CHAPTER 13

Morphology, Syntax and Morphosyntactic Typology

Anne Abeillé and Jong-Bok Kim

1 Derivational vs. Constraint-Based Views

In this positional paper, we briefly review a constructionist view of morphology and syntax that takes language as a network of form-meaning pairs or constructions. The view is also committed to a constraint-based system in which the grammar is a system of constraints governing the relation between form and meaning. There are hence no transformational or derivational operations within grammar other than constraints. Each piece of a grammatical structure is thus sanctioned by some constructions just as each node in a well-formed context-free tree structure is licensed by some grammar rule. In the constructionist view, grammatical constructions thus play a fundamental role in the morphosyntax theory of our language.

The central tasks of current morphosyntactic theories are the following:

- To discriminate between possible and impossible expressions,
- To account for morphological and syntactic preferences, since what is a hard constraint in one language may be a soft constraint in others (Bresnan et al., 2007; Bresnan & Hay, 2008; Bresnan & Ford, 2010),
- To account for an increasing variety of language data, coming from the Internet and/or field work within unified theories (Dryer & Haspelmath, 2013).

In advancing the research on morphology and syntax to resolve such tasks, there have been two main strategies: 'derivational" and "constraint-based" views.[1] Both views assume that sentences are basic units of language, representing pairings of forms and meaning. In both views, sentences are also taken to be composed of smaller expressions (words and/or morphemes) which are composed into units with hierarchical structure. Aside from these general similarities, the two views follow different tracks in many respects. The derivational view minimises what has to be learned by packing as much complexity as possible into an innate Faculty of Language so that children have less to learn. Meanwhile, the constraint-based view tries to find ways to formulate complexity in adult grammar so that children can learn more of the grammar.

One important strategy the constraint-base view takes is to minimise elements of linguistic structure that children cannot infer from overt form. This is why the constraint-based view avoids postulating abstract entities such as null elements (e.g. traces), invisible or covert syntactic structure, and movement operations. Such differences between the two views also lead to differences in formulating grammatical competence.

In the derivational view, linguistic structures are constructed by applying a sequence of rules, each applying to the output of the previous step. In the traditional derivational view, syntax is the sole generative component, the source of all combinatorial complexity while morphology is viewed as part of syntax, and phonology and semantics are "interpretive" (Chomsky, 1995, 2000, 2002). Hence there is an inherent "directionality" in the logic of sentence construction: certain rules and rule components apply after others. In particular, within the Minimalist Program, language is a "perfect" system with an optimal design in the sense that natural language grammars create structures which are designed to interface perfectly with other components – more specifically with *speech* and *thought* systems. Such a derivational view has led to the development of TG (Transformational Grammar), P&P (Principles and Parameters), and MP (Minimalist Program). In this view, words are derived from morphemes by the same kind of rules as words in syntax (Baker's mirror principle), or syntax combines morphemes directly, as advocated by

[1] The discussion here is based on Culicover & Jackendoff (2005); Sag (2012); Kim (2013, 2016).

distributional morphology (Halle & Marantz, 1993).

Meanwhile, in the constraint-based view (Müller et al., 2021; Dalrymple, 2023), each constraint determines or licenses a small piece of linguistic structure or a relation between small pieces. A linguistic structure is acceptable only if it conforms to all applicable constraints. There is no logical ordering in the grammar. Instead all levels are parallel and mutually constrained by the grammar. There are no "hidden levels" built of syntactic units. Combinatory complexity arises independently in phonology, syntax, and semantics. There is a continuum of grammatical phenomena from idiosyncratic to general rules of grammar. "Peripheral" phenomena are thus inextricably interwoven with the "core" phenomena. The constraint-based view of grammar has led to the development of LFG (Lexical Functional Grammar) (Bresnan, 1982, 2001), HPSG (Head-driven Phrase Structure Grammar) (Pollard & Sag, 1994), CxG (Construction Grammar) (Goldberg 2003, 2005), and SBCG (Sign-based Construction Grammar) (Sag, 2012). There are also constraint-based views on morphology, which consider it to be autonomous from syntax (Anderson, 1992) and which aim to capture lexemes and words relations and not morpheme combinations; for instance Information-based Morphology, which is part of HPSG and SBCG (Bonami & Crysmann, 2016; Crysmann 2021).

The key ideas of SBCG are developed from HPSG and CxG, which we adopt here in describing some English and Korean morphosyntactic phenomena. SBCG aims at formalising the descriptive properties of individual languages. The two key concepts of the SBCG includes "signs" and "constructions". In its formal foundation, SBCG takes grammars to be an inventory of signs, complexes of linguistic information that contain constraints on form, meaning and use. Linguistic signs are combined into more complex signs by "constructions" which are form-function pairings.

2 Linguistic Signs and Feature Structures

In SBCG (Sag, 2012), developed from HPSG and CxG, language is taken to be an infinite set of "signs" (whose notion is borrowed from de Saussure, 1916). Consider the following:

FIGURE 1. An example of *sign*.

The form (sound) of *arbor* ("tree" in Latin) is the signifier and its associated meaning or denotation is a signified. The linguistic sign is thus a link between a concept and a sound pattern. Adopting de Saussure's notion of sign, SBCG models signs as feature structures (FS). All the linguistic objects are thus represented by feature structures notated by attributed-value matrices (AVM), such as:

(1)
$$\begin{bmatrix} \text{FORM} & \langle \text{arbor} \rangle \\ \text{SYN} & \textit{noun} \\ \text{SEM} & \text{‘tree’} \end{bmatrix}$$

The expression *arbor* has attributes about its morphological (FORM), syntactic (SYN), and semantic (SEM) information. It also has a PHON (phonological) value too, but we will ignore it here. All these attributes have their own values. For example, *arbor* is syntactically a noun denoting an individual “tree”. Extending this feature to linguistic signs, words and phrases can be also modelled as ”typed” feature structures (FSs). For example, we may represent the information of the verb *grows* as follows:

(2)
$$\begin{bmatrix} \textit{v-word} & \\ \text{PHON} & \langle \text{grows} \rangle \\ \text{FORM} & \langle \text{grow} \rangle \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{fin} \end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}_i \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \text{NP} \rangle \\ \text{SEM} & \textit{grow}(i) \end{bmatrix}$$

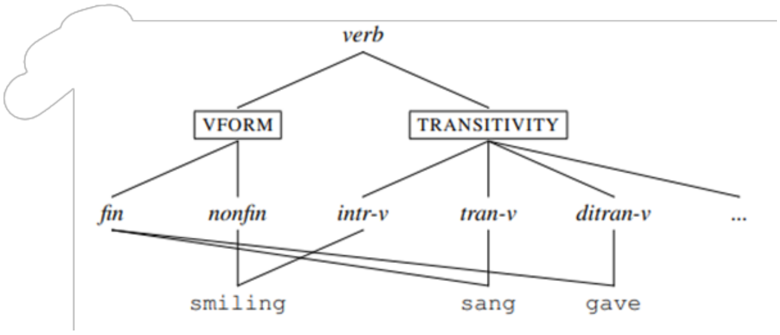
The verb *grows* is a type of *v-word* defined by its feature attributes. As given here, the attributes include its morphological (FORM), syntactic (SYN), semantic (SEM), and argument structure (ARG-ST) information. The value of these feature attributes can be either be simple (atomic) or complex (feature structure). For example, the value of the attributes POS (parts-of-speech) and VFORM (verb-form) is atomic while that of the attribute SYN is another FS. The head (HEAD) information of the verb indicates that it is a finite (VFORM) verb (POS). The argument structure (ARG-ST) tells us how many syntactic/semantic arguments each predicate (including verbs) takes. The selected arguments are in general linked to the core participants in the eventuality a verb or predicate denotes. The semantic information, simplified here, means that there is one individual (i) participating in the “grow” semantic relation and this individual is coindexed with the one denoted by the subject.

Note that in HPSG and SBCG, FSs are “typed” (e.g., *v-word*), notationally marked on the left corner of the top in the feature structure. In SBCG, each feature structure (FS) is defined in a more sophisticated way with type information. FSs are “well-typed” in the sense that every feature structure of some type includes only the features that are appropriate for that type. For example, the VFORM feature is appropriate only for *v-word* or its subtypes, and not for nouns or adjectives (but it would in other languages, to account for inflecting adjectives in Korean and inflecting nouns in Oneida. See Kim, 2016 and Diaz et al., 2019). Each *type* is thus telling us what kind of feature attributes are appropriate for the given type.

Another important property of the FSs adopted in HPSG and SBCG is that typed FSs are hierarchically organised in terms of linguistic types. The hierarchical classification is to capture cross-cutting generalisations among types. For each linguistic type, certain constraints are stated (the constraints are declared in terms of constraints on feature structures). The constraints each type carries correspond to properties shared by all members of that type.

The technique of hierarchical inheritance ensures that a subtype inherits all the constraints of its supertypes. Consider the following sample hierarchy:

(3)



The type *verb* can be subclassified in accordance with its VFORM and number of arguments (TRANSITIVITY). Each verb is specified with these two values, at least. For example, the verb *gave* carries *fin* (finite) as its VFORM value while its TRANSITIVITY value is *ditr* (ditransitive). Each of these types specifies a different subset of the information, as illustrated in (4).

(4) a.

$$fin\text{:} \begin{bmatrix} \text{HEAD} & [\text{VFORM } fin] \\ \text{SUBJ} & \langle \text{NP}[nom] \rangle \end{bmatrix}$$

b.

$$ditr\text{:} \begin{bmatrix} \text{COMPS} & \langle \text{NP}[acc], \text{PP} \rangle \end{bmatrix}$$

Each of these constraints is inherited to its subtypes like gave. Due to the organisation of the lexical signs in this hierarchical fashion, the only information we need to encode for such a word type is its own properties not inherited from the supertypes. For example, all that needs to be stated for the lexical entry of gave is the one given in (5):

(5)

$$\begin{bmatrix} \text{FORM} & \langle \text{gave} \rangle \\ \text{ARG-ST} & \langle \text{NP, NP, PP} \rangle \\ \text{SEM} & give\_rel \end{bmatrix}$$

This lexical entry bears the minimal information, its morphological FORM value, argument-structure (ARG- ST), and a semantic (SEM) relation. The multiple inheritance mechanism in the hierarchically organised lexicon then allows the lexical entry *gave* to inherit all the other general constraints from its supertypes, resulting in the more specified lexical entry:

(6)

$$\begin{bmatrix} \text{FORM} & \langle \text{gave} \rangle \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM } fin \end{bmatrix} \\ \text{SUBJ} & \langle \boxed{1}\text{NP}[nom] \rangle \\ \text{COMPS} & \langle \boxed{2}\text{NP}[acc], \boxed{3}\text{PP} \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}\text{NPi}, \boxed{2}\text{NPj}, \boxed{3}\text{PPk} \rangle \\ \text{SEM} & give\_rel(i,j,k) \end{bmatrix}$$

The three arguments will be realised as the subject and complements (COMPS) in accordance with the Argument Realisation Constraint (see the next section). All the other information is inherited from its supertypes. The notion of hierarchical classification of words and multiple inheritance, thus, enables us to eliminate the redundancy, and further to capture cross-cutting generalisations in a non-redundant, deductive fashion. In due course, we will also observe that the multiple inheritance system plays an important role in the grammar of constructions as well, where both lexical and phrasal signs are types of grammatical constructions.

3 Constructions as Primitives

SBCG follows the philosophy of the CxG (Construction Grammar) in which "constructions" are taken to be the basic units of language and central to all linguistic descriptions and theories of languages. Interpreted within the sign-based system, this means all linguistic signs are taken to be "constructions". A construction consists of a form and a meaning or a function connected with that form, which can be defined as following (Goldberg, 2005):

(7) Definition of grammatical "constructions":

> Any linguistic pattern is recognized as a construction as long as some aspect of its form or function is not strictly predictable from its component parts or from other constructions recognized to exist. In addition, patterns are stored as constructions even if they are fully predictable as long as they occur with sufficient frequency. (p. 5)

To put it simply, a construction is a form-meaning pair whose meaning we cannot predict from syntactic combinations as well as a form-meaning pair with high frequency whose meaning we can predict. Within this tenet, all levels of linguistic description, including morpheme, word, phrase, and clause, are understood to involve pairings of form with semantic or discourse functions, as long as the pairing of form and function is idiosyncratic or unpredictable. This implies that constructions vary in size and in complexity. For example, as illustrated in Table 1, any morpheme or word level expression in English (e.g., *smile, laugh, giggle, etc*) is thus a construction, since these pairings of form and function cannot be derived from any general rules of the language. The paring is unpredictable and idiosyncratic.

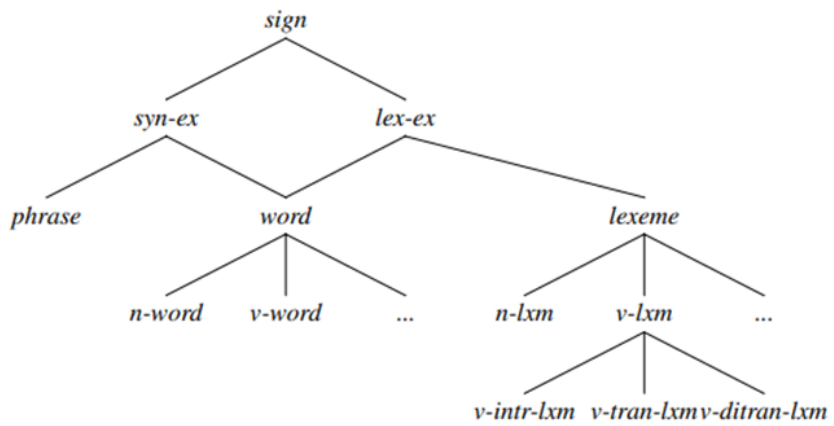
An idiom like *kick the bucket* is also a construction where the meaning of parts here (*kick*, *the*, *bucket*) does not correspond to the meaning of the whole. The pattern *What is X doing Y?* is also a construction. For example, the sentence *What is the fly doing in my soup?* has an additional, idiosyncratic implicature that the fly's being in my soup is inappropriate (see Kay & Fillmore, 1999). In addition to these distinctive constructions, sentences like *He gave her a fish taco* also involve general constructions like the HEAD-SUBJECT (combination of a subject NP with a predicate VP) and DITRANSITIVE (combination of a verb with two objects) constructions. The meaning of these two constructions are quite predictable and compositional in the sense that the whole meaning can be inferred from the meaning of its parts. The meaning of such typical constructions are predictable and occur with sufficient frequency in daily usages of the language.

One point worth noting here is that there is no principled distinction between words and phrases in terms of constructions. A lexical entry is more word-like to the extent that it is fully specified, and more rule-like to the extent that it can also have variables that have to be filled by other items in the sentence. In the CxG, language-specific generalisations across constructions are captured via inheritance networks, reflecting commonalities or differences among constructions. In what follows, we will see how this notion of inheritance hierarchy of constructions plays an important role in capturing cross-cutting generalisations among constructions.

4 Simplified Architecture of the Sign-based Construction Grammar

The framework adopted in this paper to describe the major syntactic phenomena of English is a version of SBCG, a synthesis of the two constraint-based frameworks, HPSG and CxG. The SBCG for English we sketch here starts with the following type hierarchy system in which every linguistic sign is "typed" with appropriate constraints and hierarchically organised. As exemplified in (8), the linguistic signs (*sign*) can be classified into syntactic (*syn-ex*) and lexical (*lex-ex*) expressions, and each of in turn has two subtypes[2]

(8)



Note that the type *word* is a subtype of *syn-ex* as well as *lex-ex*. What this means is that the FSs of type *word* satisfy all constraints that the grammar impose on objects of type *syn-ex* as well as type *lex-ex*.

Lexemic expressions (*lex-ex*) are abstract proto-word or root-like expressions. The lexemes can be projected into stems and then into words. For example, the lexeme *smile* will give rise to stems or genuine words through inflectional processes. That is, the *v-lxm* form can be inflected as stems or words:

(9) a. *v-int-lxm*: *smile*
b. *v-word*: *smiles*, *smiled*, *smiling, smilingly*

Only the type *v-word* can appear syntactically, indicating that this type is also cross-classified as a subtype of *word*. As such, by allowing a type to inherit from more than one supertype, we can capture cross-classifying properties of a set of linguistic objects. This notion of multiple inheritance hierarchy for signs carries over to lexical as well as phrasal constructions too, which we will discuss in what follows.

[2] The HPSG literature defines the subtypes of sign in slightly different ways. The type hierarchy given here is a simplified version of Sag (2012). See, among others, Ginzburg & Sag (2000); Sag (2003); Kim & Sells (2008); Kim (2016); and Abeillé & Borsley (2021).

4.1 *Lexical Constructions*

Let us consider the lexical constructions of English in more detail, focusing on *verb-lxm* (verb-lexeme) covering verb as well as complementiser: Like other linguistic types, lexemic expressions are organised as a type hierarchy in order to capture generalisations about similar classes. For example, all the instances of the type *v-lxm* will select at least one argument, as represented in the following feature description:

(10) $\textit{v-lxm}: \left[\text{ARG-ST} \quad \langle \text{XP},\ldots\rangle\right]$

The type declaration here ensures that the linguistic expressions belonging to the type *v-lxm* (its subtypes) have at least one argument XP in the list value of its ARG-ST (argument-structure), which would be realised as the subject. This information will be inherited to its subtypes covering intransitive, transitive, and ditransitive lexemes:

(11) a. *v-intr-lxm*: [ARG-ST <NP>] (e.g., *run*, *smile*, *laugh*,...)
b. *v-tran-lxm*: [ARG-ST <NP, NP>] (e.g., *like*, *love*, *deny*,...)
c. *v-ditran-lxm*:[ARG-ST <NP, NP, NP>] (e.g., *give, teach, send,...*)

The inheritance of such a constraint is a "default" in the sense that constraints on supertypes affect all instances of subtypes, unless contradicted by some other constraints on a given type (see Lascarides & Copestake, 1999; Sag, 2003, p. 229).

As noted before, while inheriting the constraints of its supertypes, each type also has its own constructional constraints, as illustrated in the following:

(12) a. $\textit{v-lxm}: [\text{POS} \quad \textit{verb}]$

b. $\textit{main-v-lxm}: [\text{AUX} \quad -]$

Auxiliary verb lexemes are underspecified for the feature AUX and can thus enter regular constructions (with a non-finite VP complement) as well as constructions specific for [AUX +] lexemes (subject inversion or ellipsis) (Sag et al., 2020). All instances of the *v-lxm* need to be inflected as a *word* level expression to be realised syntactically. To be more precise, only the *v-free* inflected with a mood marking can be mapped (or pumped up) to *v-word*. The main difference between lexemic expressions and word expressions is that the latter is sensitive to syntactic features such as SUBJ (subject) and COMPS (complements), as stated in terms of the following Argument Realisation Constraint:[3]

(13) Argument Realization Constraint (ARC):

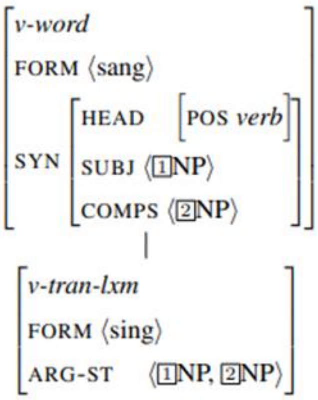
$$\textit{v-word} \quad \Rightarrow \quad \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{SUBJ} & \boxed{A} \\ \text{COMPS} & \boxed{B} \end{bmatrix} \\ \text{ARG-ST} \ \boxed{A} \oplus \boxed{B} \end{bmatrix}$$

[3] The boxed variable or integer is a variable used to "tag" certain feature values within the structure as being token-identical.

The constraint means the elements in the ARG-ST will be realized as SUBJ and COMPS syntactically. This in turn means that ARG-ST is relevant for words, while the syntactic features SUBJ and COMPS are relevant for phrases as well.

For example, when the lexeme *sing* specified only with the ARG-ST information is inflected as *sang*, its two arguments will be realised as SUBJ and COMPS as represented in (14): [4]

(14)

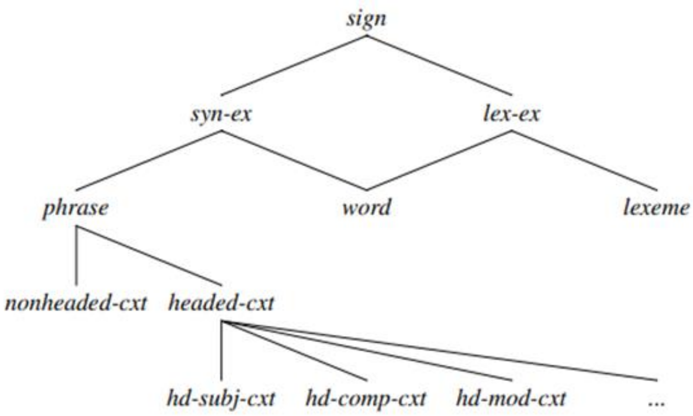


As seen from the structure, the lexeme *sing* will project into the word *sang*. In particular, the first element of the ARG-ST in the lexeme is mapped onto the SUBJ while the second one is mapped onto the COMPS in the word level expression *sang*. The morphological, inflectional processes of attaching the present tense *-es* also adds the HEAD features (e.g., POS and VFORM). This is due to the fact that within a multiple inheritance system, properties of supertypes are inherited by their subtypes.

4.2 *Phrasal Constructions*

Once we have *word* elements, these elements will be combined with other syntactic elements (either words or phrases), to form a bigger expression. It is the type *syn-ex* that places restrictions on the combination of syntactic constructions including *word*. This in turn means that the subtypes of *syn-ex* will tell us what kinds of well-formed phrases are available in language. Consider a simplified hierarchy of the phrasal constructions of English:

(15)



As defined in the hierarchy here, the phrasal constructions are divided into *headed-cxt* and *non-headed-cxt*, the

[4] The representation given here implies an inflectional process from the lexeme *sing* to the word *sang*. As illustrated in Sag (2012), this can be represented in a mother-daughter relation as constructions. That is, the word *sang* has *sing* as its stem daughter.

latter of which covers constructions like coordination. The headed construction (*headed-cxt*) carries rather universal constraints such as the HFP (Headed Feature Principle), some variant of which all X′-theories embody:

(16) Head Feature Principle (HFP):
The HEAD value of a headed phrase is identified with that of its head-daughter.

This principle can be stated in terms of the type constraint on the headed constructions as following:

(17) $$\begin{bmatrix} \textit{headed-cxt} \\ \text{SYN} \mid \text{HEAD} \quad \boxed{1} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \text{SYN} \mid \text{HEAD} \quad \boxed{1} \end{bmatrix}, \ldots$$

This construction says that the head (HEAD) value of the headed construction, consisting of a head and non-head daughters, is identical with that of its head daughter. This constraint, restricting the percolation of head value, guarantees that headed phrases are "projections" of their head daughters. This ensures that grammatical properties such as part of speech, case, verb-inflection-form (VFORM) values are systematically projected onto headed phrases from head lexical items.

The traditional X′-theory within the P&P framework is formulated in terms of hierarchical bar levels. But SBCG's X′-theory replaces this component with combinatoric saturation, governed by the Valence Principle:

(18) Valence Principle (VALP):
For each valence feature F, the F value of a headed phrase is the head-daughter's F value minus the realised non-head-daughters.

The effect of this principle, reminiscent of the category cancellation associated with functional application in Categorial Grammar, is to "check off" the subcategorisation requirements of a lexical head. Each lexical head carries specifications that determine what elements it combines with syntactically. Valence features such as SUBJ and COMPS are such specifications, whose features are linked to the elements in the ARG-ST in accordance with the Argument Realisation Constraint. This means such features are sensitive only to syntactic expressions.

Traditional X′-theory in most P&P or Minimalist work assumes two X′ schematic rules, as shown in (19).

(19) a. XP → YP (specifier), X′(head)
b. XP → X (head), ZP* (complements)

These disjunctive constraints on the immediate constituency of phrases restrict the set of well-formed phrases. Within HPSG and SBCG, we can formalise the same kind of constraints via combinatorial constraints on the phrasal constructions, illustrated in the following simplified representations:

(20) a. Head-Subject Construction:
$$\text{XP}[\textit{hd-subj-cxt}] \rightarrow \boxed{1}\text{XP}, \mathbf{H}\begin{bmatrix} \text{SUBJ} \quad \langle \boxed{1}\text{XP} \rangle \end{bmatrix}$$

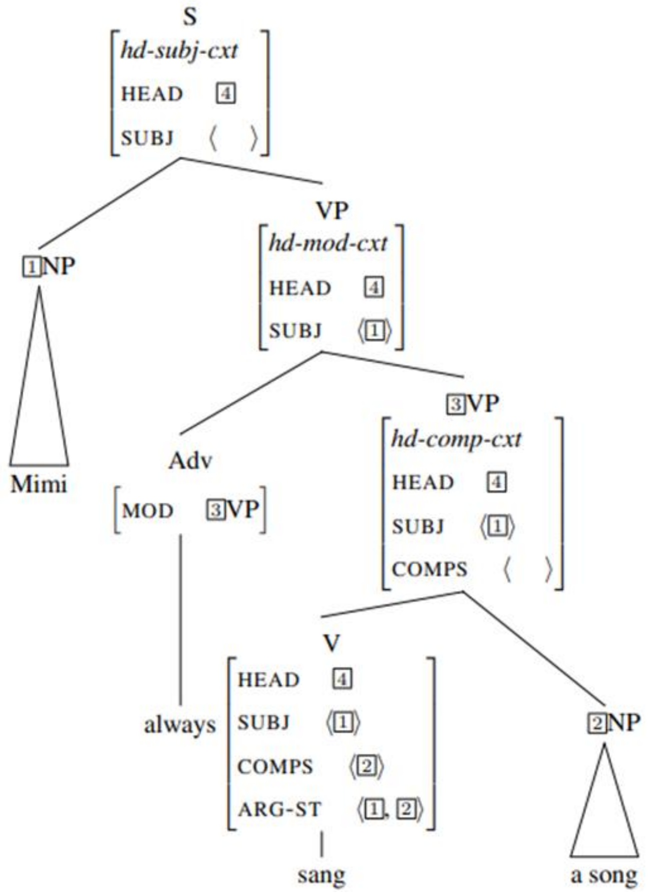
b. Head-Complement Construction
$$\text{XP}[\textit{hd-comp-cxt}] \rightarrow \boxed{1}\text{XP}, \mathbf{H}\begin{bmatrix} \text{COMPS} \quad \langle \ldots, \boxed{1}\text{XP}, \ldots \rangle \end{bmatrix}$$

c. Head-Modifier Construction

$$\mathrm{XP}[\textit{hd-mod-cxt}] \rightarrow \begin{bmatrix}\text{MOD} & \langle \boxed{1}\mathrm{XP}\rangle\end{bmatrix}, \mathbf{H}\boxed{1}\mathrm{XP}$$

These three main constructions constraining the combinatorial possibilities of expressions can license major phrases in language. The Head-Subject Construction, generating a *hd-subj-cxt*, allows a VP to combine with its subject. The Head-Complement Construction ensures a head to combine with one of its COMPS (complements) elements, forming a *hd-comp-cxt* . The Head-Modifier Construction allows a head to form a well-formed phrase with an adverbial element that modifies the head, resulting in a *hd-mod-cxt*. To observe how the universal principles (the HFP and the VALP) interact with these three phrasal constructions, let us consider the tree-style representation of sentence (20a).[5]

(21)



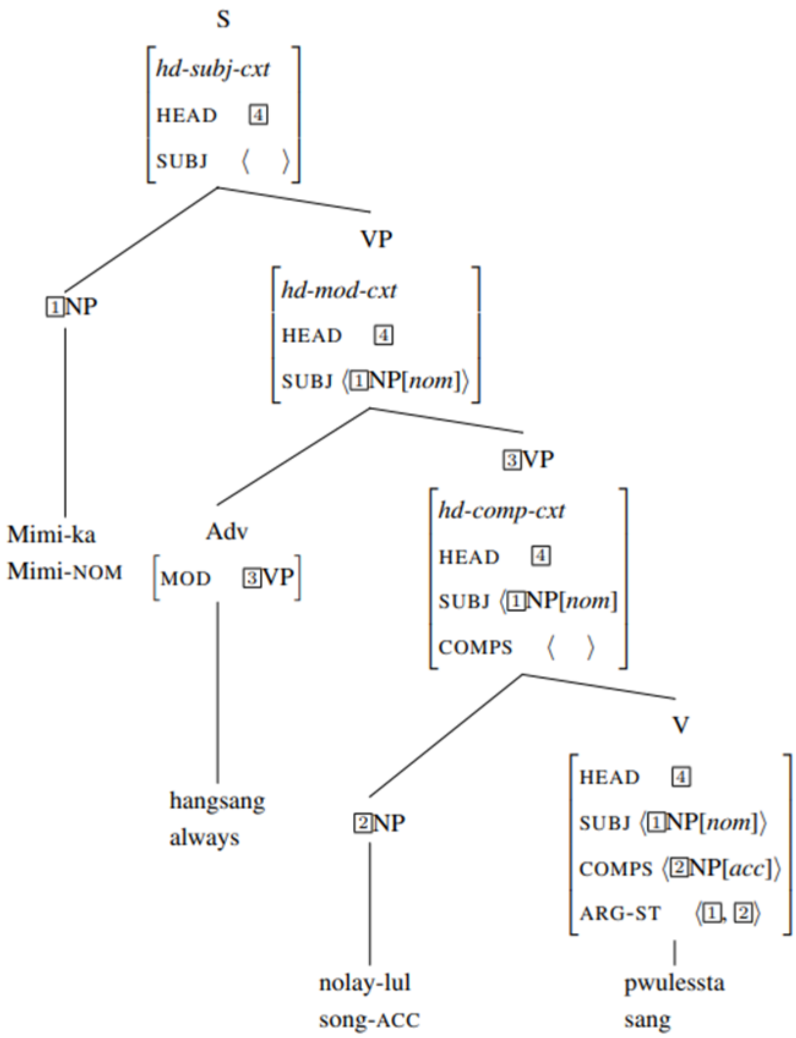
The HFP ensures that the head-daughter's HEAD information is projected in any given phrase. Thus the HEAD value (such as the part-of-speech value, *verb* and VFORM value, *fin*) of the lexical head *sing* is that of both VPs and the S. In accordance with the VALP, the head's valence information determines the elements that the maximal projection contains. The valence specifications of the head *sing* show that it requires one NP complement and a subject specifier. When it combines with the complement, its COMPS specification is satisfied, leaving the VP's COMPS value empty. The resulting VP combines with the modifier via the Head-Modifier Construction to form the top

[5] All linguistic objects are represented as feature structures in HPSG and SBCG. But for expository purposes, they are presented in terms of the familiar trappings of generative grammar – tree representations.

VP. When this top VP combines with the subject NP via the Head-Subject Construction, we obtain a completely saturated phrase, all of whose valence specifications are satisfied or discharged. Thus each subtree as well as the whole sentence conforms to the principles of the HFP, the VALP, and the combinatorial grammar rules (constructions).

The same phrasal types and principles apply to quite different languages. The following is the corresponding Korean sentence:

(22)



We could observe that just like English, the structure observes HFP and VALP, and is further licensed by the same three key phrasal constructions: Head-Subject, Head-Complement, and Head-Modifier Constructions. In this structure, the verb first combines with the object and its resulting phrase combines with the modifier *hangsang* "always". This VP will then combine with the subject NP, yielding a complete sentence. The only difference from English is the position of the lexical head (Kim, 2016).

5 Conclusion

We have sketched a constraint-based fragment for English only and briefly showed how it could be applied to typologically unrelated languages like Korean. Detailed morphosyntactic descriptions of understudied languages will broaden our perspectives on morphology and syntax,

Competing theories for morphology and syntax question the nature of morphological and syntactic units, the nature of morphological and syntactic representations, and the universality of cross-linguistic patterns. New empirical evidence for testing these theories are also coming from large corpora and databases as well as comprehension/production experiments (Abeillé, 2024). These could help us decide between competing theories.

References

Abeillé, A. (2024). The empirical turn and its consequences for theoretical syntax. *Theoretical Linguistics*, *50*(2).

Abeillé, A., Borsley, R. D. (2021). Basic properties and elements. In S. Müller, A. Abeillé, R. D. Borsley & J.-P. Koenig (Eds.), *Head-Driven Phrase Structure Grammar: The handbook* (pp. 3-45). Language Science Press.

Anderson, S. (1992). *A-Morphous Morphology*. Cambridge University Press.

Bonami, O., & Crysmann, B. (2016). Morphology in Constraint-based Lexicalist Approaches to Grammar. In A. Hippisley & G. Stump (Eds.), *The Cambridge Handbook of Morphology* (*Cambridge Handbooks in Language and Linguistics*) (pp. 609-656). Cambridge University Press.

Bresnan, J. (1982). *The mental representation of grammatical relations*. MIT Press.

Bresnan, J. (2001). *Lexical-Functional Syntax*. Blackwell Publishers Ltd.

Bresnan, J., Cueni, A., Nikitina, T., & Baayen, H. (2007). Predicting the dative alternation. In G. Bouma, I. Krämer & J. Zwarts (Eds.), *Cognitive foundations of interpretation* (pp. 69-94). KNAW Academy Colloquium.

Bresnan, J., & Ford, M. (2010). Predicting syntax: processing dative constructions in American and Australian varieties of English. *Language*, *86*(1), 168-213.

Bresnan, J., & Hay, J. (2008). Gradient grammar: an effect of animacy on the syntax of give in New Zealand and American English. *Lingua*, *118*(2), 245-259.

Chomsky, N. (1995). *The minimalist program*. MIT Press.

Chomsky, N. (2000). Minimalist inquiries: The framework. In D. Michaels, R. Martin & J. Uriagereka (Eds.), *Step by step: essays on minimalist syntax in honor of Howard Lasnik* (pp. 89-155). MIT Press.

Chomsky, N. (2002). *On nature and language*. Cambridge University Press.

Crysmann, B. (2021). Morphology. In S. Müller, A. Abeillé, R. D. Borsley & J.-P. Koenig (Eds.), *Head-Driven Phrase Structure Grammar: The handbook* (pp. 943-995). Language Science Press.

Culicover, P., & Jackendoff, R. (2005). *Simpler syntax*. Oxford University Press.

Dalrymple, M. (Ed.). (2023). *The Handbook of Lexical Functional Grammar*. Language Science Press.

de Saussure, F. (1916). *Course in general linguistics*. Duckworth.

Diaz, T., Koenig, J.-P., & Michelson, K. (2019). Oneida prepronominal prefixes in information- based morphology. *Morphology*, 29, 1-4.

Dryer, M. S., & Haspelmath, M. (Eds.). (2013). *The world atlas of language structures online*. Max Planck Institute for Evolutionary Anthropology.

Ginzburg, J., & Sag, I. A. (2000). *Interrogative investigations: The form, meaning and use of English interrogatives*. CSLI Publications.

Goldberg, A. E. (2003). Constructions: a new theoretical approach to language. *Trends in cognitive sciences*, *7*(5), 219-224.

Goldberg, A. E. (2005). *Constructions at work: the nature of generalization in language*. Oxford University Press.

Halle, M. & Marantz, A. (1993). Distributed Morphology and the Pieces of Inflection. In K. Hale & S. J. Keyser (Eds.), *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger* (pp. 111-176). MIT Press.

Kay, P., & Fillmore, C. (1999). Grammatical constructions and linguistic generalizations: The What's X Doing Y? construction. *Language*, *75*(1), 1-33.

Kim, J.-B. (2013). The Korean sluicing: As a family of construction. *Studies in Generative Grammar*, *23*(1), 103-130.

Kim, J.-B. (2016). *The syntactic structures of Korean: A Construction Grammar perspective*. Cambridge University Press.

Kim, J.-B., & Sells, P. (2008). *English syntax: An introduction*. CSLI publications.

Lascarides, A., & Copestake, A. (1999). Default representation in constraint-based frameworks. *Computational Linguistics*, *25*(1), 55-105.

Müller, S., Abeillé, A., Borsley, R. D. & Koenig J.-P. (Eds.). (2021). *Head-Driven Phrase Structure Grammar: The* handbook. Language Science Press.

Pollard, C., & Sag, I. A. (1994). *Head-driven Phrase Structure Grammer*. University of Chicago Press.

Sag, I. A. (2003). Coordination and underspecification. In J.-B. Kim & S. Wechsler (Eds.), *Proceedings of HPSG 02 conference*. CSLI Publications.

Sag, I. A. (2012). Sign-Based Construction Grammar: an informal synopsis. In H. C. Boas & I. A. Sag (Eds.), *Sign-based Construction Grammar* (pp. 69-202). CSLI Publications.

Sag, I. A., Chaves, R., Abeillé, A., Estigarribia, B., Van Eynde, F., Flickinger, D., Kay, P., Michaelis-Cummings, L. A., Müller, S., Pullum, G. K., & Wasow, T. (2020). Lessons from the English auxiliary system. *Journal of Linguistics*, *56*(1), 87-155.