A cross-linguistic account of subordinator and subordinate clause position

Olga Zamaraeva[1], Kristen Howell[1], and Emily M. Bender[1]

[1]Department of Linguistics, University of Washington, Seattle, WA, U.S.A.
{olzama, kphowell, ebender}@uw.edu

Abstract

We describe additions to the Grammar Matrix customization system to support the development of grammar fragments for languages where the addition of subordinate clauses introduces complexity to the analyses of head and complement positions. Our analysis looks at complementizer and subordinator attachment, which does not necessarily follow the pattern of headedness for verbs in the language, and at extraposition of clausal complements. The Grammar Matrix is developed in the context of the DELPH-IN ecosystem which requires rules to have fixed number and order of daughters. Despite these constraints, our analysis covers the range of variation documented in the typological literature with the addition of just two features and one to two additional phrase-structure rules for any given language.

1 Introduction

The LinGO Grammar Matrix customization system (Bender, Flickinger, & Oepen, 2002; Bender, Drellishak, Fokkens, Poulson, & Saleem, 2010) is a typologically informed toolkit for the creation of precision implemented grammars, as well as a platform for cross-linguistic hypothesis testing. It provides analyses for a variety of grammatical phenomena but previously did not support subordinate clauses. The analysis described here forms part of our new libraries for adverbial clausal modifiers (Howell & Zamaraeva, in press) and declarative clausal complements both of which have implications for the analysis of word order, as the order of subordinators/complementizers within their clauses need not follow that of other head-complement constructions, on the one hand, and clausal complements need not appear in the canonical object position on the other.[1] In keeping with the Grammar Matrix's goal of modeling the typological range of each phenomenon it includes, we develop analyses of these variations that are interoperable with the analysis of the different basic word order patterns.

Below we describe the context of our work in terms of the DELPH-IN resources and the ways in which these resources constrain the space of possible analyses (§2), and then sketch the range of the attested linguistic systems (§3), before presenting our analysis (§4). The crux of the analysis is a small number of variations on the familiar head-complement (HCR) and head-subject (HSR) rules which are added depending on the particular combination of basic word order and subordinate clause facts as well as two features, INIT and EXTRA, used to constrain the application of the additional rules. For this short abstract, we only give one detailed example. Finally, we describe our three-step evaluation process (§5).

[1]We have also extended the Grammar Matrix to handle cases where the matrix clause and subordinate clause have different word order, such as V2/V-final in some Germanic languages. For that functionality, we integrate the analyses developed by Fokkens (2014).

2 Background: DELPH-IN Formalism and the Grammar Matrix

The analysis presented here was developed as an extension to the LinGO Grammar Matrix. As such, it is couched within the DELPH-IN joint reference formalism (Copestake, 2002), a fairly restrictive variant of HPSG formalization, developed to balance expressivity with computational efficiency. It does not allow relational constraints such as shuffle operators (Reape, 1994) and furthermore requires that the number and order of daughters of each phrase structure rule be fixed in the definition of the rule, precluding systems that separate immediate dominance from linear precedence (e.g. Engelkamp, Erbach, & Uszkoreit, 1992). One implication of this is that if a language allows both head-initial and head-final orders within a phrase structure type (e.g. head-complement), then the grammar for that language will need to include two separate variants of the rule. Similarly, two variants will be required if complements can attach both low and high with respect to subjects. Finally, fixed arity means that the head-complement rules can't realize all daughters at once, but instead are always binary branching and realize daughters one at a time.

The Grammar Matrix customization system (Bender et al., 2002, 2010) includes a web-based questionnaire that elicits typological and lexical information about a language from a linguist-user and a back-end customization script that outputs extensions to the Matrix core grammar according to the elicited specifications. The resulting grammar fragments are suitable for both parsing and generation and map between surface strings and Minimal Recursion Semantics (Copestake, Flickinger, Pollard, & Sag, 2005) representations. In addition to facilitating the development of broad-coverage grammars for practical applications, the customization system also can be used in linguistic hypothesis testing (Bender, Flickinger, & Oepen, 2008). The Grammar Matrix allows the user to select among 10 options for basic word order (Bender & Flickinger, 2005; Fokkens, 2014).[2] The choice determines the number and shape of the HCR and the HSR.[3] We build on this analysis to implement our analysis for word order in complex sentences.

3 Word order in complex sentences

We look at clausal complements[4] and clausal modifiers marked by subordinators (including complementizers) and at the distribution of clausal complements within sentences. The order of the subordinator with respect to its complement (a subordinate clause) does not necessarily pattern the same way as the verb and its object. Furthermore, clausal complements are often disallowed or dispreferred in sentence-initial and especially sentence-medial position (Noonan, 2007). The following examples from Uzbek [uzb] illustrate how this SOV language nonetheless has a complementizer *ki* which precedes its complements (while also having another complementizer *deb* which observes the head-final order) and allows SVO order to sometimes (1) but not always (2) avoid center embedding of clausal complements.

(1) Men bilamen ki bu ɔdam ǰoǰa-ni oğirladi
I know-1sg COMP this man chicken-obj stole-3sg
'I know that the man stole the chicken.' [uzb] (Noonan, 2007)

(2) Xɔtin bu ɔdam ǰoǰa-ni oğirladi deb dedi
woman this man chicken-OBJ stole COMP said.3SG
'The woman said that the man stole a chicken.' [uzb] (Noonan, 2007)

To establish the possible constructions in a given language, the user-linguist must indicate in the questionnaire whether the complementizer is obligatory or optional and if it attaches on the left or on the right

[2] SVO, SOV, OSV, OVS, VSO, VOS, V-final, V-initial, free, and V2.

[3] The user's choice here determines basic word order; a fixed word order choice can still be paired with specific word-order altering constructions.

[4] In this work, we start with clausal complements that contain a complete proposition and leave other types for future work.

as well as whether the clausal complement is extraposed or not (or if both options are possible). In the next section we present the analyses that are necessary to produce a grammar that will correctly reflect the word order in complex sentences for most languages described in terms of these parameters.[5]

4 Analysis

Our analysis relies on an additional HCR and, in some cases, an additional HSR constrained for two boolean features INIT and EXTRA. INIT (a HEAD feature) is used on lexical types and on the head daughters of phrase structure rules. It is used to account for word order variations associated with subordinator attachment and with extraposition of objects from OV languages; [INIT +] is associated with head-initial rules and [INIT −] with head-final rules. EXTRA is used to handle extraposition of clausal complements in VO languages and is constrained on COMPS list elements of clausal-complement selecting heads and on the non-head daughter of HCRs.[6] While similar features have been used before (Keller, 1995), we incorporate them into a new customization logic so that sets of correct constraints are emitted automatically based on user choices.

Extraposition from objects in generally OV languages can be accounted for by an additional head-initial HCR and the INIT feature. The general HCR's head daughter is [INIT −], the new head-complement rule's head daughter is [INIT +], and the lexical types are constrained accordingly.

The INIT feature is irrelevant for **extraposition in VO orders** since both the general and the additional HCR must be head-initial. Instead we posit the EXTRA feature. It is a boolean feature for which we can constrain the complement of the clausal complement-taking verb. We can then say that the general HCR, which will be triggered for non-clausal complements, constrains its non-head daughter to be [EXTRA −]. The additional HCR which is needed for clausal complements insists on low subject attachment (with [SUBJ ⟨ ⟩]) and requires its non-head daughter to be [EXTRA +].

Whenever **subordinators** cannot use the same head-complement rule as transitive verbs, we use an additional HCR and the INIT feature to preclude spurious parses. Finally, If the analysis of subordinate clauses requires an additional phrase structure rule, all **lexical types** should be constrained with respect to INIT and EXTRA so that they can only go through the appropriate rule.

Consider one sample customization scenario.[7] Suppose the user says the language is VOS and has extraposition. Our analysis of **extraposition in VOS languages** requires two additional rules: a HCR and a HSR (see Figs. 1–2). This is because we need to license a VP constituent (Fig. 1) while also licensing low subject attachment (Fig. 2).[8] At the same time, we want to preclude any constituent from being spuriously licensed by more than one rule, so we constrain the head daughters' valence lists.

5 Evaluating the libraries

The number of possible relevant choices combinations ('pseudolanguages')[9] to cover in our implementation is rather big: 161 for the word order variations associated with clausal complements alone.[10] To test

[5] Extraposition from free and V2 word orders and displacement to the front of the matrix clause are future work.

[6] An alternative approach would be to use head types (including disjunctive types) instead of features. However, this is not sufficient when phrase structure rules need to distinguish between subtypes of one head type, such as *verb*.

[7] Our additions to the customization system cover many scenarios of this sort.

[8] The AVMs shown are highly schematized displays of the actual constraints emitted by the Grammar Matrix.

[9] E.g.: SOV, obligatory complementizer attaching before and after the clausal complement, obligatory extraposition.

[10] Assuming just one complementation strategy per pseudolanguage. Six (6; SOV, OVS, OSV, VOS, V-final, V-initial) word orders participate in extraposition, for three (3; SVO, VSO, free) it does not make sense or is not supported, and finally V2, for which we do not support extraposition, can feature V-final word order in the subordinate clause, which we do support. Then there are 7 choices for the complementizer and its optionality and position, and finally there are 3 options for extrapotion itself (strict, flexible, or no extraposition). $6 \cdot 7 \cdot 3 + 3 \cdot 7 + 2 \cdot 7 = 161$

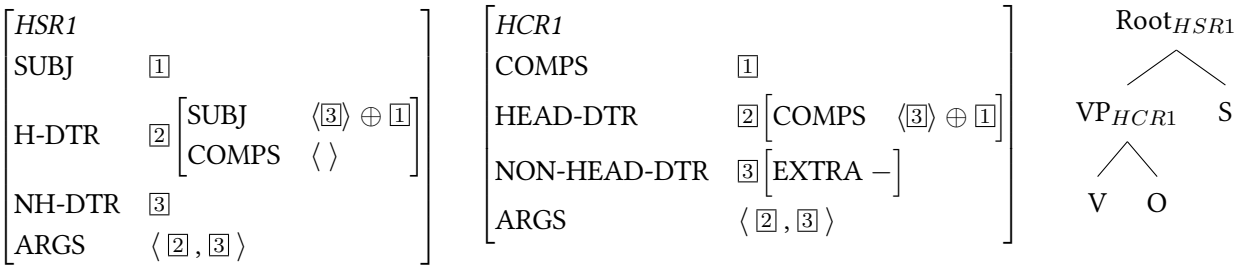

Figure 1: Abbreviated general head-initial HSR and HCR for a VOS language with obligatory extraposition.

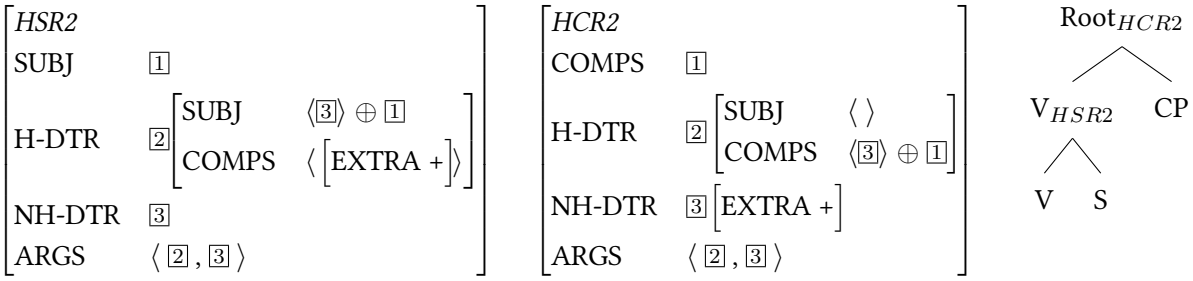

Figure 2: Abbreviated additional HSR and HCR for a VOS language with obligatory extraposition.

the typological legitimacy and the rigor of our analysis, we apply a three-stage evaluation process. In each stage, we create a testsuite illustrating grammatical and ungrammatical strings for each test language, and then create a grammar specification (choices) file using the Grammar Matrix web questionnaire to model the language as documented in the testsuite. Then we automatically create the grammars corresponding to each choices file using the customization system and run them on the testsuites to obtain coverage and overgeneration figures.

In the first stage, we sampled 50 pseudolanguages and made sure that the grammars behave correctly with respect to these choices combinations. In the second and third stages, the testsuites come from multiple real languages, from different language families. The second stage is still test-driven development rather than evaluation, since it involves languages considered in development. In this stage, we modify the system until we obtain 100% coverage and 0% overgeneration on the testsuite. The results of the true evaluation stage performed after development was frozen and involving languages not considered during development are presented in Table 1. In Jalkunan (SOV), a 3SG pronoun stays in the sentence-medial position while the clausal complement is extraposed to the end (Heath, 2017). We did not come across this strategy in our typological survey and our system did not parse half of Jalkunan examples. On other testsuites,[11] we have 100% coverage and 0% overgeneration.

6 Conclusion

In this abstract we present a typologically robust analysis that extends previous work on basic word order in the Grammar Matrix to account for complex sentences. We implement a customization logic which outputs streamlined grammars taking a wide variety of typologically possible user choices as input. Our use of the INIT and EXTRA features could be extended to cover variations of the word order beyond

[11]Grammar sources: pab (Brandão, 2014), yak (Jansen, 2010), heb (Zuckermann, 2006), wgg (Hercus, 1994). The testsuites are available at `https://students.washington.edu/olzama/gp2.html`

Language	iso639	family	WO	comp	order	extrap	Cov.	Overgen.
Jalkunan	bxl	Nig-Cong	SOV	opt	comp S	strict	4/8	0/12
Paresi-Haliti	pab	Arawak	SOV	-	-	strict	4/4	0/6
Yakima Sahaptin	yak	Plateau-Penutian	free	-	-	-	10/10	0/6
Modern Hebrew	heb	Afro-Asiatic	SVO	oblig	comp S	-	2/2	0/9
Wangkangurru	wgg	Pama-Nyungan	free	-	-	-	10/10	0/3

Table 1: Coverage and overgeneration on sentences from held-out language families.

of what we describe here. The implementation of these analyses are open source and the testsuites and choices files discussed in (§5) are available for download. Our analysis also incorporates existing work on V2/V-final German-like word order variation, which is not discussed in this abstract but can be extended to account for the differences in the matrix and embedded clauses' word order more generally. Future work will include clausal subjects, word order with respect to relative clauses, and embedded *wh*-questions.

References

Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L., & Saleem, S. (2010). Grammar customization. *Research on Language & Computation*, *8*(1), 23–72. Retrieved from `http://dx.doi.org/10.1007/s11168-010-9070-1` (10.1007/s11168-010-9070-1)

Bender, E. M., & Flickinger, D. (2005). Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In *Proceedings of the 2nd international joint conference on natural language processing ijcnlp-05 (posters/demos).* Jeju Island, Korea.

Bender, E. M., Flickinger, D., & Oepen, S. (2002). The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In J. Carroll, N. Oostdijk, & R. Sutcliffe (Eds.), *Proceedings of the workshop on grammar engineering and evaluation at the 19th international conference on computational linguistics* (pp. 8–14). Taipei, Taiwan.

Bender, E. M., Flickinger, D., & Oepen, S. (2008). Grammar engineering for linguistic hypothesis testing. In *Proceedings of the texas linguistics society x conference: Computational linguistics for less-studied languages* (pp. 16–36).

Brandão, A. P. B. (2014). *A reference grammar of Paresi-Haliti (Arawak)* (Unpublished doctoral dissertation).

Copestake, A. (2002). Definitions of typed feature structures. In S. Oepen, D. Flickinger, J. Tsujii, & H. Uszkoreit (Eds.), *Collaborative language engineering* (pp. 227–230). Stanford, CA: CSLI Publications.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on language and computation*, *3*(2-3), 281–332.

Engelkamp, J., Erbach, G., & Uszkoreit, H. (1992). Handling linear precedence constraints by unification. In *Proceedings of the 30th annual meeting on association for computational linguistics* (pp. 201–208).

Fokkens, A. S. (2014). *Enhancing empirical research for linguistically motivated precision grammars* (Unpublished doctoral dissertation). Department of Computational Linguistics, Universität des Saarlandes.

Heath, J. (2017). *A grammar of Jalkunan (Mande, Burkina Faso).* Language Description Heritage Library.

Hercus, L. A. (1994). *A grammar of the Arabana-Wangkangurru language: Lake Eyre basin, South Australia.* Australian National Univ.

Howell, K., & Zamaraeva, O. (in press). Clausal modifiers in the LinGO Grammar Matrix. In *Coling*.

Jansen, J. W. (2010). *A grammar of Yakima Ichishkíin/Sahaptin* (Unpublished doctoral dissertation). University of Oregon.

Keller, F. (1995). Towards an account of extraposition in HPSG. In *Proceedings of the seventh conference on european chapter of the association for computational linguistics* (pp. 301–306).

Noonan, M. (2007). Complementation. In T. Shopen (Ed.), *Language typology and syntactic description* (Vol. 2). Cambridge, UK: Cambridge University Press.

Reape, M. (1994). *A formal theory of word order: A case study in West Germanic* (Unpublished doctoral dissertation). University of Edinburgh.

Zuckermann, G. (2006). Complement clause types in Israeli. Oxford University Press.