Case alternation in lexicalized grammar

Symon Stevens-Guille
The Ohio State University

Elena Vaikšnoraitė
The Ohio State University

Proceedings of the 28th International Conference on
Head-Driven Phrase Structure Grammar

Online (Frankfurt/Main)

Stefan Müller, Nurit Melnik (Editors)

2021

Frankfurt/Main: University Library

pages 205–219

Keywords: HPSG, HTLCG, case alternation

Stevens-Guille, Symon & Elena Vaikšnoraitė. 2021. Case alternation in lexicalized grammar. In Stefan Müller & Nurit Melnik (eds.), *Proceedings of the 28th International Conference on Head-Driven Phrase Structure Grammar, Online (Frankfurt/Main)*, 205–219. Frankfurt/Main: University Library. DOI: 10.21248/hpsg.2021.11.

Abstract

In this paper, we propose to extend the Przepiórkowski's 2000 analysis of Long Distance Genitive of Negation to the same phenomenon in Lithuanian. We discuss the features that have their origin in Categorial Grammar. We then develop a novel analysis of the case alternation in Categorial Grammar incorporating features of the HPSG analysis. The two accounts show a surprising convergence in basic assumptions and predictions.

1 Introduction

This paper presents two analyses of genitive-accusative case alternation in Lithuanian in two lexicalist grammars: Head-driven Phrase Structure Grammar (HPSG) and Hybrid Type Logical Categorial Grammar (HTLCG). In Lithuanian, a direct object of a transitive verb is canonically accusative case marked. In the presence of verbal negation *ne*, the same argument surfaces with genitive case marking instead of accusative. This phenomenon in Balto-Slavic linguistics is called the Genitive of Negation.

(1) a. Vaiva nusipirko knyg-ą / *knyg-os.
Vaiva.NOM buy.PST.3 book-ACC book-GEN
'Vaiva bought a book.'

b. Vaiva ne-nusipirko *knyg-ą / knyg-os.
Vaiva.NOM neg-buy.PST.3 book-ACC book-GEN
'Vaiva didn't buy a book.'

Developing an empirically adequate and theoretically sound analysis of this case alternation has long animated linguists working on Balto-Slavic languages. In this paper, we propose that an analysis of Polish Genitive of Negation by Przepiórkowski (2000) can be extended to Lithuanian. Przepiórkowski's 2000 analysis is implemented in HPSG. We show that the HPSG analysis extends to Lithuanian. We highlight those components in the analysis that are historically related to developments in Categorial Grammar (CG). Subsequently, we propose an account in a contemporary Categorial Grammar, emphasizing the components of the account that are inspired by HPSG. The upshot we argue is a convergence between proof-theoretic and model-theoretic syntactic accounts, which suggests the potential for a renewed exchange of ideas between HPSG and CG communities.

In Section 2, we introduce the basic facts about Genitive of Negation in Lithuanian. In 3, we briefly summarize Przepiórkowski's 2000 analysis of Genitive of Negation in Polish and show how it can capture the Lithuanian data. Then we introduce Hybrid Type Logical Categorial Grammar (HTLCG) and propose an account of the data. Subsequently, we show how limitations in the CG analysis can be rectified by borrowing concepts from HPSG. Section 4 concludes the paper.

2 Genitive of Negation in Lithuanian

In this section we describe Genitive of Negation in Lithuanian. Genitive of Negation in Lithuanian is said to be obligatory. All normally accusative case marked objects obligatorily occur in the genitive in the presence of verbal negation, which in Lithuanian is realized as a prefix *ne*. Non-accusative case marked objects do not participate in the case alternation as shown in 2. For example, the verb *džiaugtis* 'to rejoice' selects for an instrumental case-marked NP. In the presence of verbal negation case-marking on *pergale* 'victory' does not change.

(2) a. Vaiva džiaugėsi pergal-e / *pergal-ės.
Vaiva.NOM rejoice.PST.3 victory-INST victory-GEN
'Vaiva rejoiced in victory.'

b. Vaiva ne-sidžiaugė pergal-e / *pergal-ės.
Vaiva.NOM neg-rejoice.PST.3 victory-INST victory-GEN
'Vaiva didn't rejoice in victory.'

Genitive of Negation is clause-bound as shown in 3.

(3) Vaiva ne-sakė, kad nusipirko knyg-ą / *knyg-os.
Vaiva.NOM neg-say.PST.3 that buy.PST.3 book-ACC book-GEN
'Vaiva didn't say that she bought a book.'

While Genitive of Negation is clause-bound, it is not limited to local contexts. In Long Distance Genitive of Negation, an argument of an infinitival verb occurs in the genitive when the selecting verb is negated. A sentence containing a subject control verb is used as an example in 4.

(4) a. ? Vaiva ne-pažadėjo nupirkti šit-ą knyg-ą.
Vaiva.nom neg-promise.pst.3 buy.inf this-acc book-acc
'Vaiva didn't promise to buy this book.'

b. Vaiva ne-pažadėjo nupirkti šit-os knyg-os.
Vaiva.nom neg-promise.pst.3 buy.inf this-gen book-gen
'Vaiva didn't promise to buy this book.'

While Local Genitive of Negation is obligatory, Long Distance Genitive of Negation is often optional. There is at present a dearth of information concerning the factors that influence the choice of case (though see Arkadiev 2016).

Long Distance Genitive of Negation can in principle affect multiple direct objects. In 5, negation on the matrix verb *ne-išmokė* 'didn't teach' triggers genitive case on its direct object vaikų 'children' and also on the embedded object of the infinitival verb *tvoros* 'fence'.

(5) Tėvai ne-išmokė vaik-ų / *vaik-us dažyti
parent.NOM NEG−teach.PST3 children-GEN children-ACC paint.inf
tvor-os / ?tvor-ą.
fence-GEN fence-ACC
'Parents did not teach their children to paint the fence.' (Arkadiev, 2016, 86)

An emprically adequate analysis of Genitive of Negation in Lithuanian thus needs to capture the following three empirical generalizations:

(6) Empirical generalizations
a. Local Genitive of Negation is obligatory
b. Long distance Genitive of Negation is optional
c. Long distance Genitive of Negation can trigger genitive case on multiple (non-)local arguments

3 Analyses

In this section, we introduce two analyses of the Genitive of Negation aiming to capture the empirical generalizations in 6.

3.1 Przepiórkowski (2000)'s analysis in HPSG

We propose that Przepiórkowski (2000)'s analysis of Genitive of Negation in Polish can be extended to Lithuanian. This result is expected given that Lithuanian patterns with Polish in that local Genitive of Negation is obligatory.

We adopt Przepiórkowski (2000)'s case division and case type hierarchy, which we present in 7. Notice that according to these assumptions, Polish and consequently Lithuanian has three structural cases: nominative, accusative and genitive.

(7) Case division:
a. Structural cases: snom, sacc, sgen
b. Lexical cases: lacc, lgen, ldat, lins, lloc

Lexical entries of predicates are assumed to distinguish between a structural argument and a lexical argument as shown in the toy lexicon in 8.

(8) a. *nupirkti* 'to buy': [AGR-ST ⟨NP[str], NP[str] ⟩]
b. *didžiuotis* 'to be proud of': [AGR-ST ⟨NP[STR], NP[LINS]⟩]

Structural case is resolved to a particular morphological case by the (simplified) set of constraints in 9 and 10.

(9) $[\text{NEG}-, \text{ARG-ST}[\boxed{1}_{nelist} \oplus \langle[\text{CASE } str]\rangle \oplus \boxed{2}_{list}]] \rightarrow$
$[\text{ARG-ST}[\boxed{1} \oplus \langle[\text{CASE } acc]\rangle \oplus \boxed{2}]]$

(10) $[\text{NEG}+,\text{ARG-ST}[\boxed{1}_{nelist} \oplus \langle[\text{CASE } str]\rangle \oplus \boxed{2}_{list}]] \rightarrow$
$[\text{ARG-ST}[\boxed{1} \oplus \langle[\text{CASE } gen]\rangle \oplus \boxed{2}]]$

The rule in (9) ensures that the structural case is resolved to accusative when the NP is selected by something with the neg− property. More precisely, it states that for any non negated verbal category its non-initial structural argument must bear accusative case. The rule in (10) ensures that the structural case is resolved to genitive when the NP is selected by something with the NEG+ property. These sets of assumptions provide a simple analysis of Local Genitive of Negation as shown in Figure 1.

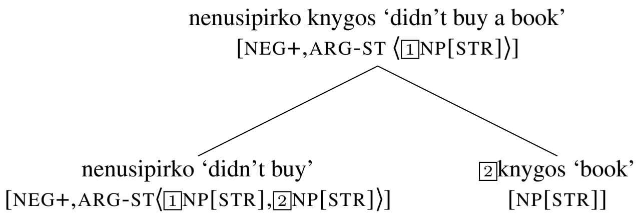

Figure 1: Local Genitive of Negation in sentence 1 in HPSG

Now turning to the optional Long Distance Genitive of Negation. Accusative case in the infinitival complement is accounted for straightforwardly as shown in Figure 2.

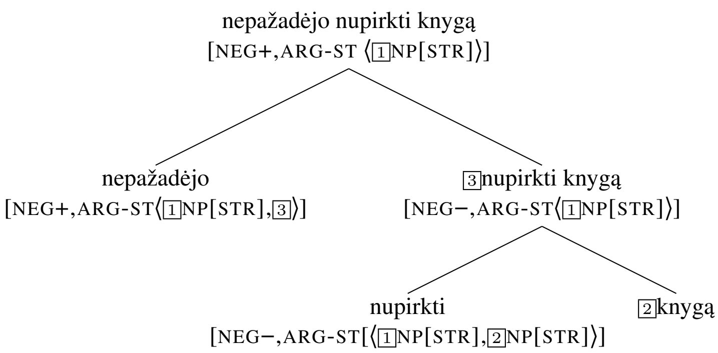

Figure 2: Local accusative in the infinitival complement in HPSG

To acccount for the genitive case in the infinitival complement further assumptions need to be made. In Przepiórkowski's 2000 analysis, the verb cluster consisting of a verb and its infinitival complement are analyzed in terms of argument inheritance. In other words, there is object raising to the complement of a complex predicate. In Figure 3, while *nupirkti* 'to buy' selects an NP to form an infinitive, the complement of *nepažadėjo* 'didn't promise' is only required to be headed by an infinitive missing a subject and including a (possibly empty) list of complements. When *nepažadėjo* 'didn't promise' selects the infintive *matyti* it subsequently inherits the selection of *tom-o* by inheriting the list of complements of the infinitive.

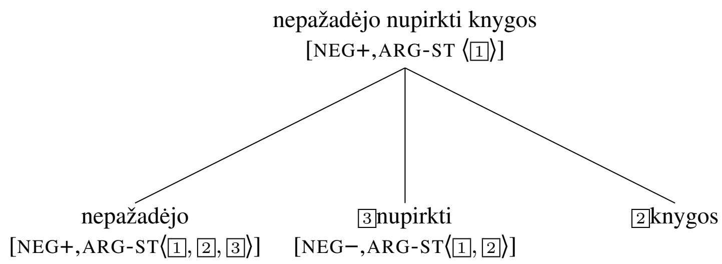

Figure 3: Long Distance Genitive of Negation in HPSG

Since in principle multiple arguments can raise, this analysis captures the fact that Long distance Genitive can trigger genitive case on multiple (non-)local arguments.

Przepiórkowski (2000)'s analysis thus captures all the empirical generalizations listed in 6.

3.2 HTLCG

HTLCG (Kubota, 2014; Moot & Stevens-Guille, 2019, 2020; Kubota & Levine, 2020) is a lexical theory of grammar based on linear logic (Girard, 1987). HTLCG differs from the standard Lambek Categorial Grammar $\mathbb{L}$ (Moortgat, 1997) in dividing syntax between the 'pheno' and 'tecto' components–roughly word order and argument structure. Moreover, it distinguishes directed and undirected implication. A sentence is generated by the grammar if and only if there is a proof of the proposition S(entence) with the premises corresponding to the lexical entries.

HTLCG lexical entries consist of tuples of pheno term, tecto type and semantic term. The tecto type reduces to a linear logic formula, the type of which can be recovered in the linear lambda calculus by the Curry-Howard Correspondence. Both the pheno and semantic components of the rules therefore correspond to inferences in the linear lambda calculus, reduction of which yields propositions in some target logic.[1] The target logic of our semantics is just first order logic. The target language of our pheno is likewise just a logic over strings or structures. The logic underlying the grammar is studied in depth in (Moot & Stevens-Guille, 2019, 2020).

Unlike earlier work in HTLCG, we import the universal quantifier from first order multiplicative linear logic (MILL1) into the logic of HTLCG. While the first order extension slightly complicates the underlying logic, we will mostly forego discussion of matters of logic, restricting ourselves to presenting the theory for which the quantifier is invoked.[2]

[1]Note however that since types from $\mathbb{L}$ correspond to terms of type *s*(tring) or *s*tructure, the directed implication introduction rule, unlike the the undirected implication introduction rule, doesn't correspond to introducing a function in the pheno.

[2]We are not the first to employ MILL1 to extend $\mathbb{L}$. Moot & Piazza (2001) show how MILL1 can be used to account for a wide variety of phenomena that are otherwise difficult to account for

In the first order version of HTLCG, case is uniformly represented by an argument of N or NP which expones it. We provide a lexicon in Table 1.

pheno	tecto	semantics
$saruno^s$	NP(gen)	$saruno^e$
$vaiva^s$	NP(nom)	$vaiva^e$
$nori^s$	(NP(nom)\S)/(NP(nom)\INF)	$\lambda P^{e\to t} x^e.want(x, P(x))$
$pamatyti^s$	(NP(nom)\INF)/NP(acc)	$\lambda x^e y^e.see(y, x)$
$pamate^s$	(NP(nom)\S)/NP(acc)	$\lambda x^e y^e.see(y, x)$

Table 1: HTLCG Lexicon.

We first introduce the directed fragment of HTLCG in Gentzen-style natural deduction, which is a pheno-decorated version of the directed implications of $\mathbb{L}$ with the addition of the distinction between Lex and Id, on which see (Moot & Stevens-Guille, 2019, 2020). These rules are shown in Figure 3.2.

$$\frac{}{p^s : w \vdash M ; A}\text{Lex} \qquad \frac{}{x^\alpha : A \vdash x^\alpha : A}\text{Id}$$

$$\frac{\Gamma \vdash M^s ; B/A \qquad \Delta \vdash N^s ; A}{\Gamma, \Delta \vdash (M + N)^s : B}/E \qquad \frac{\Delta \vdash N^s : A \qquad \Gamma \vdash M^s : A\backslash B}{\Delta, \Gamma \vdash (N + M)^s : B}\backslash E$$

$$\frac{\Gamma, p^s : A \vdash (M + p)^s : B}{\Gamma \vdash M^s : B/A}/I \qquad \frac{p^s : A, \Gamma \vdash (p + M)^s : B}{\Gamma \vdash M^s : A\backslash B}\backslash I$$

Figure 4: Gentzen-Style ND Inference Rules for directed HTLCG

We will omit the left side of the the Lex rule in the proofs to follow, since it is required just for technical reasons. In the syntactic proofs we will implicitly reduce the pheno terms. In the semantic term reductions we will perform β reduction on the fly, too, but mark it explicitly in the rule.

To get a sense of how the grammar works, we provide the syntactic proof of *Vaiva pamatė Šaruną* 'Vaiva saw Sarunas' in Figure 5 followed by the corresponding semantic proof in Figure 6. These proofs suffice to show how word order and case is determined in the theory; the semantic proof mirrors the syntactic proof,

in $\mathbb{L}$. In fact, adding the quantifiers of MILL1 to multiplicative linear logic doesn't change the complexity of deciding provability from the multiplicative propositional fragment: deciding MILL1 is NP-complete. One further point is that the Curry-Howard Correspondence for quantifiers is dependent types. In the interests of keeping exposition of the logic to a minimum we therefore suppress the rules for ∀, rolling uses of the elimination rule (which is the only rule we use) into some uses of the implication elimination rule (effectively implementing universal modus ponens). But nothing prevents us from constructing the proofs without suppressed rules.

$$\dfrac{\dfrac{}{\vdash vaiva^{e} \ ; \ NP(nom)}\text{Lex} \qquad \dfrac{\dfrac{}{\vdash pamate^{s} \ ; \ (NP(nom)\backslash S)/NP(acc)}\text{Lex} \qquad \dfrac{}{\vdash saruna^{s} \ ; \ NP(acc)}\text{Lex}}{\vdash pamate + saruna \ ; \ NP(nom)\backslash S}/E}{\vdash vaiva + pamate + saruna \ ; \ S}\backslash E$$

Figure 5: Lithuanian transitive in HTLCG

$$\dfrac{\dfrac{}{\vdash vaiva^{e}}\text{Lex} \qquad \dfrac{\dfrac{}{\vdash \lambda x^{e} y^{e}.see(y, x)}\text{Lex} \qquad \dfrac{}{\vdash saruna^{e}}\text{Lex}}{\vdash \lambda y^{e}.see(y, saruna)}app, \beta}{\vdash see(vaiva, saruna)}app, \beta$$

Figure 6: Semantic term for Lithuanian Transitive in HTLCG.

with the leaves of the tree corresponding to the semantic and syntactic content of the lexical items.

3.2.1 Negation

The centerpiece of our theory of Genitive of Negation is the lexical entry for negation. The lexical entry scheme for negation, like for conjunction and disjunction, is polymorphic.

(11) $\lambda t^{(s\to)_n s} q^{s} \ldots q_n^{s}.ne + t(q \ldots q_n)$;
$\forall x.T[x := f(x)] \upharpoonright T$;
$\lambda P^{e\to(e\to)_n t} z^{e} \ldots z_n^{e} x^{e}.\neg P(x, z \ldots z_n)$

Here T is a meta-variable in the style of Steedman (2000) over:

(12) {NP(nom)\S^{int}, (NP(nom)\S)↾E, (NP(nom)\INF)↾E}

E is a meta-variable over:

(13) {NP(x), NP(x)↾NP(x)$_1$...↾NP(x)$_{n\geq 1}$}

It is worth noting that the since the number of recursively embedded clauses to a negated verb is bounded in practice (if not in principle) and since this bound limits the number of NPs which could be 'raised' to be selected by the complement of *ne*, one could dispense with the ∀ completely, simply enumerating the set of possible lexical entries.

The most important part of the entry for *ne* is the axiom restricting *f*:

(14) $\forall x.(x \neq acc \to f(x) = x) \wedge (x = acc \to f(x) = gen)$

The axiom ensures the function *f* is the identity function on every input but acc, for which it returns gen.

$$\dfrac{\dfrac{}{\vdash vaiva^{s}}\,\text{Lex} \qquad \dfrac{\dfrac{}{\vdash nori^{s} \,;\, (NP(nom)\backslash S)/(NP(nom)\backslash INF)}\,\text{Lex} \qquad \dfrac{\dfrac{}{\vdash pamatyti^{s} \,;\, (NP(nom)\backslash INF)/NP(acc)}\,\text{Lex} \qquad \dfrac{}{\vdash saruna^{s} \,;\, NP(acc)}\,\text{Lex}}{\vdash pamatyti + saruna \,;\, NP(nom)\backslash inf}\,/E}{\vdash nori + pamatyti + saruna \,;\, VP}\,/E}{\vdash vaiva + nori + pamatyti + saruna \,;\, S}\,\backslash E$$

Figure 7: Embedded acc from *matyti* in HTLCG

$$\dfrac{\dfrac{}{\vdash vaiva}\,\text{Lex} \qquad \dfrac{\dfrac{}{\vdash \lambda P^{e\to t} x^{e}.want(x, P(x))}\,\text{Lex} \qquad \dfrac{\dfrac{}{\vdash \lambda x^{e} y^{e}.see(y, x)}\,\text{Lex} \qquad \dfrac{}{\vdash sarunas^{e}}\,\text{Lex}}{\vdash \lambda y^{e}.see(y, sarunas)}\,app, \beta}{\vdash \lambda x^{e}.want(x, see(x, sarunas))}\,app, \beta}{\vdash want(vaiva, see(vaiva, sarunas))}\,app, \beta$$

Figure 8: Semantic term for Figure 7.

Before seeing how *ne* contributes to solving (long) GN, we derive the embedded accussative of a complex predicate in which *ne* isn't present. Complex predicates are modelled in the spirit of Kubota (2014)'s account of Japanese complex predicates in CG. In the proofs in Figure 7 and Figure 8 the embedded infinitive is selected by the higher verb *nori*, which supplies the embedded verb with its subject in the semantics; the case of the embedded object is determined entirely by the embedded verb.

3.2.2 Full HTLCG

Full HTLCG is obtained by adding the following connective $\upharpoonright$ and its inference rules in Figure 9. We can add the rule for ∀E and the derived rule of universal modus ponens, too, but in keeping with our earlier comments, we suppress these rules and their exposition for space.

$$\dfrac{\Gamma \vdash M^{\alpha\to\beta} \,;\, A \upharpoonright B \qquad \Delta \vdash N^{\alpha} \,;\, B}{\Gamma, \Delta \vdash (MN)^{\beta} \,:\, A}\,\upharpoonright E$$

$$\dfrac{\Gamma, x^{\alpha} \,:\, A \vdash M^{\beta} \,:\, B}{\Gamma \vdash (\lambda x.M)^{\alpha\to\beta} \,:\, B \upharpoonright A}\,\upharpoonright I$$

Figure 9: ND for $\upharpoonright$.

Given the foregoing rules, we can now derive the Genitive of Negation. We first present the syntactic proof of local Genitive of Negation in Figure 10. The semantic proof corresponding to the syntactic proof in Figure 10 is found in 11. Note that, in the interests of intelligibility, we suppress the tecto types in some Lex rules, since these can be recovered from the tectos which combine with them–the tecto types are further present in the Figure 1. We suppress corresponding terms in the

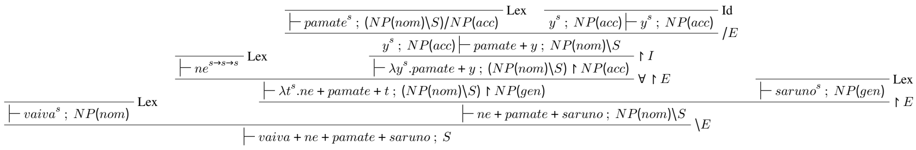

Figure 10: Gen from *ne-pamatė* 'didn't see' in HTLCG

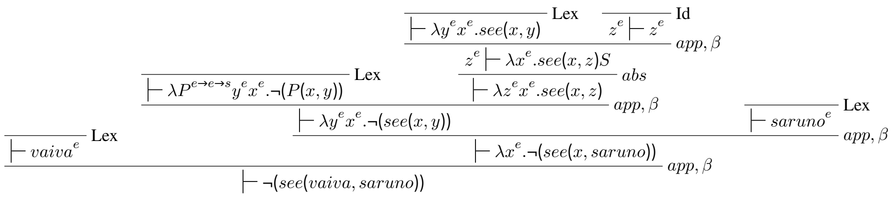

Figure 11: Semantic term for Figure 10

semantic proofs, simply writing the word which denotes the term and its type. The important step of the proof is $\forall \upharpoonright E$, the conclusion of which converts the $NP(acc)$ argument of the verb complex to $NP(gen)$, thereby licensing composition with the 'raised' genitive object.

With the extended version of HTLCG, we now step through the proof of long Genitive of Negation in *Vaiva nenori pamatyti Šarūno* 'Vaiva doesn't want to see Sarunas'. Figure 12 shows the syntactic proof of long Genitive of Negation by raising of the embedded object to the negated verb complex. The corresponding semantic proof of the syntactic proof in Figure 12 is represented in Figure 13.

3.3 Interim conclusions

Lithuanian Genitive of Negation can be given an off-the-shelf analysis in HPSG by adopting the theory developed for Polish Genitive of Negation in Przepiórkowski (2000). The HPSG theory makes significant use of function composition, which is a theorem of HTLCG. Function composition underlies the HPSG account of complex predicates, where complex predicates are the source of long GN. We have

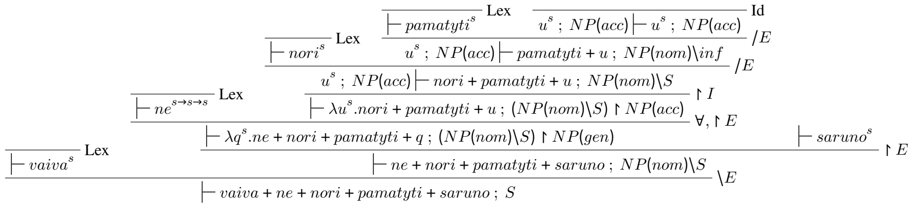

Figure 12: Gen from *ne-nori* in HTLCG

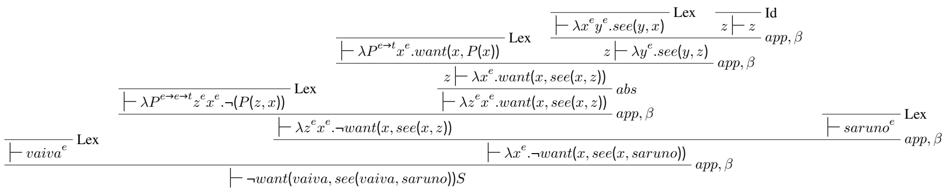

Figure 13: Semantic term for Figure 12

developed an HTLCG account in which complex predicates are the source of long Genitive of Negation too. This points to a convergence between the two theories.

The HTLCG account is successful in capturing Lithuanian Genitive of Negation due to the presumptions that $(N)P$ can be modelled in terms of a property of case.[3] The property view of categories is underdeveloped in the CG literature, perhaps in part due to there being no standard account of features (Kubota, 2021). But the property view could be extended to properties of sequences of arguments corresponding to features in HPSG; in fact we will shortly argue that further features borrowed from HPSG improve the HTLCG account. Perhaps the use of features in the present work will spur renewed interest in incorporating concepts from HPSG–which surely enjoys the most developed account of features among rigorous grammar formalisms–into CG.

Despite the proposed convergence between HPSG and HTLCG, both accounts of (long) GN, without further restrictions, overgenerate.

The following schemes represent optional embedded acc/gen:

(15) NP(nom) ne-V NP(gen) inf NP(acc)

(16) NP(nom) ne-V NP(gen) inf NP(gen)

The HTLCG theory derives this scheme. However, by virtue of deriving the scheme above, which involves uniform licensing of $\upharpoonright$I, the theory erroneously derives the following scheme:

(17) NP(nom) ne-V NP(acc) X.

Where X is schematic for one of the following:

(18) NP(obl)

(19) NP(y) INF NP({acc, gen})

(20) INF NP({acc, gen})

(21) NP(y) INF NP({acc, gen}) NP(obl)

(22) INF NP({acc, gen}) NP(obl)

[3]It should be obvious that agreement could be modelled by means of the property view of categories too.

The schemes are derived due to the fact that the NP being selected by the *ne*-V complex doesn't need to be the NP adjacent to the *ne*-V complex; it could be some embedded NP with case of any type. However, this issue can be resolved by incorporating a feature from the HPSG account.

Before proceeding to fixing the HTLCG account, we note some limits of both the HPSG and HTLCG accounts. Both accounts predict that embedded acc and gen are in free variation. For Lithuanian this predicts unattested acc, since in some environments embedded gen is overwhelmingly preferred (Arkadiev, 2016).

More difficult for the HPSG account is the following erroneous scheme:[4]

(23) NP(nom) V ne-INF NP(acc)

This scheme is erroneouly generated by building V *ne*-INF into a complex predicate and 'raising' the embedded NP, which will then get acc from V.
Przepiórkowski (2000) noticed this issue and correctly ruled it out by restricting raising to [NEG−] environments. Thus, if the lower verb is [NEG+], the embedded object is forced to be resolved with respect to *ne*-INF. This restriction is implemented by invoking raising features:

(24) $[\text{NEG+}, \text{ARG-ST}[\langle \boxed{0} \rangle \oplus \boxed{1}]] \rightarrow \boxed{1} = \text{list}[\text{XP}^{-}]$

In brief, the head with the property [NEG+] requires its arguments (except the first) to be of the form XP^{-} where the feature - means the argument is raised no further.

Moreover, raising features prevent case resolution mismatch between an embedded verb and the head which selects it to form a negated complex predicate. The resulting resolution principles then get the following form:

(25) $[\text{NEG-}, \text{ARG-ST}[\boxed{1}_{nelist} \oplus \langle [\text{CASE } str]^{-} \rangle \oplus \boxed{2}_{list}]] \rightarrow [\text{ARG-ST}[\boxed{1} \oplus \langle [\text{CASE } acc] \rangle \oplus \boxed{2}]]$

(26) $[\text{NEG+}, \text{ARG-ST}[\boxed{1}_{nelist} \oplus \langle [\text{CASE } str]^{-} \rangle \oplus \boxed{2}_{list}]] \rightarrow [\text{ARG-ST}[\boxed{1} \oplus \langle [\text{CASE } gen] \rangle \oplus \boxed{2}]]$

Witko (2008) objects to these techniques. According to him, Przepiorkowski's use of clitic climbing data to justify the prohibition on raising from negated VPs isn't robust. However, it is worth noting that Witkos develops these arguments from the position of the Minimalist Program; in his own framework every dependency, whether long or local, produced by 'agree' or 'merge', is due to features. Consequently, Witkos's criticism is pyrrhic, since his own theory relies on non-standard distinctions between complete and incomplete (double) probes and features including +MULTIPLE, which just serve to instruct probes to multiple agree.

Witkos's features do not seem superior to raising features, which are, moreover, common to Minimalist syntax in the form of feature deficiencies. Indeed, Witkos

[4]This scheme is not produced by HTLCG since in HTLCG ne requires its complement–here INF–to be missing some object.

uses such feature deficiencies to license multiple probes in long GN.[5] While we concur with Przepiórkowski (2000) that his prohibition on raising doesn't seem to follow from other principles, we do not consider this too troubling; though we do consider whether it could be reduced to some other property of the grammar a research topic worth studying.

Before closing this section, it is worth noting a confound with the data. Despite the robust judgement that local Genitive of Negation is required, recent corpus work shows that some dialects of Lithuanian don't uniformly enforce local Genitive of Negation (Kozhanov, 2017). Consequently, the problems of over-generation could be tempered by restricting attention to particular dialects of Lithuanian. Modelling the dialects studied by Kozhanov would require generating acc even in local Genitive of Negation environments. In the present work we restrict attention to just those dialects in which local Genitive of Negation is required.

3.4 Revised HTLCG theory

The entire overgeneration paradigm of HTLCG is eliminated if, following Przepiórkowski (2000), raising features are introduced. Then we can invoke the following principles concerning the selection restrictions of types of predicates:

(27) Every non-oblique NP selected by the finite verb includes the argument − for 'doesn't raise'.

(28) Every non-oblique NP selected by the infinitive, except possibly the subject cf. (Przepiórkowski, 2000, p.151), includes the argument + for 'can raise'.

Given the foregoing principles, we provide the following new definition for *ne*:

(29)
$$\lambda t^{(s\rightarrow)_n s} q^s ... q_n^s . ne + t(q...q_n);$$
$$\forall x . T[x := f(x)] \upharpoonright T;$$
$$\lambda P^{e\rightarrow(e\rightarrow)_n t} z^e ... z_n^e x^e . \neg P(x, z...z_n)$$

Here T is the following:

(30) $\{NP(nom,-)\backslash S^{int}, (NP(nom,-)\backslash S)\upharpoonright E, (NP(nom,-)\backslash INF)\upharpoonright NP(x,+)\}$

E is the following:

(31) $\{NP(x,-), NP(x,-)\upharpoonright NP(x,+)_1 ... \upharpoonright NP(x,+)_{n\geq 1}\}$

[5] It is only by the embedded infinitive missing its case feature that long Genitive of Negation is produced. When the embedded infinitive is provided this feature by the lexicon, long Genitive of Negation is blocked (Witko, 2008, fn.38). Moreover, blocking long Genitive of Negation requires the higher probe to be −MULTIPLE. Witkos must further provide further evidence for the existence of the parameter he presumes concerning whether languages license multiple probing–without such evidence there is no principled reason why the multiple probing technology he invokes is present in Polish or seemingly restricted to the genitive dependency.

The distribution for (long) Genitive of Negation given by this version of *ne* is i) finite GN, ii) finite Genitive of Negation with embedded long GN, iii) infinitive Genitive of Negation with the higher verb selecting *ne*-INF. This suffices to predict the correct distribution of GN: it cannot cross sentence boundaries; long Genitive of Negation is possible; multiple Genitive of Negation is possible.[6]

4 Conclusion

We have exemplified the pattern of (long) Genitive of Negation in Lithuanian, providing two accounts of the phenemenon. Since the HPSG account we employ is developed for Polish, the applicability of the account to Lithuanian provides further evidence for the robustness of Przepiórkowski (2000)'s theory of case. While Lithuanian and Polish are both Balto-Slavic languages, their respective positions in different subfamilies raise questions concerning the processes that produced such similar phenomena. The immediate sister languages of Polish have only optional GN, while Latvian, one of the few other Baltic languages, employs Genitive of Negation pretty much exclusively in emphatic contexts (Arkadiev, 2016).[7] In future work we hope to discuss the cross-linguistic typology of Genitive of Negation in greater depth.

The accounts we develop here suggest a surprising convergence between HPSG and HTLCG. This convergence muddies the distinction sometimes made between model-theoretic and proof-theoretic grammar formalisms. We argue that the distinction between these perspectives, while surely useful, can obscure the similarity between the respective practices of constructing grammar fragments. Components of the HPSG account of (long) Genitive of Negation borrow from categorial grammar, while the HTLCG account proposed here very explicitly borrows from the HPSG account.

We close with an open problem: the raising features of both HPSG and, by extension, HTLCG are somewhat unsatisfactory–we have not at present found a way to motivate them or the principles which depend on them on purely empirical grounds. Future work will explore whether these features are dispensable, whether they go proxy for some property of the languages not yet noticed, or whether they just reflect the highly specific distribution of GN.

References

Arkadiev, Peter. 2016. Long-distance genitive of negation in Lithuanian. In *Argument realization in Baltic*, 37–81. Amsterdam: John Benjamins.

[6]We have not discussed how to treat the negation over infinitives which embed infinitives, but this distribution is quite difficult to judge and currently underdocumented.

[7]Slovene is the closest Slavic language we know of which patterns with Polish in Genitive of Negation distribution, while Arkadiev suggests Latgalian, another Baltic language, patterns with Lithuanian in Genitive of Negation distribution (Arkadiev, 2016).

Girard, Jean-Yves. 1987. Linear logic. *Theoretical computer science* 50(1). 1–101.

Kozhanov, K. 2017. Studying variation in case marking: The genitive of negation in aukštaitian dialects of lithuanian. *Baltic Linguistics* 8. 97–114.

Kubota, Yusuke. 2014. The logic of complex predicates. *Natural Language & Linguistic Theory* 32(4). 1145–1204.

Kubota, Yusuke. 2021. HPSG and Categorial Grammar. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* Empirically Oriented Theoretical Morphology and Syntax, Berlin: Language Science Press.

Kubota, Yusuke & Robert D Levine. 2020. *Type-logical syntax*. Cambridge, MA: MIT Press.

Moortgat, Michael. 1997. Categorial type logics. In Johan van Benthem & Alice ter Meulen (eds.), *Handbook of logic and language*, 93–177. Amsterdam: North-Holland.

Moot, Richard & Mario Piazza. 2001. Linguistic applications of first order intuitionistic linear logic. *Journal of Logic, Language and Information* 10(2). 211–232.

Moot, Richard & Symon Stevens-Guille. 2020. Logical foundations for Hybrid Type-Logical Grammars. *arXiv preprint arXiv:2009.10387* .

Moot, Richard & Symon Jory Stevens-Guille. 2019. Proof-theoretic aspects of Hybrid Type-Logical Grammars. In Raffaella Bernardi, Greg Kobele & Sylvain Pogodalla (eds.), *Formal grammar*, 84–100. Berlin, Heidelberg: Springer.

Przepiórkowski, Adam. 2000. Long distance genitive of negation in Polish. *Journal of Slavic Linguistics* 8(1/2). 119–158.

Steedman, Mark. 2000. *The syntactic process*. Cambridge, MA: MIT press.

Witko, Jacek. 2008. Genitive of negation in Polish and single-cycle derivations. *Journal of Slavic Linguistics* 16(2). 247–287.