

In LFG a lexical predicator such as a verb selects its complements via f-structure rather than c-structure. A transitive verb selects a SUBJ and OBJ, which are features of f-structure, but it cannot select for the category 'DP' because such part of speech categories belong only to c-structure. For example the verb stem rule in 2b has a PRED feature whose value contains ($\uparrow$ SUBJ), which has the effect of requiring a SUBJ function in the f-structure. The f-structure (shown in 5) is built using the defining equations, as described above. Then that f-structure is checked against any *existential constraints* such as the expression ($\uparrow$ SUBJ), which requires that the f-structure contain a SUBJ feature. That constraint is satisfied, as shown in 5. Moreover, the fact that ($\uparrow$ SUBJ) appears in the angled brackets means that it expresses a semantic role of the 'rule' relation, hence the SUBJ value is required to contain a PRED feature, which is satisfied by the feature [PRED 'LION'] in 5.

Selection for grammatical relations instead of formal categories enables LFG to capture the flexibility in the expression of a given grammatical relation described at the end of the previous section. As noted there, in many languages the subject can be expressed either as an independent DP phrase as in English, or as a pronominal affix on the verb. As long as the affix introduces a PRED feature and is designated by the grammar as filling the SUBJ relation, then it satisfies

[1]For function f, attribute a and value v, $(f\ a) = v$ iff $(a\ v) = f$.

the subcategorization requirements imposed by a verb. A more subtle example of flexible expression of grammatical functions can be seen in English constructions where an argument can in principle take the form of either a DP (as in 7a) or a clause (as in 7b) (example 7 is from Bresnan et al. (2015) PAGE).

(7) a. That problem, we talked about for days.
    b. That he was sick, we talked about for days.
    c. We talked about that problem for days.
    d. *We talked about that he was sick for days.

The variant of *talk* taking an *about*-PP selects neither a DP nor a clausal complement, but rather an object (OBJ) of an oblique (OBL_{about}) function:

(8) *talk*: V $(\uparrow \text{`)'} = \text{`talk}\langle(\uparrow \textsc{subj})(\uparrow \textsc{obl}_{about}\textsc{obj})\text{'}$

It is not the verb but the local c-structure environment that conditions the category of that argument: the canonical object position right-adjacent to *about* can only house a DP, while the topic position allows either DP or clause (as seen by comparing 7c and 7d). In LFG the grammatical functions such as SUBJ and OBJ represent equivalence classes across various modes of c-structure expression.

HPSG captures this variability in the expression of arguments in essentially the same way as LFG, despite some terminological differences. The HPSG correspondent of the LFG SUBJ specification is not an HPSG SUBJ valence list item, but rather the item of the ARG-ST list that the (optional) SUBJ item maps to, when it appears. The same applies to complements. The disjunction between DP and clausal expression of an argument is encoded in the ARG-ST; the restriction to DP (observed in examples 7c,d) is encoded on the relevant valence list.

4 Head mobility

The lexical head of a phrase can sometimes appear in an alternative position apparently outside of what would normally be its phrasal projection. Assuming that an English finite auxiliary verbs is the (category I) head of its (IP) clause, then that auxiliary appears outside its clause in a yes/no question:

(9) a. [$_{IP}$ she is mad].
    b. Is [$_{IP}$ she ___ mad]?

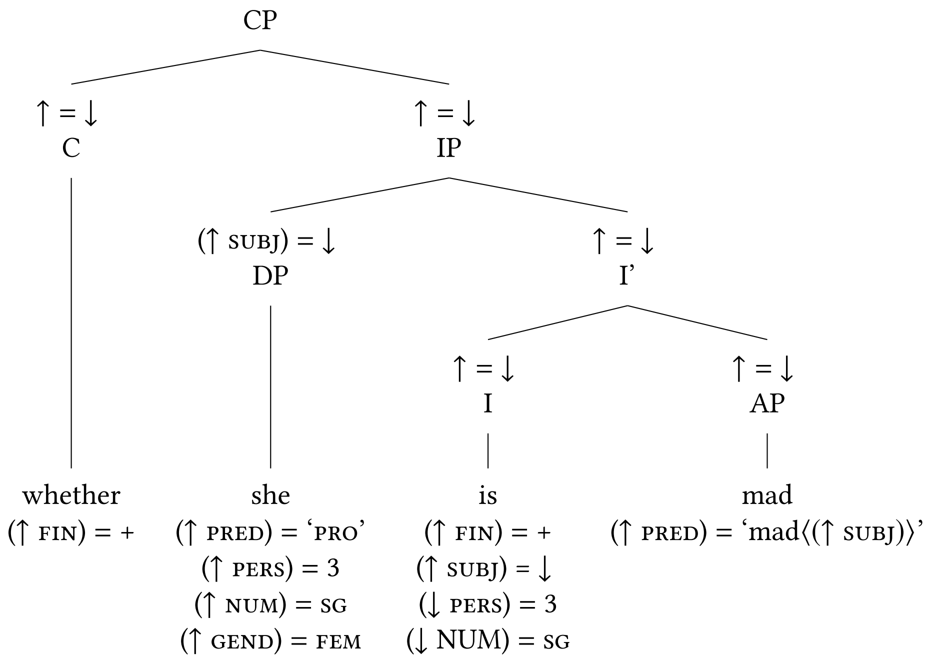
Transformational grammars capture the systematic relation between these two structures with a head-movement transformation that leaves the source IP structure intact, with a trace replacing the moved head. The landing site of the moved clausal head is often assumed to be C, the complementizer position, as motivated by complementarity between the fronted verb and a lexical complementizer observed most strikingly in German but also found in other languages, including some English constructions such as the following:

(10) a. I wonder whether [$_{IP}$ she is mad].
b. I wonder, is [$_{IP}$ she ___ mad]?
c. *I wonder whether is she mad.

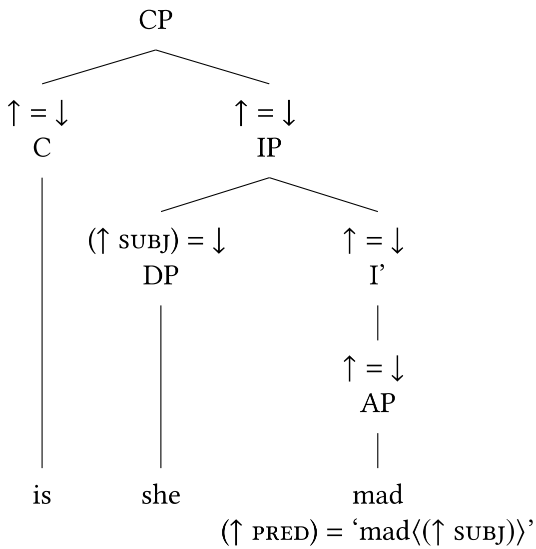
In HPSG the sentences in 9 are treated as displaying two distinct structures generated by the grammar. For example, assuming ternary branching in 9b then the subject DP *she* and predicate AP *mad* would normally be assumed to be sisters of the fronted auxiliary *is*. On that analysis the structure is flattened out so that *she mad* is not a constituent. In fact for English the fronting of *is* can even be seen as a consequence of that flattening: English is a head-initial language so the two dependents *she* and *mad* are expected to follow their selecting head *is*. (SAG REFS)

Although LFG is non-transformational, it can express the intuition behind the I-to-C movement analysis due to the separation of autonomous c- and f-structures. Recall from the above discussion of the DP in 3 that functional heads such as determiners, auxiliaries and complementizers do not introduce new f-structures, but rather map to the same f-structure as their complement phrases. The finite auxiliary can therefore appear in either I or C without affecting the f-structure, as we will see presently. Recall also that c-structure nodes are optional and can be omitted as long as a well-formed f-structure is generated. Comparing the non-terminal structures of 11 and 12, the I node is omitted from the latter structure but otherwise they are identical.

(11)



(12)



(Most of the lexical equations are omitted from 12 for clarity.) Given the ↑ =

↓ annotations, the C, I, and AP nodes (as well as IP and CP) all map to the same f-structure, namely the one shown in 13.

(13) $\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{PRED} & \textit{`PRO'} \\ \text{NUM} & \textit{SG} \\ \text{PERS} & \textit{3} \\ \text{GEN} & \textit{FEM} \end{bmatrix} \\ \text{PRED} & \textit{`mad}\langle(\uparrow \text{SUBJ})\rangle\textit{'} \\ \text{FIN} & + \end{bmatrix}$

The C and I positions are appropriate for markers of clausal grammatical features such as finiteness ([FIN +]), such as auxiliary verbs like is and complementizers like finite *that* and infinitival *for*: *I said that/*for she is present* vs. *I asked for/*that her to be present.* English has a specialized class of auxiliary verbs for marking finiteness from the C position, while in languages like German all finite verbs, including main verbs, can appear in a C position that is unoccupied by a lexical complementizer. Summarizing, the LFG framework enables a theory of head mobility based on the intuition that a clause has multiple head positions where inflectional features of the clause are encoded.

5 Case, agreement, and constraining equations

The basic theory of agreement is the same in LFG and HPSG (see Wechsler 2020, Chapter 6 of this volume): agreement occurs when multiple feature sets arising from distinct elements of a sentence specify information about a single abstract object, so that the information must be mutually consistent (Kay 1984b). The two forms are said to agree when the values imposed by the two constraints are compatible, while ungrammaticality results when they are incompatible. An LFG example is seen in (3), where the noun, determiner and verbal suffix each specify person and/or number features of the same SUBJ value.

The basic mechanism for case marking works in essentially the same way as agreement, in both frameworks: in case marking, distinct elements of a sentence specify case information about a single abstract object, hence that information must be compatible. To account for the contrast in 14a, nominative CASE equations are associated with the pronoun *she* and added to the entry for the verbal agreement suffix *-s*:

(14) a. She/*Her/*You rules.

b. *-she*: *Pron* $(\uparrow \text{CASE}) = \text{NOM}$
$(\uparrow \text{PERS}) = 3$
$(\uparrow \text{NUM}) = \text{SG}$
$(\uparrow \text{GEND}) = \text{FEM}$

c. *her*: *Pron* $(\uparrow \text{CASE}) = \text{ACC}$
$(\uparrow \text{PERS}) = 3$
$(\uparrow \text{NUM}) = \text{SG}$
$(\uparrow \text{GEND}) = \text{FEM}$

d. *-s*: *infl* $(\uparrow \text{TENSE}) = \text{PRES}$
$(\uparrow \text{SUBJ}) = \downarrow$
$(\downarrow \text{PERS}) = 3$
$(\downarrow \text{NUM}) = \text{SG}$
$(\downarrow \text{CASE}) = \text{NOM}$

The variant of 14a with *her* as subject is ruled out due to a clash of CASE features within the value of SUBJ in the f-structure. The variant with you as subject is ruled out due to a clash of PERS features. This mechanism is essentially the same as in HPSG, where it operates via the VALENCE feature.

This account allows for underspecification of both the case assigner and the case bearing element, and of both the controller and target of agreement. In English, for example, gender is marked on some pronouns but not on a verbal affix; and case is marked on the verbal affix but not on nominals, with the exception of the pronouns. But certain case and agreement phenomena do not tolerate underspecification, and for those phenomena LFG offers an account using a *constraining equation*, a mechanism absent from HPSG and indeed ruled out by current principles of HPSG theory. (Some early precursors to HPSG included a special feature value called ANY that functioned much like an LFG constraining equation (Shieber 1986: pp. 36-7), but that device has been eliminated from HPSG.) The functional equations described so far in this chapter function by building the f-structure, as illustrated in 3 and 5 above; such equations are called *defining equations*. A constraining equation has the same syntax as a defining equation, but it functions by checking the completed f-structure for the presence of a feature. An f-structure lacking the feature designated by the constraining equation is ill-formed.

The following lexical entry for *she* is identical to the one in 14b above, except that the CASE equation has been replaced with a constraining equation, notated with $=_c$.

(15) *she*: *Pron* (↑ CASE) $=_c$ NOM
(↑ PERS) = 3
(↑ NUM) = SG
(↑ GEND) = FEM

The f-structure is built from the defining equations, after which the SUBJ field is checked for the presence of the [CASE NOM] feature, as indicated by the constraining equation. If this feature has been contributed by the finite verb, as in 14, then the sentence is predicted to be grammatical; if there is no finite verb (and there is no other source of nominative case) then it is ruled out. This predicts the following grammaticality pattern:

(16) Who won the popular vote in the 2016 election?
She did! / *Her did!
b. *She! / Her!

English nominative pronouns require the presence of a finite verb, here the finite auxiliary *did*. Constraining equations operate as output filters on f-structures and are the primary way to grammatically specify the obligatoriness of a form, especially under the assumption that all daughter nodes are optional in the phrase structure. As described in Section 3 above, obligatory dependents are specified in the lexical form of a predicator using existential constraints like (↑ SUBJ) or (↑ OBJ). These are equivalent to constraining equations in which the particular value is unspecified, but some value must appear in order for the f-structure to be well-formed.

A constraining equation for case introduced by the case-*assigner* rather than the case-bearing element, predicts that the appropriate case-bearing element must appear. A striking example from Serbo-Croatian is described by Wechsler & Zlatić (2003: p. 134), who give this descriptive generalization:

(17) Serbian/Croatian Dative/Instrumental Case Realization Condition.

If a verb or noun assigns dative or instrumental case to an NP, then that case must be morphologically realized by some element within the NP.

In Serbo-Croatian most common nouns, proper nouns, adjectives, and determiners are inflected for case. An NP in a dative position must contain at least one such item morphologically inflected for dative case, and similarly for instrumental case. The verb *pokloniti* 'give' governs a dative object, such as *ovom studentu* in 18a. But a quantified NP like *ovih pet studenata* ‘these five students’ is marked for invariant genitive case, and can appear in any case position— except when it fails to satisfy the condition in 17, such as this dative position (Wechsler & Zlatić 2003: p. 125):

(18) a. pokloniti knjige ovom studentu
give.INF books.ACC this.DAT.SG student.DAT.SG
‘to give books to this student’

b. * pokloniti knjige [ovih pet studenata]
give.INF books.ACC this.GEN.PL five student.GEN.PL
(‘to give books to these five students’)

Similarly, certain foreign names such as *Miki* and loanwords such as *braon* ‘brown brunette’ are undeclinable, and can appear in any case position, except those ruled out by 17. Thus the dative example in (19)a is unacceptable unless the inflected possessive adjective *mojoj/mojom* ‘my’ appears. When the possessive adjective realizes the case feature, it is acceptable. In (19)b we contrast the undeclined loan word *braon* ‘brown’ with the inflected form *lepoj* ‘beautiful’. The example is acceptable only with the inflected adjective (Wechsler & Zlatić 2003: p. 134).

(19) a. Divim se *(mojoj) Miki.
admire.1SG REFL my.DAT.SG Miki
‘I admire (my) Miki.’

b. Divim se *braon/ lepoj Miki.
admire.1sg REFL brown/ beautiful.DAT.SG Miki
(‘I admire brunette/ beautiful Miki’)

This complex distribution is captured simply by positing that the dative (and instrumental) case assigning equation on verbs and nouns, such as the verbs *pokloniti* and *divim* in the above examples, is a constraining equation:

(20) $(\uparrow \text{OBL}_{dat}\ \text{CASE}) =_c \text{DAT}$

Any item in dative form within the NP, such as *ovom* or *studentu* in (18)a or *mojoj* or *lepoj* in (19), could introduce the [CASE DAT] feature that satisfies this equation, but if none appears then the sentence fails. In contrast, other case-assigning equations (e.g. for nominative, accusative, or genitive case, or for cases assigned by prepositions) are defining equations, which therefore allow the undeclined NPs to appear.

6 Agreement and affixal pronouns

Agreement inflections that include the person feature derive historically from incorporated pronominal affixes.

(21) a. Njûchi zi-ná-*wá*-lum-a a-lenje.
10.bee 10.SM-PST-2.OM-bite-FV 2-hunter
'The bees bit them, the hunters.'

(Chichewa; Bresnan and Mchombo 1987) Bresnan and Mchombo argue that (i) the Chichewa object marker (OM) is an incorporated pronoun; (ii) the subject marker (SM) alternates: can be agreement marker, with an associated subject NP (njûchi 'bees' in (2)), or an incorporated pronoun, when no subject NP appears.

7 Lexical mapping

8 Long distance dependencies

9 Control and raising

10 Anaphoric binding

11 Semantics

12 Conclusion

Abbreviations

Acknowledgements

References

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Mark Wechsler. 2015. *Lexical-functional syntax*. 2nd edn. Oxford: Blackwell Publishers Ltd. DOI:10.1002/978111910566
Dalrymple, Mary. 2001. *Lexical Functional Grammar* (Syntax and Semantics 34). New York, NY: Academic Press.

Kay, Martin. 1984a. Functional Unification Grammar: A formalism for machine translation. In Yorick Wilks (ed.), *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, 75–78. Stanford University, California: Association for Computational Linguistics.

Kay, Martin. 1984b. *Functional unification grammar: a formalism for machine translation*. Association for Computational Linguistics. 75–78.

Müller, Stefan & Stephen Mark Wechsler. 2014. Lexical approaches to argument structure. *Theoretical Linguistics* 40(1–2). 1–76. DOI:10.1515/tl-2014-0001

Shieber, Stuart M. 1986. *An introduction to unification-based approaches to grammar* (CSLI Lecture Notes 4). Stanford, CA: CSLI Publications.

Wechsler, Stephen. 2020. Agreement. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 161–184. Berlin: Language Science Press. DOI:??

Wechsler, Stephen Mark & Larisa Zlatić. 2003. *The many faces of agreement* (Stanford Monographs in Linguistics). Stanford, CA: CSLI Publications.