Function Words in Database Semantics (DBS)

Roland Hausser
Universität Erlangen-Nürnberg (em.)

Abstract

The vocabulary of a natural language is divided into content words like book or read (autosemantica), and function words like the or and (synsemantica).[1] Examples of content word categories are noun, verb, and adj, those of function words determiner, preposition, auxiliary, and conjunction.

Typologically, isolating languages like English and Chinese prefer function words and word order for coding semantic relations within and between noun, verb, and adj contents, while inflectional languages like classical Latin and agglutinating languages like Korean prefer morphology, i.e., affixes attached to content word surfaces.

This paper concentrates on the grammatical role of function words in English, and compares it with corresponding constructions in a language which uses more morphology than English, i.e., German. In line with the agent-based data-driven ontology of DBS, the syntactic-semantic mechanism of function words is shown in the hear and speak mode.

keywords:
Coordination and subordination, intra- and extra-propositional concatenation, function word absorption and precipitation, proplet normalization

1 Introduction

Natural languages differ in the way in which complex contents are coded. For example, in classical Latin the partial content pro1 see′ has the single surface video, but in English the two surfaces I see. The following DBS analyses show what the two codings have in common and where they differ:

1.1 DBS PROPLET PRESENTATION OF I see IN LATIN AND ENGLISH

Latin: morphology

sur: video
verb: see
cat: #s1′ a′ v
sem: pres ind
arg: pro1
mdr:
nc:
pc:
prn: 93

English: syntactic-semantic composition (cross-copying **SUBJ**×**PRD**)

sur: I	sur: see		sur:		sur:	
noun: **pro1**	verb: **see**		noun: **pro1**		verb: **see**	
cat: snp	cat: n′ a′ v		cat: snp		cat: #n′ a′ v	
sem: s1	sem: pres ind		sem: s1		sem: pres ind	
fnc:	arg:	⇒	fnc: **see**		arg: **pro1**	
mdr:	mdr:		mdr:		mdr:	
nc:	nc:		nc:		nc:	
pc:	pc:		pc:		pc:	
prn: 93	prn		prn: 93		prn: 93	

In Latin, the surface and its syntactic-semantic content are selected from the verbal paradigm of the inflectional morphology. It provides variations of person, number.

[1] Marty 1918, pp.205 ff.

tense, and verbal mood, e.g., vides, videam, videbam, viderem. In English, in contrast, two content proplets with the surfaces I and see are connected by the cross-copying operation **SBJ**×**PRD** of the hear mode. For variations of verbal mood and tense other than indicative present, English uses function words, e.g., have/has seen or could have seen. The grammatical objects, in contrast, i.e., te in Latin and you in English, are treated alike in the two languages, namely by syntactic-semantic composition: Te video[2]

In addition to *affixing* (morphological composition) in regular nouns (e.g., book, book+s), verbs (e.g., correct, correct+ed), and adjs (e.g., fast, fast+er, fast+est), there is *allomorphy*, i.e., variation of the word stem (FoCL 13). Examples of English allomorphy are the nouns foot, feet; mouse, mice, the verbs see, saw, seen; buy, bought, bought, and the adj good, better, best (suppletion).

For syntactic-semantic composition, the analyses of grammatically corresponding regular and irregular forms are coded alike (proplet normalization):

1.2 REGULAR VS. IRREGULAR VERB FORMS IN ENGLISH

regular verb form

sur: correct+ed
verb: correct
cat: n′ a′ v
sem: past ind
arg:
...
prn

irregular verb form

sur: saw
verb: see
cat: n′ a′ v
sem: past ind
arg:
...
prn

The regular and the irregular verb form share corresponding positions in their respective paradigms and their proplets differ only in the sur and core values. The empty slots are used by syntactic-semantic composition.

Proplet normalization may also be applied between different but typologically similar languages, as shown by the following English_German counterparts correct+ed_ korrigier+te (both regular) and saw_sah (both irregular):

1.3 CORRESPONDING FORMS IN ENGLISH AND GERMAN

regular verb form

sur: correct+ed	sur: korrigier+te
verb: correct	verb: correct
cat: n′ a′ v	cat: s13′ a′ v
sem: past ind	sem: past ind
arg:	arg:
...	...
prn	prn

irregular verb form

sur: saw	sur: sah
verb: see	verb: see
cat: n′ a′ v	cat: s13′ a′ v
sem: past ind	sem: past ind
arg:	arg:
...	...
prn	prn

In other respects, the proplet definitions of English-German counterparts may diverge. For example, German noun proplets require grammatical gender specification for determiner+noun agreement, which would not be appropriate for English.

[2]The choice between morphology and syntax occurs also within a language: awaiting the decision vs. waiting for the decision. A language may use a function word and an affix, e.g., Latin et and -que, for the same meaning, i.e., *and*.

2 Interpreting Determiner Noun Combination in the Hear Mode

A syntactic-semantic operation of the DBS hear mode combines a sentence start with a next word. There are three kinds of functor-argument[3] combination: (i) *cross-copying* between two proplets (connective $\times$), (ii) *absorption* of a content word into a function word (connective $\cup$), and (iii) *suspension* when an application has to be postponed because the word form to be connected with has not yet arrived (connective $\sim$).

The absorption of a content word into a function word may be shown by the following application of the hear mode operation **DET∪CN**:

2.1 PLURAL DETERMINER+NOUN COMPOSITION IN ENGLISH

DET∪CN

pattern level

noun: N_*n*		noun: α		noun: α
cat: CN′ NP		cat: CN	⇒	cat: NP
sem: Y		sem: Z		sem: Y Z
prn: K		prn:		prn: K

CN′ ε {nn′, sn′, pn′}, CN ε {sn, pn}, and NP ε {np, snp, pnp}.
If CN′ = sn′, then CN = sn and NP = snp. If CN′ = pn′, then CN = pn and NP = pnp.
If CN′ = nn′ and CN = sn, then NP = snp. If CN′ = nn′ and CN = pn, then NP = pnp.

content level

sur: The	⇑	sur: dogs		sur:
noun: n_1		noun: dog		noun: dog
cat: nn′ np		cat: pn		cat: pnp
sem: def		sem: pl		sem: def pl
fnc:		fnc:		fnc:
…		s …		…
prn: 12		prn:	⇓	prn: 12

The variable restriction If CN′ = sn′, then CN = sn and NP = snp ensures that a singular determiner must take a singular noun argument, e.g., a dog and every dog. The restriction If CN′ = pn′, then CN = pn and NP = pnp ensures that a plural determiner must take a plural noun argument, e.g., all dogs. In both, it is the determiner (functor) which determines the grammatical number of the result.

The restriction If CN′ = nn′ and CN = sn, then NP = snp ensures that a definite determiner and a singular noun result in a singular noun phrase, e.g., the dog. The restriction If CN′ = nn′ and CN = pn, then NP = pnp ensures that a definite determiner and a plural noun result in a plural noun phrase, e.g., the dogs. Here it is the noun (argument) which determines the grammatical number of the result.[4]

That a dog and the dog denotes a single individual and all dogs, the dogs as well as every dog denote plural sets is coded lexically as the sem value of

[3] For coordination see 7.

[4] The asymmetry between English indefinite and definite determiners regarding the source of grammatical number may be a problem for the *head-dependent* distinction (Osborne&Maxwell 2015) in Dependency Grammar (Mel′čuk 1988), but not for the semantically more neutral notions of functor (slot) and argument (filler).

the determiner proplet. The lexical properties of the English determiners and the variable restrictions of the hear mode operation 2.1 result in the following proplets:

2.2 PROPLETS OF a dog, the dog, every dog, all dogs, AND the dogs

sur: a dog	sur: the dog	sur: every dog	sur: all dogs	sur: the dogs
noun: dog	noun: dog	noun: dog	noun: dog	noun: dog
cat: snp	cat: snp	cat: snp	cat: pnp	cat: pnp
sem: indef sg	sem: def sg	sem: pl	sem: indef pl	sem: def pl
fnc:	fnc:	fnc:	fnc:	fnc:
mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:
prn: 12	prn: 12	prn: 12	prn: 12	prn: 12

The nouns a dog and the dog share the cat value snp and the sem value sg, but differ in the sem values indef and def. All dogs and the dogs share the cat value pnp and the sem value pl, but differ in the sem values indef and def. Every dog and all dogs share the sem value pl but differ in the cat values snp and pnp.

The German counterparts to the English examples in 2.2 are defined as follows:

2.3 PROPLETS OF ein Hund, der H., jeder H., alle Hunde, die Hunde

sur: ein Hund[5]	sur: der Hund	sur: jeder Hund	sur: alle Hunde	sur: die Hunde
noun: dog	noun: dog	noun: dog	noun: dog	noun: dog
cat: s3 m	cat: s3 m	cat: s3 m	cat: p3	cat: p3
sem: indef sg	sem: def sg	sem: pl	sem: indef pl	sem: def pl
fnc:	fnc:	fnc:	fnc:	fnc:
mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:
prn: 12	prn: 12	prn: 12	prn: 12	prn: 12

The definite article the in English has only one form for singular and plural, while the definite article in German has the forms der, die, das, des, dem, den for coding case, number, and gender.

Case is needed for filling the correct valency slot of the predicate. Number is needed for the nominative, as in der Hund bellte vs. die Hunde bellten. Gender is needed in the singular for coreference with a possible personal pronoun, as in die Frau...sie or ihr.[6] The differentiated determiner+noun combinations of German regarding case, number, and gender require variable restrictions which are substantially different from English and constitute a challenge for translating from English to German.

[5] In DBS hear mode derivations, the sur slot of sentence start proplets is empty, but elsewhere sur values may be used.

[6] In English, there is rudimentary grammatical gender of personal pronouns in indexical and anaphoric use (CLaTR 11), as when calling a ship a she.

3 Producing Determiner Noun Combination in the Speak Mode

As a minimal requirement for successful language communication, the content used as input to the speak mode and the content produced as output of the hear mode must be the same. To show a content *per se*, DBS uses two formats. One is a set of concatenated proplets as the output of the hear mode and used for storage in and retrieval from the agent's on-board database. The other is an equivalent semantic relations graph as the conceptual schema for guiding sequencing in the think-speak mode.

For example, the content of The dog barked. is defined as follows:

3.1 FORMAT 1: CONTENT OF The dog barked. AS A SET OF PROPLETS

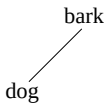
sur:	sur:
noun: **dog**	verb: **bark**
cat: snp	cat: #n′ decl
sem: def sg	sem: ind past
fnc: **bark**	arg: **dog**
mdr:	mdr:
nc:	nc:
pc:	pc:
prn: 14	nprn: 14

For purposes of storage and retrieval in the agent's content-addressable onboard database (A-memory), the proplets of a content must be order-free. They are connected by a shared prn value, here 14, and the semantic relations of structure, here subject/predicate, shown by the values in bold face.

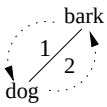
Navigating from the *dog* to the *bark* proplet is based on the address (bark 14) derived from the *dog* proplet. Navigating from the *bark* proplet back to the *dog* proplet is based on the address (dog 14) derived from the *bark* proplet. This is shown by the following graphical representation of the content:

3.2 FORMAT 2: CONTENT OF The dog barked. AS A GRAPH

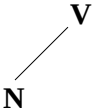
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1 2
The_dog barked_.
V/N N/V

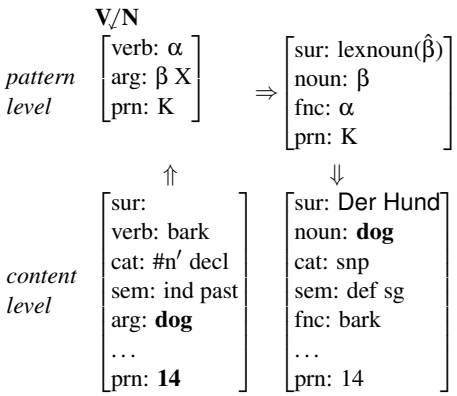
The semantic relation of subject/predicate is shown by the ⁄ lines in the graphs. There are four views on a content: the (i) SRG (semantic relations graph) connects the core *values* of the proplets; the (ii) signature connects the core *attributes*; the

(iii) NAG (numbered arcs graph) supplements the SRG with numbered arcs, which are used in the linear notation of the (iv) surface realization.

Language-dependent surfaces are realized from the *goal proplet* of a traversal step. Thus, The dog is realized from the goal proplet of arc 1, and barked_. from the goal proplet of arc 2. Both traversals are along the subject/predicate relation, but arc 1 is in the downward direction ↙ and arc 2 in the upward direction ↗.

While the operations of the hear mode take two proplets as input and produce one or two proplets as output, the navigation rules of the think-speak mode take one input proplet and retrieve one output proplet. Consider the think-speak mode operation **V↙N**, which produces the German surface Der Hund for The dog:

3.3 APPLYING THE THINK-SPEAK OPERATION V↙N



For retrieval of the output, the navigation step uses the address value (dog 14) of the input proplet *bark*. The surface is realized by the lexicalization rule lexnoun(β̂), which sits in the sur slot of the goal proplet. It uses the language-dependent variant Hund of the core value dog and the sem values def sg for realizing the German surface Der Hund. In nonlanguage navigation (e.g., activation, reasoning) the lex-rules are switched off.

4 Prepositional Phrases

Prepositional phrases consist of a preposition as the functor and a noun as the argument. The semantic kind of the noun is unrestricted in that it may be a concept, e.g., in the water, a name, e.g., in Paris, or an indexical, e.g., in here.

4.1 LEXICAL EXAMPLES OF PREPOSITIONS IN GERMAN

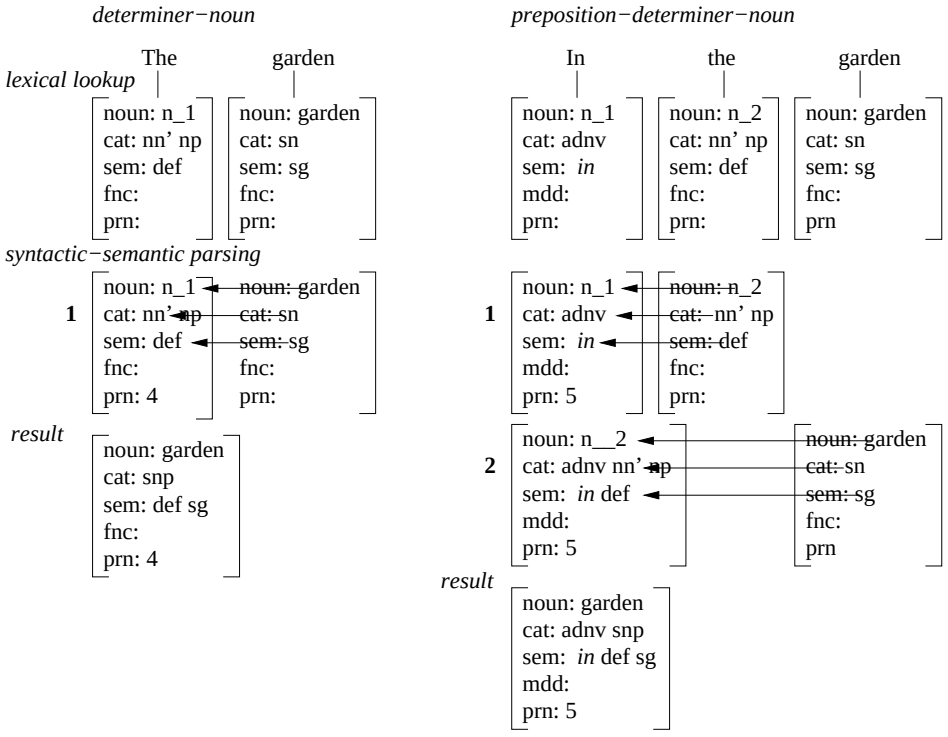
sur: auf	sur: über	sur: unter	sur: in	sur: von
noun: n_1	noun: n_1	noun: n_1	noun: n_1	noun: n_1
cat: adnv	cat: adnv	cat: adnv	cat: adnv	cat: adnv
sem: *on*	sem: *above*	sem: *below*	sem: *in*	sem: *of*
mdd:	mdd:	mdd:	mdd:	mdd:
mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:
prn:	prn:	prn:	prn:	prn:

The core value of a preposition is a substitution variable. Because prepositions like above, below, before, after, etc., are less abstract than the determiner sem values sg, pl, indef, and def, the language-independent counterpart of a preposition is stored as the initial value of the sem slot, using English place holders in *italics*, followed by the determiner values (4.2).

The argument of a preposition may be of unlimited complexity, e.g., in+the_little_red_house_by_the_lake. Like determiners, prepositions have the core attribute noun, which facilitates the time-linear processing of phrases as in Paris, in the city, in the big old city, in the big old city by the river, etc., with unlimited length.

If a preposition takes a determiner+noun composition (instead of a name or an indexical) as its argument, the time-linear hear mode derivation first combines the preposition and the determiner, e.g., in+the, and then adds the noun, e.g., in+the+garden. The following examples compare the time-linear hear mode derivations of a determiner+noun with a preposition+determiner+noun composition:

4.2 DIFFERENT FUNCTION WORD ABSORPTIONS (CLaTR 7.2.5)



Determiner and preposition proplets are alike in that their core attribute is noun. They differ in that determiners take lexical cat values like sn′ snp while the lexical cat value of prepositions is adnv, for adnominal or adverbial modification.

On the left, the determiner+noun derivation (i) substitutes the n_1 value of *the* with the core value of *garden*, (ii) cancels the nn′ position with the sn value, (iii)

replaces the np value with snp, (iv) adds the sg value to the sem attribute of the former *the* proplet, and (v) discards the *garden* proplet (NLC 13.3.3). The substitution-variable n_1 as the core value of the determiner is used for finding the determiner when it is separated from the noun argument by arbitrarily many modifiers, as in the large, beautiful ... garden.

On the right, the lexical preposition proplet introduces the continuation attribute mdd (modified). Step 1 of the time-linear preposition+determiner+noun derivation combines the two lexical function word proplets *in* and *the* into a single noun proplet.[7] Thereby the substitution variable n_1 in the preposition proplet is replaced with the incremented value n_2 of the determiner proplet, the def value of the determiner proplet is added to the preposition's sem slot, and the determiner proplet is discarded. Step 2 fuses the proplet resulting from step 1 with the lexical *garden* proplet: the n_2 substitution variable is replaced by the core value of the *garden* proplet, which is then discarded.

In linear notation, the adverbial use of an elementary adjective, as in Julia slept there, is represented as A|V, while the corresponding construction with a prepositional phrase, as in Julia slept in the garden., is represented as N|V. Graphically, the two constructions differ in the category node of the adverbial:

4.3 ELEMENTARY ADVERBIAL VS. PREPOSITIONAL PHRASE

In linear notation, the adnominal use of an elementary modifier is represented as A|N and the phrasal counterpart as N|N (CLaTR 7.3.6; NLC 7.3, 7.4).

5 Auxiliaries

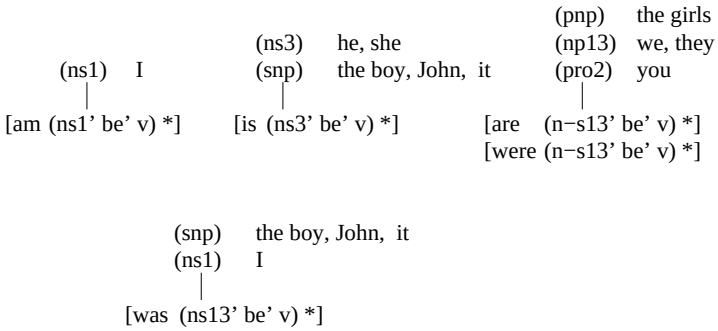
There are three kinds of auxiliaries in English, namely do, have, and be, and a larger number of modals, such as can, could, shall, should, will, would, may, might, and must, ought. In the present tense, the auxiliaries have special agreement, i.e., does, has, and is, while the modals do not.[8] Also, the auxiliaries have a progressive form, e.g., doing, having, and being, while the modals do not.

[7] DBS uses the cat values adn (adnominal) for elementary modifiers restricted to nouns, e.g., beautiful, adv (adverbial) for elementary modifiers restricted to verbs, e.g., beautifully, and adnv for elementary modifiers which may be applied equally to verbs or nouns, e.g., fast (CLaTR 3.5.5). Because prepositional phrases may be used adnominally or adverbially, their cat value is adnv as well. Elementary and phrasal adnvs differ in their core attribute, i.e., adj vs. noun.

[8] German auxiliaries and modals have several inflectional forms. For example, the German counterparts to have are habe, hast, hat, haben, habt, and to had are hatte, hattest, hatten, hattet.

The auxiliaries do and have have three finite forms do, does, did, and have, has, had, respectively, which are morphologically parallel to the forms of the main verbs and share their pattern of nominative agreement. The auxiliary be has the five finite forms am, is, are, was, and were, which require a special pattern for nominative agreement and may be described schematically as follows:

5.1 NOMINATIVE AGREEMENT OF THE AUXILIARY be (FoCL 17.3.1)



Finite forms of the auxiliaries combine with nonfinite forms of the main verbs into complex verb forms. The nonfinite forms are the infinitive, e.g., (to) give, the past participle, e.g., (has) given, and the present participle, e.g., (is) giving.

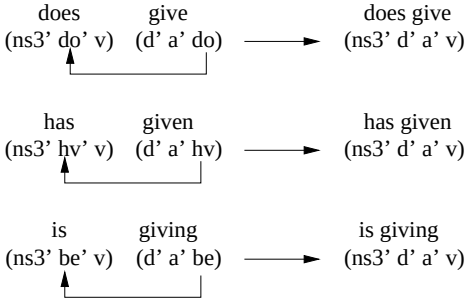
English infinitives (CLaTR 15.4) resemble the unmarked present tense form of the main verb, e.g., give. The past participle is marked morphologically in some irregular verbs, e.g., given, but usually coincides with the past tense of the main verb, e.g., worked. The present participle is always marked, as in giving.

The infinitive combines with the finite forms of do into the emphatic, e.g., does give or did give. The past participle combines with the finite forms of have into the present perfect, e.g., has given or had given. The present participle combines with the finite forms of be into the progressive, e.g., is giving and was giving.

The finite auxiliary forms all have variants with integrated negation, namely don't, doesn't, didn't, haven't, hasn't, hadn't, isn't, aren't, wasn't, and weren't. They have the same combinatorial properties as their unnegated counterparts.

The basic categorial structure of combining a finite auxiliary with a nonfinite main verb may be shown schematically as follows:

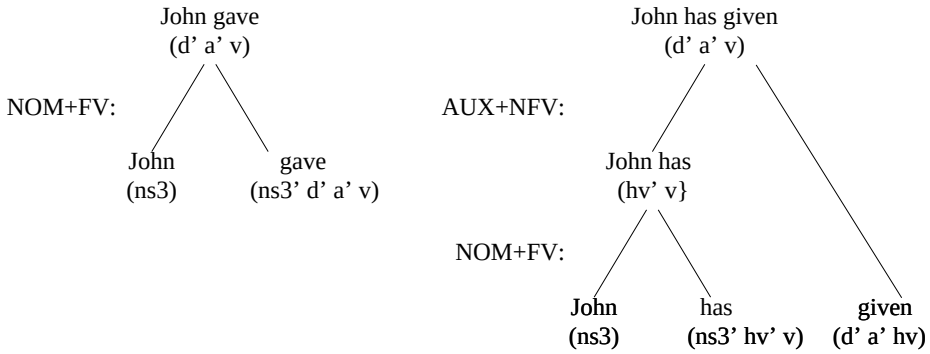
5.2 COMPLEX VERB FORMS OF ENGLISH (FoCL 17.3.2)



The nominative agrees with the finite auxiliary, which is why its valency position (here ns3′) is located in the category of the auxiliary. The oblique valency positions d′ and/or a′, in contrast, originate in the nonfinite main verb. That the above auxiliaries are finite is marked lexically by the presence of the v segment in their categories. That the main verb forms are nonfinite is marked lexically by the absence of the v segment. The identity-based agreement between the finite auxiliary and the nonfinite main verb form is expressed in the cat slot of the auxiliary by the segments do (for 'do'), hv (for 'have'), and be (for 'be'), respectively.

The combination of an auxiliary with a nonfinite main verb form, e.g., has given, results in a complex verb form which has the same properties in terms of nominative agreement and oblique valency positions as the corresponding finite form of the main verb in question, here gave:

5.3 DERIVING BASIC AND COMPLEX VERB FORM (FoCL[9] 17.3.3)



The two partial derivations end in the same state and may be continued the same.

In English, the auxiliary and its nonfinite main verb take the same adjacent surface positions in main and corresponding subclauses:

5.4 ADJACENT POSITION IN ENGLISH MAIN AND SUBCLAUSES

He *had read* the book.
After he *had read* the book, ...

He *did not do* the dishes.
Because he *did not do* the dishes, ...

He *is walking* the dog.
Because he *is walking* the dog, ...

The auxiliaries have take a past participle, do an infinitive, and be a progressive as their nonfinite counterpart.

German, in contrast, has only two auxiliaries, sein and haben, which combine with the past participle of the main verb. A finite auxiliary and a nonfinite transi-

[9]NEWCAT, CoL, and FoCL are still sign-based and valency positions are canceled by deletion (as in CG) instead of #-marking, but the derivation order is already bottom up, time-linear.

tive verb take different positions in corresponding main and subclauses: in main clauses, the nonfinite verb is in final position ('Distanzstellung'), but in a subordinate clause the nonfinite verb and the auxiliary are adjacent in final position ('Kontaktstellung'):

5.5 COMPARING 'DISTANZ' AND 'KONTAKT' POSITION IN GERMAN

Er *hat* das Buch *gelesen*.
Nachdem er das Buch *gelesen hat*, ...

Er *ist* zur Schule *gelaufen.*
Weil er zur Schule *gelaufen ist*, ...

Er *soll* die Teller *spülen*.
Weil er die Teller *spülen soll*, ...

'Distanzstellung' in German main clauses is known as 'Satzklammer' (sentence brace). German auxiliaries combine uniformly with the past participle of the main verb, while modals combine with the infinitive, as shown by the third example with *sollen*.

6 Subordinating Conjunctions

Examples of subclauses are (i) clausal subjects and objects using, e.g., that, (ii) clausal adnominals with a subject or object gap, using, e.g., who, and (iii) clausal modification using, e.g., when, as their subordinating conjunction. As function words, subordinating conjunctions use a substitution variable as their core value.

6.1 LEXICAL SUBORDINATING CONJUNCTIONS

sur: that	sur: who	sur: when
verb: v_1	verb: v_1	verb: v_1
cat:	cat:	cat:
sem: *that*	sem: *who*	sem: *when*
arg:	arg: ∅	arg:
fnc:	mdd:	mdd:
mdr:	mdr:	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn: 14	prn: 15	prn: 16

The proplets of subordinating conjunctions are special in that they have 10 attributes instead of the standard 9. For example, the additional fnc attribute in the *that* proplet is normally used for connecting an elementary or phrasal subject (3.1) or object to the predicate, but needed in subject and object clauses for the same purpose. The mdr attributes are still needed for examples like That John ate the cookie *slowly* surprised Mary.

The following examples have been analyzed in TExer in full declarative detail, which is canonized as the seven to-do's of DBS (6.3):

6.2 THE SUB-CLAUSE EXAMPLES ANALYZED IN TEXER

1. *clausal subject* (TExer 2.5)
   That Fido barked amused Mary.
2. *clausal object* (TExer 2.6)
   Mary heard that Fido barked.
3. *Clausal adnominal modifier with subject gap* (TExer 3.3)
   The dog which saw Mary barked.
4. *Clausal adnominal modifier with object gap* (TExer 3.4)
   The dog which Mary saw barked.
5. *Clausal adverbial modification* (TExer 3.5)
   When Fido barked Mary laughed.

The seven **To-do**s are defined in TExer 1.5.2 as follows:

6.3 THE **To-do**'S OF BUILDING A DBS GRAMMAR

1. **<to-do 1>**
   Definition of the content for an example surface
2. **<to-do 2>**
   Graphical hear mode derivation of the content
3. **<to-do 3>**
   Complete sequence of explicit hear mode operation applications
4. **<to-do 4>**
   Canonical DBS graph analysis underlying production
5. **<to-do 5>**
   List of speak mode operation names with associated surface realizations
6. **<to-do 6>**
   Complete sequence of explicit speak mode operation applications
7. **<to-do 7>**
   Summary of the system extension and comparison of the hear and speak mode operation applications

English and German are alike in that the grammatical roles of clausal *arguments* as subject, e.g., That Fido barked amused Mary, and as object, e.g., Mary heard that Fido barked, are encoded by word order and the choice of the higher verb. They differ in clausal *adnominals*: English encodes the role as subject, e.g., man who saw Mary, and as object, e.g., man who[10] Mary saw, by word order, but German by means of morphology: der Mann *der* Maria sah (subject) vs. der

[10] With optional use of whom (morphological relict), the word order difference remains.

Mann *den* Maria sah (object). Variation in clausal *modification* is similar in English and German in that it relies on different conjunctions such as when, since, while (temporal), because (reason), where (locational), into (directional), etc.

7 Coordinating Conjunctions

The functor-argument relations subject/predicate, object\predicate, and modifier|modified are encoded by the values of the noun, fnc, verb, arg, mdr, and mdd attributes. The conjunct−conjunct relations, in contrast, are encoded by the values of the nc (next conjunct) and pc (previous conjunct) attributes. Function words of coordination are and, or, but. In the medium of writing, DBS uses the interpunctuation signs **.**, **?**, and **!** for extrapropositional conjunction (Ballmer 1978).

Intrapropositionally, conjuncts must be grammatically similar (Bruening and Al Khalaf 2020), while no such constraint holds for extrapropositional coordination: declaratives may follow interrogatives and imperatives, imperatives may follow declaratives and interrogatives, and interrogatives may follow imperatives and declaratives. Intra- and extrapropositional coordination differ also in that intrapropositional coordination connects conjuncts bidirectionally by cross-copying, while extrapropositional coordination is unidirectional in the direction of time and uses inferencing for occasional backward traversal when rhetorically desired.

In running text, unidirectional extrapropositional forward coordination based on interpunctuation signs may continue without limit; for a minimal example in complete declarative detail see TExer 2.1.5–2.1.19. For intrapropositional coordination see TExer 3.6.

8 Conclusion

In a well-designed software solution, computer scientists distinguish (i) the *declarative specification* and (ii) the *procedural implementation*. The declarative specification presents the conceptual aspect: it must be easily read by humans and at the same time easily implemented in a programming language of choice. This includes the definition of input and output, the functional flow, the abstract data structure, the abstract operation schema, etc., in short, the *necessary* properties of the software solution.

A declarative specification may have an open number of procedural implementations which differ in *accidental* properties, i.e., properties inherent in different programming languages and programming styles. A procedural implementation is not only needed practically for using the software solution in applications, but also theoretically as the method of *verifying* the declarative specification.

A topic in computational linguistics well-suited for demonstrating the descriptive power of a declarative specification is the morpho-syntactic mechanisms of syntactic-semantic composition, which natural language controls with a precise mix of (i) function words, (ii) morphology, and (iii) word order. In this paper, it is

demonstrated with detailed declarative specifications of concrete constructions in classical Latin, English, and German.

Bibliography

Ballmer, T. (1978) *Logical Grammar*, Amsterdam: North Holland

Bruening, B., and E. Al Khalaf (2020) "Category mismatches in coordination revisited," *Linguistic Inquiry*, Vol. 31.1:1–36

CC = Hausser, R. (2019) *Computational Cognition: Integrated DBS Software Design for Data-Driven Cognitive Processing*, pp. i–xii, 1–237, `lagrammar.net`

CoL = Hausser, R. (1989) *Computation of Language*, Springer

CLaTR = Hausser R. (2011) *Computational Linguistics and Talking Robots; Processing Content in DBS*, pp. 286. Springer (preprint 2nd ed. at `lagrammar.net`)

FoCL = Hausser R. (1999) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language, 3rd ed. 2014*, pp. 518. Springer

Marty, A. (1918) *Untersuchungen zur Grundlegung der allgemeinen Grammatik und Sprachphilosophie*. Niemeyer, Halle an der Saale

Mel′čuk, I. (1988) *Dependency Syntax: Theory and Practice*, Albany: State University of New York Press

Osborne, T., and D. Maxwell (2015) "A Historical Overview of the Status of Function Words in Dependency Grammar," *Proceedings of the Third International Conference on Dependency Linguistics,* 241–250, Sweden: Uppsala University