PVP Fronting in German and Syntax/Semantics Interface

Hiroyuki IZUO

**Keywords:** German word order, incomplete category, split constructions, categorial semantics, incremental parsing, prediction

**Abstract:** This paper attempts to provide a model of the process of interpretation along with the left-to-right recognition process. Data of left periphery phenomena of German sentences include the fronted verbal complex which consists of a main verb and a part of its complements. This topicalized verbal complex is called fronted partial verb phrases (PVPs). For the purpose of parsing the left periphery of the sentence, the paper proposes that the grammar formalism should model the dynamics of language processing. For this purpose, this paper explains the fronted PVP by using a device of incremental processing realized by prediction rules. Syntax and semantics of Categorial Grammar (CG) will be applied to model the dynamic processing. The prediction mechanism which is proposed in this paper and constructed by using the syntax/semantics interface in the CG-style is a good device to capture how utterances in the sentence initial position give rise to meanings incrementally. By using the syntax/ semantics-interface, the ill-formedness of the incomplete PVP concerning split constructions can also be explored.
1 Introduction

In the light of the semantic characterization of left periphery phenomena of German sentences, this article investigates split constructions related to the VP topicalization. Verb phrases consisting of a head verb and some of its arguments are called partial verb phrases (PVPs). German partial-VPs show incomplete categories and a relatively free selection of complements. Fronted PVPs can be observed as an unsaturated verb phrase in the Vorfeld of a verb-second (V2) sentence. The Vorfeld is the field before the finite verb in V2 sentences. The fronted verb phrase in (1) is partial, because it consists of a head verb *geschenkt* (‘presented’) and one of its complements *seiner Mutter* (‘his mother’), although an accusative noun *ein Buch* (‘a book’) appears in the Mittelfeld. The Mittelfeld of V2 sentences is the field between the finite verb and the right-sentence bracket where non-finite verbal elements can appear.

(1) Seiner Mutter geschenkt hat Hans ein Buch.
    to-his Mother,dat presented has Hans nom a book acc
    ‘He has presented a book to his mother.’

Why is word order such as PVP fronting used in German V2 sentences? The central idea proposed in this paper is that the fronted PVPs in German cause an economical and efficient computation of syntactic/semantic structures of sentences. During human speech recognition, partial interpretations can be extracted rapidly, even before phrasal constituents are completed, i.e., meaning can be accumulated before the end of a sentence. This paper will argue that fronted PVPs provide syntactic and semantic information for the efficient and economical recognition process. The information of each input string in the German Vorfeld is accumulated from
left to right, and specifies the category of strings that follow. In this paper, German PVP topicalization and related split constructions will be scanned from left to right in a piecemeal fashion, and interpreted incrementally. Research argues that this incremental processing can be supported by each predicting process, which makes it easier to select possible lexical entries. Prediction rules will be constructed by applying the functor-argument structure of the Categorial Grammar (CG). For example, the category of noun phrases can be treated as functors, and applied not only to the verbal elements, but also to other nominal elements. Using those functor categories, left-to-right incremental parsing can be handled by the algorithm which accumulates syntactic/semantic information as the sentence is parsed from left to right.

The paper proceeds as follows: first, some basic observations concerning the German data of split constructions related to PVPs are described. Next, CG’s syntax/semantics interface will be introduced to give a uniform treatment for the PVP fronting in German. For the purpose of providing model-theoretic interpretations of the split construction related to PVPs, a prediction mechanism will be added to combinatory rules. Finally, combinatory rules and prediction rules based on CG will be applied to capture the characteristics of the split construction related to PVPs. The paper concludes with an illustration of how the data of split constructions can be accounted for explicitly by the difference in the efficiency of the predictive recognition related to the head and its modifier.

2 PVP fronting and non-head fronting

This section describes some basic observations concerning the German data of the split constructions related to the fronted PVP. German V2 sentences can involve verb-first structures which are derived by fronting partial constituents. Not only non-subject constituents, but even double infinitives
can appear in the initial position of V2 sentences, as shown in (2).\textsuperscript{2} The double infinitive in (2) which comprises a base form head verb (H) *lesen* (‘read’) and a base form auxiliary (AUX) *können* (‘be-able-to’), is fronted. However, its complement *das Buch* appears in the Mittelfeld. The question is why the fronted partial VP *lesen können* is allowed.

(2) Lesen können wird er das Buch.

read be-able-to will he the book

‘He will be able to read the book.’

The base form head verb alone can be fronted by itself separated from the auxiliary which appears in the sentence-final position. As exemplified in (3a) and (3b), the H-AUX split in the double infinitive construction is also possible. However, the AUX without the H cannot be fronted by itself, as illustrated in (3c) and (3d). The problem of the non-head fronting will be treated in section 4.4.

(3) a. Lesen wird er das Buch können.

read will he the book be-able-to

‘He will be able to read the book.’

b. Das Buch lesen wird er können

c. *Können wird er das Buch lesen.

d. *Das Buch können wird er lesen.

In the examples of (4), the noun phrase *ein Buch über Syntax* (‘a book about syntax’) can be a complement of the transitive verb *ausgeliehen* (‘borrowed’). However, in the V2 sentence (4a), the PP *über Syntax* is fronted independently of its nominal head *ein Buch*. Such NP-PP split constructions can occur also in the PVP topicalization. De Kuthy and Meurers (2001) point out that the embedded constituent *über Syntax* in (4b) is missing from
the nominal complement *ein Buch* of the fronted verbal head *schreiben* (*‘write’*).³

(4) a. Über Syntax hat Hans sich ein Buch ausgeliehen.  
   about Syntax has Hans himself a book borrowed  
   ‘Hans borrowed a book about syntax.’

b. Ein Buch schreiben will niemand über Syntax.⁴  
   a book write want-to no-one about syntax  
   ‘Nobody wants to write a book about syntax.’

The fronted partial VP including a NP-PP split is also possible for verbs like *ausleihen* which require an obligatory NP and no PP complement, as shown in (5a). With respect to the fronted PVP and the NP-PP split construction, the non-head fronting illustrated in (5b) is impossible. Why can the non-head PP such as *über Syntax* in (5b) not be fronted separated from its head noun? Although (5c) has the same syntactic structure as that of (5b), (5c) is grammatical. This paper will explore what causes the difference between (5b) and (5c).

(5) a. Ein Buch ausleihen will er über Syntax.  
   a book borrow wants-to he about syntax  
   b. *Über Syntax ausleihen will er ein Buch.*⁵  
   about syntax borrow wants-to he a book  
   c. Über Syntax lesen will er ein Buch.  
   about syntax read wants-to he a book

   In the GB framework, G. Müller (1996) argued that the fronted element of the incomplete category contains a trace of the argument and the trace occurs by the movement of a constituent. The constituent by which the trace in the fronted element is bound is adjoined to the tree that occurs
configurationally lower. In the HPSG framework, Hinrichs and Nakazawa (1989, 1994) and Pollard (1996) discussed that a trace is not included in constructions in which the verbal complement is fronted. Their idea can be summarized as follows: When a verbal complement is fronted and combines with its verbal governor, the resulting construction must realize the unsaturated subcategorization requirements of its verbal complement. This framework is expanded by De Kuthy and Meurers (2001), and De Kuthy (2002).

In contrast to those syntactic analyses, this paper will adopt the categorial semantics and its combinatory rules for the purpose of constructing the apparatus which can predict and accumulate interpretations in the course of the incremental parsing of sentences. Using the syntax/semantics interface in the CG-style, the difference in grammaticality between (5b) and (5c) will be explored. Typed structures of the meaning which will be proposed in this paper show the different semantic structures with respect to their split constructions, related to the PVP topicalization. Similar difference in grammaticality can be observed between (6a) and (6b) which contain extraposition. In (6a), the PP über Syntax, which is an embedded constituent of a head noun ein Buch, can be extraposed. However, in (6b), the head noun ein Buch is separated from the embedded PP. The extraposition of the head NP without the embedded PP is ungrammatical.

(6) a. Er hat ein Buch ausgeliehen über Syntax.

b. *Er hat über Syntax ausgeliehen ein Buch.

German word order phenomena such as fronted PVPs and split constructions are dependent on being integrated in the semantics related to focus. Although in the framework of GB (G. Müller (1996), Grewendorf and Sabel (1994), Haider (1990)) and HPSG (De Kuthy and Meurers (2001), Hinrichs and Nakazawa (1989, 1994), Nerbonne (1994)), the related
problems are discussed mainly related to their syntactic properties, the issue of formal semantic properties of the split construction related to PVPs has not received much try in literature. In this paper, such word order phenomena will be formalized from the model theoretic semantics point of view. Using the incremental parsing based on the prediction and the model theoretic semantics, this paper will clarify the difference in grammaticality of extraposition between (6a) and (6b).

In the next section, the framework of the syntax/semantics interface of CG will be introduced and its semantic rules will be adopted to parse the possible meaning incrementally. With this device, expressions in the left periphery of clauses and the split construction can be incrementally interpreted from left to right. For the purpose of adopting the mechanism which licenses fronted constituents to predict the following meanings, prediction rules will be constructed and applied to give reasons why PVPs occur in the sentence-initial topic position and why some split constructions related to PVPs are not allowed.

3 Combinatory rules and PVP fronting

This section proposes to apply the syntax/semantics interface of CG to describe incremental interpretation, and to accumulate the meaning recognized word by word. For the purpose of incremental interpretation, prediction rules fashioned after the combinatory rules will be proposed.

The organization of this section is as follows: in section 3.1, combinatory rules of classical categorial grammars will be introduced, including the analyses of Steedman (1988, 1996), Buszkowski (1988), and Moortgat (1988). Combinatory rules are equipped with a syntax/semantics interface, to map surface forms onto interpretable meaning representations. In section 3.2, combinatory rules will be applied to represent the head-complement linear order across related categories. A framework of syntax/semantics
interface is applied to deal with a grammar fragment for representing the treatment for free word order. In section 3.3, combinatory rules will be applied to the incremental semantic parsing of German PVP fronting. German case-marked NPs will be treated in the frame of the syntax/semantics interface realized by CG. In section 3.4, categorial semantics concerned with PVPs and auxiliaries will be constructed. Auxiliaries play an important role in the construction of the PVP fronting. The framework proposed in this section will be applied in section 4 to analyze the split constructions related to the fronted PVPs. Finally, in section 3.5, prediction rules will be constructed to license fronted constituents to predict the semantic structure which follow. For this purpose, the functor-argument scheme of CG will be applied to specify a set of possible categories which can occur.

3.1 The syntax/semantics interface in combinatory rules

In CGs, syntactic categories and semantic types can be defined recursively. They are of two types: functor categories and argument categories. Functor-argument relations serve to combine categories while preserving a transparent relation between syntax and semantics. The functor category written with a directional slash notation such as X/Y represents a function looking for an argument category Y on its right and resulting in the value category X, where X and Y are variables over the set of categories. By using another slash notation, X\Y looks for Y on its left and results in X. Application and composition are rules which combine two categories. In CG (Moortgat 1988, Steedman 1996), the combinatory rules of syntactic categories correspond with those of semantic structures. The functor-argument scheme of syntactic categories corresponds to the functor-argument scheme realized by the typed lambda calculus, which can also be used to refer to the objects in model-theoretic interpretation. In (7), each syntactic rule is paired with a
rule of semantic interpretation. The notation X:f means that a syntactic
category X has an interpretation f. If X:f and Y:g is reduced to Z:v, this
reduction is represented as X:f Y:g ˠ Z:v.

   Backward Application Y:a X\Y:f  X:f(a).

b. Forward Composition X/Y:f Y/Z:g  X/Z:λv.f(g(v)), where Z:v.
   Backward Composition Y\Z:g X\Y:f  X:f(g(v)), where Z:v.

c. Associativity (XY)/Z:f  (X/Z)\Y:λv1 λv2.f(v2(v1)), where Y:v1, Z:v2.

d. Type-raising X:a Y/(Y\X):λv.v(a), where Y\X:v.

e. Division X/Y:f (X/Z)/(Y/Z):λv1 λv2.f(v1(v2)), where Y/Z:v1, Z:v2.
   X\Y:f (X\Z)/(Y\Z):λv1 λv2.f(v1(v2)), where Y\Z:v1, Z:v2.

3.2 Combinatory rules and head-complement relations

The combinatory rules based on the functor-argument scheme are applied to
construct head-complement rules and complement-composition rules. If a
syntactic category C is paired with a semantic structure c, and C works as
functor category X/Y or X\Y, where X:v1 and Y:v2, the semantic structure c
is a function from v2 to v1. If T is a mapping from categories to types, the
type of c is defined as follows, where the vertical slash “|” used in the
functor category indicates that the position of its argument is unspecified.

(8) If C is X|Y and C : c, the type of c is <T(Y),T(X)>, where c is λv2.v1,
    and <T(Y),T(X)> is a function from T(Y) to T(X).

The syntactic category IV(=S|NPnom) of intransitive verbs corresponds to
the semantic type T(IV), namely <e,t> which was defined by Montague
(1974) as a function from entities e to truth value t, where S for sentences,
NPnom for nominative noun phrases, N for nouns. The meaning of German
intransitive verb *schwimmen* (‘swim’) can be defined as a set of individuals who swim. If an individual *a* swims, a proposition like *schwimmen*(‘*a*) is true, where *schwimmen* is a constant of type `<e,t>` and *a* is a constant of type `e`. Using the Lambda abstractor, the semantic structure of *schwimmen* is represented as *λx*.schwimmen*(‘x) with respect to the variable *x* of type `e`. German transitive verb (TV) like *lesen* (‘read’) combines with an accusative noun phrase (NPacc) to become IV, its semantic type T(TV) is therefore `<e,<e,t>>`. The meaning of *lesen* is realized as *λx2 λx1*.lesen*(‘x1,x2). As for the German ditransitive verb (DTV) like *geben* (‘give’), the combination of DTV with a dative noun phrase (NPdat) plays a role of TV, which looks for an NPacc further. Therefore, the semantic type T(DTV) is `<e,<e,<e,t>>`. The meaning of *geben* is realized as *λx3 λx2 λx1*.geben*(‘x1,x2,x3). Generally, the head verb Hn takes n complements and yields a sentence S as its value. Therefore, each head verb is represented as a functor category which specifies a set of arguments that it subcategorizes for, as shown in (9a). Cn is an argument category of Hn. For example, the ditransitive head verb (HDTV) has a functor form, as illustrated in (9b).

(9) a. If Hn = S|C1|...|Cn, then T(Hn)= <cn,...,<c1,t>...>, where T(Cn)=cn.

   b. If HDTV = S|NPnom|NPacc|NPdat, then T(HDTV)=<e,<e,<e,t>>>.

Argument categories are determined by subcategorization requirements of the head verb. This functor-argument relation creates a head-complement scheme in the CG-style, as indicated in (10a), where the variable vn has a type of head verb which takes n complements inclusive of the subject, and hn is the n place relation corresponding to a concrete head verb, for example, h1=swimmen’ in λx*.schwimmen*’(‘x), or h3=geben’ in λx3 λx2 λx1.gaben*(‘x1,x2,x3), etc. The composition rule can be applied to combine two functor-categories, as indicated in (10b), where Vn is a category of n
place verb phrases, for example, IV is represented as V¹, TV as V², and DTV as V³ corresponding to the number of arguments, and semantic structures are abbreviated to f and g. The tree (10c-1) is constructed by the head-complement rule, and the tree (10c-2) is an example of the complement-composition rule, where V¹|V² and V²|V³ correspond to NP_{acc} and NP_{dat} respectively. Both of these trees are binary, and the binary structure has the potentiality to predict the sentence structure incrementally, which will be explored later.

(10) Head-Complement Rule and Complement-Composition Rule

a. S[C¹]...[C_i]...[C_n] : v^n, C_i : c_i □ S[C¹]...[C_n] : v^n(ci)
   where 1 ≤ i ≤ n and v^n = λx_n...λx_i...λx_1.h^n(x_1,...,x_i,...,x_n) and v^n(ci) = λx_n...λx_1.h^n(x_1,...,c_i,...,x_n).

b. V^{n-1}|V^n : f, V^n|V^{n+1} : g □ V^{n-1}|V^{n+1} : λv^{n+1}.f(g(v^{n+1})), where V^{n+1} : v^{n+1}

c-1. S[C¹]...[C_n] : v^n(ci) c-2. V¹|V³ : λv³.f(g(v³))

3.3 Type raising of NP and PVP fronting

In this section, combinatory rules will be applied to construct the meanings of the fronted PVP incrementally. Above all, the type raising of the case-marked NP is useful to compose the meanings of two NPs directly. If type-raising rules are applied, case-marked nouns in German can be converted into functors over the verbs, and can be combined by composition rules. In Montague (1974), the semantic type of the noun phrase is defined as <<e,t>,t>, if the intension is ignored. This type corresponds to the functor category V⁰|(V⁰|NP_{nom}), namely V⁰|V¹, where V⁰ is a sentence category and indicates that all complements are saturated. In this paper, other case-marked NPs are also defined as functor categories, which do not correspond
to Montague’s type $<\langle e,t\rangle,t>$, as shown in the examples in (11).

(11) a. $\text{NP}_{\text{acc}}$ is type-raised to $V^1|(V^1|\text{NP}_{\text{acc}})$, namely to $V^1|V^2$.

b. $\text{NP}_{\text{dat}}$ is type-raised to $V^2|(V^2|\text{NP}_{\text{dat}})$, namely to $V^2|V^3$.

Because the type-raising of the case-marked NPs is activated by n-place verb $v^n$, each interpretation of the case-marked NPs obtains a functor type. Montague (1974) proposed that the noun phrase is translated into a function from one place predicate of type $<e,t>$ to a proposition of type $t$. This paper applies his translation to the nominative NP. In (12a), the nominative noun phrase \textit{der Mann} (‘the man’) is translated into a function from one place verb $v^1$ to a sentence, where the common noun \textit{Mann} is translated into a constant \textit{man}’ of type $<e,t>$, if the intension is ignored. Except (12a), concerning the other case-marked NPs like $\text{NP}_{\text{acc}}$ and $\text{NP}_{\text{dat}}$, this paper proposes different translations from those of Montague (1974) and GPSG (Gazdar et al. 1985). As indicated in (12b), an accusative NP \textit{ein Buch} (‘a book’) is translated into a function from two place verb $v^2$ into one place verb. (12c) shows that a dative NP \textit{jedem Kind} (‘every child’) is translated into a function from three place verb $v^3$ into two place verb. In (12), ‘$\text{Phon} \Rightarrow \text{Syn}:\text{Sem}$’ indicates that an expression whose phonological form is $\text{Phon}$ has a syntactic category $\text{Syn}$ and a semantic structure $\text{Sem}$.

(12) a. $\text{der Mann}_{[\text{nom}]} \sqsubseteq V^0|V^1 : \lambda v^1. \quad x[ \sqsubseteq w[\text{man}’(w) \leftrightarrow w=x] \quad \sqsubseteq v^1(x)]$

b. $\text{ein Buch}_{[\text{acc}]} \sqsubseteq V^1|V^2 : \lambda v^2 \lambda x. \quad \sqsubseteq w[\text{buch}’(w) \sqsubseteq v^2(x,w)]$

c. $\text{jedem Kind}_{[\text{dat}]} \sqsubseteq V^2|V^3 : \lambda v^3 \lambda x_2 \lambda x_1. \quad \sqsubseteq w[\text{kind}’(w) \rightarrow v^3(x_1,x_2,w)]$

The following tree (13) demonstrates that the composition rule can be applied to the functor-types of two case-marked noun phrases, namely $<dtv, tv>$ of $\text{NP}_{\text{dat}}$ and $<tv, vp>$ of $\text{NP}_{\text{acc}}$, and the semantic structure of a complex noun \textit{jedem kind ein Buch} is constructed, where $iv$, $vp$, $tv$ and $dtv$
are abbreviations of $T(IV)$, $T(VP)$, $T(TV)$ and $T(DTV)$ respectively. The semantic type of this complex noun is a functor type $<dtv, vp>$, which is applied to its argument dtv, namely to the type of the expression $geben$. Then the semantic structure of $jedem kind ein Buch geben$ obtains a type $vp$, as illustrated in (13).

(13)

$vp$

$jedem Kind ein Buch geben$  $\xrightarrow{V^i} f_i(f_4) = \lambda x_1 \cdot w_2[buch'(w_2) \quad \square \quad w_1[kind'(w_1) \rightarrow geben'(x_1,w_2,w_1)]]$

$dtv$

$jedem Kind ein Buch$  $\xrightarrow{\text{V}^i[V^i]} V^i : \lambda v^i.f_2(f_1(v^i)) = geben$  $\xrightarrow{\text{V}^3} : \lambda v^3 \lambda x_1 \lambda w_2[buch'(w_2) \quad \square \quad w_1[kind'(w_1) \rightarrow v^3(x_1,w_2,w_1)]]=f_3 \lambda y_3 \lambda y_2 \lambda y_1.geben'(y_1,y_2,y_3) =f_4$

$<dtv, tv>$

$jedem Kind$  $\xrightarrow{\text{V}^i[V^i]} V^i : ein Buch$  $\xrightarrow{\text{tv}, vp}$

$\lambda v^i \lambda x_2 \lambda \cdot \quad w_1[kind'(w_1) \rightarrow v^i(x_1,x_2,w_1)]=f_1$  $\lambda v^2 \lambda x. \quad w_2[buch'(w_2) \square v^2(x,w_2)]=f_2$

3.4 Combinatory rules for PVP fronting and auxiliaries

If a head verb in German V2 sentences has a feature $bse$ (base form) or a $prt$ (past participle form) and precedes a finite auxiliary (AUX-finite), and if some of its complements are saturated in front of this head verb, this topicalized verbal complex is called PVP. If the finite auxiliary does not follow the fronted string of this incomplete category, this string cannot be a PVP. Therefore, in this section, semantic rules which operate on semantic structures of auxiliaries related to PVPs will be proposed.

If $V^m$ is a category of m place verbs, the category of the auxiliary verbs is given as $V^m[V^m] because of the argument inheritance by the auxiliary. If $V^m$ is $S|C_1|...|C_m$, the complex composed of $V^m$ and AUX should have the same
category S|C_1|...|C_m. The syntactic category of m place verbs, namely S|C_1|...|C_m, corresponds to the semantic structure \( \lambda x_m...\lambda x_1.h^n \) \((...x_1,...,x_m,...)\). In the arguments \(...,x_1,...,x_m,...\) of \( h^n \), there are \( n - m \) constants which correspond to saturated complements of the head verb. The semantic structure of the auxiliary verbs can be given as \( \lambda v^m\lambda x_m...\lambda x_1.aux(v^m(x_1,...,x_m)) \), because the basic semantic type of auxiliaries like \( haben \) (‘have’), \( werden \) (‘will’), \( können \) (‘can’) is defined as a sentence modifier of type \( <t,t> \), which can be divided further into \( <v^m,v^m> \). Note that \( aux \) should be replaced by a concrete constant of auxiliary like \( haben’, werden’, können’ \) of type \( <t,t> \). For example, the semantic structure of \( können \) can be defined as \( \lambda v^m\lambda x_m...\lambda x_1.können’(v^m(x_1,...,x_m)) \), where both \( v^m(x_1,...,x_m) \) and \( können’(v^m(x_1,...,x_m)) \) have the same type \( t \). The combinatory rule related to the auxiliary can be given as indicated in (14).

(14)  AUX : f    V^m : g    S|C_1|...|C_m : f(g),

where AUX = V^m|V^m and V^m = S|C_1|...|C_m,

\[
f = \lambda v^m\lambda x_m...\lambda x_1.aux(v^m(x_1,...,x_m)),
g = \lambda y_m...\lambda y_1.h^n(...y_1,...,y_m,...),
f(g) = \lambda x_m...\lambda x_1.aux(h^n(...x_1,...,x_m,...)).
\]

For example, if AUX is combined with a one place predicate \( V^1 \), AUX becomes \( V^1|V^1 \), which is equal to the category \( (S|NP_{nom})|IV \). If AUX is combined with a two place predicate \( V^2 \), AUX becomes \( V^2|V^2 \). By applying the rule (14), each semantic structure of the verbal complexes composed of a head-verb and an auxiliary can be represented as follows.

(15)  a. können ⇒ V^m|V^m : \( \lambda v^m\lambda x_m...\lambda x_1.können’(v^m(x_1,...,x_m)) \) = f

b. schwimmen □ V^1 : \( \lambda y.schwimmen’(y) \) = g1
lesen □ V^2 : \( \lambda y_2\lambda y_1.leisen’(y_1,y_2) \) = g2

c. schwimmen können □ V^1 : f(g1)= \( \lambda x_1.können’(schwimmen’(x_1)) \)
lesen können ⇒ \( V^2 : f(g2)= \lambda x_2 \lambda x_1. \text{können}'(\text{lesen}'(x_1, x_2)) \)

In order to construct combinatory rules for PVP, Izuo (2002) classified PVPs into two types, namely in PVP1 and PVP2. PVP1 is composed of a head verb \( H^n \) with a feature \( bse \) or \( prt \) and some members of its complements which precede the \( H^n \). PVP2 is a verbal complex composed of the PVP1 and a finite auxiliary (AUX-finite). PVP1 can be regarded as an inner PVP because of the base (or past participle) form of its head verb, and PVP2 as an outer PVP because of the finite auxiliary. Because of the argument inheritance caused by the auxiliary, PVP2 inherits the partially saturated complements from the PVP1. The infinite modal auxiliary (AUX-modal:bse) can also appear in the PVP1. The PVP1 without AUX-modal:bse is also possible. However, AUX-finite is inevitable to construct PVP2. With respect to the AUXs in PVP, AUX-modal:bse should be differentiated from the finite auxiliary embedding a double infinitive (AUX-double:finite). For example, in (2) *Lesen können wird er das Buch*, PVP1 is composed of the double infinitive which consists of *lesen* and *können*. For the purpose of constructing PVP1, the combinatory rule is defined as follows.

(16) Combinatory rules for PVP1

a. \( C^* : c^* \quad H^n : H^n \quad \Box \quad \text{PVP1} : \text{pvp1} \)

b. \( \text{PVP1} : \text{pvp1} \quad \text{AUX-modal:bse} : \text{AUX}\text{modal} \quad \Box \quad \text{PVP1-double} : \text{pvp1}\text{double} \)

In (16a), \( C^* \) is a syntactic category of the fronted expressions composed of \( k \) complements of \( H^n \). \( H^n \) is \( S|C_1|...|C_n \), and \( H^n = \lambda x_n...\lambda x_1.h^n(x_1,...,x_n) \). PVP1 is an \( m \) place verb phrase \( V^m \), where \( m=n-k \). Therefore, \( \text{pvp1} \) is an \( m \) place functor \( \lambda x_j...\lambda x_i.h^n(...,x_i,...,x_j,...) \), where \( 1 \sqsubseteq i \sqsubseteq j \sqsubseteq n \). If an \( n \) place head verb is saturated by \( k \) complements \( C^* \) and if variables in \( ...,x_i,...,x_j,... \) represent \( m \) unsaturated complements, then \( v^m \) has a type of \( m \) place verb, i.e., \( \lambda x_j...\lambda x_i.h^n(...,x_i,...,x_j,...) \).\(^8\) \( AUX\text{modal} \) is \( \lambda v^m \lambda x_m...\lambda x_1.aux\text{-modal}(v^m(x_1,...,x_m)) \).
PVP1\textit{double} is also an m place verb phrase V\textsuperscript{m}. Therefore, \textit{pvp1\textit{double}} is also an m place functor \(\lambda x_1\ldots\lambda x_i.\text{aux}-\text{modal}(h^n(...,x_i,...,x_j,...))\). Because the verb on the right-hand side of PVP1 has a base (or past participle) form, PVP1 should be combined with a finite auxiliary (AUX-\textit{finite}) to make PVP2. The semantic structure of the finite auxiliary in PVP2 can be illustrated as follows.

(17) Semantic structure of the finite auxiliary in PVP2

\begin{enumerate}
\item PVP2 : \textit{pvp2}\\
\hspace{1em} PVP1 : \textit{pvp1} \quad \text{AUX-\textit{finite}} : \text{AUX-\textit{finite}}
\item AUX-\textit{finite} : \text{AUX-\textit{finite}}, \\
where AUX-\textit{finite} = V^n|V^n or AUX-\textit{finite} = PVP2 \setminus PVP1, and \\
AUX-\textit{finite} = \lambda y_1\ldots\lambda y_i.\text{aux}-\textit{finite}(v^n(x_1,...,x_m)).
\item PVP1 : \textit{pvp1}, \\
where PVP1 = S[C_1|...|C_j] and \textit{pvp1} = \lambda y_j\ldots\lambda y_i.h^n(...,y_i,...,y_j,...) or \textit{pvp1\textit{double}} = \lambda y_j\ldots\lambda y_i.\text{aux}-\textit{modal:bse}(h^n(...,y_i,...,y_j,...)).
\item PVP2 : \textit{pvp2}, \\
where PVP2 = AUX-\textit{finite}(PVP1) = S[C_1|...|C_j] = V^n and \textit{pvp2} = AUX-\textit{finite}(\textit{pvp1}), \\
\textit{pvp2} = \lambda x_m\ldots\lambda x_1.\text{aux}-\textit{finite}(h^n(...,x_1,...,x_m,...)) or \\
\textit{pvp2\textit{double}} = \lambda x_m\ldots\lambda x_1.\text{aux}-\textit{double:finite}(\text{aux}-\textit{modal:bse}(h^n(...,x_1,...,x_m,...))).
\end{enumerate}

(17a) represents a derivation tree of PVP. By applying a functor category AUX-\textit{finite} to PVP1, PVP2 is obtained. (17b) shows that the auxiliary can also be combined with PVP1 which has an incomplete category. (17c) shows that the n place head verb \(h^n\) in \textit{pvp1} has m unsaturated complements C\textsubscript{i}...C\textsubscript{j}, and can be embedded in \textit{aux}-\textit{modal:bse}. (17d) shows the argument inheritance from PVP1 to PVP2 because of the unification of meaning variables \(x_1,...,x_m\) and \(y_i,...,y_j\) in the process of lambda-conversion. Therefore, semantic features of the arguments of \textit{pvp2} are inherited from \textit{pvp1}. The relation of auxiliaries to a head verb is reflected in each semantic structure explicitly.
3.5 Prediction mechanism and PVP

Each functor category can predict possible categories which follow it. If the functor category A/B is applied to an argument category B, then a value A is obtained. This means that the functor category A/B predicts not only the existence of an argument expression of category B, but also the existence of a value expression of category A. This prediction mechanism of functor categories makes it possible to parse sentences from left to right incrementally. Based on this idea, the following prediction rules are proposed as a variant of the combinatory rules.

(18) Prediction rules

a. $S/C_1/.../C_n : \lambda c_n ... \lambda c_1. h^n(c_1, ..., c_n)$ predicts on its right the existence of a set of arguments, namely $\{C_1 : c_1, ..., C_n : c_n\}$. The existence of its value can also be predicted, i.e., $S : h^n(c_1, ..., c_n)$ is also predicted.

b. $C_2/C_3 : f$ and $C_3/C_1 : g$ predict the existence of an argument $C_1 : c_1$ on its right, and a value $C_2 : f(g(c_1))$, because a functor $C_2/C_1 : \lambda c_1. f(g(c_1))$ is composed.

c. $C_1 : c_1$ predicts a functor $C_2 \setminus C_1 : f$ on its right, and its value $C_2 : f(c_1)$.

d. $S|C_1 : f_1$ can predict not only a set $\{C_1, S\}$, but also a set of sets $\{\{C_1|C_2, S|C_2\}, ..., \{...(C_1|C_2)|...|C_n\}, \{...(S|C_2)|...|C_n\}\}$, because, in general, $S|C_1$ can be divided into $(S|C_2)\|(C_1|C_2) : \lambda v_1 \lambda v_2. f_1(v_1(v_2))$, which can be divided further into $\{(...(S|C_2)|...|C_n)\|(...(C_1|C_2)|...|C_n)\) : \lambda v_{2n-1} \lambda v_{2n}. f_n(v_{2n-1}(v_{2n})) = f_{n+1}$, where the semantic structures $v_1, v_2, v_{2n-1}$, and $v_{2n}$ are defined in $C_1|C_2 : v_1$, $C_2 : v_2$, $((C_1|C_2)|...|C_n) : v_{2n-1}$, and $C_n : v_{2n}$ respectively.

(18a) is a general prediction rule in CG style. (18b) indicates that (18a) can be applied to the result category of the composition. (18c) can be applied to
PVP, because PVP1 predicts AUX-finite (=PVP2\PVP1). (18d) shows that a functor category can predict possible categories in the lexical rules, if the division is applied. For example, \(\text{NP}_{\text{nom}}\) can predict all verbs in the lexical rules, because it has a functor category \(S|\text{IV}\), and if this is divided, for example, by \(\text{NP}_{\text{acc}}\), then the divided category \((S|\text{NP}_{\text{acc}})|(\text{IV}|\text{NP}_{\text{acc}})\) predicts \(\text{IV}|\text{NP}_{\text{acc}}\), namely TV. \((S|\text{NP}_{\text{acc}})|\text{TV}\) can be divided further by \(\text{NP}_{\text{dat}}\), then the divided category \(((S|\text{NP}_{\text{acc}})|\text{NP}_{\text{dat}})|(\text{TV}|\text{NP}_{\text{dat}})\) predicts \(\text{TV}|\text{NP}_{\text{dat}}\), namely DTV. In this way, a set of verb categories in lexical rules can be predicted, for example, as in \(\text{VP}|\text{PP}, \text{VP}|\text{PP}|\text{NP}_{\text{acc}}, \text{VP}|\text{NP}_{\text{dat}}, \text{etc.}\)

Some examples of possible ranges of prediction are illustrated in (19), where the arrow in “\(X:x \rightarrow Y:y\)” indicates that \(X:x\) predicts \(Y:y\). (19a) shows a possible range of prediction made by a fronted \(\text{NP}_{\text{acc}}\), which is type-raised to a functor category \(\text{VP}/\text{TV}\) and predicts a set \{TV,VP\}. This VP predicts further an existence of \(\text{NP}_{\text{nom}}\) and \(S\). The prediction “\(\text{VP}:v^1 \rightarrow H^v:C_1:...,C_n:c_n\)” in (19a) represents that a verb phrase predicts both a head verb \(H^v\), and its complements \(C_1:C_1:...,C_n:c_n\). (19b) is a prediction process started from a composition of two noun phrases. (19c) indicates that PVP1 predicts an AUX-finite and PVP2. As indicated by these tree diagrams, the prediction is executed from bottom to top.

(19) a. \(S:s\) b. \(S:s\)

\[
\begin{align*}
= &\text{S}/\text{NP}_{\text{nom}}: \lambda n_{\text{nom}},s \rightarrow \text{NP}_{\text{nom}}:n_{\text{nom}} \\
\rightarrow &\text{VP}:v^1 \rightarrow H^v:C_1:...,C_n:c_n \\
\rightarrow &\text{VP}/\text{TV}: \lambda ,v^2,v^1 \rightarrow \text{TV}:v^2 \\
\rightarrow &\text{NP}_{\text{acc}}:n_{\text{acc}} \\
\end{align*}
\]

\[
\begin{align*}
= &\text{S}/\text{NP}_{\text{nom}}: \lambda n_{\text{nom}},s \rightarrow \text{NP}_{\text{nom}}:n_{\text{nom}} \\
\rightarrow &\text{VP}:v^1 \rightarrow H^v:C_1:...,C_n:c_n \\
\rightarrow &\text{VP}/\text{DTV}: \lambda ,v^1,v^1 \rightarrow \text{DTV}:v^1 \\
\rightarrow &\text{TV}/\text{DTV}: \lambda ,v^2,v^2 \\
\rightarrow &\text{NP}_{\text{dat}}:n_{\text{dat}} \\
\rightarrow &\text{NP}_{\text{acc}}:n_{\text{acc}} \\
\end{align*}
\]
4 Semantic structures of fronted PVPs and split constructions

In this section, the prediction rules in CG style are applied to split constructions related to the PVP. In the above section, the PVP structure is classified into two types according to the properties of auxiliaries. This classification will be applied to explore the semantic characteristics of fronted PVPs. By exploring the difference in efficiency of the predictive recognition related to the head and its modifier, the data of split constructions related to the PVP will be accounted for.

4.1 PVP fronting and NP-PP split

In (20) (see De Kuthy (2002) and G. Müller (1998)), the noun phrase *ein Buch über Syntax* in PVP of (20a) is split in (20b). In (20b), the NP *ein Buch* is fronted in PVP. However, its modifier *über Syntax* is separated from it. In order to explore the semantic characteristics of the split construction such as partial NP fronting in the PVP in (20b), the combinatory rules mentioned above will be applied.

(20)  a. Ein Buch über Syntax ausleihen will er.

    b. Ein Buch ausleihen will er über Syntax.

For the purpose of analyzing the split construction of (20b), the semantic structure of the noun phrase *ein Buch über Syntax* should be explored, as indicated in (21).
(21) a. Buch □ N : buch’
   b. ein Buch[acc] □ NPacc : λv² λx. □ w[buch’(w) Í v²(x,w)]
   c. über Syntax □ N\N : λP λx.(P(x) Í sem-über(x, syntax’)) = f-21c
   d. Buch über Syntax □ N : f-21c(buch’) =
      λx.(buch’(x) Í sem-über(x, syntax’)) = f-21d
   e. ein[acc] □ NPacc/N : λP λv² λx. □ w[P(w) Í v²(x,w)] = f-21e
   f. ein Buch über Syntax □ NPacc : f-21e(f-21d) =
      λv² λx. □ w[buch’(w) Í sem-über(w, syntax’) Í v²(x,w)] = f-21f

Firstly, in (21a), a common noun Buch is translated into buch’ whose type is <e,t>. (21b) proposes that ein Buch[acc], namely, an accusative noun phrase can be translated into a function from a two place verb v² into a one place verb, where x and w are variables of type e. Because the prepositional phrase über Syntax in (20) modifies the noun Buch and specifies its meaning, its semantic type is <<e,t>,<e,t>>. As indicated in (21c), this paper proposes that an individual x has a property P of type <e,t> like Buch, and at the same time there is a set of semantic relations sem-über between x and the property syntax’ of type <e,t>. For example, there are such relations as x liest-über syntax (‘x reads about syntax’) and x informiert-über syntax (‘x informs about syntax’). By applying the semantic structure in (21c) to that of Buch in (21a), the semantic structure of Buch über Syntax is constructed as shown in (21d). The semantic structure of an accusative determiner ein in (21e) is applied to f-21d. A semantic structure of ein Buch über Syntax, as indicated in (21f), shows that an entity w has a property buch’, and there is a relation sem-über between w and syntax’, and at the same time w is an object of a two-place verb v² which is predicted by this accusative NP ein Buch über Syntax.

By using (21), the semantic structure of the sentence (20a) is constructed incrementally, as illustrated in procedure (22).
Following *ein Buch über Syntax*, a base form verb *ausleihen* (‘borrow’) is recognized as a verb of category V\(^2\) which is predicted by (21f). The semantic structure of this verb is indicated in (22a). Since *ausleihen* is a base form and a finite auxiliary *will* (‘want to’) follows *ausleihen*, the expression *ein Buch über Syntax ausleihen* is recognized as PVP1, whose semantic structure is constructed in (22b). PVP1 predicts the existence of AUX. This prediction is satisfied by the appearance of the finite auxiliary *will*. In (22c-1), the semantic structure of *will* is constructed, where *wollen* is a base form of *will*. Since PVP1 in (22b) has a type <e,t>, \(V^n\) and \(v^n\) in (22c-1) is adjusted to \(n=1\), namely (22c-2). The semantic structure of *wollen* is then able to be applied to that of PVP1 in (22b), and the semantic structure of the PVP2 *ein Buch über Syntax ausleihen will* is obtained, as indicated in (22d). The predicted NP\(_{\text{nom}}\) is satisfied by the nominative pronoun *er* (‘he’). (22e) shows that the meaning of *er* can be a set of properties of an individual \(a\) who is already specified in the context where the pronoun *er* is used. The semantic structure of (20a) is completed, as indicated in (22f).

This method can also be applied to the interpretation of split con-
structuress. In (20b) (repeated below as (23a)), the noun phrase *ein Buch über Syntax* is split, i.e., although the head noun *ein Buch* is fronted in PVP, its modifier *über Syntax* is separated from it and appears in the Mittelfeld. Applying the same incremental procedure as observed in (22), the semantic structure of *ein Buch ausleihen will er* is constructed. See (23b).

(23) a. Ein Buch ausleihen will er über Syntax.
   b. ein Buch ausleihen will er □

\[
S : □ w[\text{buch}'(w) \circ \text{wollen}'(\text{ausleihen}'(a,w))] 
\]

The prediction process of the NP-PP split related to the PVP of the sentence (23a) can be constructed as follows: although all complements of *ausleihen* are saturated in *ein Buch ausleihen will er* and it has already completed the meaning of a sentence, the PP *über Syntax* is added in the sentence-final position. This PP as an NP-modifier is predicted by *ein Buch*, not by the pronoun *er*, because the PP attachment to the pronoun is not possible. Since, for example, *x informiert über Syntax* and *das Buch informiert über y* can be unified, *ein Buch* and *über Syntax* have semantic features in common. For the purpose of unification, the interpretation of *ein Buch* in PVP is copied to the interpretation of *über Syntax* in the Mittelfeld, as shown in (24a). In (24a), □□□ is used to denote the semantic value of □, and □□□□□□□□ denotes the unification of the semantic values of □ and □. Then the interpretation of (23a) is constructed by the unification of □ *ein Buch ausleihen will er* □ in (24b) and □ *ein Buch über Syntax* □ in (24c).

For the purpose of unifying them, the verbal part in (24c), namely \(v^2(x,w)\), should be abstracted like (24d). Because \(\text{wollen}'(\text{ausleihen}'(a,w))\) in (24b) is a proposition, \(\lambda v^2 \lambda x. v^2(x,w)\) in (24d) does not need to be abstracted by \(v^2\) and \(x\). As a result of unification, the interpretation of (23a) is completed, as indicated in (24e).
(24) a. ein Buch ausleihen will er
b. ein Buch ausleihen will er
   = w[buch’(w) ⊗ wollen’(ausleihen’(a,w))]
c. ein Buch über Syntax
   = λv² λx. w[buch’(w) ⊗ sem-über(w,syntax’) ⊗ v²(x,w)]
d. w[buch’(w) ⊗ sem-über(w,syntax’) ⊗ λv² λx.v²(x,w)]
e. w[buch’(w) ⊗ sem-über(w,syntax’) ⊗ wollen’(ausleihen’(a,w))]

4.2 Topicalized PP and split constructions

Each prepositional phrase has potentiality to become a verb modifier or a noun modifier. With respect to the prediction, the fronted PP über Syntax in (25) can predict Vⁿ whose head verb satisfies the relation sem-über. For example, lesen (‘read’), wissen (‘know’), informieren (‘inform’) belong to such head verbs. At the same time, über Syntax in the initial position has potentiality to play the role of NP-modifier, i.e., it predicts a set of NPs whose head nouns satisfy the relation sem-über. For example, Information (‘information’), Buch (‘book’), Bericht (‘report’) belong to such head nouns. In the examples of (25), the accusative NP ein Buch appears in the Mittelfeld.

(25) a. Über Syntax hat er ein Buch ausgeliehen.
   b. *Über Syntax ausleihen will er ein Buch.
   c. Über Syntax lesen will er ein Buch.

In (25a), über Syntax is topicalized in the verb-second sentence. In contrast to (25a) which is grammatical, the example (25b) is not grammatical. In (25b), über Syntax appears in the fronted PVP1, but its head noun ein Buch is in the Mittelfeld. Is the sentence (25b) ungrammatical because of the split construction in the fronted PVP? Although the sentence (25c) has the same
syntactic structure as that of (25b), (25c) is grammatical. What causes the difference in this grammaticality? In this section, semantic factors that influence the acceptability of the split constructions related to the fronted PVP will be explored. For this purpose, the incremental recognition process of the grammatical sentence (25a) is firstly explored, as illustrated in procedure (26).

(26) a. über Syntax □ N\N : λ P λx.(P(x) □ sem-über(x,syntax')) = f-26a
    b. hat □ Vⁿ | Vⁿ = S|C₁|...|Cₙ | Vⁿ :
       λvⁿ λx₁n...λx₁. haben'(vⁿ-t(x₁,..., xₙ)) = f-26b
    c. hat er □ S|V¹ : λv¹. haben'(v¹-t(a)) = f-26c
    d. hat er ein Buch □ S|V² : λv². haben'( □ w[buch'(w) □ v²-t(a,w)])
    e. □ [Über Syntax] hat er □ □ □ ein Buch über Syntax □
       stack copy
    f. ein Buch über Syntax □ NPacc = V¹|V² :
       λ v² λ x. □ w[buch'(w) □ sem-über(w,syntax') □ v²(x,w)] = f-26f
    g. [Über Syntax] hat er ein Buch über Syntax □
       S|V² : λ v²-t.f-26c(f-26f(v²-t))
       = λ v²-t.haben'( □ w[buch'(w) □ sem-über(w,syntax') □ v²-t(a,w)])
       = λ v²-t. □ w[buch'(w) □ sem-über(w,syntax') □ haben'(v²-t(a,w))]
       = f-26g
    h. [Über Syntax] hat er ein Buch über Syntax ausgeliehen □ S :
       f-26g(f-22a)
       = □ w[buch'(w) □ sem-über(w,syntax') □ haben'(außleihen':t(a,w))]

The semantic structure of über Syntax in (26a) is stored in the stack, because each head verb or head noun predicted by über Syntax should satisfy the semantic relation sem-über(x,syntax'). However, neither head follows über Syntax directly. A finite auxiliary hat follows this PP, and this word order predicts a V2 structure. A subject should agree with hat, and this
auxiliary verb predicts a verb in the past participle form at the end of the clause. Semantic features of the next recognized pronoun er cannot match with those of über Syntax, and therefore, the semantic structure of über Syntax remains in the stack. The semantic structure of hat and that of hat er are represented in (26b) and (26c) respectively. Then an accusative NP ein Buch is recognized, and the semantic structure of hat er ein Buch is constructed in the process (26b,c,d) incrementally. Because the semantic relation between sem-über(x,syntax’) and buch’(x) is acceptable, a copy of the semantic structure of über Syntax is used to establish this relation, as illustrated in (26e). Applying the composition rule to the semantic structure of ein Buch über Syntax in (26f) and to that of hat er in (26c), a semantic structure of [Über Syntax] hat er ein Buch über Syntax is obtained in (26g), which is applied to that of ausleihen. Then the semantic structure of (25a) is completed, as shown in (26h).

In (25a), the PP-attachment to the NP ein Buch in the Mittelfeld works well by copying the fronted PP über Syntax to hat er ein Buch. They can be merged because they have semantic features in common. But in (25b), this copy and merge cannot be executed, because über Syntax and ausleihen in the fronted PVP do not have common semantic features. The interpretation of über Syntax ausleihen therefore is rejected. In contrast with (25b), über Syntax in (25c) has common semantic features with those of the verb lesen, and thus, über Syntax can fill the role of adverb, and the PVP in (25c) is accepted. The procedure (27) shows that the incremental semantic parsing of (25b) and (25c) indicates the difference in their grammaticality.

(27) a. über Syntax □ PPV-modifier = V^n|V^n :  
\[ \lambda x^n \lambda x_1 ...(\lambda^n(x_1,...,x_n) □ sem-über(x_1,...,x_n,syntax’)) \]

b. lesen □ S|NPnom|NPacc : \lambda x_2 \lambda x_1 lesen’(x_1,x_2)  

ausleihen □ S|NPnom|NPacc : \lambda x_2 \lambda x_1 ausleihen’(x_1,x_2)  
c. (1) über Syntax lesen □  
\[ \lambda x_2 \lambda x_1.(lesen’(x_1,x_2) □ sem-über(x_1,x_2,syntax’)) \]
In (27a), a semantic structure of über Syntax which functions as an adverb is proposed. The results of the application of (27a) to the semantic structures of verbs lesen and ausleihen in (27b) are represented in (27c-1) and (27c-2) respectively. (27a) and (27c) show that the semantic structure of über Syntax as V-modifier is similar to that of über Syntax as NP-modifier. However, the semantic structure of über Syntax lesen and that of über Syntax ausleihen are different. If a head verb hⁿ(x₁,...,xₙ) and sem-über(x₁,x₂,syntax’) have semantic features in common, they can be integrated as indicated in (27d). In (27c-1), lesen’(x₁,x₂) and sem-über(x₁,x₂, syntax’) have common semantic features, and therefore, a relation lesen’-über(x₁,x₂, syntax’) in (27e-1) is semantically consistent, while, in (27c-2), ausleihen’(x₁,x₂) and sem-über(x₁,x₂, syntax’) have no semantic features in common in the usual context, and they cannot be integrated into ausleihen’-über(x₁,x₂, syntax’). This means that über Syntax in (25b) cannot work well as an adverb of ausleihen. Using (27c-1), the meaning of (25c) can be constructed incrementally, as illustrated in (27e-
1,2,3,4). However, über Syntax ausleihen in (27c-2) cannot have the same procedure as (27e) in ordinary context. Über Syntax in (25b) cannot be interpreted as an adverb of ausleihen, and therefore, sem-über (x1,x2,syntax’) in (27c-2) should be reconstructed as an NP-modifier relative to x1 and x2, as in (27f), and stored in the stack. The recognition procedure started from ausleihen is illustrated in (27g-1,2,3,4), which shows that each step of the interpretations makes a tied frame of the semantic structure. Above all, in (27g-4), all complements are saturated and the NP ein Buch is embedded in this sentence frame. Therefore, it is difficult for the stacked interpretations of über Syntax which remains outside the frame to be attached to its head noun ein Buch, which is embedded in the frame. Generally, the modifier which has a functor category X|X in the fronted PVP should find its head which has an argument category X inside the PVP. In the case of (25b), the PP über Syntax as an NP-modifier has a functor category NP|NP. However, its head NP ein Buch exists outside of the PVP.

4.3 PVP fronting and the double infinitive

In the following example (28), the double infinitive lesen können is fronted. Since any complement of lesen is not saturated on its left, the base form verb lesen alone fills the role of PVP1. In (28), the fronted double infinitive PVP1-double combines with a finite auxiliary wird, and constructs a PVP2-double. The auxiliary wird which combines with the double infinitive is called the finite double-auxiliary (AUX-double:finite), where double:finite indicates double and finite.

(28) Lesen können wird er das Buch.
read be-able-to will he the book
‘He’ll be able to read the book.’
A prediction process realized by the fronted double infinitive is given as follows: since \textit{lesen} is a base form verb, it predicts the existence of an auxiliary that follows. This prediction is satisfied by the modal auxiliary \textit{können}, then a double infinitive \textit{lesen können} is composed, which predicts not only the appearance of AUX-double:finite, but also NP\textsubscript{nom} and NP\textsubscript{acc}, i.e., it predicts the possible structure of the sentence efficiently.

\begin{align*}
\text{(29) a. lesen} & \triangleq TV = S|\text{NP}_{\text{nom}}|\text{NP}_{\text{acc}} : \lambda x_2 \lambda x_1 . \text{lesen}^\ast(x_1,x_2) \\
\text{b. können} & \triangleq \text{AUX}_{\text{modal:bse}} = V^n | V^n = S|C_1|...|C_n | V^n : \lambda v^n \lambda x_n \ldots \lambda x_1 . \text{können}^\ast(v^n(x_1,\ldots,x_n)) \\
\text{c. lesen können} & \triangleq S|\text{NP}_{\text{nom}}|\text{NP}_{\text{acc}} : \lambda x_2 \lambda x_1 . \text{können}^\ast(\text{lesen}^\ast(x_1,x_2)) = \text{f.29c}
\end{align*}

PVP\textsubscript{2}-double:finite comprises PVP\textsubscript{1}-double and AUX-double:finite. Applying the combinatory rule for PVP, PVP\textsubscript{1}-double and PVP\textsubscript{2}-double are represented as (30b-1) and (30b-2). The AUX-double:finite, namely \textit{wird}, can be defined as a functor category, as indicated in (30c), where \textit{werden} is a base form of \textit{wird}. If PVP\textsubscript{1}-double consists of n-place head verb H\textsuperscript{a}bse and AUX\textsubscript{bse}, where ‘bse’ is ‘base form’, and if k complements are saturated in PVP\textsubscript{1}-double, then PVP\textsubscript{1}-double is a verbal complex V\textsubscript{m} with m unsaturated complements (where m=n - k). If the double auxiliary \textit{wird} combines with this V\textsubscript{m}, then it yields again a verbal complex with m unsaturated complements, namely V\textsubscript{m}. Therefore, the category of \textit{wird} is defined as PVP\textsubscript{2}-double\textbackslash{}PVP\textsubscript{1}-double, namely V\textsuperscript{m}\textbackslash{}V\textsuperscript{m}. In (30a,b), the set \{...,ch,...,ck,...\} has k constants, and the set \{...,x_i,...,x_j,...\} has m variables.

\begin{align*}
\text{(30) The category of double-auxiliaries and PVPs} \\
a. \text{H}^{n}\text{bse} : \lambda x_n \ldots \lambda x_1 . \text{h}^\ast(x_1,...,x_n) \\
b. \text{(1) PVP}\textsubscript{1}-\text{double} : \\
\lambda x_j \ldots \lambda x_i . \text{aux-\text{modal:bse}}(\text{h}^n(...,ch,...,x_i,...,x_j,...,ck,...)) \\
\text{(2) PVP}\textsubscript{2}-\text{double} :
\end{align*}
According to the prediction rule (18c), PVP1-double predicts AUX-double: finite and PVP2-double. By applying f-30c in (30c) to f-29c in (29), the semantic structure of lesen können wird is obtained, as shown in (31a). To apply the semantic structure of er in (31b) to f-31a in (31a), vⁿ in f-31b should be adjusted to a two place verb, namely \( \lambda v^2 \lambda x_2. v^2(a, x_2) \). Then the semantic structure of lesen können wird er is obtained, as illustrated in (31c). Next, the semantic structure f-31d of the accusative NP das Buch is applied to f-31c. By using the variable \( v^1 \) of a verb which has one unsaturated complement, the semantic structure of das Buch[acc], namely f-31d in (31d), is able to be applied to f-31c. Then a total semantic structure of (28) is obtained, as shown in (31e).

\[
(31)\ a. \ \text{lesen können wird} \ \overset{\text{PVP1-double: finite}}{\longrightarrow} \ S[NP_{\text{nom}}|NP_{\text{acc}} : f-30c(f-29c) = \\
\lambda x_2 \ \lambda x_1. \text{werden-double: finite}'(\text{können}'(lesen'(x_1, x_2))) = f-31a \\
b. \ \text{er} \ \overset{\text{PVP1-double: finite}}{\longrightarrow} \ S[V^n \ \overset{\text{PVP2-double: finite}}{\longrightarrow} \ V^n : \ \lambda v^n \ \lambda x_n...\lambda x_1. \text{werden-double: finite}'(v^n bse(x_1, x_n))) = f-30c \\
c. \ \text{lesen können wird} \ \overset{\text{PVP2-double: finite}}{\longrightarrow} \ S[NP_{\text{acc}} : \lambda x_2. \text{werden-double: finite}'(\text{können}'(lesen'(a, x_2))) = f-31c \\
d. \ \text{das Buch[acc]} \ \overset{\text{PVP2-double: finite}}{\longrightarrow} \ NP_{\text{acc}} : \lambda v^1. \ \square \ y[ \square \ w[\text{buch}'(w) \leftrightarrow w=y] \ \square \ \text{werden-double: finite}'(\text{können}'(lesen'(a, w)))] = f-31d \\
e. \ \text{lesen können wird} \ \overset{\text{PVP2-double: finite}}{\longrightarrow} \ S : f-31d(f-31c) = \\
\square \ y[ \square \ w[\text{buch}'(w) \leftrightarrow w=y] \ \square \ \text{werden-double: finite}'(\text{können}'(lesen'(a, w))))]
\]

4.4 Semantic constraints on H-AUX split related to PVP fronting

This section explores the Head-Auxiliary split (H-AUX split) in the double infinitive which is related to the fronted PVP.13 The fronted PVP in (32) das Buch lesen consists of a base form verb and its accusative NP complement.
However, a modal auxiliary können occurs in the sentence final position. The split double infinitive in (32) seems to suggest that the auxiliary können in the sentence-final position and the verb lesen in the fronted PVP cannot form a constituent. However, by applying combinatory rules, verbal members of the split double infinitive can be related to each other. The category proposed for AUX-double:finite in (30c) can be applied to the split double infinitive.

(32) Das Buch lesen wird er können
the book  read  will he be-able-to
‘He’ll be able to read the book.’

The auxiliary wird demands the PVP1-double optionally, and therefore, das Buch lesen wird has the possibility that it misses a base form auxiliary. Therefore, the category AUX-double:finite should be ready to be applied to the chipped category of das Buch lesen, namely, PVP1-double split by AUX-modal:bse, that is, PVP1-double/AUX-modal:bse. See (33b). By applying the division rule to AUX-double, namely, to PVP2-double\PVP1-double in (30c) (repeated in (33a)), the category of the double infinitive auxiliary related to the split construction can be constructed as indicated in (33c). This divided functor category is applied to the PVP1-double/AUX-modal:bse in (33b). (33d) is an example of this application. In this way, it is possible to construct split double infinitives like (32).

(33) a. werden1 □ AUX-double:finite = PVP2-double \ PVP1-double = V^n\V^n :
\[ \lambda v^n\text{bse} \lambda x_1 \ldots \lambda x_m. \text{werden}-double:finite'(v^n\text{bse}(x_1, \ldots, x_m)) = f\text{werden1} \]
b. PVP1-double/AUX-modal:bse :
\[ \lambda \text{aux-modal:bse} \lambda x_i \ldots \lambda x_i. \text{aux-modal:bse}(h^n(\ldots, c_h, \ldots, x_i, \ldots, x_j, \ldots, c_k, \ldots)) = f\text{-33b} \]
c. werden2 □ AUX-double:finite-split =
(PVP2-double/AUX-modal:bse)(PVP1-double/AUX-modal:bse) :
In order to obtain the semantic structure of (32), procedure (34) is necessary. A semantic structure (34b) is constructed by applying (34a) to \( \lambda x_2 \lambda x_1. \text{lesen}'(x_1,x_2) \). The PVP1 in (34b) is changed to the PVP1-double/AUX-modal:bse in (34c). The semantic structure of \textit{das Buch lesen wird} is obtained by the \( \lambda \)-conversion, as indicated in (34d).\(^{14}\) If the subject \( \text{er} \) is read, then f-34e in (34e) is realized, which is further applied to the semantic structure of \( \text{können} \). Because all the arguments of \( \text{lesen}' \) in f-34e have been saturated, \( n \) in \( \lambda v^n \lambda x_m...\lambda x_1. \text{können}'(v^n(x_1,...,x_n)) \) should be adjusted to 0 to obtain \( \lambda v^0. \text{können}'(v^0) \). By applying f-34e to this, the semantic structure of (32) is completed, as shown in (34f).

\[(34) \quad \begin{align*}
\text{a. } & \text{das Buch[acc]} \not\in \text{NP[acc]} : \lambda v^2. \not\in y[ \not\in w[\text{buch}'(w) \leftrightarrow w=y] \not\in v^2(w)] \\
\text{b. } & \text{das Buch lesen} \not\in \\
& \text{PVP1} : \lambda x_1. \not\in y[ \not\in w[\text{buch}'(w) \leftrightarrow w=y] \not\in \text{lesen}'(x_1,w)] \\
\text{c. } & \text{das Buch lesen} \not\in \text{PVP1-double/AUX-modal:bse} : \\
& \lambda \text{aux-modal:bse } \lambda x_1. \not\in y[ \not\in w[\text{buch}'(w) \leftrightarrow w=y] \not\in \text{aux-modal:bse}(\text{lesen}'(x_1,w))] \\
& = f-34c \\
\text{d. } & \text{das Buch lesen wird} \not\in \text{PVP2-double/AUX-modal:bse} : f\text{werden2}(f-34c) = \\
& \lambda \text{aux-modal:bse } \lambda x_1. \text{werden-double:finite}'( \not\in y[ \not\in w[\text{buch}'(w) \leftrightarrow w=y] \not\in \text{aux-modal:bse}(\text{lesen}'(x_1,w))]) = f-34d \\
\text{e. } & \text{das Buch lesen wird} \not\in \text{er} \\
& \not\in V'/\text{AUX-modal:bse} : f-34d \not\in S|V' : f_{\text{er}} = \lambda v^1.v^1(a) \\
& \not\in S/\text{AUX-modal:bse} : \\
& \lambda \text{aux-modal:bse}.f_{\text{er}}(f-34d(\text{aux-modal:bse})) \\
& = \lambda \text{aux-modal:bse}.\text{werden-double:finite}'( \not\in y[ \not\in w[\text{buch}'(w) \leftrightarrow w=y] \\
& \not\in \text{aux-modal:bse}(\text{lesen}'(a,w))]) = f-34e
\end{align*}\]
In this way, the category AUX-double:finite of \textit{wird} can be divided by the AUX-modal:bse for the purpose of recognizing the split double infinitive related to the fronted PVP. This category change supports the evidence that the split double infinitive is a constituent. However, the acceptability of the fronted PVP is reduced according to verbal elements in PVP. In (35), the Vorfeld contains a noun phrase \textit{das Buch} and a modal auxiliary \textit{können}, but the main verb \textit{lesen} appears in the sentence-final position separated from \textit{können}. The ungrammaticality of (35) can also be explored by analyzing the prediction process induced by \textit{das Buch können} in the Vorfeld.

\begin{align*}
\text{(35) } & \ast \text{Das Buch können wird er lesen.}
\end{align*}

In (35), the prediction process induced by \textit{das Buch können} is inefficient, since the predictions fail again and again. In a V2 sentence like this, the fronted NP\textsubscript{acc} \textit{das Buch} predicts that a finite TV follows it. This prediction fails, because the AUX-modal \textit{können} follows instead of the TV as predicted by NP\textsubscript{acc}. In the V2 sentence, \textit{können} in V2 position is finite and predicts a nominative NP that agrees with it. However, this prediction fails, because a finite auxiliary \textit{wird} follows it, which makes clear that \textit{können} is a base form auxiliary, and therefore, \textit{das Buch können} in front of \textit{wird} can be a PVP1. However, this PVP1 contains no main verb. This unsuccessful prediction process causes the inefficiency of the incremental processing of (35).

The semantic structure of \textit{das Buch können} can be realized as follows. In
(35), *können* should combine with a base form verb TV (=V^2) whose complement is the accusative noun phrase. Because of the fronted AUX-modal:bse in *das Buch können* and the predicted base form verb TV, the auxiliary *wird* should be a double auxiliary.

(36) das Buch ________ können ________
    □ IV|TV : f_{das_Buch}  □ V^n|V^n : f_{können}
    □ IV|TV = V^1|V^2 : λ_v^2 bse.f_{das_Buch}(f_{können}(v^2 bse))
    = λ_v^2 bse λx. □ w[buch’(w) □ können’(v^2 bse(x, w))] = f-36

As mentioned above, in (33c), the category of *wird* in (33a) is adjusted to the split double infinitive in the PVP, i.e. (33a) is divided by the category AUX-modal:bse. However, in the case of (36), *das Buch können* predicts a transitive verb. Therefore, the category of *wird* may be divided by TV_{bse}, as shown in (37a). Although the double auxiliary AUX-double:finite divided by AUX-modal:bse causes an effective prediction as mentioned, the division by TV_{bse} is invalid, because the verb phrase without its head verb contradicts. From the point of view of semantics, the ungrammaticality of (35) can be explored as follows: using the semantic structure f-37a which corresponds to the AUX-double divided by TV_{bse}, the expression *das Buch können wird* can be translated into f-37b in (37b). However in *werden*-double:finite’(...können’ (tv-bse(x1, w))) in f-37b, the relation tv-bse between x1 and w is still a variable, namely unspecific. Therefore, w which has a property buch’ cannot specify the relation to x1. Further, neither *werden*-double:finite’ nor *können’ can specify this relation. This shortage of semantic information in the Vorfeld makes the incremental processing inefficient.

(37) a. werden3 □ (V_m-double2 / TV_{bse}) \(\backslash\)(V_m-double1 / TV_{bse})
    : λ_g λ_tv^-bse. f_{werden1}(g(tv^-bse)) = f-37a
    where V_m-double1/ TV_{bse} : g and TV_{bse} : tv^-bse.
With respect to the incremental parsing by the prediction approach, (38) is less efficient than (35). Procedure (39) shows that the fronted non-head verb alone cannot have sufficient semantic information to predict the sentence meaning.

(38) *Können wird Hans das Buch lesen.

(39) a. können wird □ V^1|V^n :
\[ \lambda v^bse \lambda x_1...\lambda x_n. \text{werden-double:finite'}(\text{können'}(v^bse(x_1,...,x_n))) \]
\[ = f-39a \]

b. Hans □ NP_nom = S|VP □ V^a-1|V^n :
\[ \lambda v^a \lambda x_n...\lambda x_2. v^a(\text{hans}',x_2,...,x_n) = f_{\text{hans}} \]

c. können wird Hans □ V^a-1|V^n bse : \lambda v^bse.f_{\text{hans}}(f-39a(v^bse)) =
\[ \lambda v^bse \lambda x_n...\lambda x_2. \text{werden-double:finite'}(\text{können'}(v^bse(\text{hans}',x_2,...,x_n))) \]

The semantic structure of the subject NP Hans, as illustrated in (39b), shows that the head verb predicted by Hans can only be an arbitrary n place verb. Therefore, what is predicted by können wird Hans is a variable of n place verbs and variables of its n - 1 arguments, as shown in (39c). It contains insufficient information to predict the sentence meaning. This inefficiency in prediction process prohibits a PVP without its head verb from being fronted. On the contrary, the ordinary PVP, namely the partial VP which consists of a head verb and its partial complements is useful for the purpose of making the prediction process efficient.

4.5 NP-PP split and extraposition
As has been mentioned, the NP-PP split in (40a) is grammatical, while (40b) is ungrammatical, because the NP in the Vorfeld of (40b) is split, i.e., its head noun *ein Buch* is separated from its modifier *über Syntax* in PVP and appears in the Mittelfeld. Such difference in grammaticality can also be observed in the examples (41a,b) which contain extrapositions. De Kuthy (2002) observes that, the modifier PP of the head noun can be extraposed. See (41a). However, the head noun separated from its modifier PP in the Mittelfeld cannot be extraposed, as shown in (41b).

(40)  a. Ein Buch ausleihen will er über Syntax.
    b. *Über Syntax ausleihen will er ein Buch.
(41)  a. Er hat ein Buch ausgeliehen über Syntax.
    b. *Er hat über Syntax ausgeliehen ein Buch.

By using the syntax/semantics interface in CG, the difference in the grammaticality of extrapositions in (41a,b) can be explored. Firstly, with respect to (41a), the semantic structure of the expression *Er hat ein Buch ausgeliehen* which is obtained by removing the extraposed *über Syntax* can be completed in a few steps, as shown in (42).

(42)  a. er hat □ S|V : λv^1.t.haben^*(v^1.t(a))
    b. er hat ein Buch □ S|V^2 : λv^2.t. □ w [buch^*(w) □ haben^*(v^2.t(a,w))]
    c. er hat ein Buch ausgeliehen □ S :
      □ w[buch^*(w) □ haben^*(ausleihen-t^*(a,w)))] = f-42c

The obtained meaning f-42c in (42c) shows that (42c) is a complete sentence by itself. The extraposed PP *über Syntax* can be attached to f-42c, because *das Buch* and *über Syntax* have common semantic features. The PP attachment is executed by copying the NP *ein Buch* to the PP *über Syntax*, then the meaning of *ein Buch über Syntax* is realized as f-43a in (43a).
Unifying the meanings $f_{-42c}$ and $f_{-43a}$, namely $v^2(x,w) \sqsubseteq \text{haben}'(\text{ausleihen}-t'(a,w))$, the semantic structure of (41a) is obtained, as shown in (43b).

\[(43)\]
\[
a. \lambda v^2 \lambda x. \sqsubseteq w[\text{buch}'(w) \sqsubseteq \text{sem-über}(w,\text{syntax}') \sqsubseteq v^2(x,w)] = f_{-43a} \\
b. \sqsubseteq w[\text{buch}'(w) \sqsubseteq \text{sem-über}(w,\text{syntax}') \sqsubseteq \text{haben}'(\text{ausleihen}-t'(a,w))] \\
\]

Secondly, on the other hand, in (41b), the head noun is extraposed, while its modifier PP stays in the Mittelfeld. The meaning of (41b) cannot be realized easily from the viewpoint of semantics. As indicated in (44), a semantic structure of the partial expression *Er hat über Syntax ausgeliehen* which is obtained by removing the extraposed NP *ein Buch* is complicated and unspecified.

\[(44)\]
\[
a. \text{er} \sqsubseteq \text{NP}_{\text{nom}} = S|V^1 \sqsubseteq V^n|V^n : \lambda v^n \lambda x_n...\lambda x_2.v^n(a,x_1,...,x_n) \\
b. \text{er hat} \sqsubseteq S|V^1 : \lambda v^1-t.\text{haben}'(v^1-t(a)) \\
\quad \sqsubseteq V^n|V^n : \lambda v^n-t \lambda x_n...\lambda x_2.\text{haben}'(v^n-t(a,x_2,...,x_n)) = f_{-44b} \\
c. \text{über Syntax} \sqsubseteq \text{PPV-modifier} = V^n|V^n-1 : \\
\quad \lambda v^n-1 \lambda x_{n-1}...\lambda x_1.(v^n-1(x_1,...,x_{n-1}) \sqsubseteq \text{sem-über}(x_1,...,x_{n-1},\text{syntax}')) \\
\quad = f_{-44c} \\
d. \text{er hat über Syntax} \sqsubseteq V^n|V^n : \lambda v^n-t.f_{-44c}(f_{-44b}(v^n-t)) = \\
\quad \lambda v^n-t \lambda x_{n-1}...\lambda x_1.(\text{haben}'(v^n-t(a,x_1,...,x_{n-1})) \sqsubseteq \text{sem-über}(x_1,...,x_{n-1},\text{syntax}')) \\
e. \text{er hat über Syntax ausgeliehen} \sqsubseteq S|\text{NP}_{\text{acc}} : \\
\quad \lambda x_1.(\text{haben}'(\text{ausleihen}-t'(a,x_1)) \sqsubseteq \text{sem-über}(x_1,\text{syntax}')) = f_{-44e} \\
\]

In procedure (44), the semantic structure of *er hat* is firstly constructed as shown in (44a,b). For the purpose of combining the semantic structure $f_{-44b}$ of *er hat* with the semantic structure $f_{-44c}$ of the V-modifier *über Syntax*, each type of *er* and *er hat* should be adjusted to be a function from $V^n$ to $V^{n-1}$, because the subject *er* is saturated. The procedure from (44a,b) to (44d) is
mediated by the V-modifier über Syntax in (44c), not by the N-modifier über Syntax. In this procedure, semantic features of each argument have been inherited by unification. As has already been illustrated in (27), the unacceptability of über Syntax ausleihen in (40b) has been elucidated from the viewpoint of semantics. Über Syntax ausgeliehen in (41b) shows the same unacceptability from the viewpoint of the accumulation of semantic information for the prediction process.

What happens if this partial expression er hat über Syntax ausgeliehen is interpreted? (44e) shows its semantic structure, where a variable x₁ is contained in both relations ausleihen′ and sem-über, which means that the obtained interpretation is incomplete in spite of its complicated semantic procedure. This fact makes the predictive recognition of the extraposition like (41b) inefficient.

Another reason for interpreting (41b) as ungrammatical can be given as follows: if the accusative noun phrase ein Buch has an ordinary category/ type like (12b), repeated in (45a), the semantic structure of Er hat über Syntax ausgeliehen in (44e) cannot be combined with that of ein Buch. If the accusative noun phrase is type-raised by S, it can combine with f-44e in (44e) to form a semantic structure of (41b). See (45b) and (45c). However, the category adjusted for this purpose, namely S|(S|NPacc) in (45b) contains a category S|NPacc which cannot be regarded as a constituent. This category causes the meaning of über Syntax to stay in the same scope of er hat ... ausgeliehen, as illustrated in (45c), and therefore, über Syntax has no chance to become an NP modifier, although über Syntax cannot work well as an adverb of ausleihen because of the reason explained in (27).

(45) a. ein Buch[acc] □ V′|(V′|NPacc) = IV|TV :
  λv² λx. □ w[buch′(w) □ v²(x,w)]

  b. ein Buch[acc] □ S|(S|NPacc) :
  λs-np. □ w[buch′(w) □ s-np(w)] = f-45b
In this way, even though the semantic computation of the extraposition like (41b) is possible, it is executed with difficulty because of the unusual and unspecified category. This parsing fact corresponds to the fact that humans can manage to understand an example like (41b), but will not accept it as grammatical, because it is neither easy nor efficient to compute its syntax and semantics.

5 Conclusion

This article has applied the syntax/semantics interface realized in CG to the analysis of the split construction related to PVP fronting in German. Using formal semantics in the CG style, i.e., assigning each surface expression a model-theoretic interpretation, this article has accounted for reasons why the head should not be separated from its modifier in the fronted PVP i.e., why the fronted verbal complex without its head cannot be accepted in German. Research proposed a prediction mechanism to parse the PVP fronting incrementally on the ground of the CG formalism and its combinatory rules. Prediction rules in the frame of CG have been developed to make the recognition of free word order efficient. The grammar model proposed in this paper shows that combinatory rules modified to involve the predicting mechanism make it possible to explore the reason for the existence of the fronted PVP. PVPs are often used in the form of discourse which prefers the efficiency of comprehensions. The PVP plays the role of a prompter to make the comprehension efficient. If a fronted PVP does not contain the head, but contains only its modifier, such a PVP cannot fill the
role of prompter. By applying the composition rule, the semantic structure of the fronted PVP and its arguments invested semantic features can be inherited and accumulated step by step incrementally. In this sense, the incremental recognition process and the prediction mechanism proposed in this paper reflect the characteristics of language processing by humans. This investigation will hopefully show that the incremental computation with prediction mechanism of the CG style can become a candidate for an adequate linguistic formalism which explores some aspects of the head-modifier split in the fronted PVP in German.

Notes

1 According to De Kuthy and Meurers (2001), Haider (1990) pointed out that it is possible to realize a subject as part of fronted PVPs, as shown by Haider’s example:

   Ein Fehler unterlaufen ist ihr noch nie.
   an error crept-in is her still never
   ‘So far she has never made a mistake.’

2 The examples in this section are provided by Nerbonne (1994), De Kuthy and Meurers (2001), and De Kuthy (2002). However, some of them are modified.

3 G. Müller (1998, p.10) pointed out that, in (4b), the embedded constituent über Syntax is missing from the fronted constituent, i.e., the PP über Syntax is the “NP-internal THEME argument” of the head noun Buch.

4 De Kuthy (2002) indicates that the example of an NP-PP split such as (4b) become ungrammatical, if the embedding verb ausgeliehen (‘borrowed’) is replaced by the verb geklaut (‘stolen’).

5 (5b) is not acceptable in the ordinary context. Without context like ‘in der Bibliothek’ (‘in the library’), the semantic relations between ausleihen and über Syntax cannot be established.

6 In Izuo (2002), the PVP structure is classified into two types according to the properties of auxiliaries. This classification will be applied to explore the semantic characteristics of fronted PVPs.

7 The VP’s complement daughters in German are not always in a fixed order. The unordered sequence is made possible not only by the type-raising, but also by the
associativity in CG. The associativity relaxes the manner in which the complements are picked up by the head verb. However, the word order in PVP is under constraint of German word order rules. Uszkoreit (1987) claims that the word order rules for constituents of the Mittelfeld should hold across the partial verb phrase in the sentence-initial position.

8 Corresponding to PVP1, its semantic structure pvp1 is defined as a function from a head verb with m unsaturated arguments to a proposition in which n−m arguments are saturated.

9 In (22d), the scope of wollen’ can be narrowed to a verbal complex ausleihen’(x1,w).

10 An interpretation of hat indicates the tense by the linear-relation like -t < t, where -t indicates the past tense.

11 λ-conversion is used to obtain semantic structures of hat er and hat er ein Buch as follows:

\[
\begin{align*}
(1)\; \text{hat} & \quad \text{er} \\
\square \mathcal{V}^1 : \lambda v^1 . \lambda x. \text{haben}'(v^1, x) = g & \quad \square S | \mathcal{V}^1 : \lambda v^1 . v^1(a) = f \\
\square S | \mathcal{V}^1 : \lambda v^1 . f(g(v^1)) = \lambda v^1 . \text{haben}'(v^1, t(a))
\end{align*}
\]

(2) hat er ein Buch

\[
\begin{align*}
\square S | \mathcal{V}^1 : \lambda v^1 . \text{haben}'(v^1, t(a)) = f & \quad \square \mathcal{V}^1 | \mathcal{V}^2 : \lambda v^2 . \lambda t . \lambda x. \square w [\text{buch}'(w) \quad \square v^2(x, w)] = g \\
\square S | \mathcal{V}^2 : \lambda v^2 . f(g(v^2)) = \lambda v^2 . \square w [\text{buch}'(w) \quad \square \text{haben}'(v^2, t(a, w))]
\end{align*}
\]

12 The NP_nom can be type-raised to S|(S|C1), where C1=NP_nom, which can be divided as follows:

\[S | \mathcal{V}^1 : 0 (S | \mathcal{C}_2, ..., | \mathcal{C}_n ) | (\mathcal{V}^1 | \mathcal{C}_2, ..., | \mathcal{C}_n ), \text{where } \mathcal{V}^1 = (S | \mathcal{C}_1).\]

Therefore, the general category and the semantic structure of er can be defined as follows:

\[\text{er} \quad \square S | \mathcal{V}^i : 0 V^n : \lambda v^n . \lambda x_n . . . \lambda x_2. v^n(a, x_2, ..., x_n).\]

13 The NP-PP split and H-AUX split have a common category, i.e., a head H is separated from its modifier H|H.

14 In (34d), fwerden2 is \(\lambda f{-34c} \lambda \text{aux-modal:bse}. f\text{werden1}(f{-34c}(\lambda \text{aux-modal:bse})), \) therefore, fwerden2(f{-34c}) is \(\lambda \text{aux-modal:bse}. f\text{werden1}(\lambda x_1. \square y[ \square w [\text{buch}'(w) \leftrightarrow w = y] \quad \square \text{lesen}'(x_1, w))].\) In fwerden1, the verbal variable \(v^i\text{bse} in \lambda v^i\text{bse} \lambda x_1 . . . \lambda x_1. \text{werden-double:finite}'(v^i\text{bse}(x_1, ..., x_m))) \) will be adjusted to a one place verb \(v^i\) (namely \(m=1\)), because its argument is a verbal complex of arity 1. As a result, fwerden1 becomes \(\lambda v^i\text{bse} \lambda x_1. \text{werden-double:finite}'(v^i\text{bse}(x_1))).\)

15 The two place verb ausleihen forces the general n place head verb in (44d) to be adjusted to the two place verb.

16 The same category is used in (31d) for the purpose of combining das Buch[acc] with
lesen können wird er. However, in the case of (31d), the obtained meaning causes no difficulty with respect to the PP attachment, as in (41b).

References


Discontinuous Constituency (pp. 279-305). Berlin/New York: Mouton de Gruyter.

ドイツ語における不完全動詞句と分離構造
統語と意味のインターフェース
泉尾洋行

本稿は、ドイツ語の文において、一部の補足語を充足しない不完全動詞句が文頭に置かれ、なおかつそこに分離構造を含む現象を、形式意味論により分析したものである。分析は、ラムダ抽象化により実現された「関数 項」関係から構成される意味結合規則に基づき、不完全動詞句とその成立に関与する助動詞の意味特性を明確にした。本稿は、まずドイツ語にみられる不完全動詞句の文頭配置と、そこにおける分離構造を例示し、次に、その意味特性を、人の意味理解にみられる増進的処理という視点から分析するために、カテゴリー意味論を増進的解析と予測処理にむけて拡張した。それにより、不完全動詞句に見られる比較的自由な語順の意味処理も増進的に遂行できること、さらに、予測処理の効率と意味情報の増進的蓄積という観点から、不完全動詞句の諸形の文法的差異を説明できることを示した。さらに、分離構造が文頭の不完全動詞句に関与して現れる場合と、外置構造に現れる場合とでは、両者はその統語上の差異にもかかわらず、意味処理における効率という側面から見て、共通した特性を示すことも明らかにした。本稿は、形式的な側面からではあるが、予測に基づく増進的意味処理の過程を分析し、そのことから、不完全動詞句の文頭配置という一見すると変則的な語順が、実は
文頭近くの少数の語句に文全体の構成に関わる統語と意味の情報を集中させるたからきをもち、言語の伝達と理解のために理にかなった語順であることと示した。この見解は、もし、文頭に配置された不完全動詞句において、動詞句あるいは名詞句の語彙的主辞が欠如し、それの修飾要素としての代わりあるいはのみが残った分離構造が生じるならば、そこには語彙的主辞に含まれた統語と意味の情報が欠如することになり、その結果、文頭近くの少数の語句により文全体の構造を予測させるという機能が大幅に後退し、その文は非文法的となるという事実にその根拠を求めることができる。