ПРОТОТИП МЕТАГРАММАТИЧЕСКОГО ОПИСАНИЯ ТРЁХАРГУМЕНТНЫХ КОНСТРУКЦИЙ С МОРФОЛОГИЧЕСКИМ КАУЗАТИВОМ*

В. А. Генералова, С. Петижан
Университет им. Генриха Гейне в Дюссельдорфе

Возможность создания инструмента, позволяющего эффективно анализировать предложения на нескольких языках представляет интерес как для теоретической, так и для прикладной лингвистики. Одним из возможных подходов к этой задаче является создание формальных описаний языковых структур. В настоящей работе предлагается решение, позволяющее описывать синтаксическое строение трёхаргументных конструкций с морфологическим каузативом и его связь с их семантическим представлением. Анализ строится на формализованной версии грамматики Role and Reference Grammar. В качестве формального языка выбран инструмент eXtensible MetaGrammar (XMG), позволяющий описывать синтаксические деревья и семантические фреймы.

Ключевые слова: морфологический каузатив, актантная деривация, аргументная структура, формализованное описание, компьютерное представление.

* Первая версия этой статьи была представлена как доклад на конференции «Типология морфосинтаксических параметров 2020». Исследование является частью проекта TreeGraSP (руководитель — проф. Лаура Каллмайер), поддерживаемого грантом ERC Consolidator grant.

A PROTOTYPE OF A METAGRAMMAR DESCRIBING THREE-ARGUMENT CONSTRUCTIONS WITH A MORPHOLOGICAL CAUSATIVE[*]

Valeria Generalova, Simon Petitjean
Heinrich Heine University of Dusseldorf

The idea of a tool for parsing several languages easily and in parallel is one of the goals for both theoretical linguistics and engineering. One approach would be to create formalized descriptions of linguistic structures. This paper suggests a solution for describing the syntactic organization and its linking to the semantic structure of three-argument constructions with morphological causatives. Our analysis grounds on formalized Role and Reference Grammar. As the description tool, we use eXtensible MetaGrammar (XMG), encoding syntactic structures in the form of trees and semantic structures as decompositional frames.

Keywords: morphological causative, verbal derivation, argument structure, formalized description, computational implementation.

[*] The preliminary version of this paper has been presented as a talk at the conference "Typology of morphosyntactic parameters 2020". This research is part of the TreeGraSP project funded by the ERC Consolidator grant awarded to Prof. Laura Kallmeyer.

1. Introduction

The research area of grammar engineering has grown popular with the overall digitalization of knowledge. There are various projects seeking to create tools for parsing and generating sentences in different languages. Although projects of formalizing individual languages' grammars are more numerous, those including cross-linguistic comparisons are receiving more attention in recent years.

Multilingual linguistic descriptions bridge theoretical knowledge with practical concerns. On the one hand, they help validating previously made observations. It is difficult to formalize a research area lacking consensus upon essential questions. Besides, the scrupulous attention to every detail imposed by the implementation requirements permits to inspect some rarely discussed parts of the theory and perhaps even expand on them. On the other hand, describing several languages in a single project helps to compare data. Moreover, these descriptions can be utilized as a source for other databases and tools.

A grammar engineering project requires several modules. First of all, it has to ground on a linguistic theory, preferably formalized enough to be used for implementation purposes. Secondly, it has to comprise descriptions of languages and/or specific constructions that will produce linguistic analyses. Thirdly, a compiler must be able to read the descriptions and create a desirable output.

Further applications of the generated analyses may be numerous. In an academic context, they can be used as illustrations for some theoretical advancements. One might find it useful to create specific databases of these analyses with an option to compare languages. In this case, the fact that the analyses have been produced within the same framework by the same tools would allow linguistic information to stand out clearly without being blurred by possible methodological differences. The generated analyses can be used in other projects such as corpora or treebanks to create syntactic and semantic annotations automatically. Originating from formal grammars designed by humans, they could surpass the accuracy of machine-learning algorithms. The restriction is the time spent on the development of the linguistic description and the limited coverage of most of the solutions of this kind.

The present paper is concerned with a single step of this pipeline, which is the creation of formalized descriptions from typological and language-specific data. This step is essential as it formalizes the existing information and presents it in a machine-readable form. To ease its development and extension, the resource we develop is not a formal grammar but a more abstract description called metagrammar (see Section 2.4). The place the metagrammar occupies between theory and implementation determines the importance of its architecture. Therefore, we suggest a prototype of a novel architecture designed to minimize the efforts of grammar developers and protect the code from accidental errors.

The remainder of the paper is organized as follows. Section 2 presents the materials and methods used in our study. Primarily, it introduces the language data covered by our prototype. Besides, it gives an overview of Role and Reference Grammar, the theoretical foundation of this research, and of XMG, the language of our metagrammar. Section 3 reviews some existing projects providing multilingual grammatical descriptions. Section 4 demonstrates our solution, presenting various structures and the way they are related. The final Section 5 concludes the paper summarizing the key claims.

2. Materials and methods

In this section, we provide all the background information in order to make the presentation of our solution clear.

2.1. Necessary definitions

In the present paper, we investigate only constructions derived from (syntactically) transitive verbs by means of a morphological device, i. e., affix or clitic. We have narrowed the scope down to this set of constructions for several reasons.

Primarily, we are interested in modeling verbal derivations. To observe the derivation clearly, one would prefer three-argument constructions to two-argument ones. In most languages, there are quite a small number of underived three-argument constructions (e. g., ditransitives with verbs like *give*). Causative predicates based on transitive verbs appear to be three-argument predicates, suiting our goal well. (Once the valency increase mechanisms are well formalized and described, the solution could extend to encompass causatives of intransitives as well.)

However, sometimes causative constructions are bi-clausal, i. e., comprise two verbs, each with its own argument structure. This situation does not belong to the domain of valence-increasing devices, which we are interested in, so we do not cover this type of constructions (*periphrastic,* according to [Song 2013b]). On the other hand, we also eliminate lexical causatives as it is not always clear whether a verbal derivation has taken place or not. In other words, it is not always clear from a verb whether it is a lexical causative, or a transitive demonstrating a causative alternation, or an underived ditransitive.

Saving these complicated issues for further research, we ground the solution we present in this paper on morphological causatives only. We use the definition of morphological causatives suggested by [Nedyalkov, Sil'nitskiy 1969]. They postulate an opposition V_i:V_j, where V_i denotes a situation, and V_j denotes the same situation but having been caused. The term morphological causative is used to denote the V_j, which is morphologically more complex than the V_i. This approach differs from what is suggested by [Song 2013a], where anticausatives (verbs of type V_i that are morphologically more complex than V_j) also count as a subtype of morphological causatives. We follow [Song 2013a], counting as morphological devices suffixes, prefixes and vowel change, as well as clitics.

Lastly, a terminological note about the participants of the three-argument causative construction has to be made. The added argument is most often called **causer**, whereas the former subject is usually called the **causee**. As for the former direct object, there is no consistent tradition of referring to it, as to our knowledge. In this paper, we have chosen the term **theme**, which is in line with the naming convention used for ditransitive constructions.

2.2. Language data

This paper does not present any quantitative typological observations; therefore, there is no need to operate on a typologically representative language sample. However, we aim to demonstrate different strategies of argument marking in three-argument causative constructions. We selected languages accordingly.

2.2.1. Bashkir

In Bashkir, there are two constructions with a morphological causative [Perekhval'skaya 2017], see (1) and (2). In both of them, the causer is nominative and (usually) clause-initial. The theme is accusative (which can sometimes have no overt marking) and usually precedes the verb. The marking of the cause is different: in (2) dative case is used, while in (1) — the ablative.

(1) *Ataj* *bala-lar-**ðan*** *jeläk* *jəj-ðər-a*
father child-PL-**ABL** berry pick.up-CAUS-IPFV
'Children are picking up berries because of the orders of the father.' Lit.: 'The father makes the children pick up berries.' [Perekhval'skaya 2017: 245]

(2)

Bala	*besäj-**gä***	*höt-tö*	*es-er-ä*
child	cat-**DAT**	milk-ACC	drink-CAUS-IPFV

‘The child feeds the cat with the milk.’ [Perekhval'skaya 2017: 241]

The dative case is used in Bashkir for marking recipients in ditransitive clauses (which is not unexpected from a case labeled dative). Ablative is used in a variety of contexts (mainly in complements, though), of which we would like to report one. In passive clauses with an explicit mention of the agent, it can be marked with ablative. Although the preferred way is not a pure ablative, but a prepositional phrase, the preposition *tarafənan* (which nearly corresponds to the English *by*) requires an ablative noun after it.

So, in our sample, Bashkir is a language with multiple causative constructions, one of which can be related to a ditransitive clause, and in the other one, the cause has the same marking as the demoted agent in a passive construction.

2.2.2. Kalmyk

Kalmyk is genetically close to Bashkir, and causative constructions in these languages are very much alike. Namely, Kalmyk also has two options, one being similar to ditransitive clauses (4), and the other relatable to the passive (3) clause. However, there is a different morphological case used in Kalmyk for both these options: instrumental, see (5).

(3)

ezə-n	*ködəlməšč-**är***	*xö*	*al-ulə-v*
master-EXT	servant-**INS**	ram	kill-CAUS-PST

‘The master made the servant kill the ram.’ [Say 2009: 387]

(4)

ekə	*ürə-n-**d**-än*	*xašə*	*id-ül-ǯä-nä*
mother	child-EXT-**DAT**-POSS.REFL	porridge	eat-CAUS-PROG-PRS

‘The mother feeds the child with porridge.’ [Say 2009: 406]

(5)

üüdə-n	*Badm-ar*	*sekə-gdə-v*
door-EXT	Badma-INS	open-PASS-PST

‘The door was opened by Badma.’ [Vydrina 2009: 348]

This slight difference between Bashkir and Kalmyk serves as an illustration of some of our decisions upon the architecture of the metagrammar.

In addition, Kalmyk has a strategy where both non-causer arguments are in the accusative case, see (6). It is important that there has to be an overt marker on the causee and no explicit marking on the theme.

(6) *bagšə* *madn-**igə*** *škol-də* ***kögǯmə*** *soŋs-ul-na*
teacher we-**ACC** school-DAT **music** listen-CAUS-PRS
'At school, the teacher made us listen to music.' [Say 2009: 411]

Without going deep into the discussion of this construction (compare [Say 2009] on Kalmyk with [Letuchiy 2006] on Khakas and [Kulikov 1998] on Tuvinian), we account for this syntactic pattern in our metagrammar as to one of the possible strategies of argument marking in the constructions in question. Causative construction with double accusative marking (or doubling of other morphological devices that are generally used to encode the direct object of a transitive clause) also occurs in other languages. The most prominent examples of them would be Bantu languages (e. g. [Baker et al. 2012]).

2.2.3. Lubukusu

Lubukusu (also known as Bukusu) is a representative of the Bantu family, well known for its valency-modifying constructions. It demonstrates a completely identical marking of both non-causer constituents in constructions with a morphological causative. Such are, apparently, most of the Bantu languages; consider examples (7)–(9) from [Baker et al. 2012: 54].

(7) *Wafula* *a-nyw-esy-a* *Wekesa* *ka-ma-lwa*
Wafula SBJ.C1.TNS-drink-CAUS-FV Wekesa C6-C6-beer
'Wafula made Wekesa drink beer.'

(8) *Wekesa* *a-nyw-esy-ebw-a* *ka-ma-lwa*
Wekesa SBJ.C1.TNS-drink-CAUS-PASS-FV C6-C6-beer
'Wekesa was made to drink beer.'

(9) *Ka-ma-lwa* *ka-nyw-esy-ebw-a* *Wekesa*
C6-C6-beer SBJ.C6.TNS-drink-CAUS-PASS-FV Wekesa
'Beer was made to be drunk by Wekesa.'

In (7), a causative construction is demonstrated. There is no overt case marking on either of the nouns, yet the subject is marked on the verb. Moreover, both non-subject participants of (7), namely the causee Wekesa (class 1) and the theme the beer (class 6), can become subjects of passive constructions, which is indicated by the class marker on the verb after the sbj morpheme in (8) and (9). So, in Lubukusu, the two constituents not only look but also behave in other syntactic contexts identically.

2.2.4. Kabardian

The causative construction in Kabardian somehow mirrors the Bantu construction with double accusative marking. In (10), both the causer and the causee receive ergative marking, while the theme is absolutive.

(10) *ƚ'əźə-m* *ś'āla-m* *pχa-r* *yə-r-yə-ġa-q'ʷətaś*
old.man-ERG boy-ERG tree-ABS 3SG-3SG-3SG-CAUS-cut-PRET-AFFIRM
'The old man made the boy cut the tree.' [Matasović 2010: 50]

This pattern seems to be rather rare ([Dixon 2000] also cites Trumai as having it; to some extent, Qiang can also count as one, see [LaPolla, Huang 2008]). However, there is no reason to discard it when describing typologically varied languages.

2.2.5. Nivkh

According to [Dixon 2000], Nivkh has a very rare type of causative constructions that involve a marker claimed to be unique to this construction only. In other words, the marker *-aχ-* is claimed to be used only in constructions with causatives derived from transitive bases, see (11). [Nedjalkov et al. 1969] also call this morpheme a special marker. However, [Gruzdeva 1998] demonstrates that the same morpheme appears on subjects of converbial clauses, see (12) and (13).

(11) *n'-nanx-∅* *n'-aχ* *pxi-roχ* *vi-gu-d*
1SG-elder.sister-NOM 1SG-DAT/ACC forest-DAT/ADD go-CAUS-FIN
'My elder sister let me go to the forest.' [Gruzdeva 1998: 19]

(12) *hoĝat* *n'yṇ* *čyṇ* *ṇyṇ-d'-ra* *čyṇ-aχ*
then we you look.for-FIN-PART:PRED you-DAT/ACC

p'-ro-guin
REFL-help-CVB:PURP
'We were looking for you in order that you help [us].' [Gruzdeva 1998: 52]

(13) *if* *imṇ-aχ* *als* *p'e-ny-vur* *it-t'*
he they-DAT/ACC berry pick-FUT-CVB:RTL say-FIN
'He said [that] they would pick berries.' [Gruzdeva 1998: 57]

Our solution suggests two ways of treating this strategy, either with relation to converbial clauses, or a totally independent marking pattern.

2.3. Role and Reference Grammar representations

Grammar engineering projects have to ground on some linguistic theories. We use RRG [Van Valin, LaPolla 1997], [Van Valin 2005] as it has been initially developed for typological purposes. Moreover, syntactic and semantic representations of RRG have been formalized by [Osswald, Kallmeyer 2018], which makes the implementation easier.

Syntactic representations in RRG are realized as trees; see Fig. 1. These types of structures are called the "layered structure of the clause" [Van Valin 2005: 3–4]. The upmost level of the tree is SENTENCE, which can comprise one or several CLAUSES. These notions are used in RRG in the same way as in other theories. Lower levels of the tree are specific to RRG. The CORE encompasses the predicate (which is labeled NUCLEUS), and its arguments are usually realized as RPS. Non-arguments (adjuncts) are placed outside of the CORE, in one or several PERIPHERY nodes. In this paper, we look only at monoclausal core structures, so the representations lack some levels for the sake of brevity.

Semantic representations in classical RRG are realized as predicate logic structures [Van Valin 2005: 42]. In the formalized version, decompositional frames are used [Lichte, Petitjean 2015] as they contain the maximum of the necessary information in a concise form. Besides, deriving frames from one another is easy, which is crucial for implementation purposes. Some sample frames are shown in Fig. 1: each of the (sub) frames starts with the type of the predicate and contains its role structure. Arguments of the semantic structure of the predicate are referenced as labels. The same labels are used in the syntactic structure, allowing comprehensive syntax-semantics linking.

2.4. XMG language

A metagrammar [Candito 1996] is an abstract and compact description of a grammar. The key idea of a metagrammar is to express linguistic generalizations in order to factorize the redundant parts of the grammar (e. g. see discussion in [Clément, Kinyon 2003]). Therefore, developing a metagrammar comprises two main phases: describing smaller units (*rule fragments*) and assembling them to form the grammar rules. Once compiled, a metagrammar produces the underlying grammar, which can be used for parsing or language generation.

The framework eXtensible MetaGrammar (XMG, [Crabbé et al. 2013], and XMG-2, [Petitjean et al. 2016] offers a description language that allows generating tree grammars based on any linguistic theory. This metagrammatical language is inspired by constraint programming, for the definition of the rule fragments, and logic programming, for assembling the fragments into rules.

A rule fragment, encapsulated into a unit called *class*, can be composed of a syntactic structure (a partial tree), a semantic frame fragment, or both. The description of a tree fragment consists of the enumeration of dominance and precedence relations, which have to hold between the nodes composing it. The semantic frames are described using typed feature structures [Lichte, Petitjean 2015]. Classes are organized in an inheritance hierarchy, which determines how the fragments should be assembled to form the grammar rules. The creation of the class hierarchy is made possible by three mechanisms: inheritance, conjunction and disjunction. When inheriting a class, its entire content is added to the description of the current class. The operation of conjunction is used to inherit the content of several classes cumulatively, whereas disjunction allows expressing alternatives between different classes to be imported.

Conjunction and disjunction can also be used within classes to express combinations and alternatives at a more fine-grained level (between partial syntactic descriptions, for instance). As disjunction introduces non-determinism, XMG uses backtracking to generate the rules resulting from all possible alternatives expressed in the classes. In our study, we use this mechanism to account for several typological varieties of a given construction (see Section 4.1).

In addition to this minimal logic language and backtracking, XMG borrows from logic programming tools (such as Prolog) the extensive use of unification. All structures described in classes (syntactic nodes, feature structures) can be associated with Prolog-like variables. Unifying these variables triggers the unification of the structures, allowing fragments to complete the linguistic information missing in others. This process is key to combining classes of several kinds in our architecture; see Section 4.3.

3. Review

Previously, there have been attempts to create formalized descriptions valid for multiple languages to allow further implementations. In this section, we would like to broadly review some of the key papers having a significant influence on the conception of our solution.

Several approaches to creating multilingual metagrammars have been suggested by Alexandra Kinyon and her collaborators. Namely, [Clément, Kinyon 2003] suggest generating LFG (Lexical Functional Grammars) and TAG (Tree-Adjoining Grammars) for French and English. This paper also has a methodological discussion concerning the advantages of the metagrammatical approach over redundancies brought by LFG rules. In a later paper, [Kinyon et al. 2006] demonstrate a broader coverage of languages, including five Germanic languages and Kashmiri, yet the discussion in the paper is limited to German and Yiddish only. The paper is technically driven and demonstrates a thorough implementation of early XMG principles. The feature structures and the mechanisms of their unification are developed in much detail.

A totally different in terms of theoretical foundations and exploited tools is the LinGO Grammar Matrix project, running since 2002 at the University of Washington (see [Bender et al. 2002]). The descriptions are inspired by HPSG structures, which drives the whole analysis more towards the semantic side, letting syntactic representations look secondary. Nevertheless, the coverage of the studies related to this project is very impressive.

Among others, [Curtis 2018] is of major importance for the present research. This paper describes a library (a freely addable module of the Grammar Matrix) covering valence-modifying constructions. The central phenomenon that is modeled is the application of lexical rules. The results are illustrated with data from 8 typologically varied languages. As already mentioned, the syntactic side of the analysis is somehow shaded by the semantic one, which we consider a drawback in comparison to our approach. Also, the typological review suggested in [Curtis 2018: 13–22] seems to capture most general trends overlooking some minor varieties.

Although the realization of grammatical phenomena in our solution can hardly be compared to the above-mentioned due to differences in theoretical approaches, the very general architecture is similar. In the Grammar Matrix, there is a questionnaire eliciting all the necessary data about the grammar of a particular language. This module corresponds to the Language plugin notion (see Section 4.2). We find it advantageous to our plugins that they require less specific information than [Curtis 2018: 40] and thus are accessible to a broader audience of grammar developers.

Nevertheless, this implementation is a high-standard solution, especially powerful as a part of a larger successful project. Taking the best from it and from theories presented in Section 2, our prototype tends at least to show an equally great potential along with some keen attention to previously overlooked details.

4. Solution

This paper aims to present a solution for describing three-argument constructions with morphological causatives on the metagrammatical level. This solution has to cover a wide range of constructions encountered in languages of the world (see Section 2.2) and be expandable onto more languages once needed.

In our solution, we would like to keep cross-linguistic generalizations severed from language-specific information. This general architecture is critical for creating a solution that can be expanded in both dimensions and cover more constructions and more languages. In this section, we first present the design of metagrammatical classes describing constructions, called **construction classes**. The same classes are used for describing all languages in the sample and can be reused when other language phenomena are covered. Afterward, we present metagrammatical classes where language-specific information is stored, the so-called **language plugins**. All of them are designed in a similar way to ensure compatibility with each other and with any construction class. Finally, we couch in detail the mechanism of bringing classes of both types together to parse each particular sentence in each particular language correctly.

For the sake of clarity, we present as illustrations only prototypical situations. Namely, we talk about morphological cases, but in reality, other devices, such as clitics or vowel changes, are also covered. The full version of the metagrammar described in this paper is made available online[1].

4.1. Construction classes

Our metagrammar accounts for one single construction, a three-argument causative. Therefore, there is only one construction class. However, it is not composed from scratch but built of other classes describing construction fragments.

First, we would like to consider the structure of the causative predicate. It is made of at least two morphemes: the lexical root (or stem) and the causative affix. Each of these morphemes has its own structure. Each lexical root has its valency and role structure. These are used in non-derived verbal constructions (e. g. in a simple active non-causative clause). The causative affix always has the same frame structure, as it adds the cause subframe to the causation frame. It has no own syntactic structure, but it increases the valency of the whole sen-

[1] Please visit the GitHub repositrory for the actual version of the metagrammar: https://github.com/fleurdhelice/TMP2020_demo.

tence. In terms of RRG tree grammar, it creates a new RP node below the core. The addition of the causative affix is viewed as a regular grammatical operation of valency increase and applies to any base verb. Any constraints on this procedure encountered in individual languages will be brought in by Language plugins (see Section 4.2).

In Fig. 1, trees and frames are shown for sample intransitive, transitive and ditransitive verbs. Note that the syntactic structure of the predicate is always the same, but the semantic representations vary.

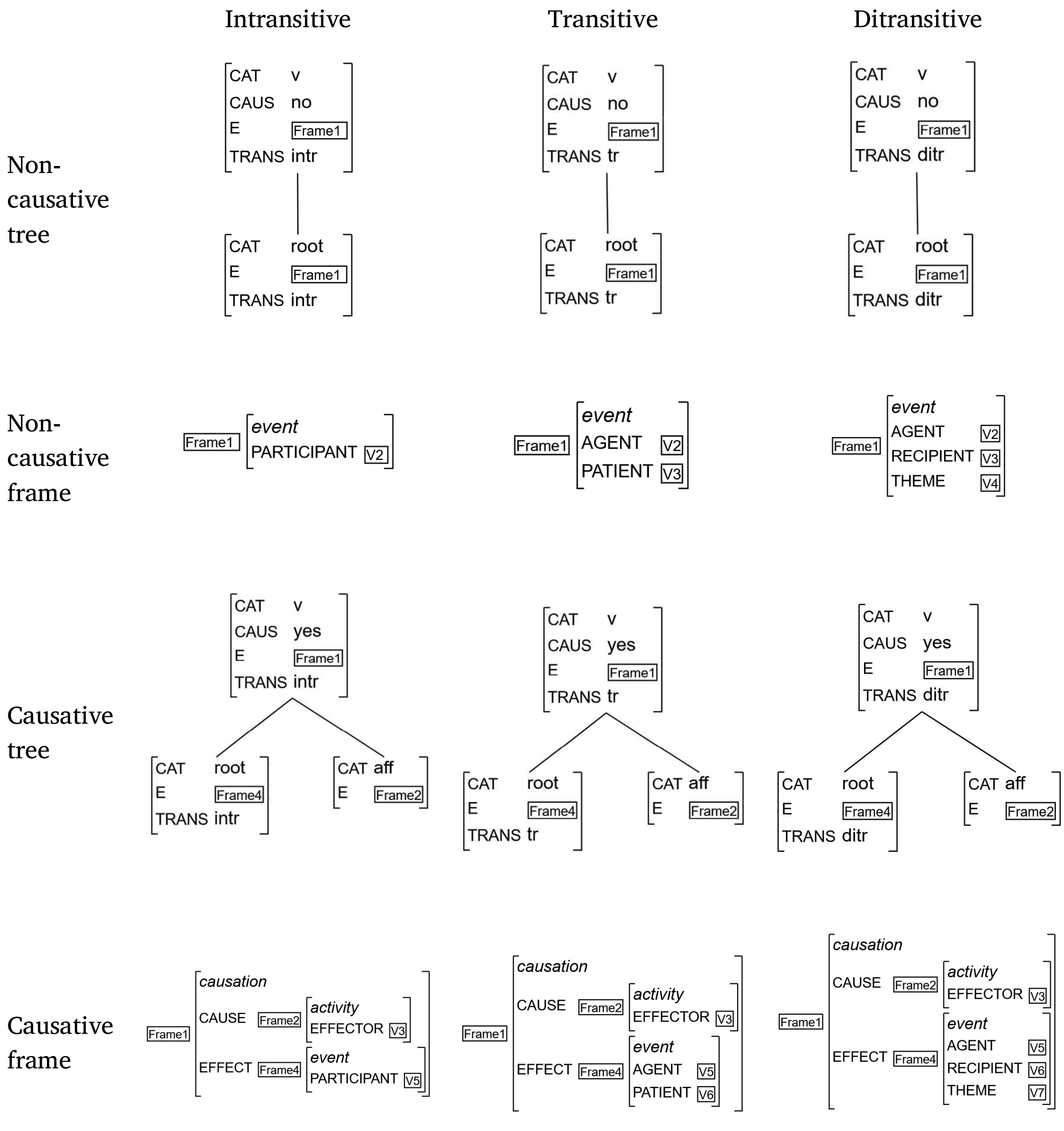


Figure 1. Syntactic and semantic structures generated by the class *Verb*

Now, let us consider the derivation of the syntactic structure. In Fig. 2, the syntactic representation of a transitive clause is shown. The two RP nodes are the subject and the direct object. Each of them has an identifier (the *i* feature). These labels are used to reference specific nodes within frames and other classes. Another feature defined for an RP is the morphological case. In Fig. 2, assigned cases are not visible. However, in the code, the label *V7* always corresponds to the case used for the syntactic subject (in RRG terms, the PSA, privileged syntactic argument) and the label *V8* — for the direct object. So, the morphological case assigned to each constituent in the non-causative transitive clause is strictly defined in the construction class.

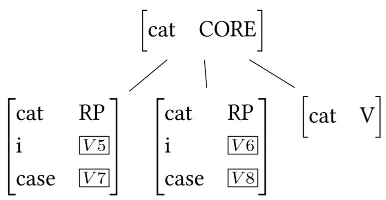


Figure 2. The syntactic tree
for a non-causative transitive clause

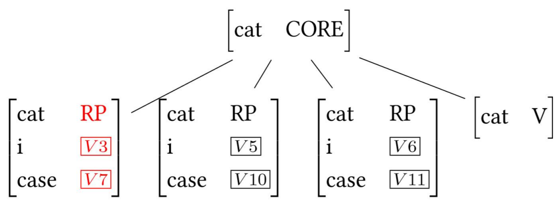


Figure 3. The syntactic tree
for a causative of a transitive clause

The situation with causative constructions is not that straightforward, as there are numerous strategies for marking the arguments. Fig. 3 shows a causative construction derived from Fig. 2. There is one RP added (see new identifier). This RP corresponds to the causer. It has been generally attested that in all languages, the causer receives the case that is used for a PSA in a non-causative transitive clause. This case is assigned to the new RP as soon as it appears in the clause (consider label *V7* on the new constituent). As for the other two RP nodes, it is not clear yet which cases they are assigned. It depends on individual language properties and thus cannot be fixed within the construction class. For this reason, free variables are assigned as a value of the case feature to these constituents. At a later stage, they unify with case values imposed by other constraints.

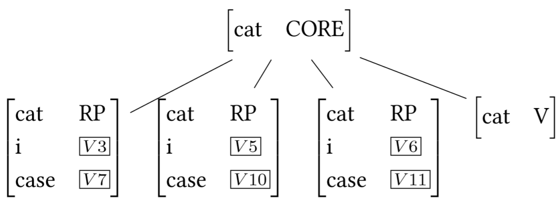
To this point, we have shown universal structures. There are, however, some characteristics of the constructions in question that hold for only some groups of languages. On the one hand, we cannot have them fixed as, for example, the morphological case of the causer, because there are several ways how each of these characteristics can be realized. On the other hand, there are salient linguistic generalization, grouping languages according to the value of these characteristics. Given that, our solution is demanded to account for some variation dependent on a particular characteristic.

To handle this, we suggest using a disjunction inside the universal construction class. For each alternative listed in the disjunction, XMG creates a new possible syntactic structure for the grammar, resulting from the combination of the existing description and the one expressed in the alternative. Whether the resulting structure will be part of the grammar or not depends on further feature unification. All the information that is used universally (e. g., node labels or information about morphological cases) is put in the first part of the class description. Afterward, there is a set of disjoint claims. Each component of disjunction can comprise one or more conjoint claims. In the first claim, the feature across which the constructions vary. In the next claims of each of the disjoint components, the specific properties of each variety are described.

A clear example of this kind of characteristics is the word order. Indeed, there is no universal word order, but the number of groups a language may fall to is fixed and small. In the sample listed in Section 2.2, there are only two word orders: SVO (Lubukusu) and SOV (all the others). So, we have to account for two disjoint options in the metagrammar.

In the description of any tree, two main types of constraints determine its shape: dominance and precedence. In all causative constructions examined in the present paper, the three RP nodes are dominated by the core node. So, the dominance constraints are formulated in the common part of the class. As for precedence, the constraints are different. In Fig. 4, one can see two components that are disjoint and the syntactic structures that result in each case (the operator “≫” indicates linear precedence).

```
[?LanguageFeatures=[wordorder=sov;
?NpCauser >> ?NpCausee;
?NpCausee >> ?NpTheme;
?NpTheme >> ?V ]]
```
(e. g. Bashkir)




```
[?LanguageFeatures=[wordorder=svo;
?NpCauser >> ?V;
?V >> ?NpCausee;
?NpCausee >> ?NpTheme ]]
```
(e. g. Lubukusu)

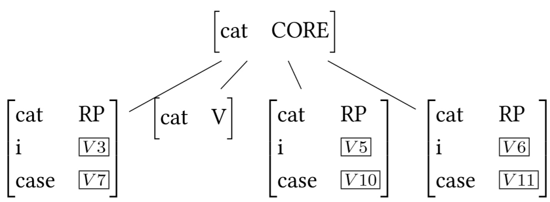


Figure 4. The example of disjoint structures

In each language plugin, this feature is assigned one single value. Once the construction class is intersected with the language plugin (see Section 4.2), the single option is chosen, and the sole description of the construction is applied to the sentence.

4.2. Language plugins

Apart from Construction classes, our metagrammar comprises Language plugins. In these XMG classes, there are no syntactic or semantic descriptions. Instead, there is one variable *LanguageFeatures* with a wide range of features. The set of features is supposed to be fixed and identical for any language. This would allow us to intersect any Language plugin with any Construction class. Having language properties as features is advantageous for one more reason. All the feature values are type-checked, which minimizes input errors.

4.2.1. Inventory features

Language plugins have two main objectives. Firstly, they have to list constructions available in each language. As shown in Section 2.2.1, there can be several causative constructions within the same language. Typological studies (e. g. [Dixon 2000]) show that there is a closed list of constructions with different marking patterns. So, they can be formulated as boolean features, each receiving positive or negative value.

Importantly, we would like to relate marking in causative constructions to marking in other constructions. This is necessary to keep patterns of constructions apart from the morphological peculiarities of each particular language. For example, both Bashkir and Kalmyk have a construction, where the causee receives the same morphological case as the demoted agent in a passive construction. Therefore we postulate a single boolean feature that stands for this kind of construction and set its value to positive for both Bashkir and Kalmyk. In Fig 5., one can see variable names and their values for each language.

In the Construction class, there is a disjunction encoding the causee marking depending on the value of the above-described boolean features, see Fig. 6. When the RP for the causee is declared in the construction class, it is assigned a free variable `?VarCauseeCase`, which serves as a label before further information is given. In the disjunction part, each conjoint claim consists of the postulation of the positive value of one boolean feature and the reassignment of the value of the variable `?VarCauseeCase`. This is done by pulling the respective feature from the variable `?LanguageFeatures`, which is declared in language plugins. The dot operator is used to access the value of a given attribute in a feature structure introduced by a variable. No overt assignment of morphological cases is done at this point because they differ across languages. However, the marking of the causee is related to features encoding the marking of other syntactic positions in each given language.

Feature	Description of the marking pattern	Bashkir	Kalmyk	Lubukusu	Kabardian	Nivkh
CauseeLikeRecip	causee is marked identically to the recipient in a ditransitive clause	+	+			
CauseeLikeDemAg	causee is marked identically to the demoted agent in a passive clause	+	+			
CauseeLikeDo	causee is marked identically to the direct object in a transitive non-causative clause		+	+		
CauseeLikePsa	causee is marked identically to the PSA in a transitive non-causative clause				+	
CauseeSpecial	causee receives a special marker					+

Figure 5. Boolean features for varieties of causative constructions

```
class CausativeConstruction
(...)
{<syn>{(...);
    node NpCausee [case=?VarCauseeCase]; (...);

    {?LanguageFeatures.ConstrCauseeLikeRecip = yes;
    ?VarCauseeCase=?LanguageFeatures.recipCase}

    | {?LanguageFeatures.ConstrCauseeLikeDemAg = yes;
    ?VarCauseeCase=?LanguageFeatures.demAgCase}

    | {?LanguageFeatures.ConstrCauseeSpecial = yes;
    ?VarCauseeCase=?LanguageFeatures.causeeCase};
    (...) }}
```

Figure 6. A piece of the code demonstrating the disjunction between boolean features

One might note the last disjunction option listing the feature `causeeCase`, which is not related to any of the other constructions. Indeed, this is the fallback strategy for languages lacking data about some constructions or having a really particular marker for the causee. In our sample, Nivkh is an example of such a language. If we follow [Nedjalkov et al. 1969], we have to postulate the exceptional causee marker, which is not used elsewhere, and thus activate this fallback strategy. However, if we follow [Gruzdeva 1997], we could add a variety "causee marked in the same way as the subject of the converbial clause" and a feature `cvSubCase` to account for the morpheme -*aχ*-. Once this strategy is encountered in another language of the world, our metagramar is already able to handle it. Nevertheless, given the lack of data, we do not eliminate the fallback option from the list of disjunctions.

4.2.2. Morphological features

Encoding morphological properties of a language is the second main objective of a language plugin. The formalization work in this area consists of converting traditional grammar descriptions into typed features. Namely, for the purposes of this study, one needs to list all morphological cases (or other devices used for argument marking) and relate them to their most common usage contexts.

Here is a sample list of these features:

`psaCase` — the case used for the privileged syntactic argument (syntactic subject) in an active transitive clause;

`doCase` — the case used for the direct object of a transitive clause;

`recipCase` — the case used for marking Recipient in a ditransitive clause;

`demAgCase` — the case for encoding the demoted agent in a passive transitive clause, etc.

Each feature of this kind takes one morphological case from the list available in this language as its value.

Identical names of the features allow accessing them from any construction class and make syntactic structures independent from morphological input. For example, values of the feature `demAgCase` are different in Bashkir and Kalmyk, but both of them are invoked within the same construction variety, namely, the `ConstrCauseeLikeDemAg` (see Fig. 5 and Fig. 6).

4.2.3. Summary

The Language plugins in our architecture comprise one single structure (accessible via a variable) with a great number of type-checked features. Boolean features affirm or negate the presence of a given variety of the causative construction in a language. In the expanded grammar, one would add similar lists for other types of constructions covered. Categorial features encode the morphology of a language. Determining the inventory of necessary attributes and their relation to each other requires typological knowledge. Once the templates for language plugins are set, values to these features can be assigned by language specialists as long as this operation does not require advanced linguistic knowledge. This accessibility makes the barrier of entry to the metagrammar relatively low, and the potential of the metagrammar to expand on a sound number of languages relatively high.

4.3. Intersection

The final step leading to creating descriptions applicable to real sentences is the intersection of a construction class with a language plugin. At this stage, all feature values unify, and all slots left empty receive a determined value. As for disjunctions, only those parts are left that comprise statements about the features conforming to the values of these features in the Language plugin.

Technically, this is realized as *inheritance* between classes. Under this term, we understand the idea of acquiring the whole structure of one class to add something on top of it within another class. For that to be possible, structural parts of a class (variables, nodes etc.) have to be named and exported. Two entities with the same name automatically count as a single one, making it possible to put additional constraints in a new class on a node imported from elsewhere. As for the features, their values unify in that respect that a determined value (the one listed in the preamble of the metagrammar as acceptable) percolates through structures and substitutes all the variables that have been referencing it. In other words, all general and placeholder values are substituted by a determined value if it is assigned elsewhere in the inherited or inheriting classes. In case of a conflict of two determined values, no unification happens.

The intersection of a construction class and a language plugin is an extreme case of inheritance. The resulting class provides no additional descriptions, apart from importing the two classes. One might consider this step as pure unification performed apart from defining the structures. The intersected classes can be created automatically, leaving this process free from possible human errors.

5. Conclusion

We have suggested a solution that presents linguistic analyses of sentences in two dimensions, creating syntactic and semantic representations. It requires only a sentence in a language as an input to generate complete analyses based on theoretical and typological reasoning. It captures cross-linguistic generalizations due to the architecture of Construction classes, which is designed to account for many varieties of structures from the beginning. The flexibility and the modularity of the architecture are provided by the XMG language and ensure easy further development.

The current prototype of the solution has no power in terms of quantitative typology, although it might turn out to be a useful tool once more languages are covered. To add a new language in the metagrammar, one needs to know what causative constructions are allowed and how morphological cases are used.

The overall idea of formalizing a well-studied field of typological knowledge has brought its insights, too. Namely, increasing the valence of the sentence has required different solutions for the syntactic and the semantic dimensions. This underlines the importance to distinguish between syntactic and semantic valency in non-computational descriptions of languages as well. Besides, the atomic approach to causative constructions has revealed some overlooked areas. To our knowledge, little has been done in the direction of relating marking in causatives to other (except for ditransitive) constructions in a language. We find that a thorough look into these relations could tell more about what axes of similarity (cf. comparative concepts by [Haspelmath 2010]) are there in typologically varied languages.

Abbreviations

3 — 3rd person; ABL — ablative; ABS — absolutive; ACC — accusative; ADD — additive; AFF — affix; AFFIRM — affirmative; C1 — noun class 1; C6 — noun class 6; CAUS — causative; CVB — converb; DAT — dative; DITR — ditransitive; ERG — ergative; EXT — stem extension; FIN — finite; FUT — future; FV — final vowel; INS — instrumental; INTR — intransitive; IPFV — imperfective; NOM — nominative; PART — particle; PASS — passive; PL— plural; POSS — possessive; PRED — predicative; PRET — preterite; PROG — progressive; PRS — present; PST — past; PURP — purpose; RP — reference phrase; RTL — re-telling; SBJ — subject; SG — singular; TNS — tense; TR — transitive.

References

Baker et al. 2012 — Baker M, Safir K, Sikuku J. Sources of (a)symmetry in Bantu double object constructions. Proceedings of the 30th West Coast Conference on Formal Linguistics, 2012. P. 54–64.

Bender et al. 2002 — Bender E.M., Flickinger D., Oepen S. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics. Carroll J, Oostdijk N, Sutcliffe R. (eds.) Taipei, Taiwan, 2002. P. 8–14.

Candito 1996 — Candito M. A principle-based hierarchical representation of LTAGs. Proceedings of the 16th International Conference on Computational Linguistics, 1996. P. 194–199.

Clément, Kinyon: 2003 — Clément L, Kinyon A. Generating parallel multilingual LFG-TAG grammars from a metagrammar. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 2003. P. 184–91.

Crabbé et al. 2013 — Crabbé B., Duchier D., Gardent C., Roux J.L., Parmentier Y. XMG: extensible metagrammar. Computational Linguistics, vol. 39(3), 2013. P. 591–629.

Curtis 2018 — Curtis C.M. A parametric implementation of valence-changing morphology in the LinGO Grammar Matrix. Master’s thesis; University of Washington, 2018.

Dixon 2000 — Dixon R.M.W. A typology of causatives: form, syntax and meaning. Changing valency: Case studies in transitivity. Dixon R.M.W., Aikhenvald A.Y. (eds.). Cambridge, Cambridge University Press; 2000. P. 30–83.

Gruzdeva 1998 — Gruzdeva E. Nivkh. Languages of the World/Materials, vol. 111. Lincom Europa, München and Newcastle, 1998.

Kinyon et al.: 2006 — Kinyon A., Rambow O., Scheffler T., Yoon S.W., Joshi A. The metagrammar goes multilingual: A cross-linguistic look at the V2-phenomenon. Proceedings of the Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms, 2006. P. 17–24.

Kulikov 1998 — Kulikov L. Causative constructions in Tuvinian: Towards a typology of transitivity. The Mainz Meeting: Proceedings of the Seventh International Conference on Turkic Linguistics. Johanson L, et al. (eds.). 1998. P. 258–64.

LaPolla 2008 — LaPolla R.J., Huang C. A grammar of Qiang: With annotated texts and glossary. Mouton de Gruyter, 2008.

Letuchiy 2006 — Letuchiy A. Case marking, possession and syntactic hierarchies in Khakas causative constructions in comparison with other Turkic languages. Case, Valency and Transitivity. Kulikov L., Malchukov A., de Swart P. (eds.), vol. 77, 2006. P. 417–441.

Lichte, Petitjean 2015 — Lichte T, Petitjean S. Implementing semantic frames as typed feature structures with XMG. Journal of Language Modelling, vol. 3(1), 2015. P. 185–228.

Matasović 2010 — Matasović R. Kabardian causatives, reflexives, and case marking domains. Suvremena lingvistika, vol. 36(69), 2010. P. 45–64.

Nedjalkov et al. 1969 — Nedyalkov V.P., Otaina G.A., Xolodovič A.A. Morfologičeskiy i leksičeskiy kauzativy v nivxskom jazyke [Morphological and lexical causatives in Nivkh]. Typology of causative constructions: Morphological causative. Xolodovič A.A. (ed.). Leningrad, Nauka, 1969. P. 179–199.

Nedyalkov, Sil’nitskiy 1969 — Nedyalkov V.P., Sil’nitskiy G.G. Tipologija morfologičeskogo i leksičeskogo kauzativov [Typoogy of morphological and lexical causatives]. Typology of causative constructions: Morphological causative. Xolodovič A.A. (ed.). Leningrad, Nauka, 1969. P. 20–50.

Osswald, Kallmeyer 2018 — Osswald R, Kallmeyer L. Towards a formalization of role and reference grammar. Applying and expanding Role and Reference Grammar (NIHIN Studies). Kailuweit R, Staudinger E, Künkel L. (eds.) Freiburg, 2018. P. 355–78.

Perekhval'skaya 2017 — Perekhval'skaya E.V. Kauzativnye konstrucii v baškirskom jazyke [Causative constructions in Bashkir]. Acta Linguistica Petropolitana Works by Institute of Linguistic Studies, vol. 13(1), 2017. P. 231–254.

Petitjean et al. 2016 — Petitjean S., Duchier D., Parmentier Y. XMG-2: describing description languages. International Conference on Logical Aspects of Computational Linguistics. Springer, 2016. P. 255–72.

Say 2009 — Say S.S. Argumentnaja struktura kalmyckix kauzativnyx konstrukcij [Argument structure of Kalmyk causative constructions]. Acta Linguistica Petropolitana Works by Institute of Linguistic Studies, 2009. P. 387–464.

Song 2013a — Song J.J. Non-periphrastic causative constructions. The World Atlas of Language Structures Online. Dryer M.S., Haspelmath M. (eds.). Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013.

Song 2013b — Song J.J. Periphrastic causative constructions. The World Atlas of Language Structures Online. Dryer M.S., Haspelmath M. (eds.). Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013.

Van Valin: 2005 — Van Valin R.D. Exploring the syntax-semantics interface. Cambridge University Press, 2005.

Van Valin, LaPolla 1997 — Van Valin R.D., LaPolla R.J. Syntax: Structure, meaning, and function. Cambridge University Press, 1997.

Vydrina 2009 — Vydrina A. Upotreblenie pokazatel'a passiva v kalmyckom jazyke [Usage of passive marker in Kalmyk]. Acta Linguistica Petropolitana Works by Institute of Linguistic Studies, 2009. P. 347–386.

Статья поступила в редакцию 16.11.2020

The article was received on 16.11.2020

Валерия Антоновна Генералова

аспирант, Университет им. Генриха Гейне в Дюссельдорфе; участник проекта TreeGraSP, поддерживаемого грантом ERC Consolidator grant

Valeria Generalova

Ph.D. student, Heinrich Heine University of Düsseldorf; performer of the TreeGraSP project funded by the ERC Consolidator grant

generalo@hhu.de

Симон Петижан
Ph.D.; постдок, Университет им. Генриха Гейне в Дюссельдорфе; участник проекта TreeGraSP, поддерживаемого грантом ERC Consolidator grant

Simon Petitjean
Ph.D.; postdoctoral researcher, Heinrich Heine University of Düsseldorf; performer of the TreeGraSP project funded by the ERC Consolidator grant

petitjean@phil.uni-duesseldorf.de